



Mplify Standard

Mplify 117.1

**SASE Service Attributes and Service Framework
Revision**

June 2025

Disclaimer

© Mplify Alliance 2025. All Rights Reserved.

The information in this publication is freely available for reproduction and use by any recipient and is believed to be accurate as of its publication date. Such information is subject to change without notice and Mplify Alliance (Mplify) is not responsible for any errors. Mplify does not assume responsibility to update or correct any information in this publication. No representation or warranty, expressed or implied, is made by Mplify concerning the completeness, accuracy, or applicability of any information contained herein and no liability of any kind shall be assumed by Mplify as a result of reliance upon such information.

The information contained herein is intended to be used without modification by the recipient or user of this document. Mplify is not responsible or liable for any modifications to this document made by any other party.

The receipt or any use of this document or its contents does not in any way create, by implication or otherwise:

- a) any express or implied license or right to or under any patent, copyright, trademark or trade secret rights held or claimed by any Mplify member which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
- b) any warranty or representation that any Mplify members will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor
- c) any form of relationship between any Mplify member and the recipient or user of this document.

Implementation or use of specific Mplify standards, specifications, or recommendations will be voluntary, and no Member shall be obliged to implement them by virtue of participation in Mplify Alliance. Mplify is a global alliance of network, cloud, cybersecurity, and enterprise organizations working together to accelerate the AI-powered digital economy through standardization, automation, certification, and collaboration. Mplify does not, expressly or otherwise, endorse or promote any specific products or services.

Table of Contents

1	List of Contributing Members	1
2	Abstract.....	2
3	Terminology and Abbreviations	3
4	Compliance Levels	9
5	Document Conventions.....	10
6	Introduction.....	11
6.1	Document Scope.....	12
6.2	Organization of Standard.....	12
6.3	Characteristics of a SASE Service.....	12
7	Key Concepts and Definitions.....	14
7.1	SASE Session	14
7.2	SASE Edge	14
7.2.1	SASE Agent.....	15
7.3	SASE UNI	15
7.4	SASE Policy End Point	15
7.5	Identity and Access Management.....	16
7.6	Actor Access Connection	16
7.6.1	Customer Termination Point.....	16
7.6.2	Network Termination Point	16
7.7	SASE Session Forwarding.....	16
7.7.1	Application Flow Specification (AFS)	16
7.7.2	Rate Limiter and Bandwidth Flow.....	16
7.8	SASE Session Monitoring	19
7.9	Security Functions	19
7.9.1	Middlebox Security Function (MBSF)	19
7.9.2	IP, Port and Protocol Filtering (IPPF).....	20
7.9.3	DNS Protocol Filtering (DPF)	20
7.9.4	Domain Name Filtering (DNF).....	20
7.9.5	URL Filtering (URLF).....	20
7.9.6	Malware Detection and Removal (MD+R).....	20
7.9.7	Data Loss Prevention (DLP).....	20
7.9.8	Protective Domain Name Service (PDNS)	21
7.9.9	Supported Application Identity and Access Management (SA-IdAM).....	21
7.9.10	Data Integrity	21
7.9.11	Proxy	21
7.10	SASE Service Notifications.....	21
7.11	Subscriber	21
7.12	Service Provider	21
7.13	Policy Driven Orchestration	22
7.13.1	Policy	22
7.13.1.1	Composite Policy.....	22
7.13.1.2	Atomic Policy	22
7.13.2	Policy Execution Order Parameter.....	22
7.14	Zero Trust Framework.....	22

7.14.1	Actor	23
7.14.2	Policy End Point	23
7.14.3	Identity Provider	23
7.15	Performance Metrics.....	23
7.15.1	Qualified Packets	23
7.15.2	One-Way Packet Delay.....	24
7.15.3	One-Way Mean Packet Delay Performance Metric.....	24
7.15.4	One-Way Mean Packet Delay Variation Performance Metric.....	24
7.15.5	One-Way Packet Loss Ratio Performance Metric	25
8	SASE Service Attributes.....	26
8.1	List of SASE Edges Service Attribute.....	26
8.2	List of SASE Network Termination Points Service Attribute.....	27
8.3	SASE Policy End Point Identifier Service Attribute	27
8.4	List of Identity Providers Service Attribute	27
8.5	List of SASE Application Flow Specifications Service Attribute.....	27
8.6	List of SASE Session State Values Service Attribute	28
8.7	List of SASE Identity Policies Service Attribute	28
8.8	List of SASE Actor Access Connection Policies Service Attribute	28
8.9	List of SASE Supported TLS Versions Service Attribute.....	28
8.10	List of SASE Supported Cipher Suites Service Attribute	28
8.11	List of SASE Supported IPSEC Security Options Service Attribute	28
8.12	List of SASE Context Policies Service Attribute	28
8.13	List of SASE Security Policies Service Attribute	28
8.14	List of SASE Security Functions Service Attribute	29
8.15	List of SASE Session Forwarding Policies Service Attribute	29
8.16	List of SASE Monitoring Policies Service Attribute.....	29
8.17	List of SASE Notification Policies Service Attribute	29
8.17.1	List of SASE Notification Recipients Service Attribute.....	29
8.18	SASE Composite Policy Levels Service Attribute.....	29
8.19	List of SASE Rate Limiters Service Attribute.....	29
8.20	List of SASE Session Business Importance Levels Service Attribute	30
8.21	List of SASE SA-IdAM Application Flow Specifications Service Attribute	31
8.22	List of SASE Data Integrity Actions Service Attribute.....	31
8.23	List of SASE RBI Actors Service Attribute	31
8.24	SASE Performance Time Intervals Service Attribute	31
8.25	SASE Service Performance Objectives Reporting Periods Service Attribute	32
8.26	SASE Policy Execution Order Parameter Range Service Attribute	32
8.27	SASE Attributes for Underlay Connectivity Services.....	32
8.27.1	SASE UCS Service Attributes.....	33
8.27.2	UCS Identifier Service Attribute	33
8.27.3	UCS Type Service Attribute	34
8.27.4	UCS Billing Method Service Attribute.....	34
8.28	SASE UCS UNI Service Attributes.....	34
9	SASE Service Framework.....	36
9.1	SASE Edge	36
9.1.1	SASE Agent.....	39

9.2	Identity and Access Management.....	39
9.2.1	IdAM Authentication of Actors	40
9.2.2	Actor Access Authorization.....	42
9.2.3	Actor Access Connections	42
9.3	SASE Session	43
9.3.1	Session Specification	45
9.3.2	Application Flow Specification (AFS)	45
9.3.3	Actor Pair.....	49
9.3.4	Session State	50
9.3.4.1	<i>Initial</i>	50
9.3.4.2	<i>Operational</i>	51
9.3.4.3	<i>Re-Evaluate</i>	51
9.3.4.4	<i>Terminal</i>	51
9.3.4.5	<i>SASE Session State Machine</i>	52
9.3.5	Ingress IP Packet Classification.....	53
9.3.6	Ingress IP Packet Classification Example.....	54
9.3.7	New SASE Session Creation Example.....	56
9.4	SASE Session Forwarding.....	58
9.4.1	ENCRYPTION Criterion.....	58
9.4.2	PUBLIC-PRIVATE Criterion.....	59
9.4.3	BILLING-METHOD Criterion.....	59
9.4.4	BUSINESS-IMPORTANCE Criterion.....	60
9.4.5	PERFORMANCE Criterion.....	60
9.4.6	BANDWIDTH Criterion	64
9.4.7	Rate Limiter	67
9.5	SASE Session Monitoring.....	68
9.5.1	Session State Change	70
9.6	Security Functions	70
9.6.1	MEF 138 Security Functions	70
9.6.2	SASE Mplify 117.1 Security Function Definitions	71
9.6.2.1	<i>Supported Application Identity and Access Management (SA-IdAM) Security Function</i>	71
9.6.2.2	<i>Data Integrity Security Function</i>	72
9.6.2.3	<i>Proxy</i>	74
9.6.2.4	<i>Cloud Access Security Broker (CASB)</i>	75
9.6.2.5	<i>Remote Browser Isolation (RBI)</i>	77
9.7	SASE Service Notifications.....	78
9.7.1	SASE Authentication or Authorization Notification (SAAN).....	78
9.7.2	SASE Security Event Notification (SSEN)	79
10	Policies.....	82
10.1	SASE Policy	82
10.2	Policy Execution Order Parameter	83
10.3	Identity and Access Management Policy.....	83
10.3.1	Actor Authentication Function	84
10.3.2	Actor Access Authorization Function.....	84
10.3.3	Actor Access Connection.....	85
10.4	Context Policy	86
10.5	Security Policy.....	87
10.6	Session Forwarding Policy	87
10.6.1	ENCRYPTION Criterion.....	88

10.6.2	PUBLIC-PRIVATE Criterion.....	88
10.6.3	BILLING-METHOD Criterion.....	88
10.6.4	BUSINESS-IMPORTANCE Criterion.....	88
10.6.5	PERFORMANCE Criterion.....	88
10.6.6	BANDWIDTH Criterion	89
10.7	Monitoring Policy.....	89
10.8	Notification Policy.....	89
10.9	SASE Edge Policy Map.....	90
11	SASE Considerations when SD-WAN is utilized for a SASE Service.....	91
11.1	SD-WAN AF-SECURITY-INGRESS Ingress Policy Criterion	91
11.2	SD-WAN AF-SECURITY-EGRESS Egress Policy Criterion.....	91
11.3	SD-WAN EGRESS-BLOCK Egress Policy Criteria	91
11.4	SD-WAN ENCRYPTION Ingress Policy Criteria	91
11.5	VIRTUAL-TOPOLOGY Ingress Policy Criterion.....	92
11.6	INTERNET-BREAKOUT Ingress Policy Criterion	92
11.7	ALLOWED-DESTINATION-ZONES Ingress Policy Criterion	92
11.8	BACKUP Ingress Policy Criterion.....	92
11.9	Application Flow Specifications	92
11.10	PUBLIC-PRIVATE Ingress Policy Criterion.....	93
11.11	BILLING-METHOD Ingress Policy Criterion.....	93
11.12	PERFORMANCE Ingress Policy Criterion	93
11.13	BANDWIDTH Ingress Policy Criterion	94
11.14	SWVC List of Application Flow Specification Service Attribute	94
11.15	BUSINESS-IMPORTANCE Ingress Policy Criterion.....	94
12	References.....	95
Appendix A	SASE Session Flow Examples (Informative)	98
A.1	Session Flow with Security Functions, a subset of which are at the Subject and Target SASE Edges.....	98
A.2	Session Flow via Security SASE Edge with Security Functions at Subject SASE Edge but not at Target SASE Edge.....	99
A.3	Session Flow with Security Functions only at Subject and Target SASE Edges	100
A.4	Session Flow with SASE in a Box deployment on Customer Premises	101
A.5	Session Flow for Cloud Only delivered SASE Service.....	102
Appendix B	Zero Trust Network Access (Informative)	103
Appendix C	Major Changes from MEF 117 to Mplify 117.1 (Informative)	105
Appendix D	Acknowledgements (Informative).....	106

List of Figures

Figure 1 – Subject and Target Actors	11
Figure 2 – SASE Service General Diagram.....	14
Figure 3 – Ingress UNI and Egress UNI Examples	15
Figure 4 – Example of Unnamed Rate Limiters	18
Figure 5 – Example of Named Rate Limiters	19
Figure 6 – SASE Service manages Subject Actor access to Target Actor	36
Figure 7 – SASE Edge	37
Figure 8 – SASE Remote Example.....	38
Figure 9 – Actor Access Connections.....	43
Figure 10 – SASE Session State Machine	52
Figure 11 – Ingress IP Packet Classification Flow Example	55
Figure 12 – New SASE Session Flow Example	57
Figure 13 – PERFORMANCE Criterion Parameters	64
Figure 14 – Intended Behavior of a Rate Limiter	68
Figure 15 – Example of Remote Browser Isolation.....	77
Figure 16 – Example of a Session Flow with Security Functions, subset at Subject and Target SASE Edges.....	98
Figure 17 – Example of Security Functions at Subject SASE Edge but not Target SASE Edge.	99
Figure 18 – Example of Security Functions only at SASE Edges.....	100
Figure 19 -- Example of SASE Service on premises with a single device	101
Figure 20 – Example of cloud delivered SASE Service	102
Figure 21 – Example of SASE Use case for Remote Access	103

List of Tables

Table 1 – Terminology.....	7
Table 2 – Abbreviations.....	8
Table 3 – Notation Conventions	10
Table 4 – Diagram Conventions	10
Table 5 – Summary of SASE UCS Service Attributes	33
Table 6 – Summary of SASE UCS UNI Service Attributes	35
Table 7 – Application Flow Specification Criteria – Support Required.....	46
Table 8 – Application Flow Specification Criteria – Support Recommended	48
Table 9 – Performance Metrics	61
Table 10 – Examples of supported Application Actions	74
Table 11 – Items to be included in a SAAN	79
Table 12 – Items to be included in a SSEN	80

1 List of Contributing Members

The following members of Mplify participated in the development of this document and have requested to be included in this list.

- Cisco Systems
- Palo Alto Networks
- Versa

2 Abstract

This document expands upon the definition and specification of the Secure Access Service Edge (SASE) Service Framework, as specified in MEF 117 [28]. For Service Providers, it has become important to deliver the SASE networking and Security Functions as a cohesive Service that can bring together a wide variety of implementations. This includes Security Functions, Policies and Connectivity Services. The document defines the behaviors of the SASE Service that are externally visible to the Subscriber irrespective of the implementation of the Service. A SASE Service based upon the framework defined in this document enables secure access and secure connectivity of Users, Devices, or Applications to resources for the Subscriber.

This document includes:

SASE Service Attributes – The enumeration and description of the information that is agreed between the Subscriber and the SASE Service Provider. The values of these Service Attributes are determined by agreement between the Subscriber and Service Provider, subject to constraints imposed by the Service Provider’s Service description.

SASE Service Framework – A framework for defining components of a SASE Service based on these Service definitions, Service components, and Service Attributes included in the document.

This document supersedes and revises MEF 117 [28] and changes to the standard are summarized in Appendix C.

3 Terminology and Abbreviations

This section defines the terms used in this document. In many cases, the normative definitions of terms are found in other documents. In these cases, the third column is used to provide the reference that is controlling in other Mplify or external documents.

In addition, terms defined in MEF 61.1 [1], MEF 66 [24], MEF 70.2 [24], MEF 138 [26], MEF 95.0.1 [28], and MEF 118.1 [28] are included in this document by reference and only terms significantly relevant to this document are repeated in the table below for the reader's convenience. However, if there is a discrepancy between this document and the referenced document, then the referenced document is authoritative and controlling.

Note that when the term “support” is used in a normative context in this document, it means that the Service Provider can enable the functionality upon agreement with the Subscriber.

Term	Definition	Reference
Actor	A User, Device, or Application.	MEF 118.1 [28]
Actor Access Connection	The Actor Access Connection is the mechanism that enables end-to-end secure access. The Actor Access Connection consists of a Customer Termination Point and a Network Termination Point.	This document
Actor Access Connectivity Policy	A named list of Policy Criteria that is incorporated into a SASE Policy and that determines how an Actor connects to a SASE Service.	This document
Agent	SASE software installed on a Device.	This document
Application Flow Criterion	A specific condition for matching an IP Packet such as a field/value pair or identification by an algorithm or heuristic.	MEF 70.2[24]
Application Flow Specification	A named set of Application Flow Criteria.	MEF 70.2[24]
Atomic Policy	A stand-alone Policy.	Adapted from MEF 95.0.1 [26]
Authentication	The process of verifying the Identity of an Actor.	Adapted from MEF 118.1 [28]
Authorization	The process that results in Allowing or Blocking a Subject Actor from accessing a Target Actor.	MEF 118.1 [28]

Term	Definition	Reference
Cloud Access Security Broker	A set of three Security Functions: <ul style="list-style-type: none"> Supported Application Identity and Access Management Data Integrity Security Function Proxy 	This document
Composite Policy	A set of related Policies (Atomic or Composite) that are organized into a hierarchical structure.	Adapted from MEF 95.0.1 [26]
Context Policy	A named list of Policy Criteria that is incorporated into a SASE Policy and that determines under which circumstances a Session between a Subject Actor and Target Actor is permitted.	This document
Customer Termination Point	The part of the Actor Access Connection in the Subscriber domain.	This document
Data Integrity Security Function	The Security Function that examines Sessions to certain supported Application and determines if the actions included in those Sessions are allowed or blocked.	This document
Identity	The set of characteristics by which a Subject or Target Actor is recognizable and that, within the scope of an Identity Provider's responsibility, is sufficient to uniquely disambiguate an instance of that Actor from an instance of any other Actor.	MEF 118.1 [28]
Identity and Access Management	The process that authenticates and authorizes an Actor to utilize a SASE Service.	This document
Identity and Access Management Policy	A named list of Policy Criteria that is incorporated into a SASE Policy and that determines if an Actor is authenticated and authorized to use a SASE Service.	This document
Identity Provider	The organization that manages the Authentication Credentials of an Actor.	MEF 118.1 [28]
Monitoring Policy	A named list of Policy Criteria that is incorporated into a SASE Policy and that determines how a SASE Service continuously evaluates the parameters of the SASE Session.	This document
Network Termination Point	The part of the Actor Access Connection in the Service Provider domain.	This document
Notification Policy	A named list of Policy Criteria that is incorporated into a SASE Policy and that determines how a SASE Service communicates to the Subscriber events which occur in a SASE Service.	This document

Term	Definition	Reference
Policy	A set of rules used to manage and control the changing or maintaining of the state of one or more managed objects.	MEF 95.0.1 [26]
Policy Criterion	A criterion that describes a specific objective or constraint.	Adapted from MEF 70.2[24]
Policy End Point	The location where one or more Policy-related functions are placed.	MEF 118.1 [28]
Policy Execution Order Parameter	A non-negative integer value for the order of processing of a Policy where the highest value is processed first.	Modified from MEF 95.0.1 [26]
Remote Browser Isolation	A Security Function that inspects and removes threats from web content within a Session.	This document
Supported Application Identity and Access Management	A Security Function that examines Sessions to certain set of supported Applications and determines if the Subject Actor is authenticated and authorized to access the set of supported Applications.	This document
SASE Agent	Software installed on a Device that enables the SASE Edge functionality.	This document
SASE Authentication or Authorization Notification	A communication of an Authentication or Authorization event, i.e., a SAAN is issued when an Actor has been denied access to the SASE Service due to an Authentication or Authorization failure or when a Session is Blocked by the SASE Policy.	This document
SASE Edge	A set of functions (physical or virtual) that are located between the SASE UNI(s) and the Underlay Connectivity Service UNI(s).	This document
SASE Policy	A Composite Policy assigned to a SASE Session that determines how a SASE Service handles the IP Packets of the SASE Session within the SASE Service.	This document
SASE Security Event Notification	A communication of a security event, e.g., a SSEN is issued when a subset of a Session is Blocked or modified.	This document

Term	Definition	Reference
SASE Service	An overlay service that secures the transport of and forwards the Subscriber's IP packets by recognizing the Session, authenticating and authorizing the Actors, implementing Security Functions, and determining forwarding behavior by applying Policies to and monitoring Sessions. MEF SASE Services are specified using Service Attributes defined in this MEF Standard.	This document
SASE Service Provider	The organization providing SASE Services as defined in this document.	This document
SASE Session	A sequence of IP Packets determined by a Session Specification and the Session State. When the term Session is used in this document it means a SASE Session unless otherwise qualified.	This document
SASE Subscriber	The entity that contracts to use a SASE Service. When the term Subscriber is used in this document it means SASE Subscriber unless otherwise qualified.	This document
SASE UNI	The demarcation point between the responsibility of the SASE Service Provider and the SASE Subscriber.	This document
Security Policy	A named Composite Policy that is incorporated into a SASE Policy and includes a set of Atomic Policies for each Security Function to be applied to a given Session.	This document
Service Provider	An entity that provides services to Subscribers. In this document, Service Provider refers to a SASE Service Provider unless otherwise qualified.	Adapted from MEF 70.2[24]
Session Forwarding Policy	A named list of Policy Criteria that is incorporated into a SASE Policy and that determines how IP packets are transmitted through a SASE Service.	This document
Session Specification	A 2-tuple consisting of a list of Application Flow Specifications and a pair of Actors.	This document
Session State	A list of Session State Values for a particular Session.	This document
Session State Value	The operational condition of the Session at a particular point in time.	This document
State Change Event	A point in time where the Session State Value changes for a given Session.	This document

Term	Definition	Reference
Subject Actor	An Actor requesting access to a Target Actor.	MEF 118.1 [28]
Subscriber	An entity that contracts to use a service. In this document, “Subscriber” should be read as meaning “SASE Subscriber”.	Modified from MEF 70.2[24]
Target Actor	An Actor that a Subject Actor wants to access.	MEF 118.1 [28]
Underlay Connectivity Service	A service providing connectivity between two or more Subscriber Locations, or between a Subscriber Location and the Internet, over which a SASE Service is provided; for example, a private IP Service or a Carrier Ethernet service.	Adapted from MEF 70.2[24]
Uniform Resource Location	The subset of URIs that, in addition to identifying a resource, provides a means of locating the resource by describing its primary access mechanism (e.g., its network "location").	RFC 3986 [13]

Table 1 – Terminology

Abbreviation	Definition	Reference
CASB	Cloud Access Security Broker	This document
CTP	Customer Termination Point	This document
IdAM	Identity and Access Management	This document
IdAMP	Identity and Access Management Policy	This document
IdP	Identity Provider	MEF 118.1 [28]
NTP	Network Termination Point	This document
RBI	Remote Browser Isolation	This document
SAAN	SASE Authentication or Authorization Notification	This document
SASE	Secure Access Service Edge	Adapted from Gartner [30]
SA-IdAM	Supported Application Identity and Access Management	This document
SEEN	SASE Security Event Notification	This document
UNI	User Network Interface	Adapted from MEF 70.2[24]
UCS	Underlay Connectivity Service	Adapted from MEF 70.2[24]
URL	Uniform Resource Location	RFC 3986 [13]
UTC	Coordinated Universal Time	RFC 3339 [10]

Table 2 – Abbreviations

4 Compliance Levels

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 (RFC 2119 [9], RFC 8174 [19]) when, and only when, they appear in all capitals, as shown here. All key words must be in bold text.

Items that are **REQUIRED** (contain the words **MUST** or **MUST NOT**) are labeled as [Rx] for required. Items that are **RECOMMENDED** (contain the words **SHOULD** or **SHOULD NOT**) are labeled as [Dx] for desirable. Items that are **OPTIONAL** (contain the words **MAY** or **OPTIONAL**) are labeled as [Ox] for optional.

5 Document Conventions

Term	Symbol	Usage
Angle Brackets	< >	Surrounds n-tuples
Square Brackets	[]	Surrounds lists
Braces	{ }	Surrounds sets
Parenthesis	()	Surrounds an acronym or example

Table 3 – Notation Conventions















	A shape with a solid outline indicates a required function
	A shape with a dotted outline indicates a desirable function
	This represents a Policy End Point
	This represents an Actor Access Connection
	This represents a SASE UNI
	This represents a UCS UNI
	This represents a private UCS
	This represents a public UCS
	This represents the SASE Service
	This represents a SASE Edge
	This represents a function
	This represents a User
	These represent Devices
	This represents a Customer or Network Termination Point

Table 4 – Diagram Conventions

6 Introduction

The paradigm where the bulk of enterprise traffic is contained within a well-defined and secure enterprise perimeter (e.g., campuses, data centers) is no longer sufficient to address modern security concerns.

This document expands upon the Gartner SASE concept [1]. For Service Providers, it has become important to deliver the SASE networking and Security Functions as a cohesive Service that combines Security Functions and network connectivity for a wide variety of implementations.

This document defines such a Service.

Defining SASE Services introduces generalized constructs that can be applied to increasingly fluid use cases and business requirements.

A SASE Service Provider delivers a SASE Service to a Subscriber (e.g., an enterprise). The SASE Service provides the secure access, the SASE Security Functions, and the secure connectivity between Subscriber's Users, Devices, or Applications and resources (Applications or Devices). This access is independent of the location (public cloud, private cloud, on-premises, Internet, etc.) of the Users, Devices, or Applications and is authorized according to Policies defined by the Subscriber. Such services are needed to cope with the increasingly complex and expanding attack surface resulting from an ever-growing range of Users, Devices, and Applications, an ever-increasing number of locations, and many requiring access to cloud services.

For this purpose, we use the concept of an Actor that is a User, Device, or Application. A SASE Service enables one Actor, the Subject Actor, to access another Actor, the Target Actor. An Actor can be a Subject Actor in one Session and a Target Actor in another Session.

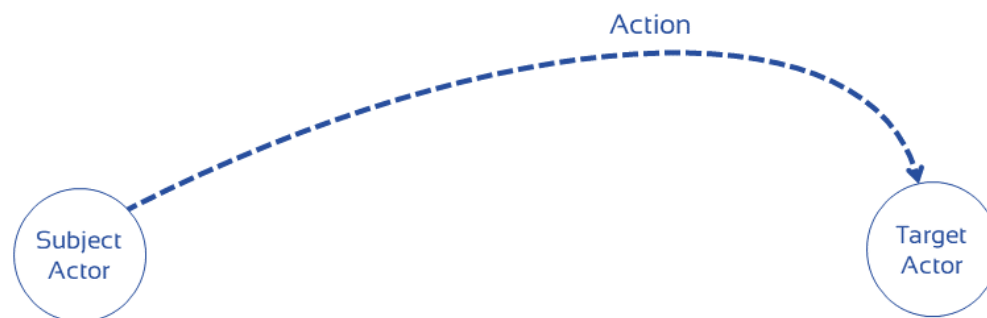


Figure 1 – Subject and Target Actors

Based on the Policies set by the Subscriber, a SASE Service determines whether a Subject Actor is trusted and is permitted to access a Target Actor. Subject Actors can be located anywhere within or outside the direct control of the Subscriber. Similarly, Target Actors may be located anywhere, including in the cloud (public or private), the Subscriber domain, or the SASE Service Provider domain. Target actor could also be on the internet (that is outside the subscriber or service provider domain).

The status of a Subject Actor is not binary (e.g., good or bad, legitimate or illegitimate) but depends on the context and can vary with time or activity (e.g., an authenticated and authorized User that keeps trying to use an Application that is not authorized at times or from locations that are not

permitted by a Subscriber Policy may incrementally lose authorized status in general). Also, the nature of trust for an Actor is not set at any point in time and needs to be continuously evaluated based upon context or activity.

A SASE Service is delineated by the SASE UNI within a SASE Edge. The SASE Edge assigns Subscriber-defined Policies to a SASE Session that the SASE Service Provider manages. The SASE Service, therefore, may or may not include the Subject Actor or the Target Actor themselves, and it may or may not be in proximity to those Actors.

This document defines Service Attributes that describe the externally visible behaviors of a SASE Service as experienced by the Subscriber and that form the basis of the agreement between the SASE Subscriber and the SASE Service Provider. It describes the behaviors from the viewpoint of the Subscriber and therefore all requirements are on the Service Provider.

This document also addresses the SASE Service performance definitions and considerations that will be critical to subscriber adoption.

6.1 Document Scope

This document provides definition and description of:

- SASE Service components.
- SASE Service functionality as viewed by the Subscriber.
- Service Attributes for SASE Edges.
- Service Attributes for Policy End Points.
- SASE Service Sessions and their attributes.
- SASE Policies and the Policy Criteria.

6.2 Organization of Standard

The document is organized as follows:

- Definitions, key concepts, and document conventions are detailed in sections 3, 5 and 7.
- An overview of the Security Functions is provided in sections 7.9.
- An overview of the SASE Service Attributes and requirements is provided in section 8.
- An overview of the SASE Service Framework and requirements is provided in section 9.
- An overview of the SASE Policies and requirements is provided in section 10.
- Considerations when SASE includes SD-WAN are provided in section 11.

6.3 Characteristics of a SASE Service

A SASE Service has the following fundamental characteristics:

- The basic unit of transport within the SASE Service is an IP Packet.
- A SASE Service applies Policy to SASE Sessions.

- Identity and Access Management of Actors utilizing the SASE Service.
- An Actor Access Connectivity Policy to influence Actor Access Connection to and from the SASE Service.
- A Policy-driven networking technology (e.g., SD-WAN).
- Security Functions to secure and protect SASE Sessions through the SASE Service.
- SASE Policies to determine the appropriate handling of IP Packets through the SASE Service.
- Monitoring of SASE Sessions.

7 Key Concepts and Definitions

A SASE Service is composed of Actors, Sessions, Actor Access Connections, SASE Edges, Identity and Access Management, Security Functions, Policy End Points, SASE Policies, the SASE Edge Connectivity, SASE Service Notifications, and Session Monitoring.

The diagram below depicts the logical constructs that make up a SASE Service and their relation to each other.

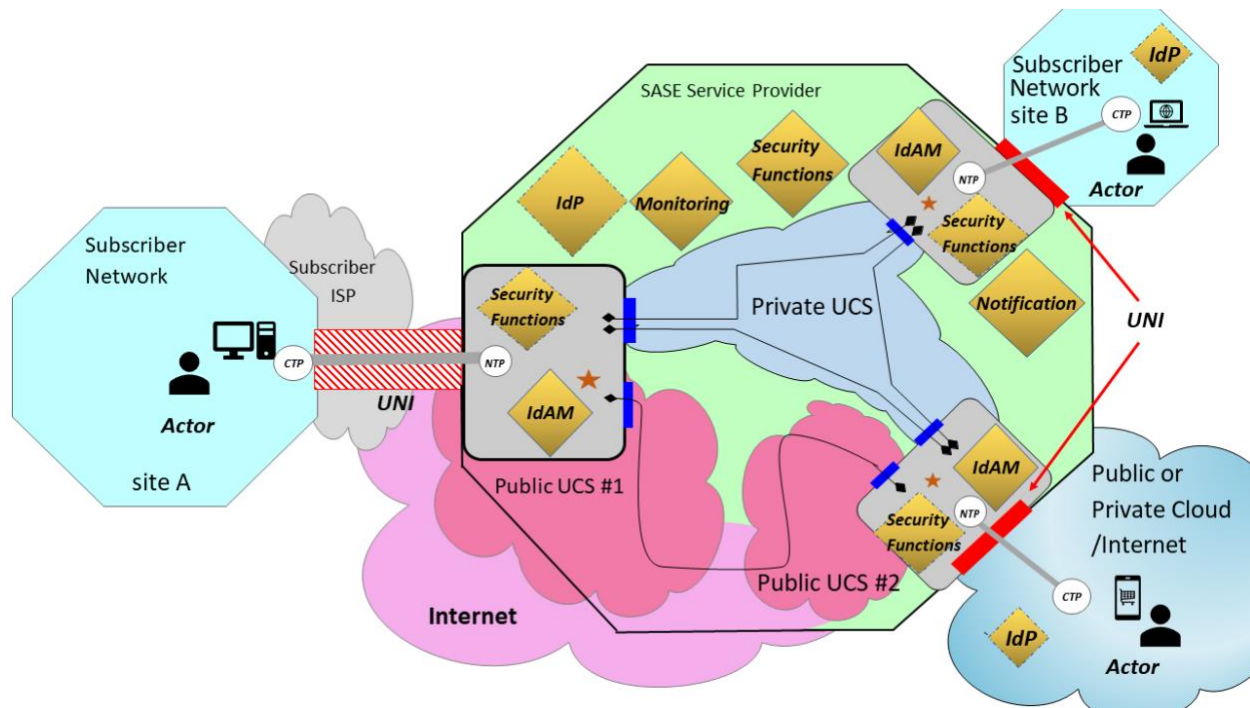


Figure 2 – SASE Service General Diagram

Note: Figure 2 uses diagram conventions which can be found in section 5.

The placement of the various constructs depicted in Figure 2 is an example. The SASE Edge is a logical construct and, as such, could be instantiated in any combination of the following: a cloud environment, on Subscriber premises, or in a Service Provider environment. The SASE Edge functions, shown in Figure 2, can be instantiated in a single Device or spread across multiple Devices. These SASE Edge functions can be instantiated on physical or virtual Devices.

7.1 SASE Session

A SASE Session, or ‘Session’, is a sequence of IP Packets determined by a Session Specification and Session State. (See sections 9.3.1 and 9.3.4)

7.2 SASE Edge

A SASE Edge is a set of network or Security Functions (physical or virtual) that are located between the SASE UNI(s) and the Underlay Connectivity Service UNI(s).

7.2.1 SASE Agent

The SASE Agent is software (installed on a Device) that provides the SASE Edge functionality (see Section 9.1.1).

7.3 SASE UNI

A SASE UNI (see Section 9.1) is the demarcation point between the responsibility of the SASE Service Provider and the SASE Subscriber.

It is common that the SASE UNI is at the SASE Edge containing the Network Termination Point with which the SASE UNI is associated.

An IP Packet that crosses the UNI from the Subscriber to the Service Provider is called an Ingress IP Packet, and the UNI is the Ingress UNI for that IP Packet. Similarly, an IP Packet that crosses the UNI from Service Provider to the Subscriber is called an Egress IP Packet, and the UNI is the Egress UNI for that IP Packet. These are shown in the Figure 3.

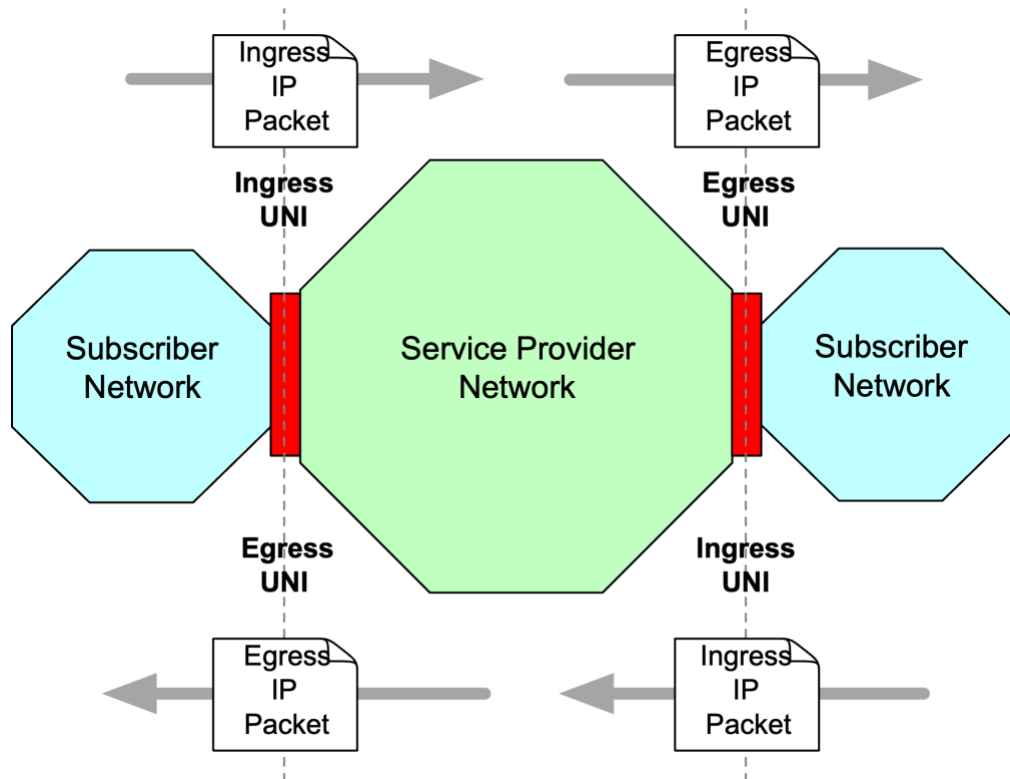


Figure 3 – Ingress UNI and Egress UNI Examples

Note: Figure 3 uses diagram conventions which can be found in section 5.

7.4 SASE Policy End Point

A SASE Policy End Point is where the Policies for the SASE Service are requested, applied, and enforced on Sessions. (See section 10.1)

7.5 Identity and Access Management

Identity and Access Management (IdAM) authenticates and authorizes an Actor to utilize a SASE Service based upon the Identity and Access Management Policies defined by the Subscriber. (See section 9.2)

While IdAM is a Security Function, IdAM is integral to the SASE Service. IdAM determines which IP Packets should be processed by the SASE Service. For this reason, IdAM is treated differently than all other Security Functions and is given its own Policy that is distinct from the Security Policy.

Identity and Access Management is different than the Supported Application Identity and Access Management (SA-IdAM) as IdAM authenticates and authorizes an Actor to utilize the SASE Service. SA-IdAM ensures that a Subject Actor is authenticated and authorized to use a particular set of supported Applications.

7.6 Actor Access Connection

The Actor Access Connection is the mechanism that enables end-to-end secure access. It is the connection between an Actor and the SASE Service. An Actor Access Connection contains a Customer Termination Point and the Network Termination Point. (See section 9.2.2)

7.6.1 Customer Termination Point

The Customer Termination Point is the part of the Actor Access Connection in the Subscriber domain.

7.6.2 Network Termination Point

The Network Termination Point is part of the Actor Access Connection in the Service Provider domain.

7.7 SASE Session Forwarding

The SASE Session Forwarding is the mechanism by which IP Packets are forwarded from one SASE Edge to another SASE Edge within the SASE Service. (See section 9.4)

7.7.1 Application Flow Specification (AFS)

Application Flow Specification (AFS) is a named set of Application Flow Criteria.

7.7.2 Rate Limiter and Bandwidth Flow

It is frequently desirable to limit the rate and/or commit to a minimum rate that data for a Session (or a set of Application Flow Specifications (AFSs)) entering or leaving the SASE Service. This is achieved by including a BANDWIDTH Criterion (see section 10.6.6) in the Session Forwarding Policy assigned to the Session.

For example, it may be desirable to:

1. commit to a minimum of 100 Mb/s and a limit of 200 Mb/s for the Session Specification with values of AFS value *Chat* and ActorPair value *CEO/Client A* at a UNI, and
2. commit to a minimum of 75 Mb/s and a limit of 150 Mb/s for the Session Specification with values of AFS value *Chat* and ActorPair value *Support Engineer A/Client A* at a UNI.

In addition, it may be desirable to limit the rate and/or commit to a minimum rate for a set of AFSs in aggregate, without regard to the set of Actor Pairs for each Session. For example, it may be desirable to:

3. commit to a minimum of 250 Mb/s and a limit of 500 Mb/s for the set of Application Flow Specifications with the value of *Social Media* for all Actor Pairs at the UNI.

The difference is that in bullets 1 and 2, each Session Specification has its own individual bandwidth constraints and thus each Session is unaffected by other Sessions. In contrast, in bullet 3, all the Sessions with the AFS value *Social Media* share the same bandwidth constraints, and, thus, each Session can be affected by the behavior of other Sessions in the set.

The term Bandwidth Flow is defined as a set of one or more Sessions that are treated as a single IP packet flow for the purpose of enforcing bandwidth constraints (commitments and limits). In the previous example, at the UNI, there are three Bandwidth Flows that have been mentioned:

- Session Specification with AFS value *Chat* and ActorPair value *CEO/Client A*
- Session Specification with AFS value *Chat* and ActorPair value *Support Engineer A/Client A*
- Session Specifications with AFS value *Social Media*

Commitments and limits for Bandwidth Flows are specified using Rate Limiters. A Rate Limiter is an abstract component of a SASE Service that ensures that the information rate (measured over time periods of duration *irduration*, see section 8.23) of a Bandwidth Flow conforms to the parameters of the Rate Limiter. A Rate Limiter meters a Bandwidth Flow and, if necessary, delays or discards IP Packets in the Bandwidth Flow so that it conforms to the Rate Limiter parameters. The effect of metering a Bandwidth Flow—that is, comparing the actual sequence of IP Packets that pass the metering point to the description in terms of the Rate Limiter parameters—is to declare IP Packets in the Bandwidth Flow either conformant or nonconformant. IP Packets that are not conformant are discarded in order to ensure that the Bandwidth Flow meets the specified bandwidth limits and commitments. A Rate Limiter commonly behaves as a traffic policer and may also include traffic shaping.

A Rate Limiter can be unnamed or named.

An unnamed Rate Limiter is specified by including a BANDWIDTH Criterion specifying the *commit* and *limit* values for the Rate Limiter (section 9.4.6) in the Session Forwarding Policy assigned to a Session. An unnamed Rate Limiter applies only to that Session. Therefore, a Session that is assigned a Policy that includes a BANDWIDTH Criterion with an unnamed Rate Limiter is always the only Session in its Bandwidth Flow.

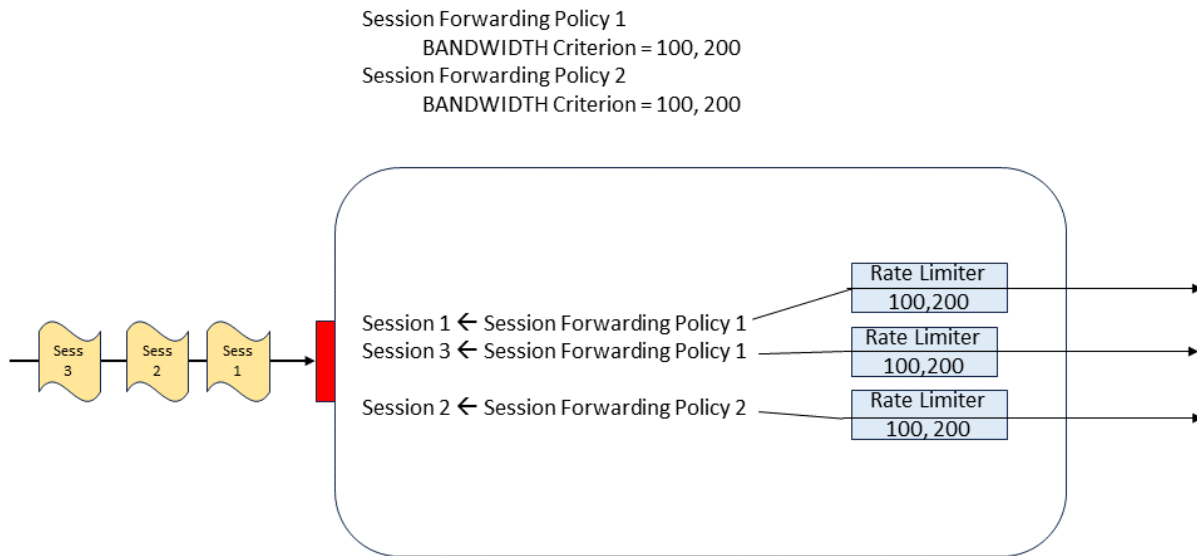


Figure 4 – Example of Unnamed Rate Limiters

In Figure 4, Policy1 and Policy2 both include a BANDWIDTH Criterion in the Session Forwarding Policy with an unnamed Rate Limiter. Session Forwarding Policy 1 is assigned to Session 1 and Session 3 at the SASE End Point, and Session Forwarding Policy 2 is assigned to Session 2. Each of the three Sessions is independently rate limited. In other words, each of these Sessions is a separate Bandwidth Flow even though the parameters for each of the Rate Limiters are the same in both Session Forwarding Policies.

A named Rate Limiter is defined in the List of SASE Rate Limiters Service Attribute (see section 8.19). Each entry in that list specifies a Rate Limiter 3-tuple *<name, commit, limit>*.

If one or more Session Forwarding Policies include a BANDWIDTH Criterion that specifies the name of the same named Rate Limiter (rather than explicit *commit* and *limit* values), only one instance of that Rate Limiter is created at the SASE End Point and all Sessions that are assigned any Session Forwarding Policy that includes a BANDWIDTH Criterion specifying that named Rate Limiter share the Rate Limiter. Therefore, all the Sessions that are assigned any Policy that includes a BANDWIDTH Criterion specifying the same named Rate Limiter are members of the same Bandwidth Flow.

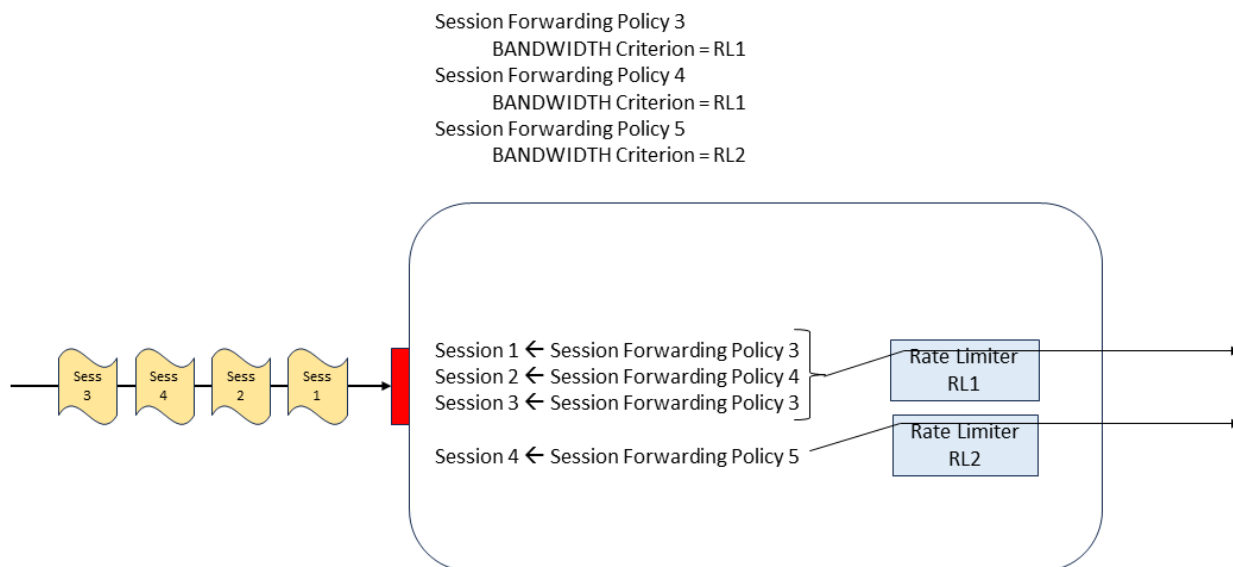


Figure 5 – Example of Named Rate Limiters

In Figure 5, three Session Forwarding Policies (Policy 2, 3 and 4) are defined. Session Forwarding Policies 2 and 3 include a BANDWIDTH Criterion with a named Rate Limiter, RL1, and Session Forwarding Policy 4 is defined with a different named Rate Limiter, RL2. Each Session Forwarding Policy is assigned to Session(s) at the SASE End Point. Sessions 1, 2, and 3 share RL1 since they are all assigned a Session Forwarding Policy that specifies RL1. RL1 views those three Sessions as a single Bandwidth Flow. Session 4 currently has RL2 to itself, but subsequent Session Forwarding Policy assignments (not shown in Figure 5) referencing that named Rate Limiter would result in other Sessions sharing that Rate Limiter.

7.8 SASE Session Monitoring

The SASE Session Monitoring evaluates the SASE Session attributes to determine the health and validity of a given Session inside a given SASE Service. (See section 9.4.7)

7.9 Security Functions

A Security Function is the logical construct that, when enabled per the Security Policy, makes a decision to Allow or Block a subset of a Session.

7.9.1 Middlebox Security Function (MBSF)

Many Security Functions can only work on Sessions that are unencrypted. Therefore, encrypted Sessions are required to be decrypted for the Security Functions to inspect the packets, and then re-encrypted after the Security Function actions are taken. Middlebox Security Function, as defined in MEF 138 [26], is the Security Function that decrypts the IP packets of a given Session for the purposes of utilizing other Security Functions and then re-encrypts the IP packets of the given Session.

7.9.2 IP, Port and Protocol Filtering (IPPF)

IP, Port, and Protocol Filtering is the Security Function, as defined in MEF 138 [26], that determines whether a Session includes a list of source IP addresses, destination IP addresses, source port numbers, destination port numbers, or IP protocols to be Allowed or Blocked.

7.9.3 DNS Protocol Filtering (DPF)

DNS Protocol Filtering is the Security Function, as defined in MEF 138 [26], that determines whether a subset of a Session contains Domain Name System (DNS) messages to be Allowed or Blocked. DNS messages are specified in RFC 1035 [5] and RFC 1996 [8].

7.9.4 Domain Name Filtering (DNF)

Domain Name Filtering is the Security Function, as defined in MEF 138 [26], that determines whether a Session contains domain names to be permitted or denied. Domain Name Filtering provides a level of protection for a Subject Actor inadvertently attempting to access a malicious Target Actor.

7.9.5 URL Filtering (URLF)

URL Filtering is the Security Function, as defined in MEF 138 [26], that determines whether a Session contains URLs to be Allowed or Blocked. URL is specified in IETF RFC 3986 [13]. URL Filtering applies to cases where the domain name is on the Domain Name Filtering Allow List, but one or more URLs associated with that domain have a security issue and need to be Blocked.

7.9.6 Malware Detection and Removal (MD+R)

Malware is software that is specifically designed to disrupt, damage, or gain unauthorized access to a computer system. Malware Detection and Removal is the Security Function, as defined in MEF 138 [26], that determines whether a Session contains Malware. If Malware is detected, Malware Detection and Removal determines whether to remove the Malware or Block the subset of the Session containing the Malware.

A typical use case for Malware Detection and Removal is where a Subscriber specifies a Policy for one or more Subject Actors where all web content, e-mails, file-attachments, and downloads detected in their Sessions are to be checked, and, when Malware is detected, it is removed.

7.9.7 Data Loss Prevention (DLP)

Data Loss Prevention (DLP) is a Security Function, as defined in MEF 138 [26], that determines whether a Service Flow, or subset of a Service Flow, contains confidential, sensitive, or important data, and prevents such data from being exfiltrated by people or systems either intentionally or unintentionally. This Security Function does not cover the case where the Subscriber no longer has access to the data.

7.9.8 Protective Domain Name Service (PDNS)

Protective DNS is the Security Function, as defined in MEF 138 [26], that examines DNS request and response records, and which can allow/block/alter them to protect the recipient.

7.9.9 Supported Application Identity and Access Management (SA-IdAM)

Supported Application Identity and Access Management (SA-IdAM) is the Security Function that examines Sessions to certain supported Applications, as agreed between the Subscriber and Service Provider, determines if the Subject Actor is authenticated and authorized to access the supported Application.

SA-IdAM differs from IdAM as SA-IdAM ensures that a Subject Actor is authenticated and authorized to use a particular supported Application while IdAM authenticates and authorizes an Actor to utilize the SASE Service.

7.9.10 Data Integrity

The Security Function that examines Sessions to a certain set of supported Applications and determines if the actions included in those Sessions are allowed or blocked.

7.9.11 Proxy

A Proxy is a virtual function that acts as an intermediary between the Subject Actor and the Target Actor. The use of a Proxy allows control over the Sessions between Actors.

7.10 SASE Service Notifications

SASE Service Notifications are the alerts and communications that are sent to the Subscriber by the Service Provider as defined by the SASE Notification Policy. (See section 9.7)

7.11 Subscriber

The Subscriber is the entity purchasing or using a SASE Service. The Subscriber defines the requirements that are used to reach agreement on the set of Service Attribute values (see section 8) that a SASE Service Provider uses to implement the SASE Service.

These include but are not limited to:

- The SASE Policies.
- The Security Functions required for the SASE Service.
- Parameters identifying and authenticating the Actors.
- The necessary business logic to develop Session Specifications.

7.12 Service Provider

The SASE Service Provider is the organization providing the SASE Service to a Subscriber. In this document, the use of Service Provider always refers to a SASE Service Provider unless it is

otherwise identified (e.g., UCS Service Provider, Cloud Service Provider, Security Service Provider, SD-WAN Service Provider). The SASE Service Provider configures the SASE Service in a manner that complies with the Subscriber's intent with regards to Service Attributes, Policy, Actors (both Subject and Target), Security Functions, and Actor Access Connectivity.

7.13 Policy Driven Orchestration

A SASE Service incorporates terminology from the MEF 95.0.1 Policy Driven Orchestration, (PDO), Amendment 1 [26].

Policy Driven Orchestration includes:

- The definition of Policy.
- The definition of Composite Policy.
- The definition of Atomic Policy.
- The definition of the Policy Priority.

The SASE Service incorporates these definitions, as needed, to provide the SASE Policy structure to assure secure connectivity between Actors.

7.13.1 Policy

Policy is a set of rules used to manage and control the changing or maintaining of the state of one or more managed objects. There are two types of Policy: Composite and Atomic.

7.13.1.1 Composite Policy

A Composite Policy is a set of related Policies (Atomic or Composite) that are organized into a hierarchical structure.

7.13.1.2 Atomic Policy

An Atomic Policy is a stand-alone Policy.

7.13.2 Policy Execution Order Parameter

Policy Execution Order Parameter is the value for order of execution of a Policy where the highest value is executed first. (See section 10.2)

7.14 Zero Trust Framework

The SASE Service incorporates a Zero Trust Framework, as defined by MEF 118.1 [28].

The Zero Trust Framework includes:

- The definition of Actors with all the associated relationships and attributes.
- The definition of Policy End Points with all the associated attributes.

- The definition of the Identity Management Function with all the associated attributes.

The SASE Service incorporates these attributes as needed to provide the secure connectivity between Actors.

7.14.1 Actor

An Actor is a User, Device, or Application. In line with the Zero Trust principle “never trust, always verify,” Actors are always assumed to be untrusted unless the Actor has passed Authentication and Authorization. An Actor deemed to have malicious intent, by failing Authentication, failing Authorization, or as identified by Continuous Monitoring, is commonly referred to as a “threat actor.”

7.14.2 Policy End Point

A location where one or more Policy-related functions are placed.

7.14.3 Identity Provider

The Identity Provider (IdP) is the entity which authenticates an Actor’s credentials and can provide the Roles that are assigned to the Actor by the Subscriber. The List of Identity Providers Service Attribute (see section 8.2) is used to identify the IdP associated with a given Actor.

7.15 Performance Metrics

The PERFORMANCE Criterion (see section 10.6.5) provides the means for Policies to specify real-time performance requirements for Sessions. The SASE Service attempts to forward IP Packets over paths that meet or exceed the performance goals specified in the Session Forwarding Policy PERFORMANCE Criterion, consistent with the requirements and restrictions posed by other SASE Policy Criteria. The Session Forwarding Policy PERFORMANCE Criterion supports three different Performance Metrics: One-way Mean Packet Delay, One-way Mean Packet Delay Variation, and One-way Packet Loss Ratio.

Section 7.15.1 defines Qualified Packets that are the packets for which the Performance metrics apply. Section 7.15.2 defines One-Way Packet Delay, on which two of the three metrics are based. The subsequent subsections define the three Performance Metrics used in this document.

7.15.1 Qualified Packets

The Performance Metrics defined in the sections below apply to Qualified Packets. A Qualified Packet between two UNIs, referred to as x and y , across a path, referred to as p , is any unicast IP Packet that satisfies the following conditions:

- The IP Packet ingresses at UNI x
- The IP Packet is destined to egress at UNI y
- The IP Packet is not discarded per other requirements
- The path p is chosen to carry the IP Packet
- The IP Packet is not fragmented

- The IP Packet does not incur significant intentional delay as a result of applying a Security Policy agreed to between the Service Provider and the Subscriber.

This covers all cases in which policy or routing prevents the forwarding of a packet, including discarding packets to enforce virtual topologies, allowed destination zones, or security policies. Such packets are therefore never considered to be Qualified Packets.

The final bullet in the list above is focused on delays as a result of Security Functions that require extended analysis (e.g., quarantining or malware detection and removal). Small increases in delay due to processing are expected to be factored into the agreed-on performance goals for the Service.

Note: This standard discusses One-Way Mean Packet Delay and One-Way Mean Packet Delay Variation which are derived from One-Way Packet Delay and One-Way Packet Delay Variation measurements. MEF 66 [24] describes methods for measuring One-Way Packet Delay and One-Way Packet Delay Variation. How the Service Provider actually measures One-Way Packet Delay and One-Way Packet Delay Variation is beyond the scope of this document.

7.15.2 One-Way Packet Delay

The One-way Packet Delay for a Qualified Packet that is sent from UNI x to UNI y across path p is defined as the time elapsed from the reception of the first bit of the packet at the Ingress UNI until the transmission of the last bit of the first corresponding Packet at the Egress UNI. If the packet is duplicated in transit (either erroneously or intentionally, e.g., packet loss remediation), the delay is based on the first copy that is delivered.

Note that this definition of One-way Packet Delay for a packet includes the delays encountered as a result of transmission across the Ingress and Egress UNIs as well as that introduced by the network that connects them.

7.15.3 One-Way Mean Packet Delay Performance Metric

- [R1]** If a Session Forwarding Policy PERFORMANCE Criterion includes a reference to One-Way Mean Packet Delay in the *primary* or *secondary* element, it **MUST** be defined as follows for each path p between UNIs x and y :

Let $\Delta = \{\delta_1, \delta_2, \delta_3, \dots, \delta_n\}$ represent the One-Way Packet Delays of the n Qualified Packets sent from UNI x to UNI y across path p during a time interval whose duration is the value of the *evalinterval* element of the SWVC Performance Time Intervals Service Attribute. Then the One-Way Mean Packet Delay for p over that interval is the arithmetic mean of the values $\delta_1 \dots \delta_n$. If $n=0$ during the time interval, the One-Way Mean Packet Delay for that time interval is zero.

7.15.4 One-Way Mean Packet Delay Variation Performance Metric

- [R2]** If a Session Forwarding Policy PERFORMANCE Criterion includes a reference to One-Way Mean Packet Delay Variation in the *primary* or

secondary element, it **MUST** be defined as follows for each path p between UNIs x and y :

Let $\Delta = \{\delta_1, \delta_2, \delta_3, \dots, \delta_n\}$ represent the One-Way Packet Delays of the n Qualified Packets sent from UNI x to UNI y across path p during a time interval whose duration is the value of the *evalinterval* element of the SWVC Performance Time Intervals Service Attribute. Let Δ' be the set of all pairs of elements $\{\delta_r, \delta_s\}$ in Δ such that $s > r$ and the difference in the arrival time at the Ingress UNI of packets s and r equals the value of the *arrivalinterval* element in the SWVC Performance Time Intervals Service Attribute. If Δ' is *null*, then the One-Way Mean Packet Delay Variation for the time interval is zero. Otherwise, let v_{rs} be the absolute value of the difference in One-Way Packet Delay for each pair, $\{\delta_r, \delta_s\}$ in Δ' , i.e., $v_{rs} = |\delta_r - \delta_s|$. Then the One-Way Mean Packet Delay Variation for p over that interval is the arithmetic mean of the values v_{rs} for each element in Δ' .

7.15.5 One-Way Packet Loss Ratio Performance Metric

[R3] If a Session Forwarding Policy PERFORMANCE Criterion includes a reference to One-Way Packet Loss Ratio in the *primary* or *secondary* element, it **MUST** be defined as follows for each path p between UNIs x and y :

Let s represent the total number of Qualified Packets sent from UNI x to UNI y across path p during a time interval whose duration is the value of the *evalinterval* element of the SWVC Performance Time Intervals Service Attribute. Let r represent the total number of unique (not duplicate) Qualified Packets received from UNI x at UNI y on p that were sent during the same period. Then the One-Way Packet Loss Ratio over that interval for p is defined as follows:

- If $s=0$, then the One-Way Packet Loss Ratio is 0.
- If $s>0$, then the One-Way Packet Loss Ratio is $(s-r)/s$

The One-Way Packet Loss Ratio is usually represented as a percentage.

8 SASE Service Attributes

Mplify Services, such as SASE, are specified using Service Attributes. A Service Attribute captures specific information agreed on between the Service Provider and the Subscriber of a Mplify Service, and it describes some aspects of the service behavior. How such an agreement is reached, and the specific values agreed upon, may have an impact on the price of the service or on other business or commercial aspects of the relationship between the Subscriber and the Service Provider; this is outside the scope of this document. Some examples of how an agreement can be reached are given below, but this is not an exhaustive list.

- The Service Provider mandates a particular value.
- The Subscriber selects from a set of options specified by the Service Provider.
- The Subscriber requests a particular value, and the Service Provider accepts it.
- The Subscriber and the Service Provider negotiate to reach a mutually acceptable value.

Service Attributes describe the externally visible behavior of the service as experienced by the Subscriber as well as the rules and Policies associated with how traffic is handled within the SASE Service. However, they do not constrain how the Service Provider implements the service, nor how the Subscriber implements their network. The Subscriber and the Service Provider agree upon the initial value for each Service Attribute in advance of the Service deployment. The Subscriber and the Service Provider may subsequently agree on changes to the values of certain Service Attributes. This document does not constrain how such agreement is reached; for example, if the Service Provider allows the Subscriber to select an initial value from a pre-determined set of values, they might further allow them to change their selection at any time during the lifetime of the service.

8.1 List of SASE Edges Service Attribute

The value of the List of SASE Edges Service Attribute is a non-empty list of SASE Edge Identifier Service Attribute values. The list contains one SASE Edge Identifier for each SASE Edge in the SASE Service.

- [R4] The List of SASE Edges Service Attribute **MUST** contain at least two SASE Edge Identifier values.

The value of the SASE Edge Identifier is a string used to allow the Subscriber and Service Provider to uniquely identify the association of SASE Service with SASE Edges.

- [R5] The value of the SASE Edge Identifier **MUST** be an Identifier String.
- [R6] The value of the SASE Edge Identifier **MUST** be unique across all SASE Edges in the SASE Service.
- [R7] A SASE Edge Identifier **MUST NOT** appear more than once as a value in the List of SASE Edges Service Attribute.

8.2 List of SASE Network Termination Points Service Attribute

The value of the List of SASE Network Termination Points Service Attribute is a non-empty list of SASE Network Termination Point Identifier values.

- [R8] The List of SASE Network Termination Points Service Attribute **MUST** contain at least two SASE Network Termination Point Identifier values.

The value of the SASE Edge Identifier is a string used to allow the Subscriber and Service Provider to uniquely identify the association of SASE Network Termination Points with SASE Edges.

- [R9] The value of the SASE Network Termination Point Identifier **MUST** be an Identifier String.
- [R10] The value of the SASE Network Termination Point Identifier **MUST** be unique across all SASE Network Termination Points in the SASE Service.
- [R11] A SASE Network Termination Point Identifier **MUST NOT** appear more than once as a value in the List of SASE Network Termination Points Service Attribute.

8.3 SASE Policy End Point Identifier Service Attribute

The value of the SASE Policy End Point Identifier Service Attribute is a string used to allow the Subscriber and Service Provider to uniquely identify the association of the SASE Policy End Point with a SASE Edge.

- [R12] The value of the SASE Policy End Point Identifier Service Attribute **MUST** be an Identifier String.
- [R13] The value of the SASE Policy End Point Identifier Service Attribute **MUST** be unique across all SASE Policy End Points in the SASE Service.

8.4 List of Identity Providers Service Attribute

The Identity Provider (IdP) is the entity that authenticates the Actor's credentials and provides the Roles and Privileges assigned to the Actor by the Subscriber. The List of Identity Providers Service Attribute is used to identify the IdP. The List of SASE Identity Providers Service Attribute is a non-empty list of Identity Provider values utilized in a given SASE Service

8.5 List of SASE Application Flow Specifications Service Attribute

As defined by MEF W70.2 [24], an Application Flow Specification (AFS) is a named set of Application Flow Criteria. An Application Flow Specification matches specific fields or patterns in each IP packet to classify the IP packets. The List of SASE Application Flow Specifications Service Attribute contains all the values and parameters to use in the Session Specification.

8.6 List of SASE Session State Values Service Attribute

The List of SASE Session State Values Service Attribute contains all the Session State Values that a Session can realize within a SASE Service.

[R14] The List of SASE Session State Values Service Attribute **MUST** contain at least the *Initial*, *Operational*, *Re-Evaluate*, and *Terminal* values.

8.7 List of SASE Identity Policies Service Attribute

The List of SASE Identity Policies Service Attribute is a non-empty list of Identity Policy identifier values utilized in a given SASE Service. (See section 10.3.1)

8.8 List of SASE Actor Access Connection Policies Service Attribute

The List of SASE Actor Access Connection Policies Service Attribute is a non-empty list of Actor Access Connection Policy identifier values utilized in a given SASE Service. (See section 10.3.3)

8.9 List of SASE Supported TLS Versions Service Attribute

The List of SASE Supported TLS Versions Service Attribute is a non-empty list of TLS versions that the Service Provider supports for the Actor Access Connection (e.g., TLS 1.2, TLS 1.3, etc.).

8.10 List of SASE Supported Cipher Suites Service Attribute

The List of SASE Supported Cipher Suites Service Attribute is a non-empty list of cipher suites that the Service Provider supports for the Actor Access Connection (e.g., TLS_DHE_RSA_WITH_AES_128_GCM_SHA256).

8.11 List of SASE Supported IPSEC Security Options Service Attribute

The List of SASE Supported IPSEC Security Options Service Attribute is a non-empty list of IPSEC security options that the Service Provider supports for the Actor Access Connection.

8.12 List of SASE Context Policies Service Attribute

The List of SASE Context Policies Service Attribute is a non-empty list of Context Policy identifiers utilized in the SASE Service. (See section 10.4)

8.13 List of SASE Security Policies Service Attribute

The List of SASE Security Policies Service Attribute is a non-empty list of Security Policy identifiers utilized in the SASE Service. (See section 10.5)

8.14 List of SASE Security Functions Service Attribute

The List of SASE Security Functions Service Attribute is a non-empty list of Security Functions utilized in the SASE Service. (See section 9.6)

[R15] The List of SASE Security Functions Service Attribute **MUST** include all the Security Functions identified in [R92].

8.15 List of SASE Session Forwarding Policies Service Attribute

The List of SASE Session Forwarding Policies Service Attribute is a non-empty list of Session Forwarding Policy identifiers utilized in the SASE Service. (See section 10.6)

8.16 List of SASE Monitoring Policies Service Attribute

The List of SASE Monitoring Policies Service Attribute is a non-empty list of Monitoring Policy identifiers utilized in the SASE Service. (See section 10.7)

8.17 List of SASE Notification Policies Service Attribute

The List of SASE Notification Policies Service Attribute is a non-empty list of Notification Policy identifiers utilized in the SASE Service. (See section 10.8)

8.17.1 List of SASE Notification Recipients Service Attribute

The List of SASE Notification Recipients Service Attribute is a non-empty list of recipients that the Subscriber identifies for receiving SASE Notifications. This format for this List is beyond the scope of this document.

8.18 SASE Composite Policy Levels Service Attribute

The SASE Composite Policy Levels Service Attribute is an integer value representing the number of Composite Policy Levels within a SASE Policy that a given SASE Service supports. Since Composite Policies may contain other Composite Policies, the number of Composite Policies that can be iteratively contained within a given Composite Policy (e.g., SASE Policy) needs to be agreed.

[R16] The SASE Composite Policy Levels Service Attribute value **MUST** greater than zero.

8.19 List of SASE Rate Limiters Service Attribute

The List of SASE Rate Limiters Service Attribute specifies the named Rate Limiters (see section 7.7.2) that can be referenced in the BANDWIDTH Criterion (see section 10.6.6).

Each named Rate Limiter is instantiated, at most, once at a SASE End Point and the set Sessions whose Session Forwarding Policy refers to a named Rate Limiter (referred to as a Bandwidth Flow) share the bandwidth constraints specified by the parameters of the Rate Limiter.

The value of the List of SASE Rate Limiters Service Attribute is a list (which may be empty) of 3-tuples, $\langle name, commit, limit \rangle$ where:

- *name* is an Identifier String that is the name of the Rate Limiter.
- *commit* is the threshold information rate (bits per second) at or below which the SASE Service Provider commits to deliver IP Packets in the Bandwidth Flow with high probability under all traffic conditions.
- *limit* is the threshold information rate (bits per second) above which the Service Provider does not deliver IP Packets in the Bandwidth Flow under any traffic conditions.

Following is an example of the value of this Service Attribute:

```
[
  <RL1, 100Mbps, 200Mbps>
  <RL2, 50Mbps, 250Mbps>
]
```

8.20 List of SASE Session Business Importance Levels Service Attribute

Sessions have different levels of business importance to the Subscriber. This could be creating a specific importance for a particular Actor (e.g, the CEO) regardless of the Application Flow Specifications for the Session. Or this could be treating a specific Application Flow Specification (e.g., POS) differently regardless of the ActorPair in the Session. The List of SASE Session Business Importance Levels Service Attribute specifies an ordered list of labels that can be assigned to Sessions to indicate the relative business importance of each Session.

The value of this Service Attribute is *None* or a list of 3-tuple entries of the form $\langle importance, ActorPair, AFSList \rangle$ where:

- *Importance* is the label of business importance:
- *ActorPair* is a 2-tuple of the form $\langle Subject, Target \rangle$ where either the value of *Subject* or the value of *Target* could be *Any*.
- *AFSList* is a non-empty list of Application Flow Specifications from the SASE List of Application Flow Specifications Service Attribute (8.5).

Entries in this Service Attribute are ordered from highest business importance to lowest. Sessions which do not have the BUSINESS-IMPORTANCE Criterion included in their Session Forwarding Policy are treated as having a value lower than the last entry in the List of SASE Session Business Importance Levels Service Attribute.

In general, it is expected that the Subscriber and Service Provider have agreed on sufficient resources in the SASE Service (e.g., number of UCSs, UCS bandwidth, number of paths between SASE Edges, etc.) to meet the service commitments indicated by the Policies assigned to Application Flows. In spite of this, there might be situations where these commitments cannot be met — large simultaneous bursts of ingress traffic, network component failures, re-routing, etc. In these cases, the SASE Service can experience resource contention that might result in the need to delay or discard packets. During these periods, the Service Provider might not be able to meet the SASE Service Session Forwarding Policy commitments. Rather than allowing random decisions

about which packets to delay or discard, each Session Forwarding Policy specifies (explicitly or implicitly) a relative business importance of the Session to the Subscriber via the BUSINESS-IMPORTANCE Criterion (see section 10.6.4) and thereby provides guidance to the Service Provider in these contention situations.

An example of the value of this Service Attribute:

- $\langle [High, \langle CEO, any \rangle, chat],$
 $[Medium, \langle Support\ Engineer\ A, any \rangle, chat],$
 $[Low, \langle any, any \rangle, SocialMedia] \rangle$

This shows a list of three Sessions of business importance.

However, if the value is *None*, Policies in the SASE Service cannot include the BUSINESS-IMPORTANCE Criterion (see 9.4.4), but if the list includes a single element, the BUSINESS-IMPORTANCE Criterion can be included in Session Forwarding Policies, thus providing a basis for future expansion of the number of levels.

8.21 List of SASE SA-IdAM Application Flow Specifications Service Attribute

The List of SASE SA-IdAM Application Flow Specifications Service Attribute is a list, which may be empty, of the supported Application Flow specifications from the List of SASE Application Flow Specifications Service Attribute (see section 8.5) for which a Policy assigned to a Session contains the supported Application Identity and Access Management Security Function.

[R17] The Application Flow Specifications which appear in the List of SASE SA-IdAM Application Flow Specifications Service Attribute **MUST** also appear in the List of SASE Application Flow Specifications Service Attribute.

8.22 List of SASE Data Integrity Actions Service Attribute

The List of SASE Data Integrity Actions Service Attribute is a non-empty list of the Actions supported for each of the entries in List of SASE SA-IdAM Application Flow Specifications Service Attribute (see section 8.21) for which a Policy assigned to a Session contains the Data Integrity Security Function.

8.23 List of SASE RBI Actors Service Attribute

The List of SASE RBI Actors Service Attribute is a non-empty list of Actors for which Remote Browser Isolation will apply. This List of SASE RBI Actors Service Attribute is a subset of all the Actors within the SASE Service.

8.24 SASE Performance Time Intervals Service Attribute

The SASE Performance Time Intervals Service Attribute specifies a set of times and time intervals used in the computation of Performance Metrics or information rates.

The value of the Service Attribute is a 3-tuple $\langle evalinterval, arrivalinterval, irduration \rangle$ where:

- *evalinterval* is the interval in milliseconds over which the Performance Metrics specified in the PERFORMANCE Criterion (section 10.6.4) are evaluated.
- *arrivalinterval* is the difference in arrival times (specified in milliseconds) at the Ingress UNI between two packets used to compute the Mean One-Way Packet Delay Variation (as described in section 7.15.4).
- *irduration* is the time interval in milliseconds over which the information rate is determined in the evaluation of the BANDWIDTH Criterion (section 10.6.6). The information rate is determined over any time interval of duration *irduration*—i.e., it uses a ‘sliding window’ rather than a fixed, recurring, window.

[D1] The *irduration* used for the BANDWIDTH Criterion **SHOULD NOT** exceed 1000 milliseconds.

8.25 SASE Service Performance Objectives Reporting Periods Service Attribute

Service Performance Objectives are evaluated over a series of consecutive time periods. The value of this Service Attribute is a 2-tuple $\langle s, T \rangle$ where:

- *s* is a time that represents the date and time that evaluation of Service Performance Objectives starts for the SASE Service and its components.
- *T* is a time duration, e.g., 1 month or 2 weeks, that is used in conjunction with *s* to specify time intervals for determining when Service Performance Objectives are met.

The time periods are specified by elements *s* and *T*. One time period, denoted T_0 , starts at time *s* and has duration *T*. Each subsequent time period, denoted T_k , starts at time $s + kT$ where *k* is an integer, and has duration *T*; in other words, each new time period starts as soon as the previous one ends. Each Service Performance Objective is evaluated for each time period T_k , so one can say that for a given T_k , the Service Performance Objective is either met or not met.

Note that *T* can be specified using any time units; in particular, calendar months are allowable. In this case, if *s* is specified as, for example, midnight on January 5, 2021, and *T* is 1 calendar month, then each subsequent T_k will start at midnight on the 5th of the month. The details of possible values of *s* and *T* are beyond the scope of this document but, like all Service Attribute values, need to be agreed to by the Subscriber and Service Provider.

8.26 SASE Policy Execution Order Parameter Range Service Attribute

The SASE Policy Execution Order Parameter Range Service Attribute is the range of Policy Execution Order Parameter values, as described in section 10.2, supported by the SASE Service.

8.27 SASE Attributes for Underlay Connectivity Services

This section contains Service Attributes that apply to each of the Underlay Connectivity Services that compose the SASE Service. There is one instance of these attributes for each Underlay Connectivity Service underlying the SASE Service.

Underlay Connectivity Services are network services independent of the SASE Service and can have a large number of “characteristics” or “attributes” that define their configuration and

behavior. If the UCS conforms to a Mplify-defined Service, these characteristics and attributes are the Mplify Service Attributes.

- [R18]** The SASE Subscriber and the SASE Service Provider **MUST** agree on the attributes and characteristics of the UCS that the SASE Subscriber needs to communicate to the SASE Service Provider.
- [R19]** The SASE Subscriber **MUST** communicate the values of the attributes and characteristics of the UCS agreed per [R18] to the SASE Service Provider, if the SASE Service Provider is not already party to them.

Note that the SASE Service Provider may have knowledge of the attributes and characteristics for the UCS either because they are also the UCS Service Provider, or because they purchased the UCS on behalf of the SASE Subscriber. In these cases, there is no need for the SASE Subscriber to communicate them.

The following sub-sections define additional UCS, UCS End Point, and UCS UNI Service Attributes that are specific to the use of the UCS for SASE. The additional attributes include only those necessary to define UCS external interfaces and behavior to the extent necessary to implement SASE Policies. Attributes that are internal to the UCS itself, such as those defining the associations between the UCS and UCS End Points or between the UCS End Point and UCS UNI, are part of the agreement between the UCS Subscriber and UCS Service Provider. These attributes are not included as Service Attributes for the SASE Service and are communicated to the SASE Service Provider per [R18] and [R19].

8.27.1 SASE UCS Service Attributes

The SASE UCS Service Attributes are summarized in the following table, and each is described in more detail in the subsequent sections.

Attribute Name	Summary Description	Possible Values
UCS Identifier	Identification of the Underlay Connectivity Service for management purposes.	Unique Identifier String for the SASE Service
UCS Type	Indicates whether the UCS is a Public UCS (i.e., Internet Access Service) or a Private UCS.	<i>Public or Private</i>
UCS Billing Method	Indicates how the UCS is billed	<i>Flat-rate, Usage-based, or Other</i>

Table 5 – Summary of SASE UCS Service Attributes

8.27.2 UCS Identifier Service Attribute

The value of the UCS Identifier Service Attribute is a string that is used to allow the Subscriber and Service Provider to uniquely identify an Underlay Connectivity Service.

- [R20]** The value of the UCS Identifier Service Attribute **MUST** be an Identifier String.

- [R21] The value of the UCS Identifier Service Attribute **MUST** be unique across all UCS Identifiers in the Service Provider Network.
- [R22] The value of the UCS Identifier Service Attribute **MUST** be the same as the value used to identify the UCS when communicating the Service Attributes or characteristics of the UCS per [R19].

8.27.3 UCS Type Service Attribute

The value of the UCS Type Service Attribute indicates whether the UCS is a Public UCS (i.e., an Internet Access Service) or a Private UCS. The possible values are *Public* and *Private*.

- [R23] If the Underlay Connectivity Service is an IP Service with characteristics consistent with the following MEF 61.1 [23]. Service Attributes: IPVC Topology Service Attribute = *Cloud Access* and IPVC Cloud Service Attribute with Cloud Type parameter = *Internet Access*; then the value of the UCS Type Service Attribute **MUST** be *Public*.
- [R24] If the Underlay Connectivity Service is not an Internet Access Service as described in [R23], then the value of the UCS Type Service Attribute **MUST** be *Private*.

8.27.4 UCS Billing Method Service Attribute

The UCS Billing Method Service Attribute indicates how access to the Underlay Connectivity Service is billed. The allowed values are *Flat-rate*, *Usage-based*, and *Other*.

UCSs with the value *Other* for this Service Attribute are only used for Application Flows with policies where the BILLING-METHOD is *Either* or is not specified.

8.28 SASE UCS UNI Service Attributes

Access to an Underlay Connectivity Service is provided at the SASE Edge via the UCS UNI. Mplify uses the term UNI consistently across all service standards to represent the demarcation point between the responsibility of the Subscriber and the responsibility of the Service Provider. Although not all Underlay Connectivity Services are Mplify Services, the concept of this demarcation point is relevant to all carrier-based services. So, if the Underlay Connectivity Service is a Mplify service the UCS UNI refers to the Mplify-defined UNI for that service and if the Underlay Connectivity Service is not a Mplify-defined service the UCS UNI refers, nonetheless, to the relevant demarcation point.

The SASE UCS UNI Service Attributes are summarized in the following table, and each is described in more detail in the subsequent sections.

Attribute Name	Summary Description	Possible Values
UCS UNI Identifier	Identification of the Underlay Connectivity Service UNI for management purposes.	Identifier String

Table 6 – Summary of SASE UCS UNI Service Attributes

9 SASE Service Framework

A SASE Service is a service that combines wide-area network connectivity and Security Functions to grant a Subject Actor access to a Target Actor for a given Session as shown in Figure 6. This access is based on the Subject Actor's Identity and Privileges, the Session Context, and the Target Actor's Identity as defined in the Policies set by the Subscriber.



Figure 6 – SASE Service manages Subject Actor access to Target Actor

9.1 SASE Edge

The SASE Edge is the set of security and network functions (physical or virtual) that are located between the SASE UNI(s) and the Underlay Connectivity Service UNI(s). This set of functions can be located in a Cloud Service Provider, on-premises of a Service Provider, or on-premises of a Subscriber. A given SASE Edge may have multiple Service UNIs that accept Subscriber IP packets. The SASE Service uses one or more Underlay Connectivity Services to deliver Sessions from one SASE Edge to another.

[R25] Each SASE UNI **MUST** have a unique Identifier String.

In a given Session, there are two SASE Edge types: the Subject SASE Edge and the Target SASE Edge. But for a different Session, the same two SASE Edges could have those roles reversed. The Subject and Target SASE Edges, for a given Session, may be located on the same Device, located in the same cloud, or separated by WAN connectivity within the same SASE Service.

For a given Session, the Subject SASE Edge is the SASE Edge that controls, monitors, and evaluates the Actor Access Connection for the Subject Actor. For a given Session, the Target SASE Edge is the SASE Edge that controls, monitors, and evaluates the Actor Access Connection for the Target Actor. There is no difference in the functions of the two SASE Edges (Subject and Target), but for the discussion of order of operations and process flows, the terms Subject SASE Edge and Target SASE Edge are used as a reference.

A SASE Edge contains all the logical constructs and functions to classify IP packets into SASE Sessions by identifying the Actors, the Application Flow Specification(s), the Session State, Authenticating and Authorizing the Actors, and applying, enforcing, and monitoring SASE Policies for the given SASE Sessions. Therefore, a SASE Edge uses the following components, as illustrated in Figure 7:

- Identity and Access Management.
- Network Termination Point of the Actor Access Connection.
- Policy End Points.
- SASE UNI.

- UCS UNI.

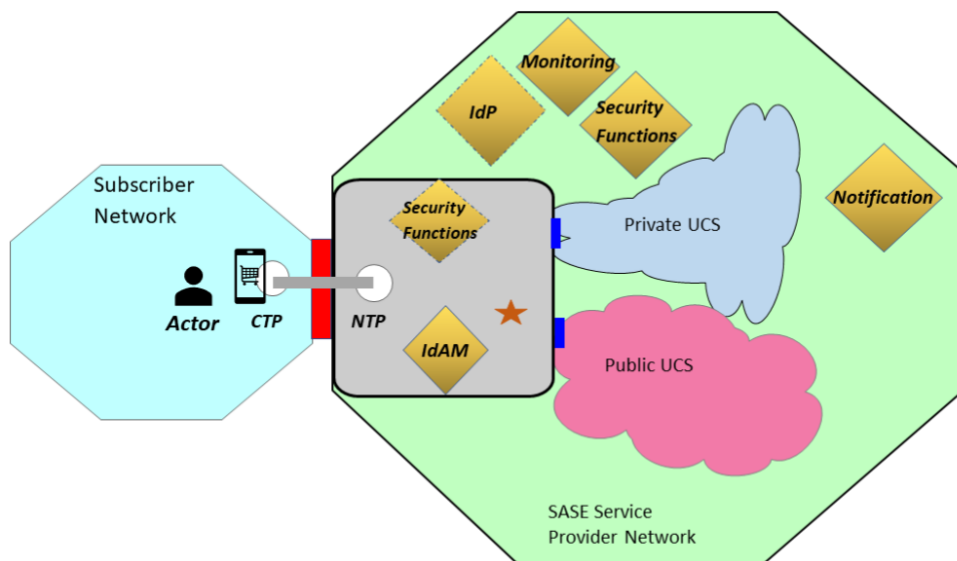


Figure 7 – SASE Edge

Note: Figure 7 uses diagram conventions which can be found in section 5.

- [R26]** Each SASE Edge **MUST** have at least one SASE UNI.
- [R27]** Each SASE Edge **MUST** connect to at least one UCS.
- [R28]** Each SASE Edge **MUST** have a SASE Edge Identifier.
- [R29]** Each SASE Edge **MUST** contain Identity and Access Management.
- [R30]** Each SASE Edge **MUST** contain one or more Network Termination Points.
- [R31]** Each Network Termination Point Identifier in the List of SASE Network Termination Points Service Attribute **MUST** be associated with one and only one SASE Edge.
- [R32]** Each Network Termination Point Identifier in the List of SASE Network Termination Points Service Attribute **MUST** be associated with one and only one SASE UNI.
- [R33]** A SASE Edge **MUST** contain at least one SASE Policy End Point.

- [R34]** If an Egress IP Packet at a SASE UNI results from an Ingress IP Packet at a different SASE UNI, the two SASE UNIs **MUST** be associated by the same SASE Service.

It is recommended that the SASE Edge contain Security Functions to provide the security for the SASE Session. However, where the Security Functions are performed within a SASE Service is at the discretion of the Service Provider provided the placement meets all the requirements in the Subscriber SASE Policy. In Appendix A, there are examples of SASE Sessions flowing through a SASE Service including use cases where Security Functions are done at the SASE Edge, in the SASE Service, and even a use case of SASE-in-a-box or SASE delivered as a Cloud Service Only.

- [D2]** A SASE Edge **SHOULD** contain Security Functions.

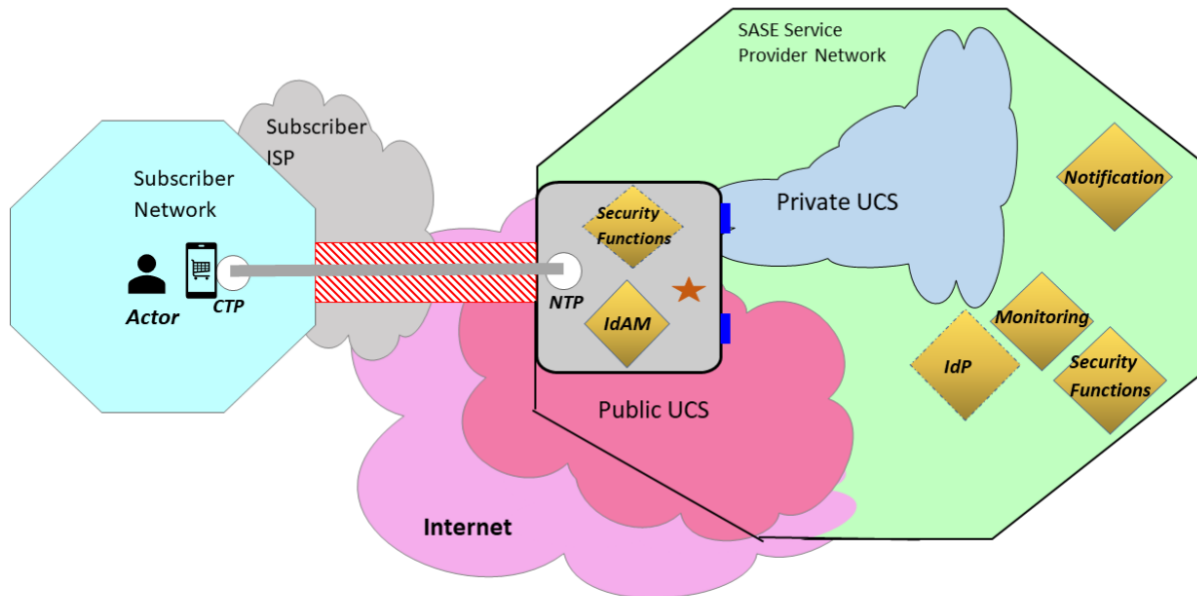


Figure 8 – SASE Remote Example

Figure 8 also shows the agentless connection of a Device to the SASE Service. Here, the Actor Access Connection traverses from the Device across the Internet to a SASE UNI on a SASE Edge instantiated in a Cloud Service Provider.

However, the responsibility of the connectivity between the Subscriber and the Service Provider is distributed between the Subscriber, the Service Provider, and the involved Internet Service Providers (ISPs). To this end, the Service Provider needs to ensure that all the appropriate mechanisms have been utilized to properly secure the Actor Access Connection.

For example, in a scenario where the Subscriber uses an Internet Access Service from their ISP to connect to the SASE Service, the Subscriber would be responsible for their access via their ISP, the SASE Service would be responsible for their Internet Access via their ISP, whether that is the

same ISP as the Subscriber or not, and the ISP or ISPs is responsible for the Access from Subscriber to Service Provider over the Internet.

9.1.1 SASE Agent

In many SASE Services, a SASE Agent is installed on a Subscriber's Device to extend the SASE Service to the Subscriber Device. Since the SASE Agent includes a Policy End Point, the SASE UNI, the IdAMP, and a UCS UNI, the SASE Agent is also considered to be a SASE Edge. The SASE Agent represents the minimal function required to be a SASE Edge. The list of SASE Edges Service Attribute includes all the SASE Agents in the SASE Service.

- [R35] If a SASE Service includes a SASE Agent, then the SASE Agent **MUST** have a SASE Edge Identifier.
- [R36] If a SASE Service includes a SASE Agent, then the SASE Agent **MUST** connect to at least one UCS.
- [R37] If a SASE Service includes a SASE Agent, then the SASE Agent **MUST** contain the Identity and Access Management.
- [R38] If a SASE Service includes a SASE Agent, then the SASE Agent **MUST** contain a SASE UNI.
- [R39] If a SASE Service includes a SASE Agent, then the SASE Agent **MUST** contain at least one SASE Policy End Point.

The size of the SASE Agent software is often restricted to reduce the impact of resources on the Device upon which it will be installed. Thus, advanced functionality might not be included. Exact implementation details are beyond the scope of this document.

As shown in Figure 7, a SASE Agent can be installed on a Device extending the SASE Service to that Device. Since the SASE Agent has a Policy End Point, the SASE UNI, and connects to a UCS, the SASE Agent is the SASE Edge as represented in Figure 7. In this case, the Actor Access Connection now is internal to the Device.

9.2 Identity and Access Management

Identity and Access Management (IdAM) has three main roles within the SASE Service. First, the IdAM authenticates the Actor Identity. Second, it authorizes the Actor to utilize a SASE Service based upon the Identity and Access Management Policies as defined by the Subscriber. Finally, the IdAM applies and enforces the Actor Access Connectivity Policy.

The Authentication and Authorization done by the IdAM assures that only authenticated and authorized Actors have access to the SASE Service. This Authentication and Authorization is not the same Authentication and Authorization for the Target Actor. (e.g., the Authentication and Authorization to login and use an Application), that Authentication and Authorization would need to be accomplished by the Target Actor to fully grant access to the Target Actor for the Subject

Actor. The Authentication and Authorization of the Target Actor is handled by the owner of the Target Actor and beyond the scope of this document.

Likewise, this IdAM, while providing the Actor with access to the SASE Service, does not authorize the Session to proceed through the SASE Service. That Authorization is accomplished via the SASE Policies applied to the Session.

[R40] A SASE Service **MUST** use Identity and Access Management to authenticate Actors utilizing a SASE Service.

9.2.1 IdAM Authentication of Actors

The IdAM relies on an Identity Provider to authenticate the Actor credentials. This Identity Provider may be a part of the SASE Service or provided by the Subscriber, either directly or through a third party. In all cases, the Identity and Access Management verifies the Identity of a given Actor.

SASE Services incorporate a Zero Trust Framework, and as such, a SASE Service needs to comply with the requirements concerning Identity as defined by MEF 118.1 [28].

[R41] The Identity and Access Management **MUST** comply with all requirements in Section 8 of MEF 118.1 [28].

The Actor initiating the Session is called the Subject Actor. The Actor receiving the Session is called the Target Actor. The Subject and Target Actors can be the same for different Sessions, but for a given Session, the Subject/Target Actor Pair is fixed. Each Actor has a unique identifier in a SASE Service.

[R42] For a given SASE Service, every Actor **MUST** have a unique identifier, *ActorID*.

[R43] A SASE Service **MUST** authenticate the Identity of all Subject Actors.

It should be noted that the process of verifying the Identity of an Actor is called Authentication. Authentication is determined by matching a set of criteria. In the case of a User Actor, the criteria often utilized in Authentication is a login and a password combination. Many times, multi-factor Authentication (MFA) is used. However, these are just different criteria used to determine the identity of an Actor through the Authentication process.

In the context of Internet of Things (IOT), the identification of a Device Actor is determined by the Authentication Policy and the numerous parameters utilized to establish the identity of a given Device. The more criteria used to authenticate a Device reduces the probability that a Device could be cloned or impersonated. The same concept holds true for all Actors. An example of Device Actor Authentication might be collecting packets from the Device Actor using either a passive or active discovery method. For a passive discovery method, packets can be collected from the Device Actor as they traverse the network. For an active discovery method, targeted queries might be sent to the Device Actor. In either discovery method, the characteristics and communication

patterns of the Device Actor are identified and establish the set of criteria for Authentication. This set of criteria represents a fingerprint which the IDP uses to authenticate the Device Actor.

[D3] A SASE Service **SHOULD** authenticate the Identity of all Target Actors.

A Target Actor exists in one of the three domains: Subscriber domain, Service Provider domain, or public domain.

When the Target Actor is in the public domain, neither the Subscriber nor the Service Provider has direct control of the Target Actor. In this case, the ability to authenticate the Target Actor might be limited. Since the ability to authenticate the Target Actor is limited, the SASE Service provides the Subscriber the ability to define a Policy that determines whether to authenticate Target Actors and the action to perform if Authentication fails or if the Target Actor cannot be authenticated.

The authentication of the Target Actor can be accomplished either by using an Identity and Access Management Policy or other means. Validation of a certificate that is part of a trusted root certificate chain of authority is one possible example of validating the Target Actor without using an Identity and Access Management Policy. There are other methods, but the method for authenticating the Target Actor is beyond the scope of this document.

Given that a SASE Edge receives and transmits IP Packets, the Source and Destination IP addresses need to be associated with the corresponding Actors. Therefore, an Actor is mandated to be associated with an IP address.

[R44] Each Actor **MUST** be associated with an IP address.

[R45] Any IP Packet that ingresses a SASE UNI which cannot be associated with an authorized Actor **MUST** be discarded.

[R46] For a given SASE Service, every *ActorID* for a Subject Actor **MUST** be associated with an IdP which is a value in the List of Identity Providers Service Attribute.

[D4] For a given SASE Service, every *ActorID* for a Target Actor **SHOULD** be associated with an IdP which is a value in the List of Identity Providers Service Attribute.

[R47] An *ActorID* **MUST** be associated with no more than one IdP within a given SASE Service.

The SASE Service could associate an IP address with an Actor based upon the dynamic or static allocation of an IP address to that actor (i.e., DHCP assignment of an IP Address to a Device) or could be the association of a User Actor to a Device IP address or an Application IP address. The methods of IP address association are beyond the scope of this document.

9.2.2 Actor Access Authorization

The second role for the IdAM is to Authorize the Actor Access. The IdAM applies the appropriate Subscriber policy to authorize the Actors to utilize the SASE Service.

[R48] A SASE Service **MUST** authorize all Actors.

[R49] Any IP Packet that ingresses a SASE UNI which cannot be associated with an authorized Actor **MUST** be discarded.

Since all Subject Actors are authenticated, Authorization of the Subject Actors is based upon Authentication and other IdAM Policy parameters. However, since Target Actors may or may not be authenticated, Authorization of the Target Actors is based upon the IdAM Policy parameters applied, which could include an Authentication parameter.

9.2.3 Actor Access Connections

The third role for the IdAM is to control the Actor Access Connection.

Actors are typically not collocated with the SASE Edges. The network connection between an Actor and the SASE Edge is defined as an Actor Access Connection as seen in Figure 9. The Actor Access Connection consists of a Customer Termination Point (in the Subscriber domain) and a Network Termination Point (located in a SASE Edge).

Traffic flowing from the Actor to the SASE Service is carried by the Actor Access Connection. The Actor Access Connection can be established with each session (i.e., new TLS Actor Access Connection with every SASE Session) or these could be pre-established Actor Access Connections (i.e., IPSEC Actor Access Connection established for the Actor to a SASE Edge and SASE Sessions ride over this established Actor Access Connection.) This standard does not preclude either option or any other option as long as the Actor Access Connection is controlled by Policy as defined in the IdAM Policy. (See section 10.3)

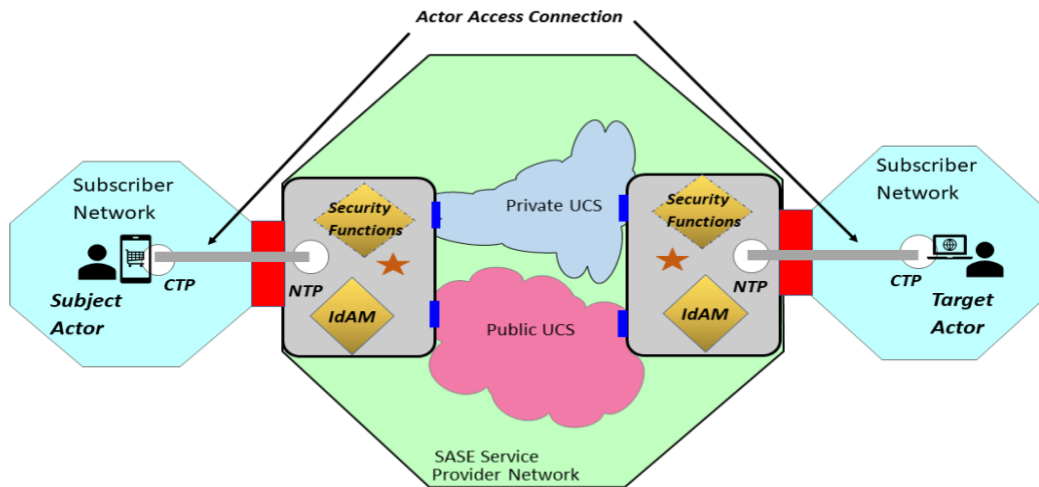


Figure 9 – Actor Access Connections

[R50] The Actor Access Connection **MUST** connect to a SASE Edge via a SASE UNI.

The Actor Access Connection attached via the SASE UNI of a SASE Agent could be represented by the memory and I/O bus within the Device itself. This information still needs to be secured against malicious attacks, especially on shared-use Devices. The exact implementation details are beyond the scope of this document.

9.3 SASE Session

A SASE Session, or ‘Session’, is a sequence of IP Packets determined by a Session Specification and Session State, as defined below.

To properly define the origination and termination points of a Session, the SASE Service uses the concept of an Actor that is a User, Device, or Application. The SASE Session has two Actors, one at each termination point of the SASE Session. The Actor that initiates the Session is called the Subject Actor. The Actor that is accessed is called the Target Actor. An Actor can be a Subject Actor for some Sessions and a Target Actor for other Sessions.

A SASE Service enables the Subject Actor to operate on the Target Actor in a given Session. SASE Policies are set by the Subscriber and determine which Subject Actors can access which Target Actors, and which operations are authorized for execution on the Target Actors.

In the SASE Service, a given Session has a time at which the Session initializes and a time at which the Session is Terminated. Each Session is unique.

[R51] Each SASE Session **MUST** have an identifier, *SessionID*.

- [R52]** The SASE *SessionID* **MUST** be unique across all SASE *SessionIDs* allocated within a given SASE Service.

The sequence of IP Packets is identified for a unique instance by a SASE Session Specification (see section 9.3.1) and Session State (see section 9.5.1).

In a SASE Service, IP Packets are classified into Sessions at the ingress SASE UNI and a SASE Policy is applied to each Session. The Policy determines how the SASE Service handles the Session.

There are several important logical constructs used in this standard to describe and define Sessions:

- Session Specification
 - Actor Pair
 - Subject Actor
 - Target Actor
 - Application Flow Specification
- Session State
 - *Initial*
 - *Operational*
 - *Re-evaluate*
 - *Terminal*

Note: The SASE Session is used in the SASE Service and is not a characteristic of the IP Packets themselves.

Different Sessions can be distinguished by the differences in Session Specification if occurring in the same time frame, or by differences in time frame for the same Session Specification.

Therefore, a given Session is represented by a 3-tuple of the form $\langle SessionID, SessionSpecID, SessionState \rangle$ where:

- *SessionID* is the unique identifier for the Session.
- *SessionSpecID* is the named set of criteria necessary to classify IP Packets into a Session
- *SessionState* is the list of Session State Values recorded for a given Session.

Several Service Attributes relate to defining a Session:

- List of Application Flow Specifications Service Attribute (8.5)
- List of Identity Providers Service Attribute (8.2)
- List of SASE Session State Values Service Attribute (9.3.4)

Several Service Attributes relate to assigning Policies to Sessions:

- List of SASE Edges Service Attribute (8.1).
- List of Policies Service Attribute (10.1).
- List of SASE Security Functions Service Attribute (8.14).
- SASE Edge Policy Map Service Attribute (10.1).

The following subsections explain these constructs in detail and their relationship to each other.

9.3.1 Session Specification

In a SASE Service, the Session Specification is one of the criteria necessary to classify IP Packets into a Session. The Session Specification is a 3-tuple of the form $\langle SessionSpecID, ActorPair, AFSList \rangle$ where:

- *SessionSpecID* is a unique identifier for the Session Specification.
- *ActorPair* is a 2-tuple of the form $\langle Subject, Target \rangle$.
- *AFSList* is a non-empty list of Application Flow Specifications from the SASE List of Application Flow Specifications Service Attribute (8.5).

The *ActorPair* (i.e., Subject Actor and Target Actor) and the *AFSList* specify a unique set of criteria and, when coupled with Session State, uniquely identifies a Session.

9.3.2 Application Flow Specification (AFS)

The List of SASE Application Flow Specifications Service Attribute is one of the criteria used to specify the Sessions that can be recognized by the SASE Service and information about how to identify IP Packets in each Session. The value of the List of SASE Application Flow Specifications Service Attribute is a non-empty ordered list of 2-tuples $\langle AFName, AFCritList \rangle$ where:

- *AFName* is an Identifier String that is used to refer to the Application Flow Specification (and resulting Sessions when included with the *ActorPair* and *State*).
- *AFCritList* is a non-empty list of Application Flow Criteria 2-tuples of the form $\langle AFCritName, AFCritValue \rangle$ where:
 - *AFCritName* is an Identifier String containing an Application Flow Criterion Name from Table 7 or Table 8.
 - *AFCritValue* contains the parameter values specific to the Application Flow Criterion specified in *AFCritName*. If there are no parameter values, *AFCritValue* is *None*.

[R53] Each Application Flow Specification name, *AFName*, in the value of the List of SASE Application Flow Specifications Service Attribute **MUST** appear, at most, once.

[R54] An *AFCritName* **MUST NOT** appear more than once in the *AFCritList* for each item in the value of the List of SASE Application Flow Specifications Service Attribute.

[R55] If the *AFCritList* element in an entry of the List of SASE Application Flow Specifications Service Attribute contains more than one Application Flow Criterion, an Ingress IP Packet **MUST** match all Application Flow Criteria in order to be associated with the Session.

As shown in the example later in this section, the criteria for one Application Flow Specification can be a subset of the criteria for another Application Flow Specification, so the order that the

Application Flow Specifications are matched, and hence the order of the Application Flow Specification definitions in the value of this Service Attribute is important and is one aspect of the agreed value of this Service Attribute.

Requirement [R53] indicates that the Application Flow Specification is defined by the conjunction of a set of Application Flow Specification Criteria. This doesn't allow for alternatives within an Application Flow Specification. This constraint is partially mitigated by the fact that most of the Application Flow Specification Criteria are ranges or lists of values.

[R56] The Application Flow Specification Criteria supported by the Service Provider **MUST** include the Application Flow Specification Criteria listed in Table 7.

<i>AFCritName</i>	<i>Layer</i>	<i>Match</i>	<i>Values for AFCritValue</i>	<i>Reference</i>
SAV4	3	IPv4 Source Address	List of IPv4 prefixes	RFC 791 [6]
DAV4	3	IPv4 Destination Address	List of IPv4 prefixes	RFC 791 [6]
PROTV4	3	IPv4 Protocol List	List of integers in the range 0 to 255 or a list of keywords from [3] or a mix of integers and keywords	IANA Protocol Numbers Registry
SAV6	3	IPv6 Source Address	List of IPv6 prefixes	RFC 8200 [20]
DAV6	3	IPv6 Destination Address	List of IPv6 prefixes	RFC 8200 [20]
NEXTHEADV6	3	IPv6 Next Header List	List of integers in the range 0 to 255 or a list of keywords from [3] or a mix of integers and keywords	IANA Protocol Numbers Registry [3]
DSCP	3	Differentiated Services Code Point	List of integers in the range 0 to 63	RFC 2474 [10]
SPORT	4	Transport Source Port ¹	List of integers in the range 0 to 65535 or a list of service names from [4] or a mix of integers and service names	IANA Service Name and Port Number Registry [4]
DPORT	4	Transport Destination Port	List of integers in the range 0 to 65535 or a list of service names from [4] or a mix of integers and service names	IANA Service Name and Port Number Registry [4]
APPID	3-7	Custom match including heuristic/algorithmic matching	List of arguments starting with the Application Identifier.	This provides the ability to reference a library of custom application flow specifications.
ANY	1-7	Match any IP Packet	<i>None</i>	

Table 7 – Application Flow Specification Criteria – Support Required

¹ This (and the other Application Flow Criteria referring to ports) was changed from “TCP/UDP Source Port” to “Transport Source Port” in MEF 70.1.

The Application Flow Specification Criteria listed in Table 7 represent the basic “IP 5-tuple” (and APPID and ANY). Note that the IPv4 criteria are optional if all the UNIs that have a SASE End Point for this SASE Service only support IPv6 addressing, and the IPv6 criteria are optional only if those UNIs only support IPv4 addressing.

The APPID Application Flow Specification Criterion provides a method for referring to named packet matching definitions (both simple and complex) defined by the Service Provider. These can include standard matches available to all the Service Provider’s Subscribers from a catalog and/or custom matches developed by the Service Provider by agreement with a particular Subscriber.

APPID can include simple protocol matches that can be accomplished with the other Policy Criteria, such as DPORT, (e.g., APPID match-dest-port 443) or *SSH* or *SNMP* or *RTP*, but they can also support more complex packet inspection, enabling identification of specific applications such as *Microsoft365*², *Webex*³, or *Facebook*⁴.

[R57] If the Service Provider defines a named packet matching definition (either standard or custom) for use with the APPID Application Flow Specification Criterion, the description provided to the Subscriber **MUST** include the following information:

- The Application Identifier
- Additional Arguments Required (beyond the Identifier)
- Details of the applications and application traffic matched by the APPID

Complex matches, for example, using deep packet inspection, often require inspection of several initial packets and may include heuristics to define the characteristics of an Application Flow Specification. These details are included in the description of the matching logic required by [R57].

For example:

- An APPID with name SIP: There are no additional arguments required, and the match is performed by inspecting the TCP or UDP source and destination port in each IP Packet for value 5060 or value 5061.
- An APPID named SIPUSER: This includes an additional argument “user-id”. The operation of this match is the same as SIP with the addition that if the port match is successful, the SIP *To* and *From* fields are matched against the “user-id”.

The Application Flow Specification Criterion ANY matches all IP Packets. This criterion allows an Application Flow Specification to be defined that includes all “unmatched” IP Packets. A Policy can then be assigned to resulting Sessions at the SASE End Point. In general, if this Application Flow Specification Criterion is used, it should be in the last in the List of SASE Application Flow Specifications Service Attribute since no IP Packets are matched against subsequent Application Flow Specifications. An example is provided later in this section.

² Microsoft365 is a registered trademark of Microsoft Corporation

³ Webex is a registered trademark of Cisco Systems Incorporated

⁴ Facebook is a registered trademark of Meta Platforms, Inc.

- [R58]** If an entry in the value of the List of SASE Application Flow Specifications Service Attribute includes the Application Flow Specification Criterion ANY, that entry **MUST NOT** contain any other Application Flow Specification Criteria.

Support for the Application Flow Specification Criteria listed in Table 8 is recommended. They provide additional capabilities or additional expressivity in Application Flow Specifications.

- [D5]** The Application Flow Specification Criteria supported by the Service Provider **SHOULD** include the Application Flow Specification Criteria listed in Table 8.

<i>AFCritName</i>	<i>Layer</i>	<i>Match</i>	<i>Values for AFCritValue</i>	<i>Reference</i>	<i>Note</i>
ETHERTYPE	2	EtherType	List of Integers in the range 0x0600 to 0xffff, e.g., 0x0800 for IPv4	802.3 [5]	Since SD-WAN is a layer 3 service, not all implementations have access to the L2 header during Application Flow matching, hence this Application Flow Criterion is optional.
SDAV4	3	IPv4 Source or Destination Address	List of IPv4 prefixes	RFC 791 [6]	This Application Flow Criterion allows an Application Flow to match a list of values for <i>either</i> the source or destination address. Support for this Application Flow Criterion is optional since it can be accomplished with two Application Flows based on the required criteria.
SDAV6	3	IPv6 Source or Destination Address	List of IPv6 prefixes	RFC 8200 [20]	Same as previous
SDPORT	4	Transport Source or Destination Port List	List of integers in the range 0 to 65535 or a list of service names from [4] or a mix of integers and service names	IANA Service Name and Port Number Registry [4]	This Application Flow Criterion allows an Application Flow to match a list of values for <i>either</i> the source or destination port. Support for this Application Flow Criterion is optional since it can be accomplished with two Application Flows based on the required criteria.

Table 8 – Application Flow Specification Criteria – Support Recommended

It is important to note that Table 7 and Table 8 are not intended to describe the implementation or “syntax” of how Application Flow Specifications are described in any product or service, but rather

the capabilities that are available. For example, no implementation is required to have a command or configuration parameter called DPORT, but rather that the implementation can “match packets with destination port =x”.

Here is an example value for this Service Attribute with four Application Flow Specifications:

```
[ <peach,      [{SAV4, [192.168.7.0/24] }, <DPORT, [80,443,8080] }] >
  <VOIP,      [{APPID, ["RTP"]} >
  <banana,    [{DPORT, [80]}] >
  <Else,      [{ANY, None}] >
]
```

In this example, Application Flow Specification *peach* matches IP Packets from any 192.168.7.x address destined to port 80 or 443 or 8080. Application Flow Specification *VOIP* selects IP Packets that are matched by the APPID “RTP”. Application Flow Specification *banana* matches any IP Packet to port 80 that is not matched by *peach*. At the end of the list is the Application Flow Specification *Else*, which matches IP Packets not matched by the other three.

In this example, it is important that *banana* is after *peach* because there are some IP Packets that would match both definitions, but the desired behavior is that they are assigned to *peach*. If *banana* were first, then IP Packets with a source address in 192.168.7.x and destination port 80 would match to Application Flow Specification *banana* rather than Application Flow Specification *peach*.

It is also important to note that many Application Flow Specifications are not symmetric. For example, *banana* above, matches traffic destined to port 80, but response traffic will be from port 80. It can be made symmetric by using SDPORT, if that is available, or by adding another Application Flow Specification, e.g., *banana*, which has SPORT 80. At each SASE Edge a Policy can be assigned to one or both.

9.3.3 Actor Pair

The Actor Pair (*ActorPair*) is the list of the Actors that make up the origination and destination points of a given Session. The Actor List is a 2-tuple of the form <*Subject*, *Target*> where:

- *Subject* is a 2-tuple of the form <*SubjectID*, *IdP*> where:
 - *SubjectID* is the *ActorID* for the Subject Actor.
 - *IdP* is the Identity Provider value, from the List of Identity Providers Service Attribute, that authenticates the Subject Actor.
- *Target* is a 2-tuple of the form <*TargetID*, *IdP*> where:
 - *TargetID* is the *ActorID* for the Target Actor.
 - *IdP* is the Identity Provider value, from the List of Identity Providers Service Attribute, that authenticates the Target Actor, or *Null* if the Target Actor cannot be authenticated.

The SASE Service controls both the sequence of IP Packets from the Subject Actor to the Target Actor and the IP packets that flow from the Target Actor to the Subject Actor; therefore, there is a high probability that the set of Application Flow Specifications will contain more than one Application Flow Specification. However, nothing precludes a single Application Flow Specification that matches the IP packets in both directions.

9.3.4 Session State

Every unique Session has multiple Session State Values. Session State Value is defined as the operational condition of the Session at a particular point in time. The Session State Value *Re-evaluate* for a given Session is communicated to every Policy End Point in the SASE Service. The method for how the Session State Value is communicated to every Policy End Point is beyond the scope of this document.

Session State is a list of Session State Values that reflect the sequence of Session State Value changes and associated Policy decisions made during the lifetime of the Session.

[R59] A Session State Value **MUST** be a value from the List of SASE Session State Values Service Attribute.

Session State is a non-empty list of 2-tuple entries of the form [*<StateValue, Timestamp>*] where:

- *StateValue* is the value from the List of SASE Session State Values Service Attribute as agreed between the Service Provider and Subscriber.
- *Timestamp* is the time that the *StateValue* occurred.

[R60] The Service Provider **MUST** support UTC for the *Timestamp*.

The need to standardize the *Timestamp* times for a Session State mandates that a standard convention is utilized for this timestamp. For this purpose, this standard mandates that UTC be supported.

[D6] The Service Provider **SHOULD** support Subscriber's time zone when recording the *Timestamp*.

Many organizations may want to record the *Timestamp* for a Session State in a manner that has more important significance to the Subscriber than UTC. Therefore, this standard recommends that the Service Provider should support the Subscriber's time zone when recording the timestamp for *Timestamp*.

9.3.4.1 Initial

The Session State Value of *Initial* is defined as the state where the SASE Service receives the first IP Packet for a given SASE Session. During the state of *Initial*, IP packets for the Session may be arriving at the Subject UNI, but since no policy has yet been applied, no IP packets are forwarded to the Target Actor. The SASE Session either transitions to *Operational* (due a Policy being applied to the Session) or *Terminal* if the Session is not authorized.

9.3.4.2 Operational

The Session State Value of *Operational* is defined as the state where the SASE Service has applied a SASE Policy. Once a Policy is applied to a Session, IP packets are forwarded and secured based upon that Policy applied.

9.3.4.3 Re-Evaluate

The Session State Value of *Re-evaluate* is defined as the state where the SASE Service detects a State Change Event (9.5.1) for a given SASE Session and re-evaluation of the Policy is mandated. IP Packets are still flowing from Subject Actor to Target Actor, vice versa, but upon a SASE Edge receiving the next IP packet with the State of *Re-Evaluate*, the SASE Edge will re-evaluate the Policy applied and determine if a new Policy needs to be applied. The SASE Session either transitions to *Operational* (due a Policy being applied to the Session) or *Terminal* if the Session is not authorized.

9.3.4.4 Terminal

The Session State Value of *Terminal* is defined as the state where the SASE Service determines the last IP Packet transmission for a given SASE Session.

Many mechanisms can be utilized to determine when a given SASE Session ends. For example, a Service could look for a Fin/FinAck/Ack termination in a TCP Session, check for an application termination message, or use a temporal dead timer to wait for any delayed packets to be received and transmitted. The method of determining the *Terminal* value of a given Session is beyond the scope of this document.

9.3.4.5 SASE Session State Machine

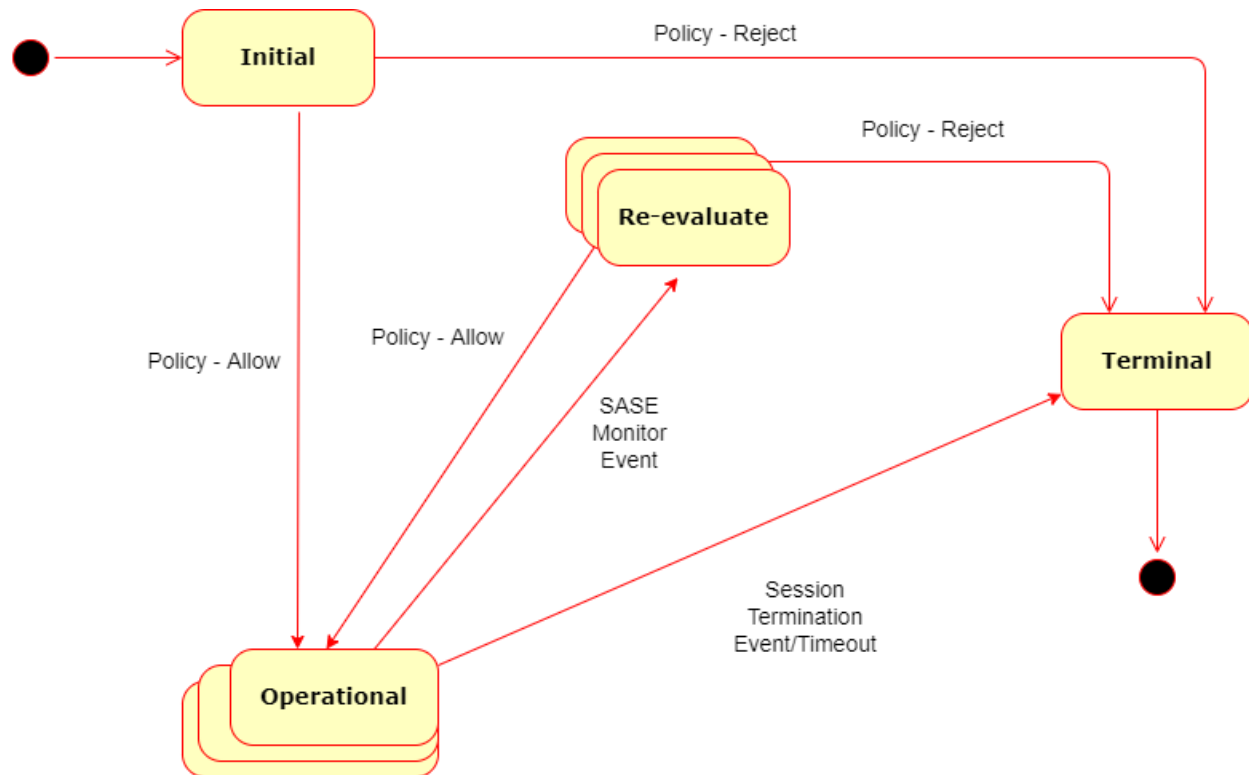


Figure 10 – SASE Session State Machine

Example: Actor A starts to send IP Packets for Application *Talk* destined for Actor Z to SASE Service at time T UTC.

Recall that a SASE Session is defined as a 3-tuple of the form $\langle SessionID, SessionSpecID, SessionState \rangle$.

Since there is no current Session between Actor A and Actor Z, a new Session ID *Example* is established.

So, Actor A and Actor Z would represent the *ActorPair*. The IdAM authenticates and authorizes both Actors. The *AFSList* would include *Talk*. Those together would provide the *SessionSpecID* of *AZ-Talk*.

So, the Session would be $\langle Example, AZ-Talk, SessionState \rangle$ Here the Session State has not been recorded yet and we will now look at the progression of Session State.

- Session = $\langle Example, AZ-Talk, [\langle Initial, T \text{ UTC} \rangle] \rangle$

Since this is the state of *Initial*, no policy had yet been assigned so there is no Policy Map in the SASE Edge for this Session yet.

SASE Service authenticates and authorizes Actor A, evaluates the Session parameters, and applies Policy “Normal” at time T+20ms.

- Session = <Example, AZ-Talk, [<Initial, T UTC>, <Operational, T+20ms UTC>]>

After 5 more minutes, the SASE Service, due to the applicable Monitoring Policy, detects a change in Session parameters and issues a State Change event. This changes the Sessions *StateValue* to *Re-Evaluate*.

- Session = <Example, AZ-Talk, [<Initial, T UTC>, <Operational, T+20ms UTC>, <Re-Evaluate, T+5min20ms UTC>]>

At this point every Policy End Point will re-evaluate the next IP Packet (received at that Policy End Point) for this Session and determine if a new Policy is needed. Once the new Policy is assigned (or if no new Policy was needed), the Session State will change to *Operational*.

The Policy End Point received the next IP Packet in 0.536 seconds after Session State changed to *Re-Evaluate*. After 10ms, the SASE Service was able to determine that new Policy “Restrict” was needed.

Note: The Session State for a given SASE Session could have multiple *Operational* and multiple *Re-evaluate* entries.

- Session = <Example, AZ-Talk, [<Initial, T UTC>, <Operational, T+20ms UTC>, <Re-Evaluate, T+5min20ms UTC>, <Operational, T+5min566ms UTC>]>

After 10 minutes of additional time, the SASE Session transmitted the Fin, Fin Ack, Ack sequence.

- Session = <Example, AZ-Talk, [<Initial, T UTC>, <Operational, T+20ms UTC>, <Re-Evaluate, T+5min20ms UTC>, <Operational, T+5min566ms UTC>, <Terminal, T+15min566ms UTC>]>

At this point any subsequent IP packet that came from Actor A destined for Actor Z for Application *Talk* would result in a new Session Specification ID as the current Session ID *Example* is in the state of *Terminal*.

9.3.5 Ingress IP Packet Classification

The SASE Session consists of a sequence of IP Packets that match a specific Session Specification and have a specific Session State. The ingress SASE Edge is responsible for determining if an IP Packet is part of an existing SASE Session, if it is a new SASE Session, or if it is an IP packet that should be discarded.

- [R61]** Each Ingress IP Packet **MUST** be assigned to a Session based on the first Session Specification it matches, if any.
- [R62]** Any Ingress IP Packet that cannot be associated with a Session based on a Session Specification **MUST** be discarded.

9.3.6 Ingress IP Packet Classification Example

Essentially, the IP Packet needs to be matched to an Actor Pair, identified as belonging to one of the Application Flow Specifications, and associated with a SASE Session, whether an existing SASE Session or the start of a new SASE Session. The methodology by which this is accomplished is beyond the scope of the document. However, two example workflows to achieve this are presented below.

Figure 11 – Ingress IP Packet Classification Flow is an example of the logic that could be utilized by a SASE Service to classify the IP Packets into different SASE Sessions at a SASE Edge.

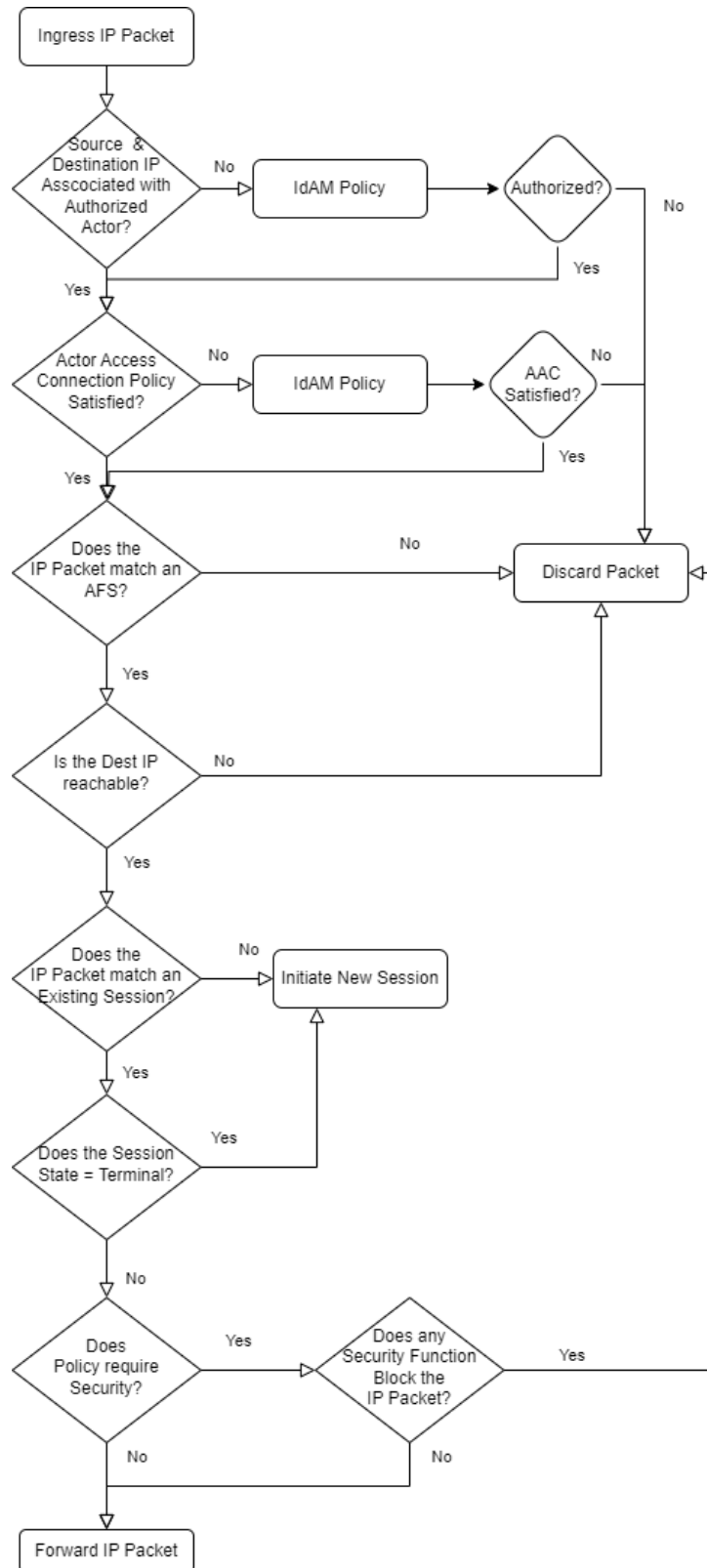


Figure 11 – Ingress IP Packet Classification Flow Example

The flow in Figure 11, shows how the source IP of the ingress IP Packet is checked against the SASE list of authorized and authenticated Actors. If the source IP is not associated with an authorized and authenticated Actor, then the Identity and Access Management is queried to determine what should be done with the IP Packet.

If the IP Packet is associated with an authorized and authenticated Actor, then the Actor Access Connection Policy is triggered to determine if the proper conditions exist for the transport of this packet. If the Actor Access Connection Policy is not satisfied, then the Identity and Access Management is queried to determine what should be done with the IP Packet.

The Destination IP needs to be reachable to process the packet via the SASE Service. If the Destination IP is not reachable, the IP Packet is discarded.

Next the IP Packet is matched against the List of Application Flow Specifications Service Attribute. The SASE Service mandates that the IP Packet match one of the AFSs in the List of Application Flow Specifications Service Attribute or discard the IP Packet.

The SASE Edge determines if the IP Packet can be classified into a known SASE Session. If this can be done, the SASE Edge determines if the SASE Session is not in the Terminal State. If the SASE Session, to which the IP Packet has been associated, is in the Terminal State, the SASE Edge will start a new SASE Session (See section 9.3.7).

If the SASE Session is not in the Terminal State, the IP Packet is then checked against the Security Policy. If the Security Policy does not block the IP packet, it is then forwarded. If the Security Policy does block the IP Packet, then the IP Packet is discarded.

9.3.7 New SASE Session Creation Example

A new SASE Session needs to be created for any IP Packet that does not match an existing Session. Figure 12 is an example of a new SASE Session flow.

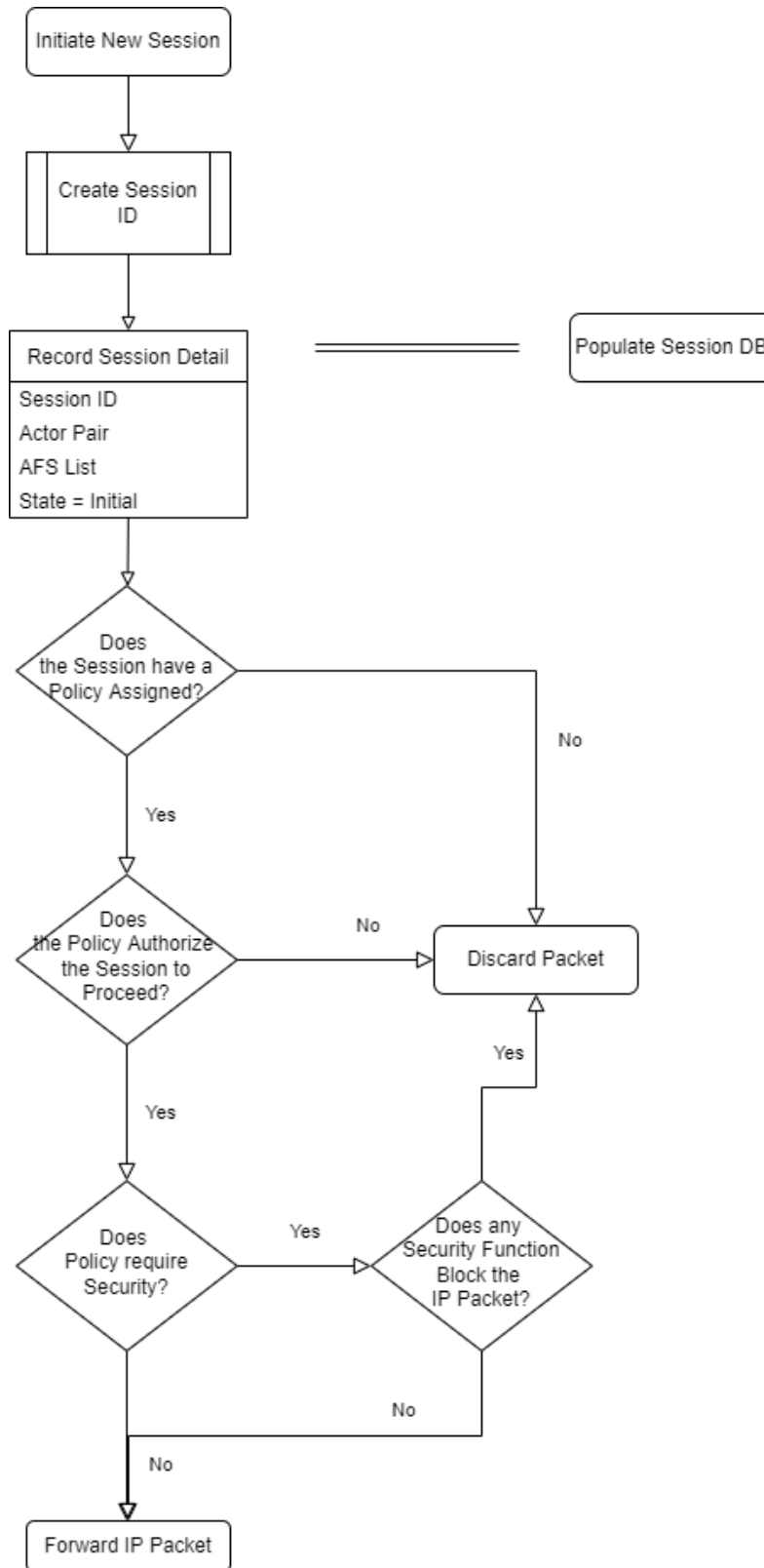


Figure 12 – New SASE Session Flow Example

The Session ID is created and the Session Specification (consisting of Actor Pair and AFS List) is recorded along with the Session State Value of *Initial*.

Next, the IP Packet is checked to see if a SASE Policy is associated with this Session. If not, the IP Packet is discarded.

If there is a SASE Policy, and if the SASE Policy authorizes the IP Packet, then the IP Packet is forwarded. This includes checking if the IP Packet is subjected to the Security Policy and not Blocked.

9.4 SASE Session Forwarding

Traffic transmitted between two SASE Edges within the SASE Service needs to be protected and separated from all other traffic traversing the SASE Service. This is required for the most secure transmission and handling of the data in the Session.

[R63] The connectivity between any two SASE Edges **MUST** be encrypted.

[D7] A Service Provider **SHOULD** support SD-WAN Service as defined in MEF 70.2 [24] for the connectivity between SASE Edges.

Note that when the term *support* is used in a normative context in this document, it means that the Service Provider can provide or enable the functionality upon agreement with the Subscriber. Conditional requirements for SD-WAN can be found in section 11.

If the SASE Edges are in different locations, a SASE Session Forwarding Policy is mandated to direct the IP packets from one SASE Edge to another SASE Edge. A SASE Service treats every individual Session discretely. This implies that two unique Sessions that include the same set of Actors and the same set of Application Flow Specifications can take different paths through the SASE Service based upon the Subscriber's Policies. It is even possible for a single Session to take different paths across the SASE Service due to different evaluation points and changes in the context or behavior as defined by the Subscriber's Policies.

While traditional networking constructs could connect these SASE Edges, and current Security Functions could be implemented to create secure connections across traditional network constructs, this network design would not provide the necessary per-Session performance benefits needed for a SASE Service.

Therefore, a SASE Service needs an advanced networking design that utilizes Policy to implement Security Functions and networking to optimize each individual Session's security and network performance. For this reason, a SASE Service needs to support MEF 70.2 [24] compliant SASE Edge Session forwarding.

9.4.1 ENCRYPTION Criterion

The ENCRYPTION Criterion provides a mechanism to specify whether a Session is required to traverse encrypted paths. It can have the value *Required-Always* or *Required-Public-Only*.

- [R64] If the ENCRYPTION Criterion with a value of *Required-Always* is applied to a Session, the path chosen for forwarding the Session **MUST** include only paths that encrypt IP Packets in the Session before forwarding over the Underlay Connectivity Service.
- [R65] If the ENCRYPTION Criterion with a value of *Required-Public-Only* is applied to a Session, any path chosen for forwarding the Session that traverse Public UCSs **MUST** encrypt IP Packets in the Session before forwarding over the Public UCS.

Requirements [R64] and [R65] mean that encryption is applied by the SASE Edge before packets are forwarded over the UCS UNI, and decryption is applied to packets received from the UCS at the destination SASE Edge.

9.4.2 PUBLIC-PRIVATE Criterion

A SASE Service can use *private* Underlay Connectivity Services such as Mplify Carrier Ethernet Services and *public* Underlay Connectivity Services, i.e., Internet Access Services. The PUBLIC-PRIVATE Criterion provides control over whether or not a Session can traverse a public Underlay Connectivity Service, i.e., the Internet. It can have the value *Private-Only* or *Either*.

- [R66] If the PUBLIC-PRIVATE Criterion with a value of *Private-Only* is applied to a Session, the path chosen for forwarding the Session **MUST** include only paths that traverse a UCS whose UCS Type Service Attribute has a value of *Private* (section 8.27.3).
- [R67] If a Session Forwarding Policy that does not include the PUBLIC-PRIVATE Criterion or a Session Forwarding Policy that includes the PUBLIC-PRIVATE Criterion with the value *Either* is assigned to a Session, then PUBLIC-PRIVATE Criterion **MUST NOT** be considered in the forwarding decision for the Session.

9.4.3 BILLING-METHOD Criterion

The cost for the use of a particular Underlay Connectivity Service can be flat rate (i.e., based on units of time such as €500/month) or usage-based (i.e., based on how much data is sent across it such as €10/TB). The BILLING-METHOD Policy Criterion provides control over the charge type of the network that can be used to forward a session. It can have the value *Flat-Rate-Only*, *Usage-Based-Only*, or *Either*.

- [R68] If the Session Forwarding Policy Criterion BILLING-METHOD=*Flat-Rate-Only* is applied to a Session, the path chosen for forwarding the Session **MUST** traverse only UCSs whose UCS Billing Method Service Attribute=*Flat-rate* (section 8.27.4).
- [R69] If the Session Forwarding Policy Criterion BILLING-METHOD=*Usage-Based-Only* is applied to a Session, the path chosen for forwarding the Session

MUST traverse only UCSs whose UCS Billing Method Service Attribute=*Usage-based* (section 8.27.4).

- [R70] If a Session Forwarding Policy that does not include the BILLING-METHOD Criterion or a Session Forwarding Policy that includes the BILLING-METHOD Criterion with the value *Either* is assigned to a Session, then BILLING-METHOD Criterion **MUST NOT** be considered in the forwarding decision for the Session.

Refer to MEF 74 [26] for examples of a broader range of billing options that are currently beyond the scope of this document.

9.4.4 BUSINESS-IMPORTANCE Criterion

The Session Forwarding Policy BUSINESS-IMPORTANCE Criterion indicates the relative business importance of this Session compared with other Sessions in the SASE Service. This determination is relevant during periods of resource contention when the Service Provider might have to choose to delay or discard IP Packets. During these periods, the Service Provider should provide better service to Sessions that have a higher business importance.

- [D8] During periods of resource contention, if it is necessary for the Service Provider to discard IP Packets or degrade their performance, these actions **SHOULD** target IP Packets in Sessions that have a lower business importance.

The value of the BUSINESS-IMPORTANCE Criterion is the name of a business importance level specified in the *importance* element in the value of the List of SASE Business Importance Levels Service Attribute (see section 8.20).

- [R71] If the value of the List of SASE Business Importance Levels Service Attribute is *None*, the BUSINESS-IMPORTANCE Criterion **MUST NOT** be included in any Session Forwarding Policy and all Sessions are assumed to have the same level of business importance.
- [R72] If the value of the List of SASE Business Importance Levels Service Attribute is not *None* and the Session Forwarding Policy assigned to a Session does not include the BUSINESS-IMPORTANCE Criterion, the effect of the Policy **MUST** be as if the Session Forwarding Policy includes the BUSINESS-IMPORTANCE Criterion with value less than the last element in the value of the List of SASE Business Importance Levels Service Attribute.

9.4.5 PERFORMANCE Criterion

One of the benefits of SASE is that, assuming that there are multiple ways of reaching a destination, the SASE Service can dynamically choose a path that best meets the Session Forwarding Policy applied to a Session, and this includes Session Forwarding Policy Criteria associated with performance.

A SASE service can monitor the performance of the various paths between SASE Edges in real time and adjust the forwarding decisions based on the most recently measured performance. This document does not specify how or when a SASE implementation measures performance or even that it does measure it; however, there is an expectation that the Service Provider can implement the PERFORMANCE Criterion if it is included in a Session Forwarding Policy.

The PERFORMANCE Criterion allows the Subscriber to indicate the important Performance Metrics for each Session. The value of the PERFORMANCE Criterion is a 2-tuple $\langle \textit{primary}, \textit{secondary} \rangle$ or *None*, where:

- *primary* is a 4-tuple $\langle \textit{metric}, \textit{threshold}, \textit{ceiling}, \textit{remediation} \rangle$ that describes the most important Performance Metric for this Session. *metric* is one of the Performance Metrics listed in 7.15 and *threshold*, *ceiling*, and *remediation* are values for this Performance Metric, each of which can be *None*.
- *secondary* is a list of zero or more 2-tuples $\langle \textit{metric}, \textit{threshold} \rangle$. *metric* is one of the Performance Metrics listed in 7.15 and *threshold* is a value for this Performance Metric described below.

[R73] A Performance Metric *metric* **MUST** appear at most once in the value of the PERFORMANCE Criterion.

Requirement [R73] indicates that each performance metric (e.g., One-Way Mean Packet Delay) can only be specified once in the value of the PERFORMANCE Criterion, either in the *primary* element or one of the entries in the list of *secondary* elements.

The result of most Policy Criteria are binary decisions, i.e., either a path meets the requirement or not, whereas the result of the PERFORMANCE Criterion is described as an ordered list of acceptable paths. The Performance Metrics specified in the value of the PERFORMANCE Criterion are evaluated for each path to the Egress UNI and the paths are ordered by how well they meet the stated performance goals.

Performance Metric Name	Description
One-Way Mean Packet Delay	See section 7.15.3
One-Way Mean Packet Delay Variation	See section 7.15.4
One-Way Packet Loss Ratio	See section 7.15.5

Table 9 – Performance Metrics

Note: For the three Performance Metrics used in this document (listed in Table 9), higher values indicate worse performance, so the normative descriptions in this document use “greater than” to express worse performance and “less than” to express better performance.

The *threshold* element in both *primary* and *secondary* is a value for the Performance Metric that indicates the point at which the use of a path for this Session is not preferred, i.e., if the measured value of this Performance Metric is worse than the *threshold*, selection of the path for this Session should be avoided. This is indicated in [R74], [R75], and [D9].

- [R74] When the value of the PERFORMANCE Criterion is not *None*, a path whose performance for any of the specified Performance Metrics (primary or secondary) is worse than the corresponding *threshold* value, if not *None*, **MUST** be placed in the list of paths that result from the evaluation of this Policy Criterion after all paths for which this is not the case.

A path as described in [R74] is not necessarily put at the end of the list but needs to be put after all of the paths whose performance is better than or equal to the threshold values (if not *None*) for all of the specified Performance Metrics.

- [R75] When the value of the PERFORMANCE Criterion is not *None*, all paths other than those described by [R74] **MUST** be ordered based on the Performance Metric specified in *primary*, from best to worst.
- [D9] When the value of the PERFORMANCE Criterion is not *None*, forwarding of IP Packets in a session to which the PERFORMANCE Criterion is applied **SHOULD** favor paths that appear earlier in the list of paths that result from evaluation of the PERFORMANCE Criterion.

A path is selected for a Session based on the *threshold* values as described in [R74], [R75], and [D9]. This path selection can be reevaluated periodically, based on changes in the measured performance of the current path, or based on other factors determined by the Service Provider. This reevaluation may or may not result in a new path selection for the Session.

Once a path is selected for a Session it may be desirable to avoid frequent path changes. Some Sessions don't react well to frequent path switching and the packet reordering and delay variation that can result. Similarly, moving several Sessions from one path to another can result in ping-ponging between paths. The *ceiling* element provides a method to control this.

- [R76] If the value of the *threshold* element for the *primary* Performance Metric is *None*, the values of the *ceiling* and *remediation* elements **MUST** also be *None*.
- [R77] If the value of the *threshold* element for the *primary* Performance Metric is not *None*, the value of the *ceiling* element for the primary Performance Metric **MUST** be greater than or equal to the value of the *threshold* element for the *primary* Performance Metric or *None*.
- [R78] If the value of the *ceiling* element is not *None* and the measured value of the *primary* Performance Metric on the current path is greater than or equal the *ceiling* element, the Service Provider **MUST** select a better path for the Session if one is available.
- [O1] For a session, if the measured value of the primary Performance Metric of the current path is less than the *ceiling* value or if the *ceiling* value is *None*, the

Service Provider **MAY** select a different path that meets the constraints specified by [R74], [R75], and [D9].

To avoid excessive path switching, the *ceiling* element allows for a range of measured performance values (between *threshold* and *ceiling*) within which the Service Provider is not obliged to attempt to switch the Session to a different path. However, requirement [O1] indicates that the Service Provider can move a Session to another path at any time regardless of the measured performance of the current path.

Another way to avoid moving a Session to another path is to attempt to remediate the performance on the current path for the Session. Remediation is a set of techniques intended to improve the performance for a Session usually at the cost of increased resource usage. Techniques used for remediation are outside the scope of this document.

The *remediation* element for the *primary* Performance Metric indicates whether the Service Provider should perform remediation for this Session and, if so, what performance level should trigger remediation.

- [R79]** Session performance remediation for the *primary* Performance Metric **MUST NOT** be performed if the value of the *remediation* element is *None* or if the measured performance of the current path without remediation for the *primary* Performance Metric is less than the *remediation* value.

Note that [R79] makes the need for remediation conditional on the measured performance of the path without remediation. This is important in order to avoid starting and stopping remediation around the value of the *remediation* element.

- [R80]** If the value of the *ceiling* element and the *remediation* element for the *primary* Performance Metric are both not *None*, the value of the *remediation* element for the *primary* Performance Metric **MUST** be less than the value of the *ceiling* element for the *primary* Performance Metric.
- [R81]** If the value of the *ceiling* element for the *primary* Performance Metric is *None* and the value of the *remediation* element for the *primary* Performance Metric is not *None*, the value of the *remediation* element for the *primary* Performance Metric **MUST** be less than or equal to the value of the *threshold* element for the *primary* Performance Metric.
- [D10]** If the value of the *remediation* element is not *None*, the Service Provider **SHOULD** perform remediation for the *primary* Performance Metric when the measured performance of the current path without remediation is greater than or equal to the value of the *remediation* element.

Figure 13 illustrates the relationship between these three parameters. In this diagram *remediation* is shown to have a value less than *threshold*, but consistent with [R81] it can have any value less than *ceiling*.

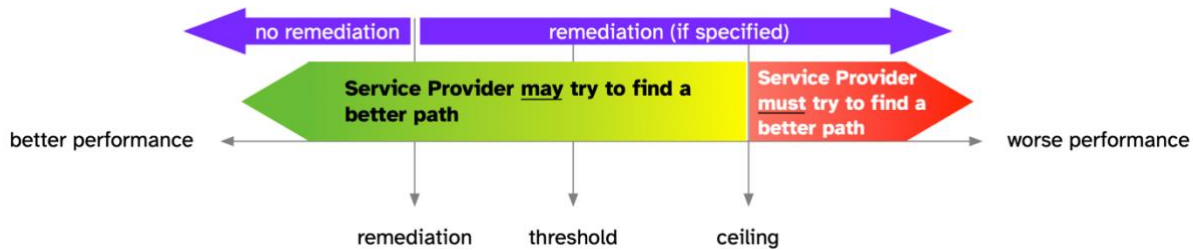


Figure 13 – PERFORMANCE Criterion Parameters

As the measured performance of the current path degrades, it will cross the *remediation* value, if specified, and remediation can be initiated (see [D10]). In the region below the *ceiling* value the Service Provider may choose a different path for the Session, if appropriate (see [O1]), but this consideration becomes more acute when the measured performance exceeds the *threshold* value (see [D10]). When the measured performance (taking into account any remediation in effect) exceeds the *ceiling* value, the Service Provider is required to choose a new path for the Session that has better performance, if one is available (see [R78]).

- [O2] If the Session Forwarding Policy assigned to a Session does not include the PERFORMANCE Criterion or the value of the PERFORMANCE Criterion is *None*, then the Service Provider **MAY** choose whether and how to consider performance in the selection of a path for the Session.

9.4.6 BANDWIDTH Criterion

The BANDWIDTH Criterion provides a method to express the intended bandwidth requirements for a Session, and the probability of packet delay or discard in the face of varying bandwidth contention for Underlay Connectivity Service resources.

The effect of applying the BANDWIDTH Criterion in the Session Forwarding Policy to a Session is to declare IP Packets in the Session either conformant or non-conformant. This can also include delaying IP Packets in order to improve conformance, i.e., traffic shaping. Non-conformant IP Packets in the Session are discarded in order to meet the Session Forwarding Policy.

The value of the BANDWIDTH Criterion is either:

- an unnamed Rate Limiter 2-tuple $\langle commit, limit \rangle$ (see section 7.7.2), or
- the *name* of a named Rate Limiter specified in the List of SASE Rate Limiters Service Attribute (see section 8.19)

- [R82] If the Session Forwarding Policy assigned to a Session does not include the BANDWIDTH Criterion, the effect of the Session Forwarding Policy **MUST** be as if the Session Forwarding Policy includes the BANDWIDTH Criterion with value $\langle 0, None \rangle$.

Requirement [R82] indicates that if a Session Forwarding Policy does not contain a BANDWIDTH Criterion, Sessions to which that Session Forwarding Policy is assigned do not have any committed bandwidth nor a bandwidth limit. This, in effect, means that no Rate Limiter is imposed on the Bandwidth Flow.

For the purpose of determining conformance, the information rate of the Session is computed over time intervals of length *irduration*, which is an element in the value of the SASE Performance Time Intervals Service Attribute (section 8.23).

- [R83] The Subscriber and Service Provider **MUST** agree on sufficient Underlay Connectivity Service capacity to ensure that the *commit* values for all Bandwidth Flows can be met.

The agreement required by [R83] should consider other traffic that shares the capacity of the UCS such as traffic from other UNIs and traffic routed through the SASE Edge. How the average information rates are determined and the behavior of the rate limiting function are described below.

This document does not specify the implementation of the bandwidth measurement/limiting by the Service Provider.

- [R84] Bandwidth metering **MUST** occur after it is determined that the IP Packet will be forwarded based on SASE Policy.

The intent of [R84] is to ensure that bandwidth metering occurs as late as possible in the packet processing in order to avoid metering IP Packets that are subsequently discarded.

The following requirements define the behavior imposed by the BANDWIDTH Criterion.

- [R85] The information rate for IP Packets in a Bandwidth Flow that are declared conformant over any time interval equal to *irduration* **MUST** be at least the lower of *commit* and the information rate for IP Packets in that Bandwidth Flow that pass the metering point over that time interval.
- [R86] IP Packets in a Bandwidth Flow **MUST** be declared non-conformant in order to ensure that the information rate for such packets over any time interval equal to *irduration* that are declared conformant is, at most, *limit*.
- [O3] If the information rate for IP Packets in a Bandwidth Flow that pass the metering point over any time interval equal to *irduration* is greater than *commit*, then IP Packets in the Bandwidth Flow **MAY** be declared non-conformant in order to ensure that the information rate over that time interval for IP Packets across all Bandwidth Flows at a UNI does not exceed the available capacity of the Underlay Connectivity Services at that UNI.
- [O4] If, given the traffic received for various Bandwidth Flows with various Policies applied, the traffic that needs to be forwarded over a given Underlay Connectivity Service over any time interval equal to *irduration* exceeds the available capacity of that Underlay Connectivity Service, IP Packets in the affected Bandwidth Flows **MAY** be declared non-conformant in Bandwidth Flows where the information rate for IP Packets in the Bandwidth Flow that pass the metering point over that time interval is greater than *commit*.

The requirements above ([R85], [R86], [O3], and [O4]) describe how packets are declared conformant to the BANDWIDTH Policy (i.e., delivered with a high probability of success) or non-conformant (subject to discard as noted in [R87]). Note that [O5] allows for shaping of the Bandwidth Flow (i.e., delaying some IP Packets in the Bandwidth Flow) prior to this determination, which may result in fewer IP Packets being declared non-conformant and therefore subject to discard than otherwise would be.

Requirement [R85] indicates that at least the commit rate needs to be declared conformant if that much traffic is presented, i.e., if the average rate is less than or equal to *commit*, all of the traffic is declared conformant. On the other hand, [R86] indicates that the Service Provider discards packets to ensure that the average rate is less than *limit*.

Requirements [O3] and [O4] deal with the case where the average rate is between *commit* and *limit*. They indicate that the Service Provider may drop packets if the average rate is in this range if that is necessary to not exceed the total amount of UCS bandwidth or to avoid congestion on any particular UCS.

Note that the above requirements specify constraints over any time interval of duration *irduration*—i.e., they require a ‘sliding window’. Constraining bandwidth using a fixed, recurring, window can have the effect of declaring bursts of traffic “green” that are twice as large as intended, as described in MEF 23.2 [22] Appendix H.2.

When the total amount of traffic received at a SASE UNI exceeds the available capacity of the associated Underlay Connectivity Services, some packets may need to be discarded, even though each individual Bandwidth Flow is operating below the maximum specified in the BANDWIDTH Criterion. The business importance (see section 10.6.4) of each Session in the Bandwidth Flows is considered when packets need to be discarded, so long as each Bandwidth Flow still gets its *commit* rate.

Similarly, even when the total amount of traffic is less than the total available capacity on the Underlay Connectivity Services, as a result of other SASE Policy Criteria affecting path selection the combination of traffic across different Sessions with different Policies might mean that the traffic that needs to be forwarded over a given Underlay Connectivity Service exceeds its capacity. As above, the business importance of the Sessions is considered when packets need to be discarded.

- [R87]** IP Packets in a Bandwidth Flow **MUST** be declared conformant unless it is declared non-conformant by a condition specified in [R86], [O3], or [O4].
- [O5]** The Service Provider **MAY** delay certain IP Packets in a given Bandwidth Flow before determining conformance in order to increase the number of IP Packets in the Bandwidth Flow that are declared conformant.

The intent of [O5] is to note that the Service Provider may include a shaping function as part of the overall rate limiting process in order to potentially increase the number of IP Packets that are declared conformant. If shaping is performed, it can have an impact on the delay and delay variation of the affected Sessions.

- [R88]** IP Packets in a Bandwidth Flow that are declared non-conformant **MUST** be discarded.

9.4.7 Rate Limiter

Rate Limiters are specified using a two-tuple, $\langle commit, limit \rangle$ where:

- *commit* – is the threshold information rate (bits per second) at or below which the SASE Service Provider commits to deliver packets in the Bandwidth Flow with high probability under all traffic conditions, i.e., regardless of the information rate for other Bandwidth Flows at this UNI, or at other UNIs in the same SASE Edge. The value for *commit* can be 0.
- *limit* – is the threshold information rate (bits per second) above which the Service Provider does not deliver IP Packets in the Bandwidth Flow under any traffic conditions, i.e., the SASE Service Provider delays or discards IP Packets in the Bandwidth Flow regardless of the bandwidth used for other Bandwidth Flows at this UNI, or at other UNIs in the same SASE Edge. The value for *limit* can be *None*.

- [R89]** In the specification of a Rate Limiter, the value of the *limit* element, if not *None*, **MUST** be greater than or equal to the value of the *commit* element.

A Rate Limiter constrains the bandwidth of a Bandwidth Flow which is the set of Sessions that are assigned to the Rate Limiter. For unnamed Rate Limiters, the Bandwidth Flow comprises exactly one Session. For named Rate Limiters, the Bandwidth Flow comprises one or more Sessions that share the bandwidth constraints of the Rate Limiter. See section 7.7.2 for an explanation of unnamed and named Rate Limiters.

Specifying a value greater than 0 for *commit* indicates that delivering at least this rate is necessary to achieve the desired or intended Bandwidth Flow behavior, whereas *commit* = 0 indicates that the Bandwidth Flow has no bandwidth requirement and operates with whatever bandwidth is available.

The intent of specifying a value other than *None* for *limit* is to indicate an upper bound on the bandwidth used by the Bandwidth Flow. The *limit* can be used, for example, to ensure that there is sufficient bandwidth for other Bandwidth Flows. Specifying *limit* = *None* means that there is no maximum imposed on the Bandwidth Flow up to the limits imposed by the SASE UNI speed (which needs to be agreed on between the Subscriber and Service Provider) and Underlay Connectivity Service bandwidth constraints.

The intended behavior is shown in the following diagram.

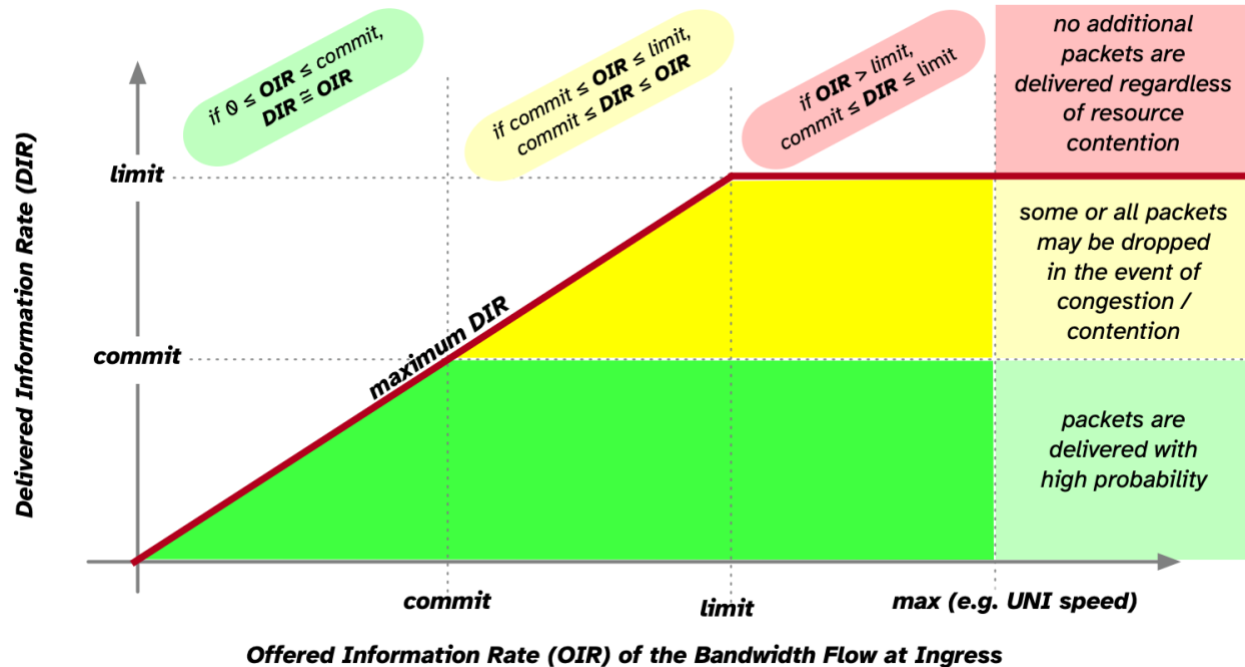


Figure 14 – Intended Behavior of a Rate Limiter

For an offered information rate from zero up to the *commit* bandwidth, the expectation is that IP Packets in the Bandwidth Flow are delivered with high probability. In the event of network congestion in the forwarding path, such packets in the Bandwidth Flow have a low probability of discard or delay. They could be discarded or delayed due to other SASE Policy Criteria (including Security Policies), but not due to bandwidth limiting.

For an offered information rate between *commit* and *limit*, IP Packets in the Bandwidth Flow are delivered with high probability unless there is contention with IP Packets in other Bandwidth Flows for the available UCS bandwidth. In this case, the Service Provider might delay or discard IP Packets in this Bandwidth Flow to reduce its impact on UCS utilization, but not such that the delivered information rate is reduced below *commit*.

For an offered information rate above *limit*, no additional packets will be delivered beyond those delivered for an offered information rate equal to *limit*.

Note that if the value of *commit* is 0, the yellow area would extend down to the horizontal axis (covering the green area) and if *limit*=None, the yellow area would extend upwards to the limit imposed by the UNI.

9.5 SASE Session Monitoring

It is important that a SASE Service identifies and monitors Sessions. Many threat Actors will attempt to find an already authenticated and authorized Session through which to infiltrate a Subscriber's network by injecting threats into the already established Session. To this end, the SASE service creates Session Monitors to evaluate the behavior of the Session. This monitoring

allows the SASE service to restrict the set of Application Flow Specifications, the Permission or Behavior of the Session, or the Target Actors to which the Sessions are allowed to flow.

SASE Session Monitoring is the logical construct that evaluates the SASE Session parameters to determine the health and validity of a given Session inside a given SASE Service.

SASE Session Monitoring allows the SASE Service Provider to evaluate the Session and trigger Session State Changes for a given SASE Session. This could include concepts like location, day of week, connectivity requirements, etc.

SASE Session Monitoring evaluates the Session Policy parameters based upon the applied Monitoring Policy.

Session Policy Parameters may include the following:

- Identity and Access Management Policy
 - Identity Risk
 - Identity Reputation
 - Roles
 - Capabilities
 - Privileges
 - Actor Access Connectivity
 - Other Identity and Access Management Policy parameters as agreed between the Service Provider and Subscriber.
- Context Policy
 - Temporal
 - Time of day
 - Day of week
 - Duration
 - Location
 - Region
 - Mobility
 - Zones
 - Other Contextual Policy parameters as agreed between the Service Provider and Subscriber.
- Security Policy
 - Threat Detection, Prevention and Remediation parameters
 - Data patterns
 - Other Security Policy parameters as agreed between the Service Provider and Subscriber.
- Session Forwarding Policy
 - Performance Metric parameters
 - UCS Parameters
 - Other Session Forwarding Policy parameters as agreed between the Service Provider and Subscriber.

9.5.1 Session State Change

Changes in the values of the Session Policy parameters, which are monitored as part of the Session Monitoring Policy, trigger a Session State Change. The Session State Value changes from *Operational* to *Re-Evaluate*. This change in Session State triggers each Policy End Point to re-evaluate, upon the next IP packet received, the SASE Policy applied to the Session. This could possibly result in a change in the Policy enforced on the Session.

[R90] For any change in the Session Policy parameters as specified in the Monitoring Policy, the Session State Value **MUST** change to the Session State Value of *Re-Evaluate*.

The Session State value *Re-evaluate* for a Session is communicated to every Policy End Point in the SASE Service. The method for how the Session State Value is communicated to every Policy End Point is beyond the scope of this document

[R91] Any Policy End Point, receiving IP packets in a Session with the Session State Value of *Re-Evaluate* **MUST** initiate a re-evaluation of the SASE Policies applied to the given Session by the SASE service.

9.6 Security Functions

A SASE Service delivers and manages Security Functions as specified by the Subscriber Policy for a specific Session. These Security Functions need to be deployable anywhere within the SASE Service to optimize the performance and security provided by the SASE Service for that Session.

The Security Functions are ‘atomic’ as they represent the basic building blocks which security Providers and vendors use to create security packages such as secure web gateway (SWG), web application firewall (WAF), or a firewall. Each of these terms is a combination of one or more of the following Security Functions.

The Security Functions needed for a SASE Service are those Security Functions adopted from MEF 138 [26] as listed in section 7.9. Additionally, this specification lists other Security Functions that are required for a SASE Service in section 9.6.2.

9.6.1 MEF 138 Security Functions

The SASE Service needs the Security Functions, as defined in MEF 138 [26], and are adapted by the SASE Service to apply to Sessions.

[R92] A SASE Service **MUST** support the following Security Functions from MEF 138 [26] to be specified in the Security Policy associated with Sessions:

- Middlebox Security Function (MBSF).
- IP, Port and Protocol Filtering (IPPF).
- DNS Protocol Filtering (DPF).
- Protective Domain Name Service (PDNS)

- Domain Name Filtering (DNF).
- URL Filtering (URLF).
- Malware Detection and Removal (MD+R).
- Data Loss Prevention (DLP)

[R93] The Security Functions in [R92] **MUST** comply with all mandatory requirements in MEF 138 [26] sections 8 and 9 as applied to a Session as the Service Flow.

[R94] A SASE Service **MUST** implement the Security Action Lists defined in MEF 138 [26] section 6.1 as applied to a Session as the Service Flow.

The parameters listed for each Security Function in MEF 138 [26] section 6 are the Policy components for the specific SASE Security Function Atomic Policy. (See section 10.5).

[R95] Each SASE Security Function, listed in [R92], **MUST** utilize the appropriate parameters for Security Functions as defined in MEF 138 [26] section 6.

9.6.2 SASE Mplify 117.1 Security Function Definitions

9.6.2.1 Supported Application Identity and Access Management (SA-IdAM) Security Function

Supported Application Identity and Access Management (SA-IdAM) is a Security Function that determines whether a Subject Actor of a Session is authenticated and authorized to access a particular supported Application.

A typical Cloud workload use case is where the Subscriber would like to control access to a particular supported Application (e.g., Office365, Salesforce, G-Suite, etc.) As an example, the SA-IdAM Security Function could block access to the data for individuals that do not have proper access within that supported Application.

SA-IdAM differs from IdAM as SA-IdAM ensures that a Subject Actor is authenticated and authorized to use a particular supported Application while IdAM authenticates and authorizes an Actor to utilize the SASE Service.

The Subscriber and Service Provider agree on a list of supported Applications which is agreed as per the List of SASE SA-IdAM Application Flow Specifications Service Attribute (see 8.21).

The SA-IdAM Security Function needs to be able to restrict certain Subject Actors from accessing the supported Applications.

A SA-IdAM Block List is a list of criteria entries used by the SA-IdAM Security Function to Block certain Subject Actors from accessing the supported Application when the Subject Actor matches one of the entries.

An example is when a Subscriber wants to Block certain Subject Actors from accessing the supported Application, the SA-IdAM Block List would include criteria entries containing roles or Subject Actor Identifiers.

A SA-IdAM Access Allow List is a list of criteria entries used by the SA-IdAM Security Function to permit Subject Actors to access a supported Application that contains a match to one of the entries.

The Subscriber referred to in the above paragraph can be informed either via a SASE Security Event Notification (SSEN) (if the reason for the Block triggers a SSEN), or via immediate communication to the client (Subject Actor) depending on the Service Policy. For notification by the SSEN, see section 9.7.2 for conditions that trigger a SSEN. How and if the Service immediately notifies the client is beyond the scope of this document.

- [R96] When a SA-IdAM Security Function is included in a Service Policy, the Service Provider **MUST** meet the mandatory requirements specified in MEF 138 [26] sections 6.1 as relating to the SA-IdAM Block List, the SA-IdAM Allow List, and the SA-IdAM Quarantine List.
- [R97] When a SA-IdAM Security Function Policy is included in a Service Policy for a given Session, the Service Provider **MUST** support both of the following actions for a subset of the Session that does not match a criteria entry on any of the SA-IdAM lists:
- Block the subset of the Session
 - Allow the subset of the Session
- [R98] When a SA-IdAM Security Function Policy is included in a Service Policy for a given Session, the SA-IdAM Security Function **MUST** perform one of the following actions for each subset of the Session that does not match a criteria entry on any of the SA-IdAM lists:
- Block the subset of the Session
 - Allow the subset of the Session

9.6.2.2 Data Integrity Security Function

The Data Integrity Security Function is the Security Function that examines Sessions to certain supported Applications and determines if the actions included in those Sessions are allowed or blocked based upon the SASE Policy.

The Data Integrity Security Function needs to be able to restrict certain Subject Actors from performing certain actions within a supported Application.

A Data Integrity Block List is a list of criteria entries used by the Data Integrity Security Function to Block certain Subject Actors from performing certain actions within a supported Application when the Subject Actor matches one of the entries.

An example is when a Subscriber wants to Block certain Subject Actors from downloading certain files or objects from a supported Application, the Data Integrity Block List would include criteria

entries containing roles or Subject Actor Identifiers, certain action identifiers, or other combinations.

A Data Integrity Allow List is a list of criteria entries used by the Data Integrity Security Function to permit Subject Actors to perform certain actions within a supported Application that contains a match to one of the entries.

The Subscriber referred to in the above paragraph can be informed either via a SASE Security Event Notification (SSEN) (if the reason for the Block triggers a SSEN), or via immediate communication to the client (Subject Actor) depending on the Service Policy. For notification by the SSEN, see Section 9.7.2 for conditions that trigger a SSEN. How and if the Service immediately notifies the client is beyond the scope of this document.

- [R99]** When a Data Integrity Security Function is included in a Service Policy, the Service Provider **MUST** meet the mandatory requirements specified in MEF 138 [26] section 6.1 as relating to the Data Integrity Block List, the Data Integrity Allow List, and the Data Integrity Quarantine List.
- [R100]** When a Data Integrity Security Function Policy is included in a Service Policy for a given Session, the Service Provider **MUST** support both of the following actions for a subset of the Session that does not match a criteria entry on any of the Data Integrity lists:
- Block the subset of the Session
 - Allow the subset of the Session
- [R101]** When a Data Integrity Security Function Policy is included in a Service Policy for a given Session, the Data Integrity Security Function **MUST** perform one of the following actions for each subset of the Session that does not match a criteria entry on any of the Data Integrity lists:
- Block the subset of the Session
 - Allow the subset of the Session

Data Integrity Security Function needs to block or allow certain actions to safeguard the Subscriber Data in a supported Application. Table 10 shows some examples of actions in supported Applications. This is only an exemplary list, and it is not meant to be comprehensive.

Action	Summary Description
Upload	Add a file to the supported Application
Download	Copy a file from the supported Application
Read	View a file in the supported Application
Create	Make a new file in the supported Application
Delete	Remove a file from a supported Application
Modify	Edit a file in the supported Application
Copy	Duplicate a file in the supported Application
Move	Change storage location of a file in the supported Application
Execute	Execute file in the supported Application
Run	Execute file in the supported Application
Get	Retrieve information from a supported Application, often via APIs
Post	Send information to a supported Application, often via APIs

Table 10 – Examples of supported Application Actions

Since the actual actions supported by a supported Application vary, the Subscriber and Service Provider need to agree on which actions are supported by the SASE Service for a particular supported Application.

[R102] For each of the entries in the List of SASE SA-IdAM Application Flow Specifications Service Attribute, the Subscriber and Service Provider **MUST** agree on the actions supported by the Data Integrity Security Function.

9.6.2.3 Proxy

A Proxy is a virtual function that acts as an intermediary between the Subject Actor and the Target Actor. The use of a Proxy allows control over the Sessions between Actors.

One of the benefits of using a Proxy for Sessions is that the Proxy can obfuscate the real IP addresses of the Subject Actor. To the Internet, the Subject Actor would appear to be originating from the Proxy public IP address.

Another benefit is that by using a Proxy for Sessions, other Security Functions can be applied to limit access and control beyond the typical network constraints. Cloud Access Security Broker and Remote Browser Isolation are two examples that utilize a Proxy.

However, some Applications do not allow for a Proxy. For example, the HTTP Strict Transport Security (HSTS) protocol, as specified in IETF RFC 6797 [16], can run over TLS and provides a mechanism implemented at the server to expressly prevent a Proxy. Another example is IPSEC, as specified in RFC 7296 [17], that uses perfect forward secrecy for secure transport.

9.6.2.4 Cloud Access Security Broker (CASB)

As enterprise workloads move to the cloud, there is an increasing need to secure the access to those workloads and control what can be done with the data. To solve this data access and control risk for workloads deployed to the Cloud, enterprises often use Cloud Access Security Broker (CASB).

CASB is a set of three Security Functions:

- Supported Application Identity and Access Management (SA-IdAM) (see section 9.6.2.1)
- Data Integrity (see section 9.6.2.2)
- Proxy (see section 9.6.2.3)

A typical Cloud workload use case is where the Subscriber would like to control the access to the supported Application, actions (i.e., create, delete, modify, etc.) performed by the Subject Actor within the supported Application, and distribution of data (i.e., upload, download, etc.) from the supported Application (i.e., Office365, Salesforce, G-Suite, etc.) As an example, CASB could block access to the data for individuals that do not have proper access within the supported Application via SA-IdAM. In CASB, the Data Integrity Security Function would block download of certain content, or it could restrict the ability of the Subject Actor to read only the information that is available as per the Subscriber's information security Policy.

While SA-IdAM Security Function and Data Integrity Security Function can each individually deny or block access to the supported Application or actions within the supported Application, the CASB needs to use both of these decisions to determine if the Access and Action are Allowed.

If either of the two Security Functions (SA-IdAM or Data Integrity) block the Session or a subset of the Session, then the effect of the CASB is to deny the subset of the Session.

[R103] When CASB is included in a Service Policy for a given Session, CASB **MUST** perform one of the following actions, based on agreement between the Service Provider and the Subscriber:

- Allow the subset of the Session that matches a criteria entry that is on both the SA-IdAM Allow List and the Data Integrity Allow List
- Allow the subset of the Session that does not match a criteria entry on any of the SA-IdAM lists, per the second bullet of [R98], and on the Data Integrity Allow List
- Allow the subset of the Session that does not match a criteria entry on any of the Data Integrity lists, per the second bullet of [R101], and on the SA-IdAM Allow List
- Block the Session that matches a criteria entry on the SA-IdAM Block List or Data Integrity Block List
- Block the subset of the Session that matches a criteria entry on the SA-IdAM Block List or the Data Integrity Block List and Allow the remainder of the Session
- Block the subset of the Session that matches a criteria entry on the SA-IdAM Quarantine List or the Data Integrity Quarantine List and Allow the remainder of the Session
- Block the subset of the Session that does not match a criteria entry on any of the SA-IdAM lists or the Data Integrity lists, per the first bullet of [R98] or [R101]

The following requirements deal with the situation where a subset of a Session cannot be inspected due to the protocol not allowing for a Proxy between the client and the server. In this case, CASB can either Block those Sessions or it can Allow those Sessions.

[R104] When CASB is included in a Service Policy for a given Session, the Service Provider **MUST** support both of the following actions for a subset of the Session that cannot be inspected:

- Block the subset of the Session
- Allow the subset of the Session

The Subscriber and Service Provider need to agree on the behavior for such cases.

[R105] When CASB is included in a Service Policy for a given Session, CASB **MUST** perform one of the following actions for a subset of the Session that cannot be inspected:

- Block the subset of the Session
- Allow the subset of the Session

9.6.2.5 Remote Browser Isolation (RBI)

Often times web traffic includes imbedded code that makes even the viewing of this traffic dangerous for the Subject Actor. To eliminate this security threat vector, Remote Browser Isolation (RBI) is used to remove malicious content and provide a clean web browsing experience to the Subject Actor.

RBI provides two benefits. First, the malicious content never reaches the Subject Actor and is executed in a secure sandbox location. Secondly, the Subject Actor is obscured from the Target Actor. This provides for secure connectivity without revealing any sensitive or compromising data between the Actors.

Remote Browser Isolation (RBI) is a set of security functions that includes:

- Web Proxy
- Security Functions

The web traffic (Figure 15 step 1) from a certain set of Subject Actors, as identified in the List of SASE RBI Actors Service Attribute, is redirected by the SASE Edge to RBI As seen in Figure 15 step 2. RBI inspects the web traffic and interacts with the appropriate Security Functions (Figure 15 step 3). The traffic, if permitted by Policy is forwarded on to the Target Actor (Figure 15 step 4). Return traffic from the Target Actor would be inspected and permitted by policy in reverse order of steps.

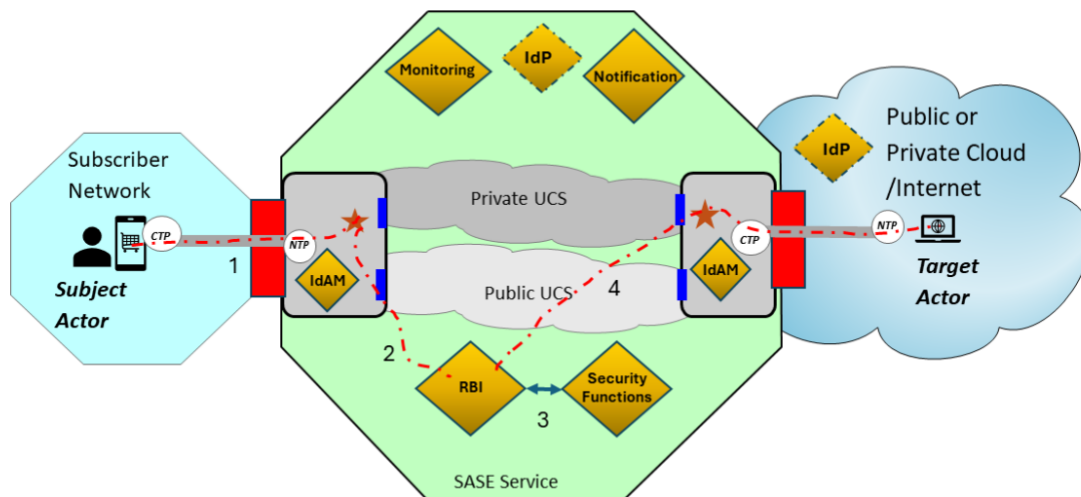


Figure 15 – Example of Remote Browser Isolation

There are four different models of RBI currently deployed in the industry.

- **Pixel Pushing Model:** Web content is rendered and processed on a remote server which captures the visual representation of the page and sends it to the users' device pixel-by-pixel. This method provides fully air-gapped security but is bandwidth-heavy, driving up costs and degrading the user experience.
- **Document Object Model:** A Document Object Model (DOM) approach reconstructs the page with HTML elements. It filters and processes web content before rebuilding the page on the user's device. This approach responds better than pixel-pushing but exposes the user to potentially malicious files and zero-day threats.
- **SKIA Graphics Model:** An open-source 2D graphics library renders the webpage through a SKIA Graphics Model rendering layer to address many of the gaps with pixel and DOM-based approaches. However, the SKIA Graphics Model has limited versatility for low-latency, highly dynamic, and interactive real-time applications.
- **Isolation Platform Model:** Unlike traditional pixel-pushing reconstruction that causes high latency or DOM-based approaches that expose security gaps, a next-generation RBI Isolation Platform Model combines the latest vector and pixel-based technologies for superior isolation while simultaneously delivering near-native UX.

[R106] RBI **MUST** support at least one of the following models:

- Pixel Pushing Model
- Document Object Model
- SKIA Graphics Model
- Isolation Platform Model

Remote Browser Isolation often is utilized in conjunction with many other Security Functions, such as Malware Detection and Removal or URL Filtering. However, these other Security Functions are distinct from RBI and would be defined separately within the SASE Security Policy as defined by the Subscriber.

9.7 SASE Service Notifications

The following two subsections include the SASE Authentication or Authorization Notification and the SASE Security Event Notification.

9.7.1 SASE Authentication or Authorization Notification (SAAN)

A SASE Authentication or Authorization Notification (SAAN) is a communication of an Authentication or Authorization event, i.e., a SAAN is issued when an Actor has been denied access to the SASE Service due to an Authentication or Authorization failure or when a Session is Blocked by the SASE Policy. The SAAN is sent to the Subscriber.

[R107] A SAAN **MUST** be issued whenever an Actor is denied access to the SASE Service due to an Authentication or Authorization failure.

[R108] A SAAN **MUST** be issued whenever a Session is Blocked by a SASE Policy.

[R109] The Service Provider **MUST** store each SAAN in a secure repository for future reference and auditing purposes.

The amount of time that a SAAN needs to be stored is agreed between the Subscriber and Service Provider, e.g., it could be in accordance with the Subscriber’s data retention policy or to comply with any applicable regulations.

[R110] A SAAN **MUST** include the items listed in Table 11.

Item	Value	Comments
Issuer	UTF-8 [12] String ⁵	Examples: SASE Service Provider Name
Timestamp of SAAN	Date-time	Example: UTC [10]
SAAN ID	UTF-8 [12] String	Example: Universally Unique Identifier [14]
Source IP Address	Human readable IPv4 dotted decimal IPv6 hexadecimal strings	Example: IP address of Subject Actor
SASE UNI ID	UTF-8 [12] String	Example: Universally Unique Identifier [14]
Policy ID	UTF-8 [12] String	Example: Policy Name
Type of SAAN	UTF-8 [12] String	Examples: Actor Authentication failure, Actor Authorization failure, Session Policy Failure
Authentication/Authorization Failure Details	UTF-8 [12] String	Examples: Username, Actor ID, Source/Destination IP address, Source Media Access Control (MAC) Address, Session ID, Source/Destination port number.

Table 11 – Items to be included in a SAAN

The format of the SAAN is not specified in this document.

9.7.2 SASE Security Event Notification (SSEN)

A SASE Security Event Notification (SSEN) is a communication of a security event, i.e., a SSEN is issued when a subset of a Session is Blocked or modified by a Security Function. The SSEN is sent to the Subscriber.

[R111] A SSEN **MUST** be issued whenever a subset of the Session is Blocked or modified by a Security Function.

⁵ UTF-8, Unicode Transformation Format 8-bit

An example of a subset of a Session that has been modified is the removal of an infected file attachment in an e-mail. Typically, the removed file attachment would be replaced with a message stating that the file was infected and removed.

[R112] The Service Provider **MUST** store each SSEN in a secure repository for future reference and security auditing purposes.

The amount of time that a SSEN needs to be stored is agreed between the Subscriber and Service Provider, e.g., it could be in accordance with the Subscriber’s data retention policy or to comply with any applicable regulations.

[R113] A SSEN **MUST** include the items listed in Table 12.

Item	Value	Comments
Issuer	UTF-8 [12] String	Examples: SASE Service Provider Name
Timestamp of IOC	Date-time	Example: UTC [10]
SSEN ID	UTF-8 [12] String	Example: Universally Unique Identifier [14]
Source IP Address	Human readable IPv4 dotted decimal IPv6 hexadecimal strings	Example: IP address of Subject Actor
SASE UNI ID	UTF-8 [12] String	
Policy ID	UTF-8 [12] String	Example: Security Policy name
IOC Type	UTF-8 [12] String	Examples: CVE [22], STIX [34], CWE [35], CAPEC [33], ATT&CK [32], RFC 7970 [15].
IOC Information ID	UTF-8 [12] String	Identifies the IOC based on type
IOC Source	URL for IOC Type	Example: CVE [22]
Type of Compromise	UTF-8 [12] String	Examples: Known vulnerability, breach, data leakage, abuse of resources, jacking, where to find more information on breach.
Compromise Details	UTF-8 [12] String	Examples: Username, Source/Destination IP address, Source Media Access Control (MAC) Address, neutralized URL, neutralized domain name, Malware, Source/Destination port number, or anomalous behavior.
Action Taken	UTF-8 [12] String	Examples: informational, quarantined, Blocked, or Malware removed.

Table 12 – Items to be included in a SSEN

The format of the SSEN is not specified in this document.

This document mandates that the URLs and domain names listed in the SSEN be neutralized. It also recommends the use of square brackets, which are reserved characters in RFC 3986 [13], to

neutralize a domain name or URL in a SSEN. For example, if the compromised detail includes www.domain.tld, the SSEN would send it as www[.]domain[.]tld.

[R114] Any domain name or URL in a SSEN **MUST** be neutralized.

[D11] The method for neutralizing the domain name or URL in a SSEN **SHOULD** use square brackets around each period.

10 Policies

A SASE Service uses Policies to determine the behavior of Sessions. SASE Policy concept is described in sections 10.1 through 10.9.

10.1 SASE Policy

A SASE Policy is a Composite Policy, as defined by MEF 95.0.1 [26] Amended PDO model, utilized in a SASE Service and applied to a given Session.

[R115] A SASE Policy **MUST** be a Composite Policy.

[R116] A SASE Policy **MUST** comply with the mandatory requirements in MEF 95.0.1 [26].

Since a SASE Service incorporates a Zero Trust Framework, the Policies within a SASE Service need to comply with the requirements as defined in MEF 118.1 [28].

[R117] A SASE Policy **MUST** comply with the mandatory requirements in MEF 118.1 [28] sections 13 and 14.

[R118] A SASE Policy **MUST** include exactly one of each the following Policies:

- Identity and Access Management Policy (IdAMP).
- Context Policy.
- Security Policy.
- Session Forwarding Policy.
- Monitoring Policy.
- Notification Policy.

Each Policy within a SASE Policy (e.g., Security Policy or Identity and Access Management Policy) is defined independently and then referenced by the SASE Policy. (See section 8 for the Service Attribute Lists of Policies)

Unless specifically stated within this document, whether these independently defined Policies are Composite Policies or Atomic Policies is beyond the scope of this document.

The Atomic and Composite Policy structure allows for a method to re-use Policies. This allows Atomic Policies to be utilized in many different Composite Policies. This leads to a reduction in the need to redefine the same values across multiple Policies.

A SASE Policy provides details on how Ingress and Egress IP Packets associated with each Session should be handled by the SASE Service, providing rules concerning security, performance, forwarding and others.

A SASE Policy might include other Policies not defined in this document.

The SASE Policy is referenced by the SASE Service and applied to a given SASE Session. Therefore, it is mandated that a SASE Policy have a unique identifier.

[R119] A SASE Policy **MUST** have a unique identifier.

In addition, each individual Policy that makes up a SASE Policy also has a unique identifier. Requirements for this uniqueness can be found in each individual Policy section below. Any SASE Edge needs to be able to use any SASE Policy from the List of SASE Policies Service Attribute. The method for distributing these SASE Policies to the SASE Edge is beyond the scope of this document.

10.2 Policy Execution Order Parameter

In a SASE Service, the order in which the Policies within the SASE Policy needs to be properly identified to assure the appropriate execution of the Policies. Therefore, a Policy within the SASE Policy is assigned a Policy Execution Order Parameter. The Policy Execution Order Parameter is a non-negative integer whose value defines the desired order of execution of the Policy where the highest Policy Execution Order Parameter value is processed first.

[R120] A Policy Execution Order Parameter value **MUST** be a value within the range of values as indicated in the SASE Policy Execution Order Parameter Range Service Attribute.

[R121] Each Policy within a SASE Policy **MUST** have a Policy Execution Order Parameter value.

10.3 Identity and Access Management Policy

Identity and Access Management Policy (IdAMP) is the set of criteria needed to properly authenticate the identity of a given Actor and authorize that Actor to utilize the SASE Service. The Identity and Access Management Policy is also responsible for enforcing the Actor Access Connection.

Identity and Access Management Policy is different than Supported Application Identity and Access Management (SA-IdAM) Policy as IdAM Policy authenticates and authorizes an Actor to utilize the SASE Service. SA-IdAM Policy ensures that a Subject Actor is authenticated and authorized to use a particular set of supported Applications.

Since the Identity and Access Management Policy will be utilized in many SASE Policies, the IdAMP needs a unique identifier

[R122] The IdAMP **MUST** have a unique identifier.

As the name implies, the Identity and Access Management Policy is composed of three functions:

- Actor Authentication Function – Authenticating the Identity of a given Actor.
- Access Authorization Function – Authorizing a given Actor to utilize the SASE Service.

- Actor Access Connection – assuring secure Actor Access to the SASE service.

The IdAMP is executed first in a SASE Service. This allows only Actors who have been authenticated, authorized, and with proper Actor Access Connection to access the SASE Service. Therefore, the Policy Execution Order Parameter of the IdAMP needs to be the highest possible value.

[R123] The Policy Execution Order Parameter for IdAMP **MUST** be the highest possible value (highest priority) for the SASE Service.

The order of execution for the other Policies within the SASE Policy is at the discretion of the SASE policy as built by the Subscriber and implemented by the Service Provider.

10.3.1 Actor Authentication Function

The Actor Authentication Function is the set of necessary criteria to properly identify the Actor. The Actor Authentication Function utilizes an Identity Provider to validate the identity of a given Actor.

[R124] The IdP within an IdAMP **MUST** be a value from the List of Identity Providers Service Attribute.

The Actor Authentication Function utilizes the parameters of the IdP to send the appropriate credentials to the IdP and then utilize the IdP response to determine if the Actor's identity has been validated

Examples of Actor Authentication Function parameters could include options such as credentials/one-time passwords, multi-factor authentication, password complexity, certificates, and others.

[R125] An IdAMP **MUST** include an Actor Authentication Function.

10.3.2 Actor Access Authorization Function

The Actor Access Authorization Function utilizes parameters returned by the IdP and the satisfaction of the Actor Access Connection parameters to determine to grant Actor Access to the SASE Service.

The IdP parameters utilized by the Actor Access Authorization Function include, but are not limited to:

- Identity Risk – the risk associated with a given Actor.
- Identity Reputation – the reputation associated with a given Actor.
- Authentication Method.
- Roles – This indicates the list of provided roles for a given Actor.
- Capabilities – This indicates the set of inherent or advertised capabilities of a given Actor.
- Privileges – this indicates the set of authorized capabilities of a given Actor.

The inclusion of these parameters in the IdAMP depends on the Subscriber's SASE Policies. Which parameters are available for SASE Policies is an agreement between the Service Provider and Subscriber.

[R126] An IdAMP **MUST** include an Actor Access Authorization Function.

This Actor Access Authorization Function only determines if the Actor is permitted to utilize a SASE Service. The SASE Policies authorize the individual Session to perform the particular action requested by the Subject Actor on the Target Actor.

10.3.3 Actor Access Connection

Since the Subject Actor Access Connection and Target Actor Access Connection carry Subscriber traffic and are outside the SASE Service Provider Domain, they may be considered vulnerable. Therefore, the Subscriber can set Policies within the SASE Service that specify the type of encryption (e.g., None; TLS 1.2 [15]; IPSEC) of traffic accepted from or transmitted to the Actor.

For the Subject Actor (which utilizes the SASE Service) the Subscriber fully controls the Actor Access Connection. If the IdAMP does not permit the Actor Access, then the Subject Actor does not gain access to the SASE Service.

[R127] The SASE Service **MUST** secure the Subject Actor Access Connection.

Since the Target Actor, in many cases, is outside of the direct control of the Subscriber or Service Provider, the SASE Service cannot mandate a secure connection. The Subscriber can only control whether to allow access to a Target Actor which refuses to use encrypted connections.

[D12] The SASE Service **SHOULD** secure the Target Actor Access Connection.

[R128] The SASE Service **MUST** support the use of TLS 1.2 [15] or greater to secure the Actor Access Connection.

[D13] The SASE Service **SHOULD** support the use of IPSEC to secure the Actor Access Connection.

The SASE Service Provider may also default to a specific encryption type on Actor Access Connections where the Subscriber SASE Policy does not specify explicitly an encryption type for a given Session between Subject Actor and Target Actor.

[R129] The SASE Service **MUST** meet the mandatory requirements of TLS 1.2, per RFC 5246 [15].

[R130] The SASE Service **MUST** meet the mandatory requirements of RFC 8446 [20] section 9.3 (Protocol Invariants).

[R131] The TLS version utilized by a SASE Service **MUST** be a value of the List of SASE Supported TLS Versions Service Attribute.

[R132] The cipher suites utilized by a SASE Service **MUST** be a value of the List of SASE Supported Cipher Suites Service Attribute.

[CR1]<[D13] The SASE Service Provider **MUST** provide to the Subscriber a list of all IPSEC security options supported by the SASE Service.

[CR2]<[D13] The IPSEC security options utilized by a SASE Service **MUST** be a value of the List of SASE Supported IPSEC Security Options Service Attribute.

In many instances, the Subscriber may wish to establish a Policy for what sort of encryption and cipher suites will be allowed for specific SASE Sessions. Therefore, the SASE Service Provider needs to utilize Policy to enforce the Subject and Target Actor Access Connections.

[R133] An IdAMP **MUST** include an Actor Access Connectivity Policy.

The exact implementation details of what connection type, encapsulation mechanisms and encryption mechanisms (not previously dictated by this document) are beyond the scope of this document.

10.4 Context Policy

Context Policy is a set of criteria that influence the authorization of a given Session within a SASE Service. The Context Policy influences whether the individual Session is authorized to proceed.

[R134] The Context Policy **MUST** have a unique identifier.

[R135] The Policy Execution Order Parameter value for a Context Policy **MUST** be lower than the IdAMP Policy Execution Order Parameter value.

The Context criteria can be defined as the following:

- Temporal
 - Time of day
 - Day of week
 - Duration
- Location
 - Zones
 - Mobility
- Other Context criteria as agreed by the Service Provider and Subscriber.

[R136] A Context Policy **MUST** include a set of Temporal criteria.

Temporal criteria may be the time of day, the duration of a Session, or even the day of the week. Perhaps certain information is unavailable on the weekends or certain Applications are restricted during business hours. Or perhaps, the Subscriber wishes to control how long a Session is permitted before timing out.

[R137] A Context Policy **MUST** include a set of location criteria.

A Session that accesses, utilizes or transports data may be constrained due to governmental, industrial or Subscriber regulations. As a result, the location of the data needs to be part of the Context Policy applied to a given Session.

While there are many methods to establish location (such as geo-location, network location, IP location, etc.), the actual method for determining location is beyond the scope of this document.

Mobility indicates if an Actor (either Subject or Target) associated with a given Session is permitted to change location.

It is not expected that a traffic camera attached to a traffic light at a given location would move. However, a dashboard camera on a fire engine would be expected to move. In both cases, the Subject Actors are cameras. They may be classified in the same Application Flow Specification as Video, but only one of the Sessions would be expected to allow Mobility.

10.5 Security Policy

A Security Policy is a Composite Policy consisting of an Atomic Policy for each Security Function needed by the SASE Policy for a given Session.

[R138] The Security Policy **MUST** have a unique identifier.

[R139] The Policy Execution Order Parameter value of a Security Policy **MUST** be lower than the IdAMP Policy Execution Order Parameter value.

Security Policy contains one or more Atomic Policies for Security Functions from the List of SASE Security Functions Service Attribute.

[R140] The Security Policy **MUST** contain at least one Atomic Policy associated with a value from the List of SASE Security Functions Service Attribute (see Section 8.14).

The actual implementation details of where the specific Security Functions are placed within the Service, the method for Service Chaining multiple Security Functions, and the particular operation of a Security Function, is beyond the scope of this document.

10.6 Session Forwarding Policy

A Session Forwarding Policy is a set of criteria that determines the forwarding and performance requirements that influence how a given Session traverses the SASE Service. The Session Forwarding Policy dictates which Underlay Connectivity Service a given Session follows from one SASE Edge to the next SASE Edge within a SASE Service.

[R141] The Session Forwarding Policy **MUST** have a unique identifier.

- [R142] The Policy Execution Order Parameter value of a Session Forwarding Policy **MUST** be lower than the Policy Execution Order Parameter value assigned to the IdAMP Policy.

10.6.1 ENCRYPTION Criterion

Under most circumstances, the SASE Service requires encryption of the Sessions when forwarding the traffic. However, the Subscriber might not want to encrypt traffic that traverses a UCS that already provides its own encryption or is a private UCS that is considered trusted and secure. Sessions that traverse such UCSs might not be encrypted. However, encryption of sessions which traverse a public UCS need to be encrypted.

- [R143] A Session Forwarding Policy **MUST** include the ENCRYPTION Criterion.

10.6.2 PUBLIC-PRIVATE Criterion

A SASE Service can use *private* Underlay Connectivity Services such as Mplify Carrier Ethernet Services and *public* Underlay Connectivity Services, i.e., Internet Access Services. The PUBLIC-PRIVATE Criterion provides control over whether or not a Session can traverse a public Underlay Connectivity Service, i.e., the Internet. It can have the value *Private-Only* or *Either*.

- [D14] A Session Forwarding Policy **SHOULD** include the PUBLIC-PRIVATE Criterion.

10.6.3 BILLING-METHOD Criterion

The BILLING-METHOD Criterion (see section 9.4.3) provides control over the charge type of the network that can be used to forward a Session. It can have the value *Flat-Rate-Only*, *Usage-Based-Only*, or *Either*.

- [D15] A Session Forwarding Policy **SHOULD** include the BILLING-METHOD Criterion.

10.6.4 BUSINESS-IMPORTANCE Criterion

The BUSINESS-IMPORTANCE Criterion (see section 9.4.4) indicates the relative business importance of a Session compared with other Sessions in the SASE Service.

- [D16] A Session Forwarding Policy **SHOULD** include the BUSINESS-IMPORTANCE Criterion.

10.6.5 PERFORMANCE Criterion

One of the benefits of SASE is that, assuming that there are multiple ways of reaching a destination, the SASE Service can dynamically choose a path that best meets the Policy applied to a Session, and this includes Policy Criteria associated with performance.

A SASE service can monitor the performance of the various paths between SASE Edges in real time and adjust the forwarding decisions based on the most recently measured performance. This

document does not specify how or when a SASE implementation measures performance or even that it does measure it; however, there is an expectation that the Service Provider implements the PERFORMANCE Criterion (see section 9.4.5) in a Session Forwarding Policy.

[R144] A Session Forwarding Policy **MUST** include the PERFORMANCE Criterion.

10.6.6 BANDWIDTH Criterion

The BANDWIDTH Criterion provides a method to express the intended bandwidth requirements for a Session, and the probability of packet delay or discard in the face of varying bandwidth contention for Underlay Connectivity Service resources.

The effect of applying the BANDWIDTH Criterion (see section 9.4.6) to a Session is to declare IP Packets in the Session either conformant or non-conformant. This can also include delaying IP Packets in order to improve conformance, i.e., traffic shaping. Non-conformant IP Packets in the Session are discarded in order to meet the Policy.

[D17] A Session Forwarding Policy **SHOULD** include the BANDWIDTH Criterion.

10.7 Monitoring Policy

SASE Monitoring Policy is a set of criteria for continually evaluating the SASE Policy parameters as to changes in those parameters, triggering Session State Changes, and, where appropriate, subsequent SASE Policy selection for a given Session as it traverses the SASE Service. The Monitoring Policy dictates the time frames for evaluation of changes to SASE Policy parameters, which SASE Policy parameters will trigger State Changes and when the SASE Service must initiate re-evaluation of the SASE Policy, for a given Session.

[R145] The Monitoring Policy **MUST** have a unique identifier.

[R146] The Policy Execution Order Parameter value of a Monitoring Policy **MUST** be lower than the IdAMP Policy Execution Order Parameter value.

The actual implementation details of what is contained in the Monitoring Policy is beyond the scope of this document.

10.8 Notification Policy

Notification Policy is a set of criteria for sending Notifications to the Subscriber about events (i.e., some examples) within the SASE service. These Notifications might contain changes to Policy, changes to Security Functions, reports of issues with the SASE Service, or information about Sessions which were blocked as part of a SASE Policy.

[R147] The Notification Policy **MUST** have a unique identifier.

[R148] The Policy Execution Order Parameter value of a Notification Policy **MUST** be lower than the IdAMP Policy Execution Order Parameter value.

[R149] A Notification Policy **MUST** include criteria for sending a SAAN.

[R150] A Notification Policy **MUST** include criteria for sending a SSEN.

The Notification Policy needs include the recipients to receive this Notification. However, the Subscriber may not mandate that a Notification be sent in some cases. In this case, the recipient value of *None* exists. The value of *None* indicates that no notification is needed for this SASE Policy.

[R151] Recipient value of a Notification Policy **MUST** be *None* or a value from the List of SASE Notifications Recipients Service Attribute.

[D18] The Service Provider **SHOULD** support the ability to configure different Recipient value for each Notification Policy criteria.

[D19] The Service Provider **SHOULD** support the ability to configure a different Timestamp for each Notification Policy criteria.

10.9 SASE Edge Policy Map

The SASE Edge Policy Map specifies the SASE Policies assigned to Sessions at Policy End Points. The value of the SASE Edge Policy Map is a non-empty list of 2-tuples of form (*SessionID*, *AssignedPol*) where:

- *SessionID* is the unique identifier for a Session ([R51]).
- *AssignedPol* is a 2-tuple of form (*SASEpolName*, *timestamp*).

where:

- *SASEpolName* is a SASE Policy identifier as defined in section 10.1.
- *timestamp* is the time when the SASE Policy was assigned to the Session.

11 SASE Considerations when SD-WAN is utilized for a SASE Service

As per Requirement [D7], an SD-WAN Service, as defined in MEF 70.2 [24], is recommended for the connectivity between SASE Edges. This section describes conditional requirements when SD-WAN is used for a SASE Service and the forwarding mechanism between SASE Edges. This does not extend to any part of the SD-WAN that exists outside of the SASE Service.

While Security Functions can be implemented in both SD-WAN and SASE, the Service Provider needs to ensure that the Security Functions in a SASE Policy for a given Session get implemented appropriately.

- [R152] If Security Functions are implemented in two or more different services (i.e., SASE and SD-WAN), the SASE Service Provider **MUST** implement such Security Functions in accordance with the Subscriber SASE Policy.

11.1 SD-WAN AF-SECURITY-INGRESS Ingress Policy Criterion

- [CR3]<[D7] If SD-WAN is not implementing Security Functions, the value of the AF-SECURITY-INGRESS Ingress Policy Criterion **MUST** be *None*.

Requirement [CR3]<[D7] indicates that the security for SASE is done within the SASE Service and that the SD-WAN Service is not performing any Security Functions.

11.2 SD-WAN AF-SECURITY-EGRESS Egress Policy Criterion

- [CR4]<[D7] If SD-WAN is not implementing Security Functions, the value of the AF-SECURITY-EGRESS Egress Policy Criterion **MUST** be *None*.

Requirement [CR4]<[D7] indicates that the security for SASE is done within the SASE Service and that the SD-WAN Service is not performing any Security Functions.

11.3 SD-WAN EGRESS-BLOCK Egress Policy Criteria

- [CR5]<[D7] The value of the EGRESS-BLOCK Egress Policy Criterion **MUST** be *Allow*.

Requirement [CR5]<[D7] indicates that the Sessions are not dropped due to an SD-WAN EGRESS-BLOCK Egress Policy Criterion.

11.4 SD-WAN ENCRYPTION Ingress Policy Criteria

- [CR6]<[D7] The value of the ENCRYPTION Ingress Policy Criterion **MUST** be either *Required-Always* or *Required-Public-Only*.

Requirement [CR6]<[D7] indicates that the values for ENCRYPTION in SD-WAN are restricted to only the values that are supported in the SASE Service.

11.5 VIRTUAL-TOPOLOGY Ingress Policy Criterion

[CR7]<[D7] The VIRTUAL-TOPOLOGY Ingress Policy Criterion **MUST NOT** be included in an SD-WAN Policy.

Requirement [CR7]<[D7] indicates that since Virtual Topologies are not supported in a SASE Service, an SD-WAN Service cannot utilize the VIRTUAL-TOPOLOGY Ingress Policy Criterion.

When SD-WAN is the connectivity between SASE Edges, there are multiple Sessions that pass through a SASE Edge and Virtual Topologies for SASE has yet to be defined. The SD-WAN topology needs to be a full mesh so all SASE Edges can send traffic to all other SASE Edges.

11.6 INTERNET-BREAKOUT Ingress Policy Criterion

[CR8]<[D7] The INTERNET-BREAKOUT Ingress Policy Criterion **MUST NOT** be included in an SD-WAN Policy.

Requirement [CR8]<[D7] indicates that since internet breakout is not supported in a SASE Service, an SD-WAN Service cannot utilize the INTERNET-BREAKOUT Ingress Policy Criterion.

A SASE Service is a UNI to UNI Service. INTERNET-BREAKOUT in SD-WAN would allow the Session to enter in a SASE UNI but exit an SD-WAN UCS (never having crossed the Egress SASE UNI); thus, violating the SASE Service definition.

11.7 ALLOWED-DESTINATION-ZONES Ingress Policy Criterion

[CR9]<[D7] The ALLOWED-DESTINATION-ZONES Ingress Policy Criterion **MUST NOT** be included in an SD-WAN Policy.

Requirement [CR9]<[D7] indicates that since zones are not supported in a SASE Service, an SD-WAN Service cannot utilize the ALLOWED-DESTINATION-ZONES Ingress Policy Criterion.

11.8 BACKUP Ingress Policy Criterion

[CR10]<[D7] The BACKUP Ingress Policy Criterion **MUST NOT** be included in an SD-WAN Policy.

Requirement [CR10]<[D7] indicates that since the concept of a backup UCS is not supported in a SASE Service, an SD-WAN Service cannot utilize the BACKUP Ingress Policy Criterion.

11.9 Application Flow Specifications

[CR11]<[D7] The values in the SASE Service List of Application Flow Specifications Service Attribute **MUST** be a subset of the values in the SD-WAN SWVC List of Application Flow Specifications Service Attribute.

Requirement [CR11]<[D7] indicates that when an SD-WAN Service is utilized by the SASE Service, the Forwarding Policies in both SD-WAN and SASE need to be synchronized so the proper handling of the IP packet can be assured. Therefore, the Session Forwarding Policy in a SASE Service needs to comply with the Application Flow Policies in an SD-WAN Service.

11.10 PUBLIC-PRIVATE Ingress Policy Criterion

[CR12]<[D7] The SD-WAN PUBLIC-PRIVATE Ingress Policy Criterion values **MUST** be the same as the values of the SASE PUBLIC-PRIVATE Criterion.

Requirement [CR12]<[D7] indicates that since SASE and SD-WAN both utilize the same set of UCSs, the values for the UCS Type Service Attributes need to be the same in both a SASE Service and an SD-WAN Service.

11.11 BILLING-METHOD Ingress Policy Criterion

[CR13]<[D7] The SD-WAN BILLING-METHOD Ingress Policy Criterion values **MUST** be the same as the values of the SASE BILLING-METHOD Criterion.

Requirement [CR13]<[D7] indicates that since SASE and SD-WAN both utilize the same set of UCSs, the values for the UCS BILLING-METHOD Service Attributes need to be the same in both a SASE Service and an SD-WAN Service.

11.12 PERFORMANCE Ingress Policy Criterion

[CR14]<[D7] The SD-WAN PERFORMANCE Ingress Policy Criterion for a given Application Flow **MUST** be consistent with the SASE PERFORMANCE Criterion for the SASE Session which is included in that Application Flow.

Requirement [CR14]<[D7] indicates that since SASE and SD-WAN both monitor the performance of Application Flow Specifications (via the Application Flow for SD-WAN and the Session for SASE), the PERFORMANCE Service Attributes need to be consistent in both a SASE Service and an SD-WAN Service for the same AFS used for SD-WAN Applications flows and the SASE Sessions. The SD-WAN Performance Service Attributes need to be defined in a manner that prevents the SD-WAN Service from impacting packets whose performance would be acceptable in the SASE Service. Conversely, the SD-WAN Service Performance Service Attribute need to be defined in a manner that does not permit path selection for Application Flows which contain SASE Sessions which do not meet the possibly more stringent SASE Service Performance metrics.

The PERFORMANCE Service Attributes in SD-WAN apply to an Application Flow. This SD-WAN Application Flow does not distinguish between the Actors that sends/receives the traffic. SASE PERFORMANCE Service Attributes apply to the Session, and, as such, apply to both the Application Flow Specification and the ActorPair. It would be detrimental if the SD-WAN Performance Service Attributes defined the acceptable loss for the Application Flow *Zoom* as 50%

while the SASE PERFORMANCE Service Attributes for the Sessions (AFS=Zoom, ActorPair=CEO,any) defined the acceptable loss as only 5%.

11.13 BANDWIDTH Ingress Policy Criterion

[CR15]<[D7] The SASE BANDWIDTH Criterion **MUST** be consistent with the SD-WAN BANDWIDTH Ingress Policy Criterion for a given Application Flow Specification.

Requirement [CR15]<[D7] indicates that the SASE Service BANDWIDTH Criterion needs to be consistent with the SD-WAN BANDWIDTH Ingress Policy Criterion. This is needed to assure that the SD-WAN BANDWIDTH can be handled by the SASE BANDWIDTH for the same AFS used for SD-WAN Applications flows and the SASE Sessions.

11.14 SWVC List of Application Flow Specification Service Attribute

When a SASE Service utilizes an SD-WAN Service, as defined in MEF 70.2 [2], the SASE List of Application Flow Specifications Service Attribute needs to be a subset of the SD-WAN SWVC List of Application Flow Specifications Service Attribute so proper IP packet classification can happen in both the SASE Service and the SD-WAN Service and appropriate Subscriber Policy intent can be realized.

[CR16]<[D7] The values in the SASE List of SASE Application Flow Specifications Service Attribute **MUST** be a subset of the Application Flows in the SD-WAN SWVC List of Application Flow Specifications Service Attribute.

[CR17]<[D7] The *AFGroup* value in all entries in the SWVC List of Application Flow Specification Service Attribute **MUST** be *None*.

Requirement [CR17]<[D7] indicates that the SD-WAN Service needs to assign an *AFGroup* value of *None* to all Application Flows that result from SASE Sessions.

11.15 BUSINESS-IMPORTANCE Ingress Policy Criterion

[CR18]<[D7] The SD-WAN BUSINESS-IMPORTANCE Ingress Policy Criterion **MUST** be consistent with the SASE BUSINESS-IMPORTANCE Criterion for SASE Sessions that result in SD-WAN Application Flows.

Requirement [CR18]<[D7] indicates that since SASE Sessions have shared Application Flow Specifications with SD-WAN, the BUSINESS-IMPORTANCE associated with a SASE Session needs to correspond to the same BUSINESS-IMPORTANCE associated with the resulting SD-WAN Application Flow.

In this scenario, it would be detrimental if the SASE BUSINESS-IMPORTANCE for the Session defined by AFS=Zoom and ActorPair= CEO,any, was set to High and the SD-WAN BUSINESS-IMPORTANCE for AFS=Zoom was set to low. This could cause the Zoom Application Flows to be dropped in SD-WAN when the CEO Zoom Sessions would be protected and prioritized in SASE.

12 References

- [1] Gartner, "Gartner's description of Secure Access Service Edge (SASE)"
<https://www.gartner.com/en/information-technology/glossary/secure-access-service-edge-sase>
- [2] Gartner, Zero Trust Network Access Reviews and Ratings,
<https://www.gartner.com/reviews/market/zero-trust-network-access>
- [3] IANA, Protocol Numbers, <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>
- [4] IANA, Service Name and Transport Protocol Port Number Registry,
<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>
- [5] IEEE Std 802.3-2018, IEEE Standard for Ethernet, August 2018
- [6] IETF RFC 791, Internet Protocol, September 1981
- [7] IETF RFC 1035, *Domain Names - Implementation and Specification*, by P. Mockapetris, November 1987.
- [8] IETF RFC 1996, *A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)*, by Paul Vixie, August 1996.
- [9] IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, by Scott Bradner, March 1997.
- [10] IETF RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, Fred Baker and David L. Black and Kathleen Nichols and Steven L. Blake, December 1998. Copyright © The Internet Society (1998). All Rights Reserved.
- [11] IETF RFC 3339, *Date and Time on the Internet: Timestamps*, by Chris Newman and Graham Klyne, July 2002. Copyright © The Internet Society (2002). All Rights Reserved.
- [12] IETF RFC 3629, *UTF-8, a transformation format of ISO 10646*, by Francois Yergeau, November 2003. Copyright © The Internet Society (2003). All Rights Reserved.
- [13] IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, by Tim Berners-Lee, Roy T. Fielding, and Larry Masinter, January 2005. Copyright © The Internet Society (2005).
- [14] IETF RFC 4122, *A Universally Unique Identifier (UUID) URN Namespace*, by Paul Leach, Michael Mealling, and Rich Salz, July 2005. Copyright © The Internet Society (2005).

- [15] IETF RFC 5246, *The Transport Layer Security (TLS) Protocol Version 1.2*, by Tim Dierks and Eric Rescorla, August 2008. Copyright © The IETF Trust (2008).
- [16] IETF RFC 6797, HTTP Strict Transport Security (HSTS), by Jeff Hodges, Collin Jackson, and Adam Barth, November 2012. Copyright © 2012 IETF Trust and the persons identified as the document authors. All rights reserved.
- [17] IETF RFC 7296, Internet Key Exchange Protocol Version 2 (IKEv2), by C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen, October 2014. Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.
- [18] IETF RFC 7970, *The Incident Object Description Exchange Format Version 2*, by Roman Danyliw, November 2016. Copyright © 2016 IETF Trust and the persons identified as the document authors. All rights reserved.
- [19] IETF RFC 8174, *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*, by Barry Leiba, May 2017. Copyright © 2017 IETF Trust and the persons identified as the document authors. All rights reserved.
- [20] IETF RFC 8200, Internet Protocol, Version 6 (IPv6) Specification, by Dr. Steve E. Deering and Bob Hinden, July 2017. Copyright © 2017 IETF Trust and the persons identified as the document authors. All Rights Reserved.
- [21] IETF RFC 8446, *The Transport Layer Security (TLS) Protocol Version 1.3*, by Eric Rescorla, August 2018. Copyright © 2018 IETF Trust and the persons identified as the document authors. All rights reserved.
- [22] MEF 23.2, Carrier Ethernet Class of Service – Phase 3, August 2016
- [23] MEF 61.1, *IP Service Attributes*, January 2019
- [24] MEF 66, *SOAM for IP Services*, Jul 2020
- [25] MEF 70.2, *SD-WAN Service Attributes and Service Framework*, October 2023
- [26] MEF 74, Commercial Affecting Attributes Technical Standard, December 2018
- [27] MEF 95.0.1, *Amendment to MEF 95: Policy Driven Orchestration*, October 2022
- [28] MEF 117, *SASE Service Attributes and Service Framework*, October 2022
- [29] MEF 118.1, *Zero Trust Framework for MEF Services*, July 2024
- [30] MEF 138, *Security Functions for IP Services*, July 2024
- [31] MITRE, <https://cve.mitre.org>
- [32] MITRE, ATT&CK, <https://attack.mitre.org>

- [33] MITRE, CAPEC, *Common Attack Pattern Enumeration and Classification*,
<https://capec.mitre.org>
- [34] STIX, *Structured Threat Information Expression*, <https://oasis-open.github.io/cti-documentation/>
- [35] NIST, National Vulnerability Database, Common Weakness Enumeration,
<https://nvd.nist.gov/vuln/categories>

Appendix A SASE Session Flow Examples (Informative)

Note: The figures in this Appendix use diagram conventions which can be found in section 5.

A.1 Session Flow with Security Functions, a subset of which are at the Subject and Target SASE Edges

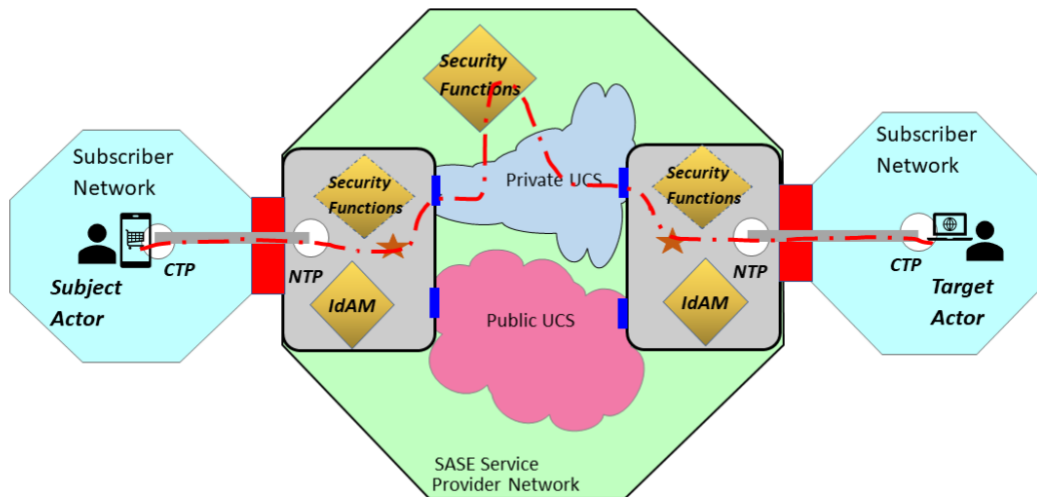


Figure 16 – Example of a Session Flow with Security Functions, subset at Subject and Target SASE Edges

Figure 16 illustrates a scenario where the Security Functions are being performed at the Subject SASE Edge, the Target SASE Edge and within the SASE Service.

A.2 Session Flow via Security SASE Edge with Security Functions at Subject SASE Edge but not at Target SASE Edge

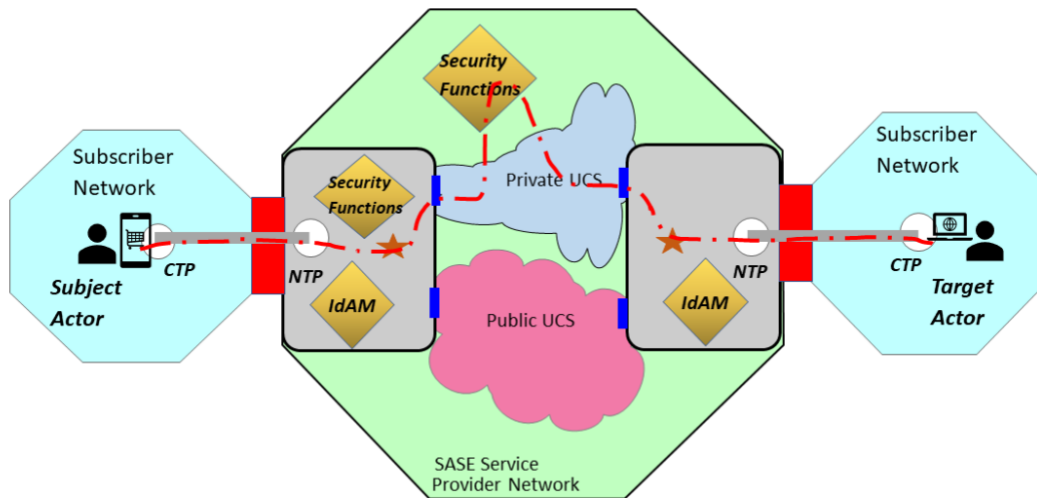


Figure 17 – Example of Security Functions at Subject SASE Edge but not Target SASE Edge

Figure 17 illustrates a scenario where some of the Security Functions are being performed at the Subject SASE Edge and others are performed within the SASE Service but no Security Functions are being performed at the Target SASE Edge.

A.3 Session Flow with Security Functions only at Subject and Target SASE Edges

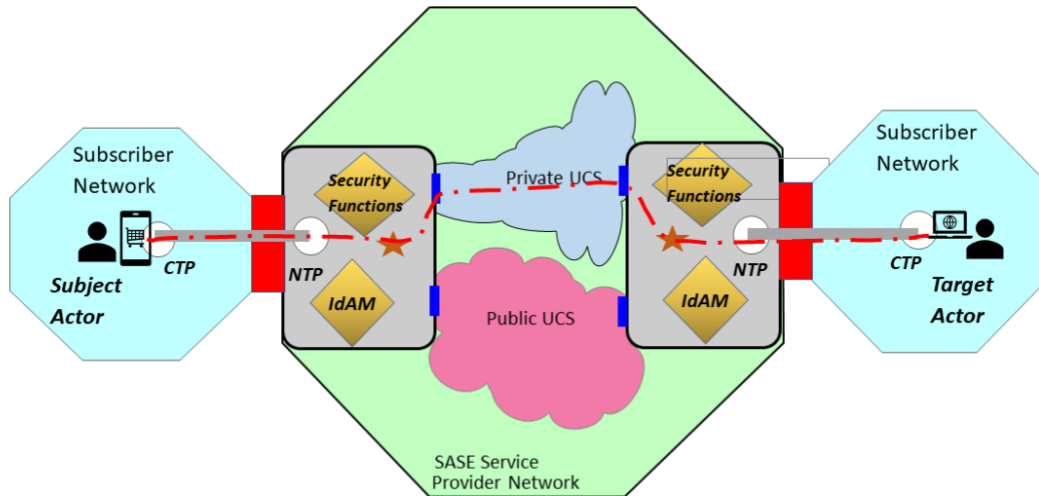


Figure 18 – Example of Security Functions only at SASE Edges

Figure 18 illustrates a scenario where all the Security Functions are being performed at only the Subject and Target SASE Edges. There is no need for any Security Functions to be performed elsewhere in the SASE service.

A.4 Session Flow with SASE in a Box deployment on Customer Premises

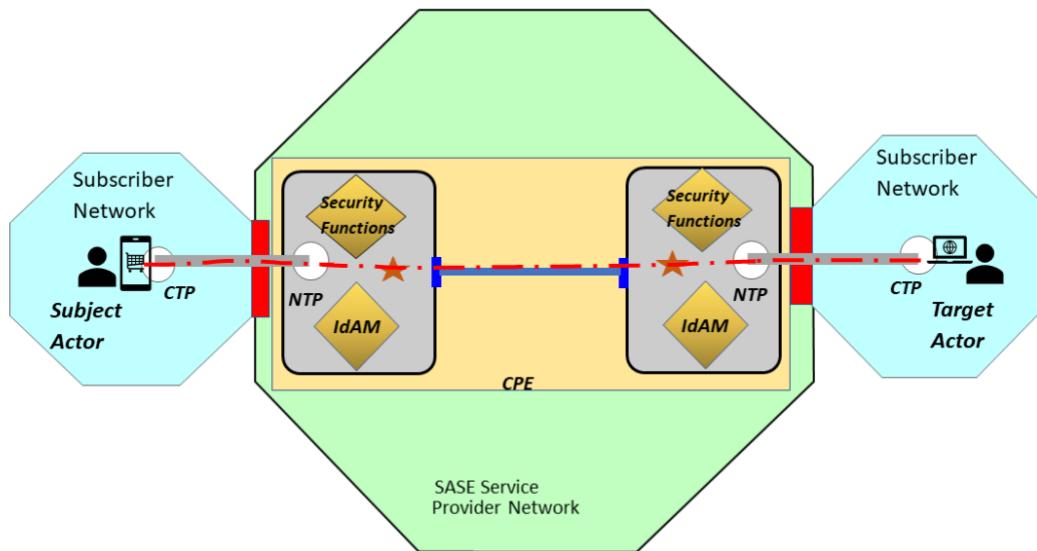


Figure 19 -- Example of SASE Service on premises with a single device

Figure 19 illustrates how a SASE Service could be delivered via an appliance on site without the need for cloud services. However, it should be noted that depending on the Security Functions needed for the SASE Policies, this device would need to be sized correctly.

A.5 Session Flow for Cloud Only delivered SASE Service

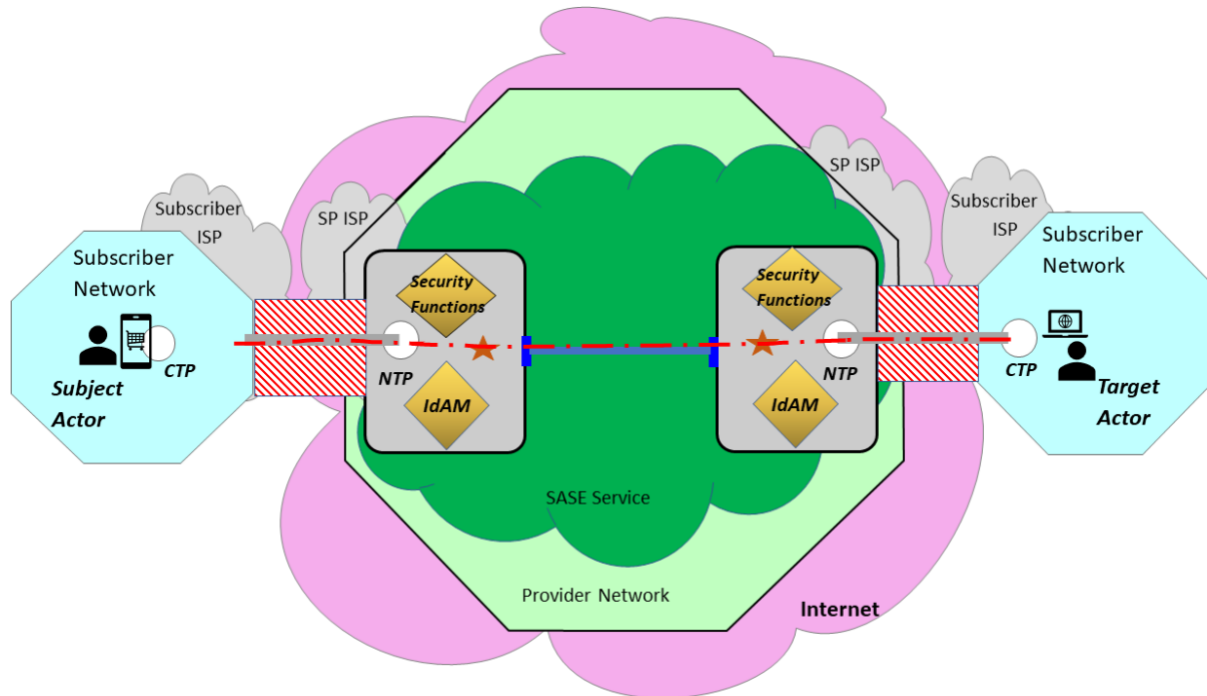


Figure 20 – Example of cloud delivered SASE Service

Figure 20 illustrates the implementation of a SASE Service as a cloud only delivered model. In this case access to the service is via Public UCS. However, this is only one example and other UCS types could be utilized to access this cloud service.

Appendix B Zero Trust Network Access (Informative)

In this document, Zero Trust Network Access (ZTNA) refers to access from any Subject Actor at any location, any time, and under any circumstance to any Target Actor at any location, any time, and under any circumstance. Just as the name implies... This implies that no network access is permitted by anyone or anything at any time, any location, or under any circumstance unless a specific Policy allows the access. As such, this would apply to any Application to Application, Device to Device, User to User, or any combination of Actors permissions for access. Even then, this access is granted with the least privilege level needed to accomplish the access. This access is also continuously monitored for compliance with the Policy.

The term “network” utilized in this document does not refer to any specific network (e.g., access network, Service Provider network, Subscriber network, core network, cloud network or any virtualized network). ZTNA considers all access between any User, Device, or Application to any other User, Device, or Application as subject to Zero Trust Policies.

ZTNA, as used in this document, differs from the industry generally accepted definition. Gartner defines Zero Trust Network Access (ZTNA) [2] as “products and services that create an identity and context-based, logical-access boundary that encompasses an enterprise user and an internally hosted application or set of applications.” In the Gartner definition, ZTNA has become synonymous with the remote access or VPN replacement model. However, this Gartner definition ignores the on-premises use case of Zero Trust Network Access and access to Target resources located on the Internet or externally hosted. Mplify includes all aspects of access in its ZTNA.

However, the Remote Access version of ZTNA is a use case within the SASE Service as can be seen in Figure 21.

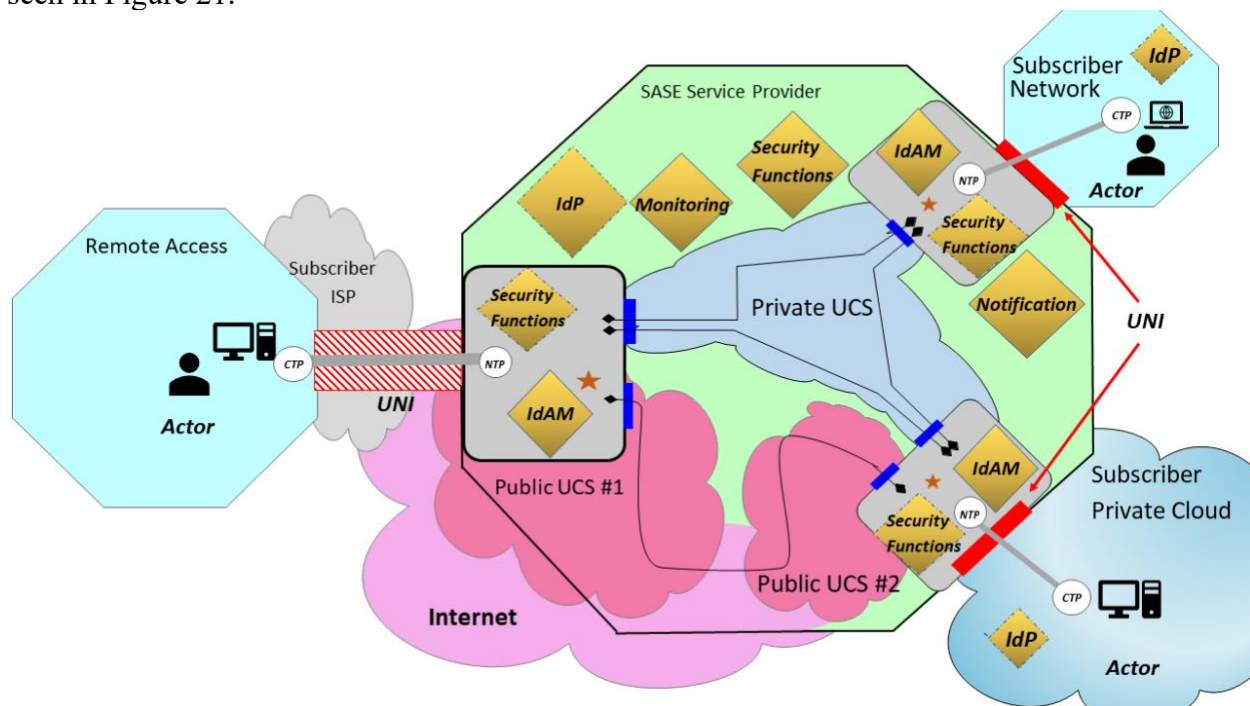


Figure 21 – Example of SASE Use case for Remote Access

Figure 21 illustrates that a SASE Service can be utilized as a Remote Access solution or a VPN replacement. Users remotely connect to a SASE Edge via the Internet, and the SASE Service determines access to which corporate locations is permitted.

Appendix C Major Changes from MEF 117 to Mplify 117.1 (Informative)

The following lists the major changes from MEF 117 [28]:

- Changed the reference for Security Functions from MEF 88 to MEF 138
- Added the following Performance Metrics:
 - One-Way Mean Packet Delay (section 7.15.2)
 - One-Way Mean Packet Delay Variation (section 7.15.4)
 - One-Way Packet Loss Ratio (section 7.15.5)
- Added the following Service Attributes: (all included in section 8)
 - List of SASE Rate Limiters Service Attribute
 - List of SASE Session Business Importance Levels Service Attribute
 - List of SA-IdAM Application Flow Specifications Service Attribute
 - List of Data Integrity Actions Service Attribute
 - SASE Performance Time Intervals Service Attribute
 - SASE Service Performance Objectives Reporting Periods Service Attribute
 - SASE UCS Service Attributes
 - UCS Identifier Service Attribute
 - UCS Type Service Attribute
 - UCS Billing Method Service Attribute
 - SASE UCS UNI Service Attributes
- Updated SASE Agent requirements to apply only if a SASE Service includes a SASE Agent. (Requirements [R35] to [R39])
- Enhanced Application Flow Specification Criteria (see section 9.3.2)
- Updated SASE Session Forwarding to include the following Criterion (see section 9.4):
 - Encryption
 - UCS type
 - UCS Billing Method
 - Session Business Importance
 - Session Performance
 - Bandwidth
- Added the following Security Functions (see section 9.6.2):
 - Supported Application Identity and Access Management (SA-IdAM)
 - Data Integrity
 - Proxy
 - Cloud Access Security Broker (CASB)
 - Remote Browser Isolation (RBI)
- Updated Session Forwarding Policy to include the new Session Forwarding Criterion noted above (see section 10.6)
- Added section 11 detailing SD-WAN Policy considerations when SD-WAN utilized by a SASE Service
- Added Appendix A SASE Session Flow Examples (Informative)

Appendix D Acknowledgements (Informative)

The following contributors participated in the development of this document and have requested to be included in this list.

- Bill **BJORKMAN**
- Neil **DANILOWICZ**
- Jeff **FANELLI**
- Mark **FISHBURN**
- Samaresh **NAIR**