



## **MEF Standard MEF 117**

# **SASE Service Attributes and Service Framework**

**October 2022**

## Disclaimer

© MEF Forum 2022. 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 MEF Forum (MEF) is not responsible for any errors. MEF does not assume responsibility to update or correct any information in this publication. No representation or warranty, expressed or implied, is made by MEF concerning the completeness, accuracy, or applicability of any information contained herein and no liability of any kind shall be assumed by MEF 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. MEF 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 MEF member which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
- b) any warranty or representation that any MEF 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 MEF member and the recipient or user of this document.

Implementation or use of specific MEF standards, specifications, or recommendations will be voluntary, and no Member shall be obliged to implement them by virtue of participation in MEF Forum. MEF is a non-profit international organization to enable the development and worldwide adoption of agile, assured, and orchestrated network services. MEF does not, expressly or otherwise, endorse or promote any specific products or services.

**EXPORT CONTROL:** This document contains technical data. The download, export, re-export or disclosure of the technical data contained in this document may be restricted by applicable U.S. or foreign export laws, regulations and rules and/or applicable U.S. or foreign sanctions ("Export Control Laws or Sanctions"). You agree that you are solely responsible for determining whether any Export Control Laws or Sanctions may apply to your download, export, reexport or disclosure of this document, and for obtaining (if available) any required U.S. or foreign export or reexport licenses and/or other required authorizations.

## Table of Contents

<b>1</b>	<b>List of Contributing Members .....</b>	<b>1</b>
<b>2</b>	<b>Abstract.....</b>	<b>2</b>
<b>3</b>	<b>Terminology and Abbreviations.....</b>	<b>3</b>
<b>4</b>	<b>Compliance Levels .....</b>	<b>8</b>
<b>5</b>	<b>Document Conventions.....</b>	<b>9</b>
<b>6</b>	<b>Introduction.....</b>	<b>10</b>
6.1	Document Scope.....	11
6.2	Organization of Standard.....	11
6.3	Characteristics of a SASE Service.....	11
<b>7</b>	<b>Key Concepts and Definitions.....</b>	<b>13</b>
7.1	SASE Session .....	13
7.2	SASE Edge .....	13
7.2.1	SASE Agent.....	14
7.3	SASE UNI .....	14
7.4	SASE Policy End Point .....	14
7.5	Identity and Access Management.....	15
7.6	Actor Access Connection .....	15
7.6.1	Customer Termination Point.....	15
7.6.2	Network Termination Point .....	15
7.7	SASE Session Forwarding.....	15
7.8	SASE Session Monitoring.....	15
7.9	Security Functions .....	15
7.9.1	Middle Box Function (MBF) .....	15
7.9.2	IP, Port and Protocol Filtering (IPPF).....	15
7.9.3	DNS Protocol Filtering (DPF) .....	16
7.9.4	Domain Name Filtering (DNF).....	16
7.9.5	URL Filtering (URLF).....	16
7.9.6	Malware Detection and Removal (MD+R).....	16
7.10	SASE Service Notifications.....	16
7.11	Subscriber .....	16
7.12	Service Provider .....	17
7.13	Policy Driven Orchestration .....	17
7.13.1	Policy .....	17
7.13.1.1	Composite Policy.....	17
7.13.1.2	Atomic Policy .....	17
7.13.2	Policy Priority .....	17
7.14	Zero Trust Framework.....	17
7.14.1	Actor .....	18
7.14.2	Policy End Point .....	18
7.14.3	Identity Provider .....	18
<b>8</b>	<b>SASE Service Attributes.....</b>	<b>19</b>
8.1	List of SASE Edges Service Attribute.....	19

8.2	List of SASE Network Termination Points Service Attribute .....	20
8.3	SASE Policy End Point Identifier Service Attribute .....	20
8.4	List of Identity Providers Service Attribute .....	20
8.5	List of Application Flow Specifications Service Attribute .....	20
8.6	List of SASE Session State Values Service Attribute .....	21
8.7	List of SASE Identity Policies Service Attribute .....	21
8.8	List of SASE Actor Access Connection Policies Service Attribute .....	21
8.9	List of SASE Supported TLS Versions Service Attribute .....	21
8.10	List of SASE Supported Cipher Suites Service Attribute .....	21
8.11	List of SASE Supported IPSEC Security Options Service Attribute .....	21
8.12	List of SASE Context Policies Service Attribute .....	21
8.13	List of SASE Security Policies Service Attribute .....	21
8.14	List of SASE Security Functions Service Attribute .....	21
8.15	List of SASE Session Forwarding Policies Service Attribute .....	22
8.16	List of SASE Monitoring Policies Service Attribute .....	22
8.17	List of SASE Notification Policies Service Attribute .....	22
8.17.1	List of SASE Notification Recipients Service Attribute .....	22
8.18	SASE Composite Policy Levels Service Attribute .....	22
<b>9</b>	<b>SASE Service Framework .....</b>	<b>23</b>
9.1	SASE Edge .....	23
9.1.1	SASE Agent .....	25
9.2	Identity and Access Management .....	26
9.2.1	IdAM Authentication of Actors .....	26
9.2.2	Actor Access Authorization .....	28
9.2.3	Actor Access Connections .....	28
9.3	SASE Session .....	29
9.3.1	Session Specification .....	31
9.3.1.1	Actor Pair .....	31
9.3.2	Session State .....	31
9.3.2.1	Initial .....	32
9.3.2.2	Operational .....	32
9.3.2.3	Re-Evaluate .....	32
9.3.2.4	Terminal .....	32
9.3.2.5	SASE Session State Machine .....	33
9.3.3	Ingress IP Packet Classification Example .....	34
9.3.4	New SASE Session Creation Example .....	37
9.4	SASE Session Forwarding .....	39
9.5	SASE Session Monitoring .....	39
9.5.1	Session State Change .....	41
9.6	Security Functions .....	41
9.6.1	SASE Security Function Atomic Policy .....	42
9.7	SASE Service Notifications .....	42
9.7.1	SASE Authentication or Authorization Notification (SAAN) .....	42
9.7.2	SASE Security Event Notification (SEEN) .....	43
<b>10</b>	<b>Policies .....</b>	<b>46</b>
10.1	SASE Policy .....	46
10.2	Policy Execution Order .....	47

10.3	Identity and Access Management Policy .....	47
10.3.1	Actor Authentication Function .....	48
10.3.2	Actor Access Authorization Function.....	48
10.3.3	Actor Access Connection.....	48
10.4	Context Policy .....	50
10.5	Security Policy .....	51
10.6	Session Forwarding Policy .....	51
10.7	Monitoring Policy .....	52
10.8	Notification Policy .....	52
10.9	SASE Edge Policy Map.....	53
<b>11</b>	<b>References.....</b>	<b>54</b>
<b>Appendix A</b>	<b>SASE Session Flow Examples .....</b>	<b>56</b>
A.1	Session Flow via Security SASE Edge with subset of Security Functions at Subject and Target SASE Edges .....	56
A.2	Session Flow via Security SASE Edge with Security Functions at Subject SASE Edge but not at Target SASE Edge.....	56
A.3	Session Flow with Security Functions only at Subject and Target SASE Edges .....	57
A.4	Session Flow with SASE in a Box deployment on Customer Premises .....	57
A.5	Session Flow for Cloud Only delivered SASE Service.....	58

## List of Figures

Figure 1 – Subject and Target Actors .....	10
Figure 2 – SASE Service General Diagram.....	13
Figure 3 – Ingress UNI and Egress UNI Examples .....	14
Figure 4 – SASE Service manages Subject Actor access to Target Actor .....	23
Figure 5 – SASE Edge .....	24
Figure 6 – SASE Remote Example.....	25
Figure 7 – Actor Access Connections.....	29
Figure 8 – SASE Session State Machine .....	33
Figure 9 – Ingress IP Packet Classification Flow Example .....	36
Figure 10 – New SASE Session Flow Example .....	38

## List of Tables

Table 1 – Terminology and Abbreviations .....	7
Table 2 – Notation Conventions .....	9
Table 3 – Diagram Conventions .....	9
Table 4 – Items to be included in a SAAN .....	43
Table 5 – Items to be included in a SSEN .....	44

## 1 List of Contributing Members

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

- AT&T
- Ciena
- IBM
- Nokia
- Orange
- PCCW Global
- Versa Networks



## 2 Abstract

This document defines a Secure Access Service Edge (SASE) Service Framework and specifies Service Attributes that need to be agreed between a Service Provider and a Subscriber for SASE Services, including Security Functions, Policies and Connectivity Services. The document defines the behavior 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.

### 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 MEF or external documents.

In addition, terms defined in MEF 61.1 [1], MEF 70.1 [2], MEF 88 [3], MEF 95.0.1 [4], and MEF 118 [5] 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 [5]
Actor Access Connection	The network connection between an Actor and the SASE Service. The Actor Access Connection consist of the Customer Termination Point and the 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 70.1[2]
Application Flow Specification	A named set of Application Flow Criteria.	MEF 70.1[2]
Atomic Policy	A stand-alone Policy.	Adapted from MEF 95.0.1 [4]
Authentication	The process of verifying the Identity of an Actor.	Adapted from MEF 118 [5]
Authorization	The process that results in Allowing or Blocking a Subject Actor from accessing a Target Actor.	MEF 118 [5]

Term	Definition	Reference
Composite Policy	A set of related Policies that are organized into a hierarchical structure.	Adapted from MEF 95.0.1 [4]
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 (CTP)	The part of the Actor Access Connection in the Subscriber domain.	This document
DNS Protocol Filtering (DPF)	The Security Function that determines whether a Session contains Domain Name System (DNS) messages that are to be Allowed or Blocked.	Adapted from MEF 88 [3]
Domain Name Filtering (DNF)	The Security Function that determines whether a Session contains domain names that are to be Allowed or Blocked.	Adapted from MEF 88 [3]
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 [5]
Identity and Access Management (IdAM)	The process that authenticates and authorizes an Actor to utilize a SASE Service.	This document
Identity and Access Management Policy (IdAMP)	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 (IdP)	The organization that manages the Authentication Credentials of an Actor.	MEF 118 [5]
IP, Port and Protocol Filtering (IPPF)	The Security Function that determines whether a Session's source or destination IP addresses, source or destination port numbers, or IP protocols are to be Allowed or Blocked.	Adapted from MEF 88 [3]
Malware Detection and Removal (MD+R)	The Security Function that determines whether a Session contains Malware and removes the Malware or Blocks the subset of the Session containing the Malware.	Adapted from MEF 88 [3]
Middle Box Function (MBF)	A function used to decrypt and re-encrypt a given Session so Security Functions can be applied to that Session.	Adapted from MEF 88 [3]

Term	Definition	Reference
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 (NTP)	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
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 [4]
Policy Criterion	A criterion that describes a specific objective or constraint.	Adapted from MEF 70.1[2]
Policy End Point	The location where one or more Policy-related functions are placed.	MEF 118 [5]
Policy Execution Order	The value for order of processing of a Policy where the highest value is processed first.	MEF 95.0.1 [4]
SASE	Secure Access Service Edge.	Adapted from Gartner [6]
SASE Agent	Software installed on a Device that enables the SASE Edge functionality.	This document
SASE Authentication or Authorization Notification (SAAN)	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.	
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 (SSEN)	A communication of a security event, e.g., a SSEN is issued when a subset of a Session is Blocked or modified.	Adapted from MEF 88 [3]

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
SASE Security Event Notification (SSEN)	A communication of a security event, e.g., a SSEN is issued when a subset of a Session is Blocked or modified.	Adapted from MEF 88 [3]
Security Function	The component that, when enabled per the Security Policy, makes a decision to Allow or Block a subset of a Session.	Adapted from MEF 88 [3]
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 70.1[2]
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

Term	Definition	Reference
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
Subject Actor	An Actor requesting access to a Target Actor.	MEF 118 [5]
Subscriber	An entity that contracts to use a service. In this document, “Subscriber” should be read as meaning “SASE Subscriber”.	Modified from MEF 70.1[2]
Target Actor	An Actor that a Subject Actor wants to access.	MEF 118 [5]
Underlay Connectivity Service (UCS)	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 70.1[2]
Uniform Resource Location (URL)	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 [12]
URL Filtering (URLF)	The Security Function that determines whether a Session contains a URL that is to be Allowed or Blocked.	Adapted from MEF 88 [5]
UTC	Coordinated Universal Time.	RFC 3339 [10]

**Table 1 – Terminology and 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 [16]) 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 2 – 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 the Actor Access Connection
	This represents the SASE UNI
	This represents the UCS UNI
	This represents the private UCS
	This represents the public UCS
	This represents the SASE Service
	This represents a SASE Edge
	This represents a function
	These represent Users
	These represent Devices
	This represents a Customer or Network Termination Point

**Table 3 – Diagram Conventions**



## 6 Introduction

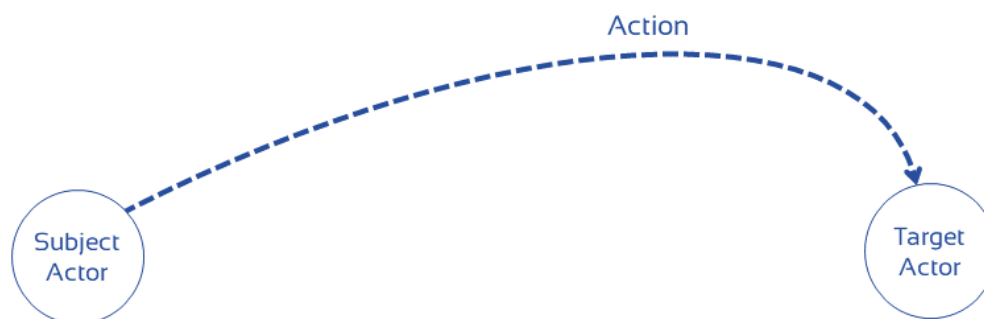
The paradigm where the bulk of an enterprise traffic is contained within a well-defined and securable enterprise perimeter (e.g., campuses, data centers) is no longer sufficient to address modern security concerns.

This document expands upon the Gartner SASE concept [6] by defining a standard SASE Service that combines Security Functions and network connectivity.

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 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 in 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.

## 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 SASE Service Attributes and requirements is provided in section 8.
- An overview of the Security Functions is provided in sections 7.9.
- 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.

## 6.3 Characteristics of a SASE Service

A SASE Service has the following fundamental characteristics:

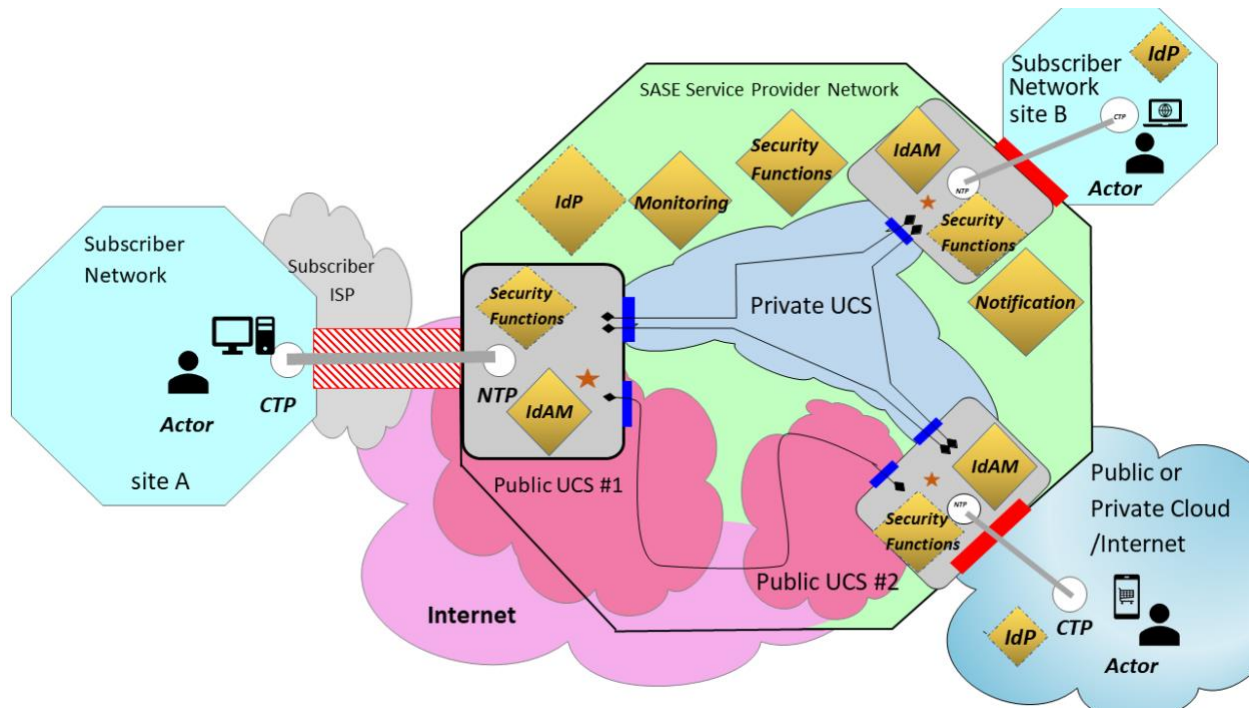
- The basic unit of transport within the SASE Service is an IP Packet.
- A SASE Service applies Policy to Sessions.
- Identity and Access Management of Actors utilizing the SASE Service.
- An Actor Access Connectivity Policy to influence Actor Access Connection to and from the SASE Service.
- A Policy-driven networking technology (e.g., SD-WAN).

- Security Functions to secure and protect the Sessions through the SASE Service.
- SASE Policies to determine the appropriate handling of IP Packets through the SASE Service.
- Monitoring of the Session.

## 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**

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. These 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.

Each of these logical constructs are defined in Section 9.

### 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 9.3.2)

### 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

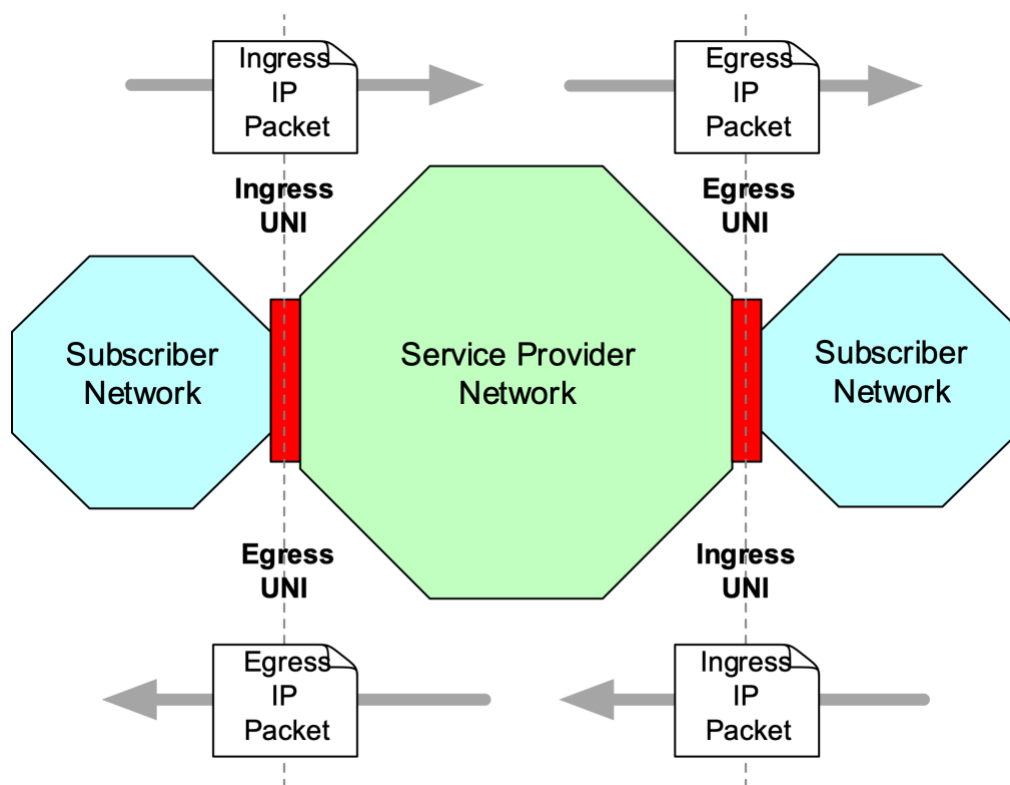
The SASE Agent is software (installed on a Device) that provides the SASE Edge functionality (see Section 9.1.1).

## 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**

## 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 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)

## **7.6 Actor Access Connection**

The Actor Access Connection is the network connection between the 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)

## **7.8 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 9.5)

## **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 Middle Box Function (MBF)**

Many Security Functions can only work on Sessions that are unencrypted. Therefore, encrypted Sessions must be decrypted for the Security Functions to inspect the packets, and then re-encrypted after the Security Function actions are taken. Middle Box Function 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 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 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 [7] and RFC 1996 [8].

### 7.9.4 Domain Name Filtering (DNF)

Domain Name Filtering is the Security Function 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 that determines whether a Session contains URLs to be Allowed or Blocked. URL is specified in IETF RFC 3986 [12]. 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 defined as the Security Function that determines whether a Session contains Malware. If Malware is detected, the 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.

## 7.10 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

The SASE Service incorporates terminology from the MEF 95.0.1 Policy Driven Orchestration, (PDO), Amendment 1 [4].

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 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 Priority

Policy Priority 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 [4].

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.

#### **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.

## 8 SASE Service Attributes

MEF 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 MEF Service, and it describes some aspect 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.

- [R1] 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.

- [R2] The value of the SASE Edge Identifier **MUST** be an Identifier String.
- [R3] The value of the SASE Edge Identifier **MUST** be unique across all SASE Edges in the SASE Service.
- [R4] 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 Service Attribute values.

- [R5] The List of SASE Network Termination Points Service Attribute **MUST** contain at least two SASE Network Termination Point Identifiers 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.

- [R6] The value of the SASE Network Termination Point Identifier **MUST** be an Identifier String.
- [R7] The value of the SASE Network Termination Point Identifier **MUST** be unique across all SASE Network Termination Points in the SASE Service.
- [R8] 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 Points with a SASE Edge.

- [R9] The value of the SASE Policy End Point Identifier Service Attribute **MUST** be an Identifier String.
- [R10] 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.

## 8.5 List of Application Flow Specifications Service Attribute

As defined by MEF W70.1 [2], 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 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.

[R11] 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 Functions 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)

**[R12]** The List of SASE Security Function Service Attribute **MUST** include all the Security Functions identified in [R46].

### **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.

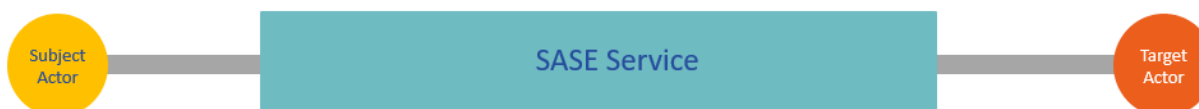
### **8.18 SASE Composite Policy Levels Service Attribute**

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.

**[R13]** The SASE Composite Policy Levels Service Attribute value **MUST** greater than zero.

## 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 4. 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 4 – 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.

**[R14]** Each SASE UNI **MUST** have a unique Identifier String.

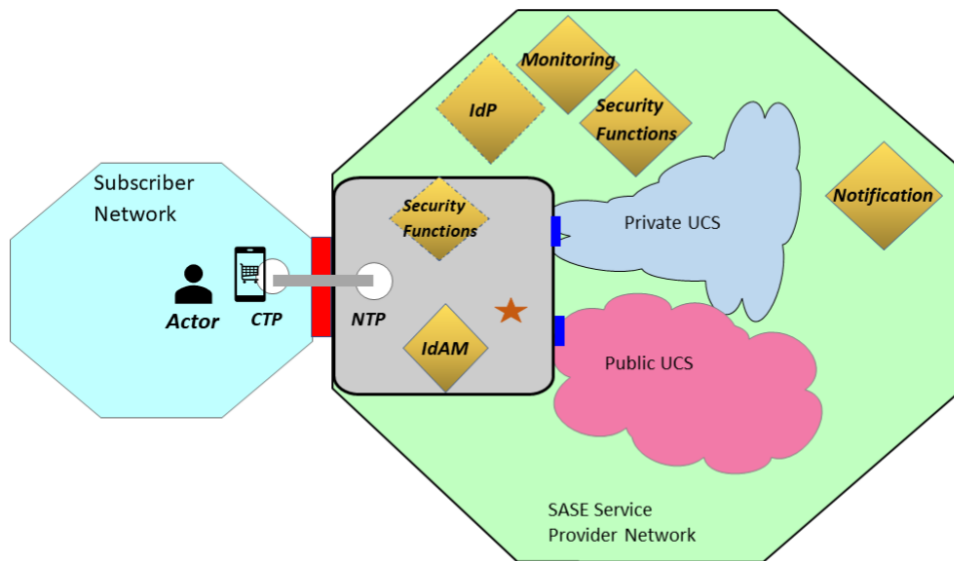
In a given Session, there are two SASE Edge types: the Subject SASE Edge and the Target SASE Edge. But for a different Session, the same two SASE Edges could have those roles reversed. The Subject and Target SASE Edges, for a given Session, may be located on the same Device, located in the same cloud, 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 5:

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

- UCS UNI.



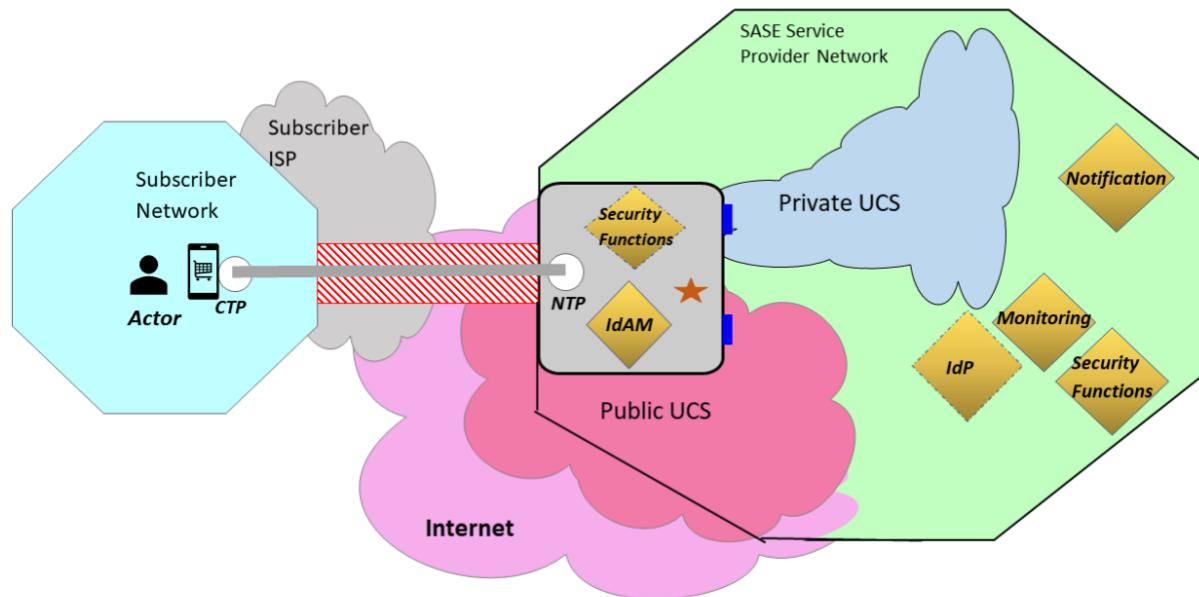
**Figure 5 – SASE Edge**

- [R15] Each SASE Edge **MUST** have to at least one SASE UNI.
- [R16] Each SASE Edge **MUST** connect to at least one UCS.
- [R17] Each SASE Edge **MUST** have a SASE Edge Identifier.
- [R18] Each SASE Edge **MUST** contain Identity and Access Management.
- [R19] Each SASE Edge **MUST** contain one or more Network Termination Points.
- [R20] 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.
- [R21] 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.
- [R22] A SASE Edge **MUST** contain at least one SASE Policy End Point.
- [R23] 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.

**[D1]** A SASE Edge **SHOULD** contain Security Functions.



**Figure 6 – SASE Remote Example**

Figure 6 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 assure 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.

- [R24] Each SASE Agent **MUST** have a SASE Edge Identifier.
- [R25] Each SASE Agent **MUST** connect to at least one UCS.
- [R26] Each SASE Agent **MUST** contain the Identity and Access Management.
- [R27] Each SASE Agent **MUST** contain a SASE UNI.
- [R28] Each 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 5, 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 5. 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 provides only access to the SASE Service. This is not the same IdAM that would need to be accomplished by the Target Actor to fully grant access to the Subject Actor. That IdAM 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.

- [R29] A SASE Service **MUST** use an 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 [5].

[R30] The Identity and Access Management **MUST** comply with all requirements in Section 8 of MEF 118 [5].

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.

[R31] For a given SASE Service, every Actor **MUST** have a unique identifier, *ActorID*.

[R32] A SASE Service **MUST** authenticate the Identity of all Subject Actors.

[D2] 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.

The authentication of the Target Actor can be accomplished either by using an Identity and Access Management 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. There are other methods, but the method for authenticating the Target Actor is beyond the scope of this document.

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.

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.

[R33] Each Actor **MUST** be associated with an IP address.

[R34] Any IP Packet that ingresses a SASE UNI which cannot be associated with an authorized Actor **MUST** be discarded.

[R35] 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.

[D3] 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.

- [R36] 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.

- [R37] A SASE Service **MUST** authorize all Actors.

- [R38] Any IP Packet that ingresses a SASE UNI which cannot be associated with an authorized Actor **MUST** be discarded.

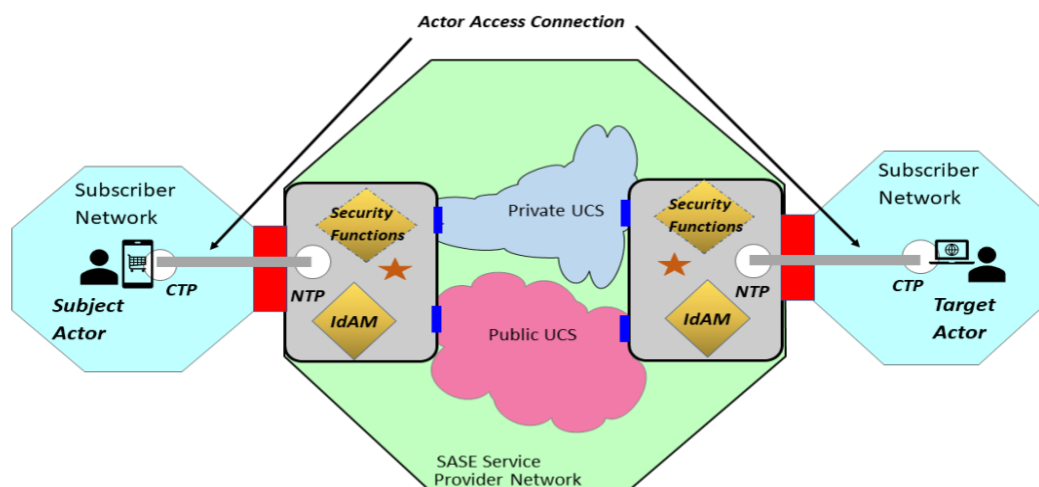
Since all Subject Actors are authenticated, Authorization of the Subject Actors is based upon Authentication and other IdAM Policy parameters. However, since Target Actors may or may not be authenticated, Authorization of the Target Actors is based upon the IdAM Policy parameters applied, which could 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 7. 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 7 – Actor Access Connections**

**[R39]** 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.

**[R40]** Each SASE Session **MUST** have an identifier, *SessionID*.

**[R41]** 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 (9.3.2)

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 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.

#### 9.3.1.1 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  $\langle Subject, Target \rangle$  where:

- *Subject* is a 2-tuple of the form  $\langle SubjectID, IdP \rangle$  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  $\langle TargetID, IdP \rangle$  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.

### 9.3.2 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.

**[R42]** 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.

**[R43]** 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.

**[D4]** 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*.

#### 9.3.2.1 *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.

#### 9.3.2.2 *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.

#### 9.3.2.3 *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 versus, 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.

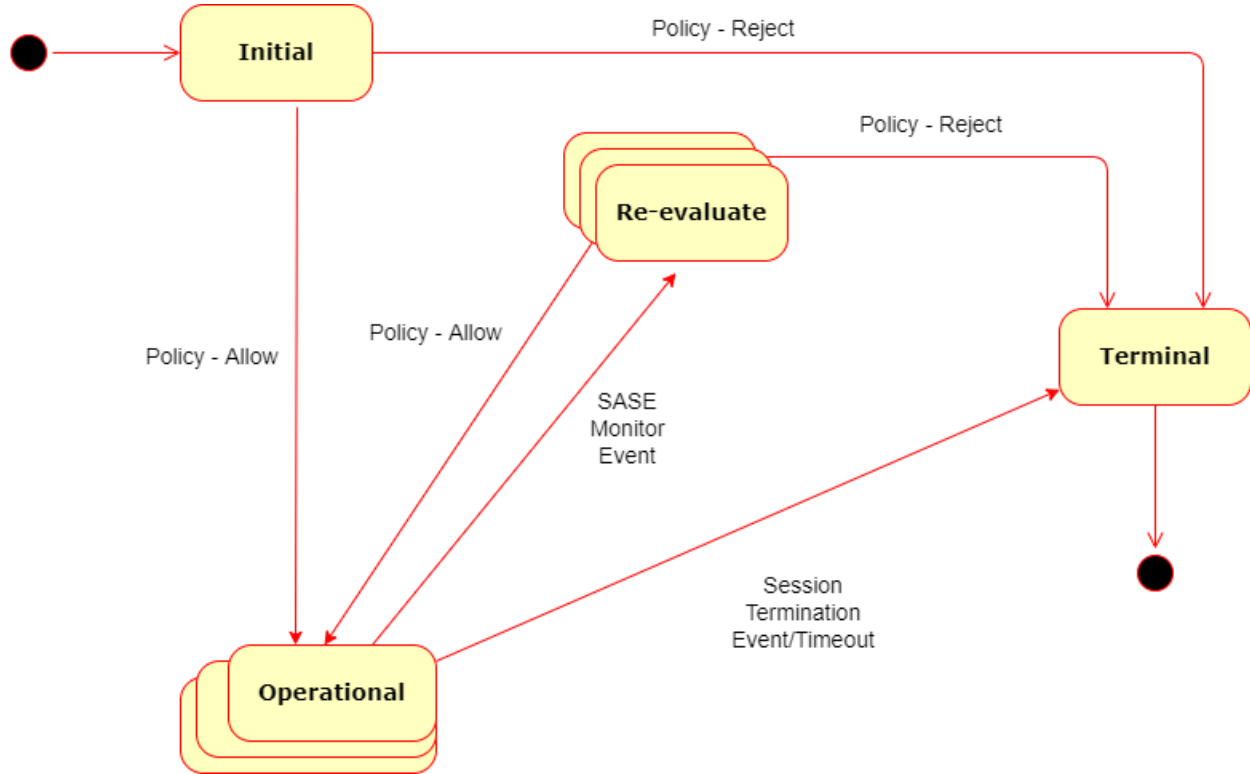
#### 9.3.2.4 *Terminal*

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.

### 9.3.2.5 SASE Session State Machine



**Figure 8 – 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  $\langle SessionID, SessionSpecID, SessionStates \rangle$ .

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  $\langle Example, AZ-Talk, SessionState \rangle \dots$  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*.

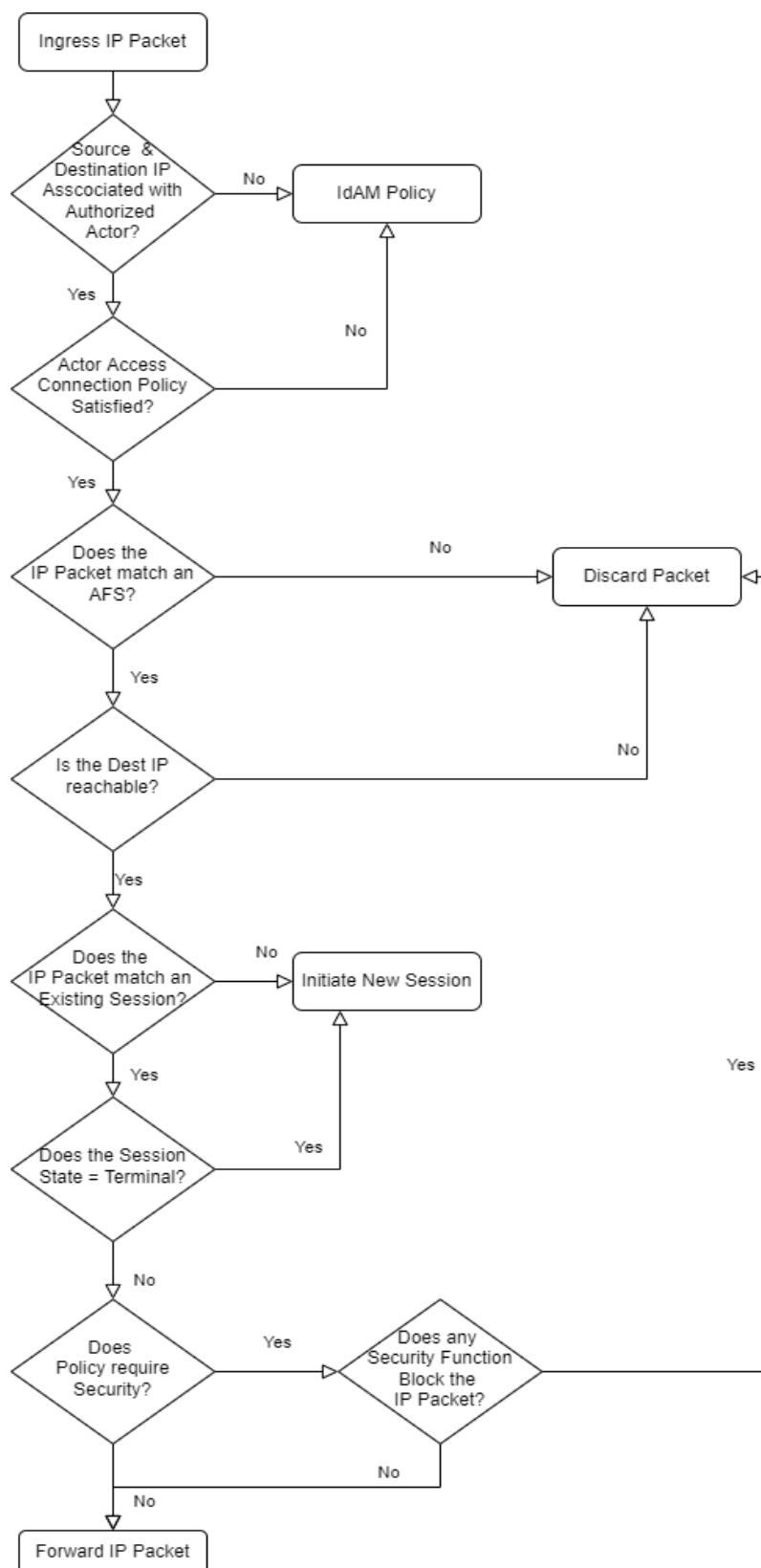
### 9.3.3 Ingress IP Packet Classification Example

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.

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 9 – 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 9 – Ingress IP Packet Classification Flow Example**

The flow in Figure 9, 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 must be reachable to process the packet via the SASE Service. If the Destination IP is not reachable, the IP Packet is discarded.

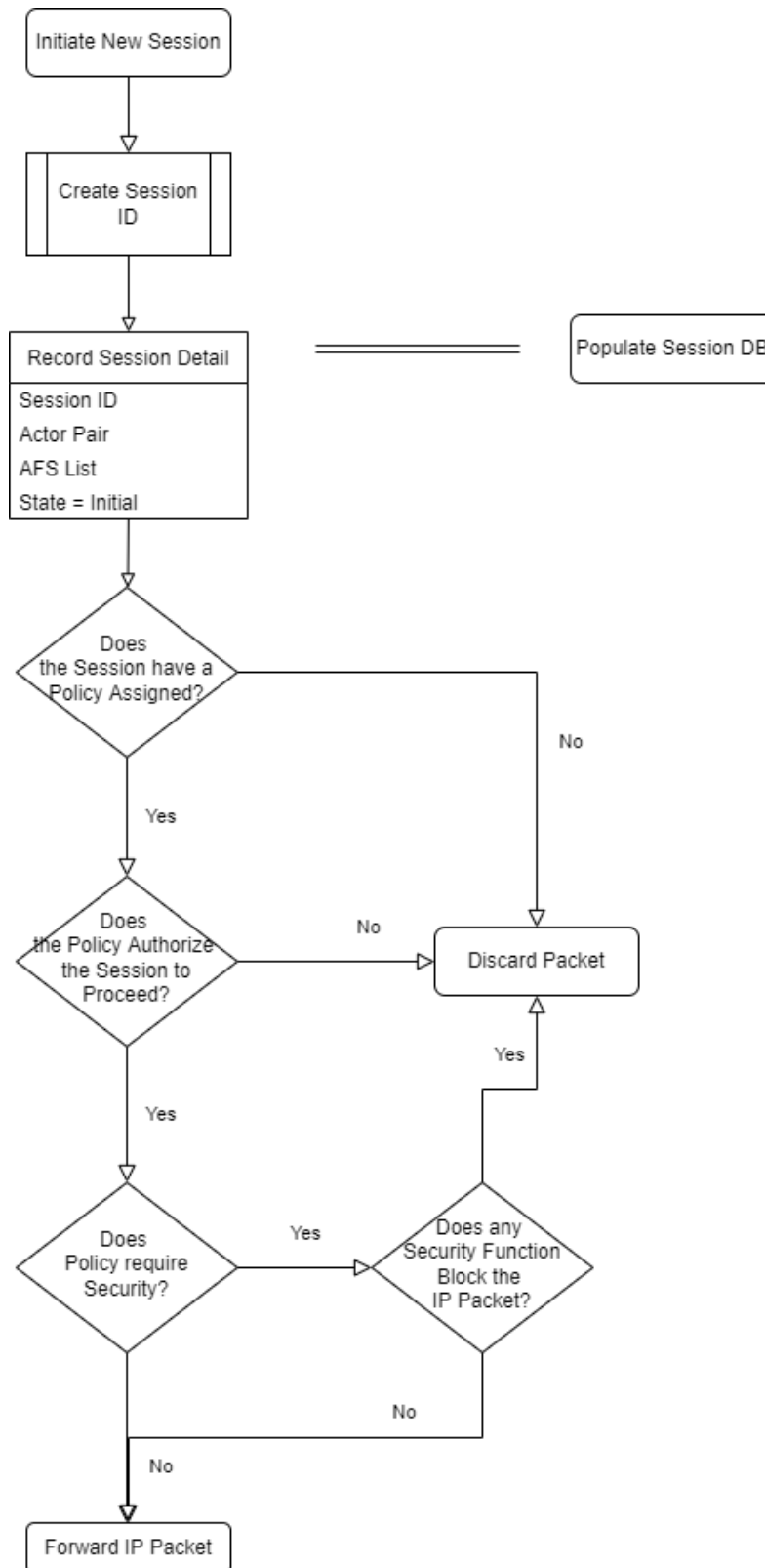
Next the IP Packet is matched against the List of Application Flow Specifications Service Attribute. The SASE Service mandates that the IP Packet match one of the AFSs in the List of Application Flow Specifications Service Attribute or discard the IP Packet.

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 9.3.4).

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.

### **9.3.4 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 10 is an example of a new SASE Session flow.



**Figure 10 – 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. The connectivity between any two SASE Edges **MUST** be encrypted.

[D5] A Service Provider **SHOULD** support SD-WAN Service as defined in MEF 70.1 [2] 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.

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 W70.1 [2] compliant SASE Edge Session forwarding.

## 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.

**[R44]** 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

**[R45]** 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 must be deployable anywhere within the SASE Service to optimize the performance and security provided by the SASE Service for that Session.

The Security Functions needed for a SASE Service are those Security Functions adapted from MEF 88 [3] as listed in section 7.9.

The Security Functions are ‘atomic’ as they represent the basic building blocks which security Providers and vendors use to create security packages such as cloud access security broker (CASB), 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 list of mandatory Security Functions is adopted from MEF 88 [3] and adapted by the SASE Service to apply to Sessions instead of Application Flows and removed from being enforced in the SD-WAN Service and are instead enforced in the SASE Service.

**[R46]** A SASE Service **MUST** support the following Security Functions to be specified in the Security Policy associated with Sessions:

- Middle Box Function (MBF).
- IP, Port and Protocol Filtering (IPPF).
- DNS Protocol Filtering (DPF).
- Domain Name Filtering (DNF).
- URL Filtering (URLF).
- Malware Detection and Removal (MD+R).



- [R47] The Security Functions in [R46] **MUST** comply with all mandatory requirements in MEF 88 [3] sections 8 and 9 as applied to Sessions instead of Application Flows.
- [R48] A SASE Service **MUST** implement the Security Action Lists defined in MEF 88 [3] section 7 as applied to Sessions instead of Application Flows.

If the SD-WAN Service provided by the SASE Service is also a secure SD-WAN that implements both MEF 70.1 [2] and MEF 88 [3], then it is quite possible that these same Security Functions would exist both in SASE and the SD-WAN. How this double Security Function paradigm is reconciled, and which Security Functions are done by which Service is at the Service Provider's discretion as long as the requirements in the Subscriber Policy are met.

#### 9.6.1 SASE Security Function Atomic Policy

The parameters listed for each Security Function in MEF 88 [3] section 6 are the Policy components for the specific SASE Security Function Atomic Policy. (See section 10.5).

- [R49] Each SASE Security Function Atomic Policy **MUST** utilize the appropriate parameters for Security Functions as defined in MEF 88 [3] section 6.

### 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.

- [R50] A SAAN **MUST** be issued whenever an Actor is denied access to the SASE Service due to an Authentication or Authorization failure.
- [R51] A SAAN **MUST** be issued whenever a Session is Blocked by a SASE Policy.
- [R52] 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.

- [R53] A SAAN **MUST** include the items listed in Table 4.

Item	Value	Comments
Issuer	UTF-8 [11] String <sup>1</sup>	Examples: SASE Service Provider Name
Timestamp of SAAN	Date-time	Example: UTC [10]
SAAN ID	UTF-8 [11] String	Example: Universally Unique Identifier [13]
Source IP Address	Human readable IPv4 dotted decimal IPv6 hexadecimal strings	Example: IP address of Subject Actor
SASE UNI ID	UTF-8 [11] String	Example: Universally Unique Identifier [13]
Policy ID	UTF-8 [11] String	Example: Policy Name
Type of SAAN	UTF-8 [11] String	Examples: Actor Authentication failure, Actor Authorization failure, Session Policy Failure
Authentication/Authorization Failure Details	UTF-8 [11] String	Examples: Username, Actor ID, Source/Destination IP address, Source Media Access Control (MAC) Address, Session ID, Source/Destination port number.

**Table 4 – 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.

The Security Event Notification needed for a SASE Service is adapted from MEF 88 [3] and applies to a subset of a Session versus a subset of Application Flow.

**[R54]** A SSEN **MUST** be issued whenever a subset of the Session is Blocked or modified by a Security Function.

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.

**[R55]** The Service Provider **MUST** store each SSEN in a secure repository for future reference and security auditing purposes.

<sup>1</sup> UTF-8, Unicode Transformation Format 8-bit

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.

**[R56]** A SSEN MUST include the items listed in Table 5.

Item	Value	Comments
Issuer	UTF-8 [11] String	Examples: SASE Service Provider Name
Timestamp of IOC	Date-time	Example: UTC [10]
SSEN ID	UTF-8 [11] String	Example: Universally Unique Identifier [13]
Source IP Address	Human readable IPv4 dotted decimal IPv6 hexadecimal strings	Example: IP address of Subject Actor
SASE UNI ID	UTF-8 [11] String	
Policy ID	UTF-8 [11] String	Example: Security Policy name
IOC Type	UTF-8 [11] String	Examples: CVE [19], STIX [22], CWE [18], CAPEC [21], ATT&CK [20], RFC 7970 [14].
IOC Information ID	UTF-8 [11] String	Identifies the IOC based on type
IOC Source	URL for IOC Type	Example: CVE [19]
Type of Compromise	UTF-8 [11] String	Examples: Known vulnerability, breach, data leakage, abuse of resources, jacking, where to find more information on breach.
Compromise Details	UTF-8 [11] 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 [11] String	Examples: informational, quarantined, Blocked, or Malware removed.

**Table 5 – 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 [12], 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.

- [R57] Any domain name or URL in a SSEN **MUST** be neutralized.
- [D6] The method for neutralizing the domain name or URL in a SSEN **SHOULD** use square brackets around each period.

## 10 Policies

### 10.1 SASE Policy

A SASE Policy is a Composite Policy, as defined by MEF 95.0.1 [4] Amended PDO model, utilized in a SASE Service and applied to a given Session.

**[R58]** A SASE Policy **MUST** be a Composite Policy.

**[R59]** A SASE Policy **MUST** comply with the mandatory requirements in MEF 95.0.1 [4].

Since SASE Service incorporates a Zero Trust Framework, the Policies within a SASE Service need to comply with the requirements as defined in MEF 118 [5].

**[R60]** A SASE Policy **MUST** comply with the mandatory requirements in MEF 118 [5] sections 13 and 14.

**[R61]** 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) are 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.

**[R62]** 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

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. The Policy Execution Order is the value for order of processing where the highest value is processed first.

**[R63]** A Service Provider **MUST** indicate to the Subscriber the possible values for Policy Execution Order.

**[R64]** All Policies within a SASE Policy **MUST** have a Policy Execution Order.

## 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.

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

**[R65]** 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 the 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 Priority of the IdAMP needs to be the highest possible value.

**[R66]** The Policy Priority 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.

**[R67]** 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.

**[R68]** 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.

**[R69]** 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 [14]; 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.

**[R70]** 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.

**[D7]** The SASE Service **SHOULD** secure the Target Actor Access Connection.

**[R71]** The SASE Service **MUST** support the use of TLS 1.2 [14] to secure the Actor Access Connection.

**[D8]** 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.

**[R72]** The SASE Service **MUST** meet the mandatory requirements of TLS 1.2, per RFC 5246 [14].

**[R73]** The SASE Service **MUST** meet the mandatory requirements of RFC 8446 [17] section 9.3 (Protocol Invariants).

**[R74]** The TLS version utilized by a SASE Service **MUST** be a value of the List of SASE Supported TLS Versions Service Attribute.

**[R75]** The cipher suites utilized by a SASE Service **MUST** be a value of the List of SASE Supported Cipher Suites Service Attribute.

**[CR1]<[D8]** The SASE Service Provider **MUST** provide to the Subscriber a list of all IPSEC security options supported by the SASE Service.

**[CR2]<[D8]** 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 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.

**[R76]** 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.

[R77] The Context Policy **MUST** have a unique identifier.

[R78] The Policy Priority value for a Context Policy **MUST** be lower than the IdAMP Policy Priority 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.

[R79] 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.

[R80] A Context Policy **MUST** include a set of location criteria.

Due to many industrial, governmental, and Subscriber regulations, or the location, for which a given set of information may be located, accessed, transported, or utilized, may need to be restricted or constrained by any Policies that grant access to said information.

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.

[R81] The Security Policy **MUST** have a unique identifier.

[R82] The Priority value of a Security Policy **MUST** be lower than the IdAMP Policy Priority value.

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

[R83] 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 Forward Policy dictates which Underlay Connectivity Service a given Session follows from one SASE Edge to the next SASE Edge within a SASE Service.

[R84] The Session Forward Policy **MUST** have a unique identifier.

[R85] The Policy Priority value of a Session Forward Policy **MUST** be lower than the IdAMP Policy Priority value.

The exact implementation details of what Underlay Connectivity Service type, encapsulation mechanisms and encryption mechanisms are beyond the scope of this document.

When a SASE Service utilizes an SD-WAN Service, as defined in MEF 70.1 [2], the SASE Service 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.

[CR3]< [D5] 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.

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.

[CR4]< [D5] The Session Forwarding Policy **MUST** comply with the SD-WAN Policies (as defined by MEF 70.1 [2]) assigned to the Application Flows in the SASE Session Specification of a SASE Service.

Where a SASE Service does not utilize an SD-WAN Service, the Session Forwarding Policy would still need to determine the proper method to transmit the IP packets from one SASE Edge to another. However, the methods used in this situation are beyond the scope of this document. It is strongly suggested that an SD-WAN Service be utilized within a SASE Service.

## 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 Monitor 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.

[R86] The Monitor Policy **MUST** have a unique identifier.

[R87] The Policy Priority value of a Monitor Policy **MUST** be lower than the IdAMP Policy Priority value.

The actual implementation details of what is contained in the Monitor 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.

[R88] The Notify Policy **MUST** have a unique identifier.

[R89] The Policy Priority value of a Notify Policy **MUST** be lower than the IdAMP Policy Priority value.

[R90] A Notification Policy **MUST** include criteria for sending a SAAN.

[R91] A Notification Policy **MUST** include criteria for sending a SSAN.

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.

[R92] Recipient value of a Notify Policy **MUST** be *None* or a value from the List of SASE Notifications Recipients Service Attribute.

- [D9] The Service Provider **SHOULD** support the ability to configure different Recipient value for each Notification Policy criteria.
- [D10] 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 ([R40]).
- *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.

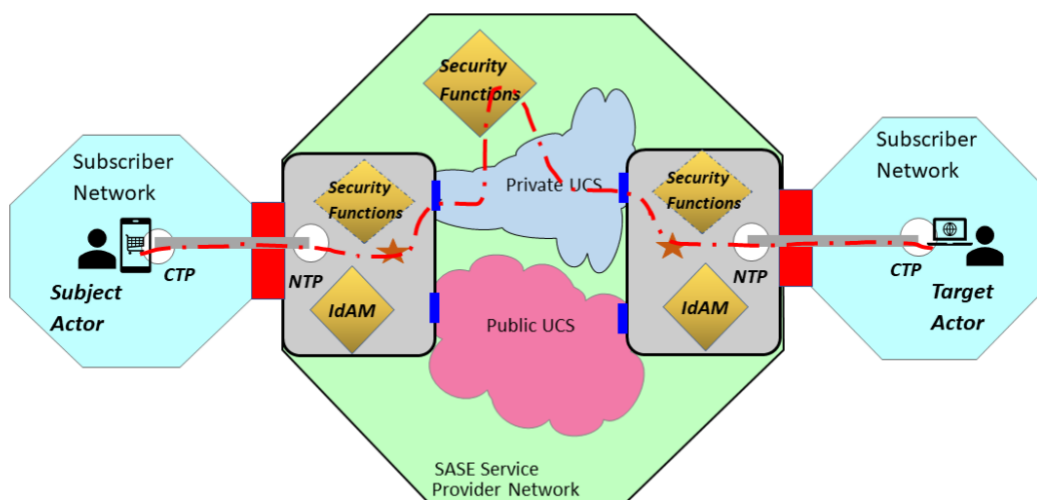
## 11 References

- [1] MEF 61.1, *IP Service Attributes*, January 2019
- [2] MEF 70.1, *SD-WAN Service Attributes and Service Framework*, November 2021
- [3] MEF 88, *Application Security for SD-WAN Services*, November 2021
- [4] MEF 95.0.1, *Amendment to MEF 95: Policy Driven Orchestration*, October 2022
- [5] MEF 118, *Zero Trust Framework for MEF Services*, October 2022
- [6] Gartner, *Say Hello to SASE*, December 2019
- [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 3339, *Date and Time on the Internet: Timestamps*, by Chris Newman and Graham Klyne, July 2002. Copyright © The Internet Society (2002). All Rights Reserved.
- [11] IETF RFC 3629, *UTF-8, a transformation format of ISO 10646*, by Francois Yergeau, November 2003. Copyright © The Internet Society (2003). All Rights Reserved.
- [12] 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).
- [13] 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).
- [14] 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).
- [15] 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.
- [16] 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.

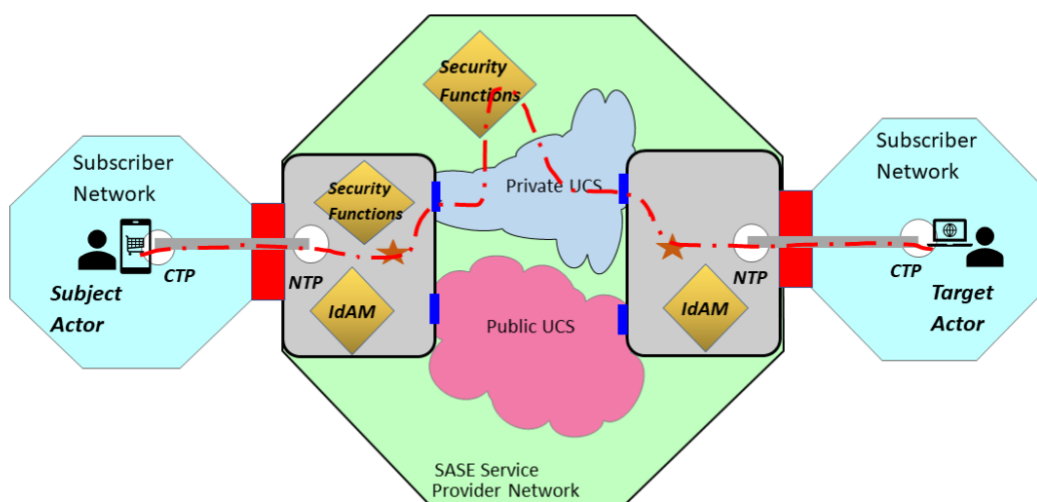
- [17] IETF RFC 8446, *The Transport Layer Security (TLS) Protocol Version 1.3*, by Eric Rescorla, August 2018. Copyright © 2018 IETF Trust and the persons identified as the document authors. All rights reserved.
- [18] NIST, National Vulnerability Database, Common Weakness Enumeration, <https://nvd.nist.gov/vuln/categories>
- [19] MITRE, <https://cve.mitre.org>
- [20] MITRE, ATT&CK, <https://attack.mitre.org>
- [21] MITRE, CAPEC, *Common Attack Pattern Enumeration and Classification*, <https://capec.mitre.org>
- [22] STIX, *Structured Threat Information Expression*, <https://oasis-open.github.io/cti-documentation/>

## Appendix A SASE Session Flow Examples

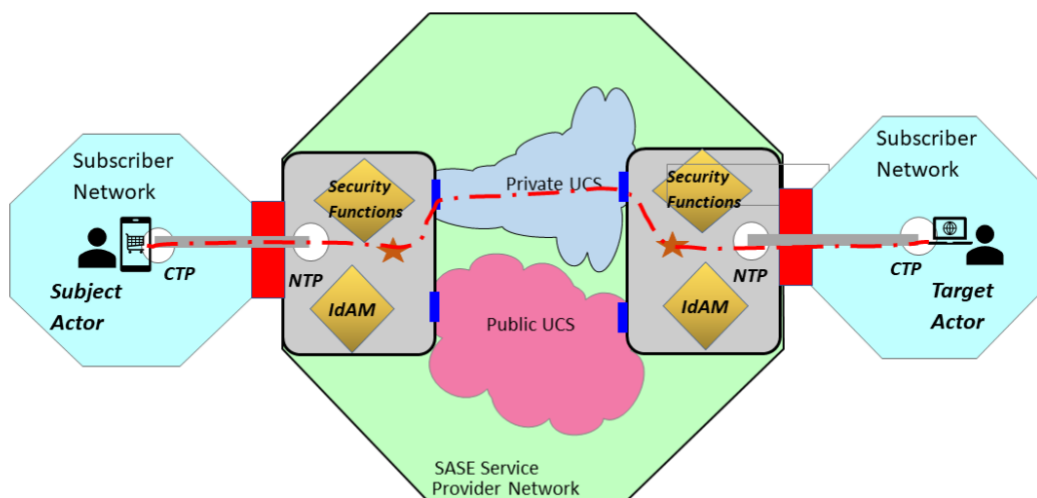
### A.1 Session Flow via Security SASE Edge with subset of Security Functions at Subject and Target SASE Edges



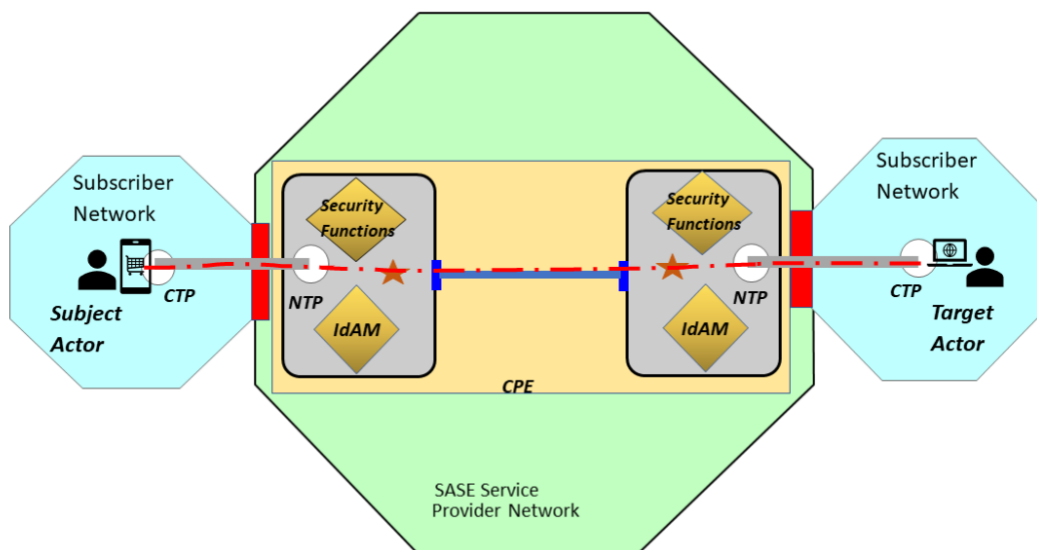
### A.2 Session Flow via Security SASE Edge with Security Functions at Subject SASE Edge but not at Target SASE Edge



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



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





## A.5 Session Flow for Cloud Only delivered SASE Service

