INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2017-2020

SG13-TD481/WP3

STUDY GROUP 13

English only

Question(s): 16/13 Virtual, 20-31 July 2020

TD

Source: Editors

Title: Draft new Recommendation ITU-T Y.OBF_Trust

Purpose: Admin

Contact: Abhay Shanker Verma
Telecom Engineering Centre (TEC)
India
Tel: + 91 9999554900
E-mail: as.verma@gov.in

Contact: Ranjana Sivaram
Telecom Engineering Centre (TEC)
India
Tel: +91 9868136990
E-mail: ranjana.sivaram@gov.in

Contact: Sharad Arora
Sensorise Digital Services Pvt Ltd
Tel: +91 9212109999
E-mail: sharad.arora@sensorise.net

Keywords: Y.OBF_Trust; Output; Q16/13; 20-31 July 2020

Abstract: This document is the output of draft Recommendation ITU-T Y.OBF_Trust: "Open Bootstrap Framework enabling trusted devices, applications and services for distributed diverse ecosystems". It includes the discussion results during the SG13/WP3/Q16 meeting sessions, which was held virtually from 20 to 31 July 2020.

The following table shows discussion results for the contributions.

Document Number	Source	Title	Meeting results
[C921]	Ministry of Communications (India)	Draft recommendation of ITU-T Y.OBF_Trust "Open Bootstrap Framework enabling trusted devices, applications and services for distributed diverse ecosystems".	The contributors/ editors were asked to provide an updated version based on discussions in the meeting dated 20 July 2020
[C921R1-R3]	Ministry of Communications (India)	Draft recommendation of ITU-T Y.OBF_Trust "~~**Open Bootstrap**~~ **Framework** **for ~~open~~ bootstrapping ~~enabling trusted~~of devices and~~,~~ applications for open access to trusted ~~and~~ services ~~for~~ in distributed ~~diverse~~ ecosystems**".	Accepted with modifications.

Observations/ comments during the meeting for further action:

- **Clause 7:** The convention for writing requirements needs to be followed in sub Clause 7.1 General requirement. The term "industry standards" and "network technology layer" used at various places need to be clarified. Further, requirements on functions and the tokens may also be added to ensure better match between Clause 7 and Clause 9.

- **Clause 8:** Naming of the Elements, specifically Application Element, may be relooked to avoid confusion.
- **Clause 9:** The description of the functions may be improved for better clarity. The paragraphs and bulleted texts may not be mixed to describe a function; it should be made uniform. The naming of the functions may be changed for better readability. Further, protocol names may be removed from the diagrams and retained as an example in the text. The bootstrap_token and the bootstrapping function are a very important part of the Recommendation. These may be described and emphasized more. Also, the section on bootstrap_token and other identifiers may be suitably re-organized as per their importance.
- **Clause 10:** The information flows to be rechecked against the functionality as in Clause 9 and the figures need to be updated keeping in view other similar workflows in other recommendations. Actors and functions may not be mixed.
- Updates/ rewording of texts may be done, wherever felt appropriate, for improved clarity and readability using 'NOTE' for examples and other important but non-normative text.

Based on the above observations, contributions may be invited to further improve the document.

Meeting result:

During this meeting, it was agreed to change the title as below.

- Change in the title of the Recommendation from "Open Bootstrap Framework enabling trusted devices, applications and services for distributed diverse ecosystems" to "**Framework for bootstrapping of devices and applications for open access to trusted services in distributed ecosystems**", has been agreed.
- The updated draft document with proposed changes has been accepted.
- However, as additional work for further improvement is necessary, the meeting has decided to postpone a decision for requesting consent to the next SG13 RGM.

_____

Annexure-I

Draft new Recommendation ITU-T Y.OBF_Trust

~~Open Bootstrap~~ Framework ~~for open~~ bootstrapping ~~enabling trusted~~of devices and~~,~~ applications for open access to trusted ~~and~~ services ~~for~~ in distributed ~~diverse~~ ecosystems

Summary

R~~R~~apid advancements in ~~and deployment of~~ ~~new~~ communications and~~-~~ associated technologies ~~and Internet of Things~~ has led to the emergence of distributed ecosystems ~~new ecosystems~~ with a large number of devices, ~~many~~ applications and use cases requiring ~~a trust framework~~ ~~that is~~ ~~which is~~ open access to trusted services ~~to all the entities of~~ ~~thein a new ecosystem~~domain~~.~~ This nature of ~~trust enablement~~ open access to trusted services in distributed ecosystems can be provisioned by using the inherent security capabilities and mechanisms already present in the devices and the underlying networks. This r~~R~~ecommendation provides ~~an Open Bootstrap framework~~ a concept ~~(~~OBF~~)~~ of~~for~~ bootstrapping of devices and applications by network operators who can share the network security capabilities with users and ~~bootstrapping of devices and applications by~~ providers of new devices and services. ~~network operators so as to enable an open and scalable trust infrastructure for the emerging ecosystems the secure provisioning of trusted services by Application Services Providers (ASPs) that have no existing trust relationship with the users.~~ It ~~introduces a concept diagram showing the trust and service interactions between the entities (devices, applications, application service providers (ASPs) and network operators) and then~~ describes the requirements to be fulfilled by the~~se~~ entities of the ecosystem such that they~~may~~ may benefit from the bootstrapping capabilities. ~~The recommendation includes the~~ ~~OBF~~ ~~concept, the requirements of the~~ ~~OBF~~ ~~as well as the pre-requisites for the devices and the application.~~ Based on the requirements, ~~It also includes~~ a reference model as well as ~~describing the OBF elements and~~ a functional architecture is provided, which together ~~describing~~ describe the elements, functions and ~~four functional groups, four~~ reference points needed for provisioning of the bootstrapping capabilities.~~and security parameters.~~ Finally, the recommendation provides the ~~specifications of the reference points and the~~ information ~~work~~flows required to enable~~for~~ the bootstrapping capabilities.~~, authentication and change of OBF realm is also provided.~~

~~This Recommendation is relevant to network operators, IoT service providers and ASPs for deployment of trusted services in the emerging 5G, smart cities, and IoT application/ services domainecosystem.~~

Keywords

Bootstrapping; ~~IoT service provider; OBF; OBF~~bootstrap_token~~; Open Bootstrap Framework; Trust Framework~~; trusted device; trusted application; authentication; authorizsation;

Contents

Page

Draft new Recommendation ITU-T Y.OBF_Trust

~~Open Bootstrap Framework enabling trusted devices, applications and services for distributed diverse ecosystems~~Framework for ~~open~~ bootstrapping ~~enabling trusted~~of devices and, applications for open access to trusted ~~and~~ services ~~for~~in distributed ~~diverse~~ ecosystems

1 Scope

This Recommendation ~~proposes an Open Bootstrap Framework (OBF)~~describes the concept, architecture and information ~~work~~flows for bootstrapping of devices and applications by network operators, by ~~and~~ providing~~es for~~:

- a bootstrapping concept ~~description of~~for entities ~~and t~~requiring open access to trusted services in their interactions~~heir interactions in a concept diagram~~;
- the requirements imposed on the entities for enabling the bootstrapping capabilities;
- a reference model showing the ~~functional~~ elements required for bootstrapping;
- ~~an~~ functional -architecture diagram showing ~~the entities, elements,~~ functions, reference points and security parameters ~~of each element and the specifications of the reference poin~~;~~ts~~ and ~~;~~ ~~and~~

information ~~work~~flows for the operation~~working~~ of the bootstrapping process~~es.es.~~ ~~for secure provisioning of trusted services by Application Services Providers (ASPs) that have no existing trust relationship with the users. OBF can be deployed by the network operators or IoT service providers to enable authentication and authorization of devices for access to trusted services provisioned by ASPs.~~

~~The scope of this Recommendation includes~~

- ~~OBF concept;~~
- ~~requirements for the OBF and OBF elements;~~
- ~~OBF reference model;~~
- ~~OBF functional architecture; and~~
- ~~information workflows of the OBF.~~

~~The recommendation offers a framework for the provisioning of trusted ASP services to the subscribers of network operators who deploy the OBF, by the use of the underlying secure elements and bootstrapping mechanisms.~~

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

~~[ITU-T X.1113] Recommendation ITU-T X.1113 (2007), *Guideline on user authentication mechanisms for home network services*~~

[ITU-T X.1124] Recommendation ITU-T X.1124 (2007), *Authentication architecture for mobile end-to-end communication*

~~[ITU-T X.1158] Recommendation ITU-T X.1158 (2014), *Multi-factor authentication mechanisms using a mobile device*~~

~~[ITU-T X.1311] Recommendation ITU-T X.1311 (2011), *Information technology – Security framework for ubiquitous sensor networks*~~

~~[ITU-R F.1399] Recommendation ITU-R F.1399 (2001), *Vocabulary of terms for wireless access*~~

~~[ITU-T Y.3052] Recommendation ITU-T Y.3052 (2017), *Overview of trust provisioning for information and communication technology infrastructures and services*~~

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1. secure element [b-ITU-T X.1158 (11/2014)]: A dedicated microprocessor system that contains an operating system, memory, application environment and security protocols intended to be used to store sensitive data and execute sensitive applications.

NOTE – A secure element may reside in a universal subscriber identity module (USIM), a dedicated chip in a phone's motherboard, an external plug in a memory card or as an integrated circuit card.

3.1.2. security degree [ITU-T X.1124 (11/2007)]: An identifier (e.g., number) that represents a set of security parameters including at least one authentication mechanism, the crypto algorithms and related parameters to reflect the security requirement of a certain service. It is defined to profile the security requirement of each service**.**

3.1.3. session key [b-ITU-T X.1113 (11/2007)]: The session key is a temporary key used to encrypt data for the current session only. The use of session keys keeps the secret keys even more secret because they are not used directly to encrypt the data. Secret keys are used to derive the session keys using various methods that combine random numbers from either the client or server or both.

3.1.4. trust [b-ITU-T Y.3052 (03/2017)]: Trust is the measurable belief and/or confidence which represents accumulated value from history and the expecting value for future.

Note – Trust is quantitatively and/or qualitatively calculated and measured, which is used to evaluate values of entities, value-chains among multiple stakeholders, and human behaviours including decision making.

3.1.5. user [b-ITU-R F.1399 (05/2001)]: Any entity external to the network which utilizes connections through the network for communication.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1. bootstrapping: ~~Refers to~~ a cryptographic process of binding the user's identity to the keying material provisioned in the secure element of the user's device, enabling the device to communicate securely with trusted services.

~~**3.2.1.** See also clause 3.2.2 of [ITU-T X.1311 (02/2011)].~~

~~**3.2.2. open bootstrap framework (OBF):** A trust framework for provisioning of trusted services by extending the security capabilities of a network technology layer to benefit distributed and unrelated devices and applications.~~

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

3GPP	3rd Generation Partnership Project
AKA	Authentication and Key Agreement
API	Application Programming Interface
ASP	Application Services Provider
FQDN	Fully Qualified Domain Name
GBA	Generic Bootstrapping Architecture
HTTP	Hyper Text Transfer Protocol
IoT	Internet of Things
IPSec	Internet Protocol Security
KYC	Know Your Customer
~~OBF~~	~~Open Bootstrap Framework~~
PSK-TLS	Pre-Shared Key Cipher suites for Transport Layer Security
SIM	Subscriber Identification Module
TLS	Transport Layer Security
UID	Universal Identifier or Public Entity Identifier

5 Conventions

In this Recommendation, requirements are classified as follows:

- The keywords "**is required to/ are required to**" indicate a requirement/ requirements, which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed;
- The keywords "**is recommended**" indicate a requirement, which is recommended but which is not absolutely required. Thus, such requirements need not be present to claim conformance; and
- The keywords "**optionally**" or "**may**" indicate an optional requirement which is permissible, without implying any sense of being recommended. These terms are not intended to imply that the vendor's implementation must provide the option; it means the vendor may optionally provide the feature and still claim conformance with the specification.

6 ~~OBF~~ Concept of bootstrapping

The rapid developments in electronics, communications and applications domain is leading to the emergence of ~~new~~~~distributed~~new ecosystems of users, devices, applications, service providers and network operators, which require open access to trusted services~~a trust mechanism that is open~~ to all the entities in these distributed ecosystems~~domain~~.

~~Users of new age devices and applications require secure mechanisms for accessing trusted services. At the same time, providers of applications and services also require mechanisms for a minimum level of authentication of the Users. From time immemorial,~~ The network operators have played ~~the~~ a critical role in provisioning of ~~providing connectivity to the premises of subscribers,~~trusted services by undertaking the subscriber and device verification ~~and then allowing the connectivity to be used for a diverse set of~~ prior to permitting access to its services. The network operator's~~is~~ trust relationship is possible to be extended to ~~for use by~~ the new users, devices, applications and service providers ~~entities in the distributed ecosystems~~that require trustful interactions for the orderly proliferation of the trusted services.

Devices use secure elements that authenticate themselves to the network's security nodes by using cryptographic processes. The network operator conducts a subscriber verification prior to allowing them the use of its network services and resources. With some enhancements in its network, network operators can add capabilities to on-board application services providers (ASPs) from distributed ecosystems to allow subscribers of its network to securely access the trusted services of the ASP. Network operators can extend the trust from the existing verification of users and devices, by bootstrapping the devices and ASP applications using the network operator trust infrastructure.

The bootstrapping capabilities built into the devices and the network elements that secure the transactions between the subscribers of the network and the services offered by the network operator can be easily extended to provide open access to trusted services to the entities within the distributed ecosystems. The entities can also be provided the facility to change the bootstrapping for trusted services when changing the network operator.

~~The Open Bootstrap Framework (OBF) makes it possible to extend the existing trust relationship between the network operator and its subscribers to enable one to many trust relationships between the many users and the diverse new age service providers.~~

~~The OBF can enable secure service interactions between users and ASPs. This may be done by utilizing the inherent security capabilities of the underlying network technology layer such as authentication, bootstrapping and authorization to create trustful interactions between devices and applications.~~The bootstrapping capabilities of the network operator realm can enable subscribers of the network to access trusted services from any service provider that is on-boarded by the network ~~operator.~~

The concept of ~~the OBF~~ bootstrapping is shown in the diagram below:

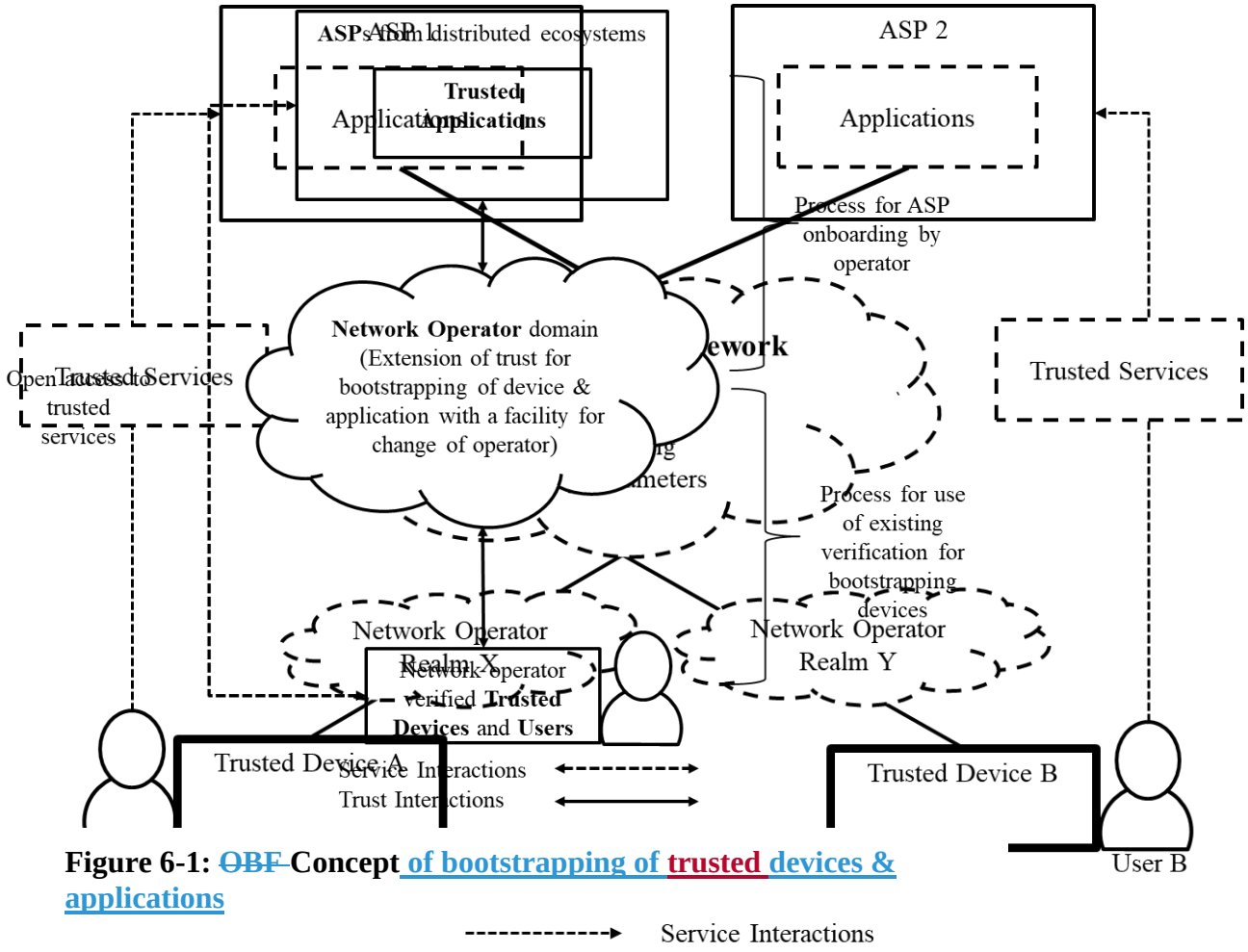


Figure 6-1: ~~OBF~~ Concept of bootstrapping of trusted devices & applications

Service Interactions

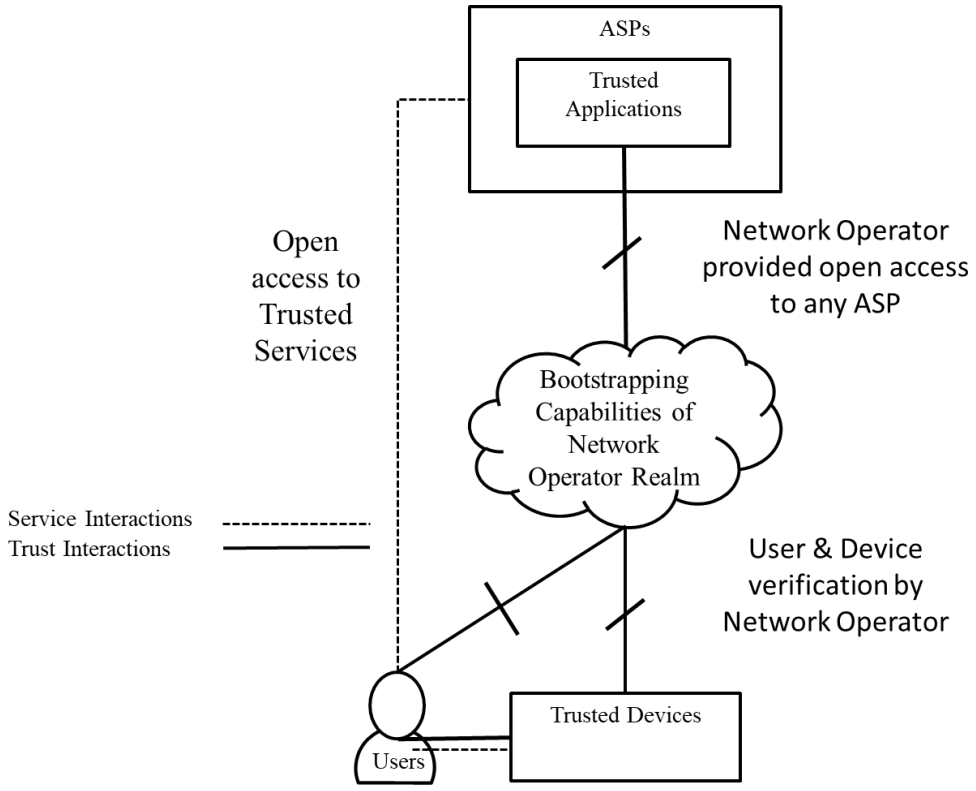


Figure 6-1: ~~OBF~~ Concept of bootstrapping of trusted devices & applications

~~The OBF is a set of requirements, functions, security parameters and mechanisms that can open up the security capabilities of the network layer to all types of trusted devices, applications and services. The OBF can be implemented by any network operator or IoT service provider independent of the underlying network technology. An implementation of the OBF is referred to as an OBF realm. Further, any user of a bootstrapped device can access the applications and services of an ASP by using the security capabilities of the OBF.~~

The bootstrapping concept involves the following entities~~An OBF realm can address the following actors and stakeholders~~:

1. **Trusted device**: A device with an associated secure element which is on-boarded by the network operator.

   NOTE – Secure element is defined in clause 3.1.1.

2. **User~~s~~s:** A person that is a verified subscriber of the network operator, desirous of using trusted services from ASPs. The user provides its credentials to the ASP, whose services it intends to consume, via the network operator or ~~IoT~~ service provider that holds the verified credentials of the user by virtue of an earlier verification process.

   ~~1.~~ NOTE – A subscriber is a person/ entity, who subscribes to the services of a network operator and whose credentials are verified by the network operator before providing services.

1. ~~**Network operator**~~**Network operator:** An entity that provides network connectivity services and undertakes the physical verification ~~process for~~of the subscriber and the device. It can share trust generated from ~~the~~ this ~~trust~~verification information to bridge new relationships

between providers of trusted services and users of trusted devices by deploying the bootstrapping capabilities in its network~~an OBF realm~~.

~~1. Network operator refers to any service provider that deploys an OBF realm, including but not limited to Mobile Network Operators, Virtual Network Operator, IoT Service Providers, etc.~~

2. **Application service~~s~~ provider~~s~~ (ASP):** An entity that develops and offers trusted services and applications, and has a requirement for a minimum level of authentication and authorization prior to the use of its application and services by the users. However, the ASP does not have a direct relationship with the users, unlike the relationship between the network operator and its ~~S~~subscriber. The ASP has an expectation of deriving its trust from the relationship between the network operator and its subscriber.

~~2.~~3. **Trusted ~~a~~Application~~s~~:** ASP ~~A~~application~~s~~ on-boarded by the network operator, which are capable of controlling access to users of trusted devices using cryptographic capabilities.

The ~~mechanisms and workflows~~interactions between ~~that enable~~ the entities that are intended for the establishment of ~~required~~ the ~~level of~~ ~~When the stakeholders engage to establish~~ trust between the entities~~and security in their transactions, these~~ are referred to as the trust interactions. When the entities ~~engage~~interact such as to ~~for~~ use ~~of~~ the functionality of the trusted applications~~In other cases, when the purpose of the engagement is to use the features and functions of the applications~~, these interactions are referred to as the service interactions.

The required solutions to enforce trustworthy interactions between the subscribers, devices and services within the network operator domain already exist. The objective of the next clauses is to provide the requirements, architecture and information flows ~~for network operators~~ to extend the underlying network and device security capabilities for use by ASP trusted applications that are outside the network operator domain. An important consideration for the~~is r~~Recommendation is that it ensures independence from a specific network technology and permits change of network operators for the user and the ASPs.

~~The bootstrapping concept described above apabilities are required to enables a device and network agnostic service and trust interactionstrust framework, creating an open and scalable trust relationshipinfrastructure for new users and service providers of new age services.~~

7 ~~OBF~~ Requirements

7.1 ~~High-level~~General requirements

The following general requirements are imposed on the overall system wherever applicable:

- use identification and numbering of trusted devices and network elements as per the network technology layer;
- use identification and numbering of trusted applications as per industry standards applicable to the distributed ecosystem to which the trusted application belongs;
- use security parameters for mutual authentication by using an identifier from each of the network, the trusted device and the trusted application domain;
- support the existence of multiple network operators offering bootstrap capabilities;
- use industry standard authentication and authorization protocols; and
- use industry standard application protocols for service interactions.

~~In order to benefit from the bootstrapping capabilities, certain requirements have to be fulfilled by the entities namely, user, trusted device, network operator, trusted application and ASP, which are identified below.~~

~~The entities and the domain are required to:~~

- ~~allow any network operator to enable the bootstrapping capabilities regardless of the underlying network technology;~~
- ~~uniquely identify and address the many available bootstrapping realms which may be implemented by multiple network operators;~~
- ~~Hhave a mechanism to identify and address the authentication and authorisation elements in the network operator realm;~~
- ~~identify and address the clientsdevices and the applicationss by using the identifiers of the underlying Information and Communication Technology (ICT) layers to ;~~
- ~~ensure that the applications and functions are accessible over the public Internet;~~
- ~~support existing bootstrapping frameworks, e.g. the 3GPP GBA [b-3GPP TS 33.220];~~
- ~~enable a network technology agnostic identification and addressing of trusted devices; and~~

~~permit a user to be authenticated by any one of the many network operators of which the user is a subscriber.~~

7.2 Requirements for the user

The user is required to:

- ~~complete the~~ registerration ~~with~~ the network operator for bootstrapping facility; and
- ~~bootstrap the device with the network operator; and~~
- subscribe to the trusted services of an ASP.

7.3 Requirements for the trusted device

~~The trusted device may host a secure element to satisfy the security degree of the application. It is recommended that the trusted device has the capability to configure the lifetime and check the validity before using the keying material.~~

The trusted device is required to have:

- capabilities to use its secure element for enabling trust interactions; and
- an application for initiation and management of bootstrapping with the network operator; and
- an application for accessing trusted applications.

~~The trusted device is required to:~~

- ~~have an implementation of secure clients in the device or its connectivity element (e.g. SIM card);~~
- ~~have configurations that make the device OBF aware, and initiate the bootstrapping process, when the OBF application requires it;~~
- ~~support the application specific protocol over the reference point between the device and the application such as HTTP, Message Queue Telemetry Transport (MQTT), Web Sockets or Constrained Object Authentication Protocol (COAP);~~
- ~~support HTTP Digest AKA protocol and optionally others as required by the underlying network technology or application; and~~

- discover, identify, address and connect to the network operator realm for bootstrapping.The trusted device may host a secure element to satisfy the security degree of the application. It is recommended that the trusted device has the capability to configure the lifetime and check the validity before using the keying material.

7.4 Requirements for the network operator

The network operator is required to:

- make network enhancements to support the on-boarding of ASPs and trusted applications;
- allow ASPs to register trusted applications to the network operator without constraints of the network technology or geographical location;
- extend the existing trust relationship and security capabilities between subscriber and network operator to that between the user and the ASP;
- publish the security parameters for bootstrapping of trusted devices and applications;
- support systems and processes to extend the existing user/ device verification for bootstrapping devices to trusted applications; and
- allow the user to change its bootstrapping registration to a different network operator.

The Network Operator is required to:

- verify and on-board users and their trusted devices;
- identify and on-board ASPs whose applications require to be protected from unauthorized usage;
- extend the inherent security capabilities of the underlying network technology layer for the benefit of ASPs;
- be accessible over the public Internet;
- permit authorization and de-authorization of applications for a set of users;
- protect the privacy of the sensitive user / identification information; and
- support industry standard authentication and authorization protocols;

7.5 Requirements for the trusted application

After the bootstrapping is completed, the trusted device and the application can run an application specific protocol, where the authentication of messages will be based on the keying material generated during the mutual authentication.

The trusted application is required to:

- have functions to benefit from network operator offered bootstrapping capabilities;
- have unique identifiers and access control capabilities;
- establish secure connections with the trusted device using the security parameters;
- have session management capabilities;
- manage keys and their lifecycles;
- have the capability to establish secure association with trusted devices;
- support industry standard protocols for key management;

- indicate to the device, the protocol and keying material required by the bootstrapping capabilities;
- implement Diameter / HTTP proxy functionality to act as a proxy towards the OBF realm in which the user is bootstrapped;
- be able to locate the user's OBF realm and communicate securely with the OBF functions;
- acquire the user's security parameters from the OBF realm; and
- implement the security parameters in its security protocol used for creating secure associations between the device and the application. After the bootstrapping is completed, the trusted device and the application can run an application specific protocol, where the authentication of messages will be based on the keying material generated during the mutual authentication.

7.6 Requirements for the ~~trusted application~~ASP

The ASP is required to:

- register with the network operators that offer bootstrapping capabilities;
- register and publish trusted applications with unique identifier and manage access control (e.g., add, delete and modify) configurations;
- expose a registration process for subscribers of network operators to discover and register to its trusted applications;
- register users that have trusted devices and want to use the ASP services; and
- support industry standard authentication and authorization protocols;

The OBF is required to:

- identify and expose address the OBF elements in an OBF realm deployed by a network operators and the OBF elements that have been deployed;
- identify and onboard ASPs whose applications require to be protected from unauthorized usage;
- identify and on-board trusted devices that are authenticated by a network operator;
- expose the inherent security capabilities of any underlying network technology for the benefit of ASPs;
- enable applications to establish secure association with trusted devices;
- identify and address the clients and the applications by using the identifiers of the underlying Information and Communication Technology (ICT) layers;
- be accessible over the public Internet;
- support industry standard protocols for key management;
- support industry standard authentication and authorization protocols;
- support existing bootstrapping frameworks, e.g. the 3GPP GBA [b-3GPP TS 33.220]; and
- enable a network technology agnostic identification and addressing of trusted devices.

The OBF is recommended to:

- permit authorization and de-authorization of applications for a set of users;
- protect the privacy of the sensitive user / identification information;

- allow any network operator to enable the trust framework regardless of the underlying network technology; and
- enable multiple OBF implementations to exist simultaneously.

The OBF may permit a user to be authenticated by any one of the many network operators of which the user is a subscriber.

7.2 Pre-requisites for the trusted devices

In order to use the OBF, the trusted devices are required to:

- have an implementation of secure clients in the device or its connectivity element (e.g. SIM card);
- have configurations that make the device OBF-aware, and initiate the bootstrapping process, when the OBF application requires it;
- support the application specific protocol over the reference point between the device and the application such as HTTP, Message Queue Telemetry Transport (MQTT), Web Sockets or Constrained Object Authentication Protocol (COAP);
- support HTTP Digest AKA protocol and optionally others as required by the underlying network technology or application; and
- discover, identify, address and connect to the OBF realm.

The trusted devices may host a secure element to satisfy the security degree of the application.

It is recommended that the trusted devices have the capability to configure the lifetime and check the validity before using the keying material.

7.3 Pre-requisites for the applications

After the bootstrapping is completed, the trusted device and the application can run an application specific protocol, where the authentication of messages will be based on the keying material generated during the mutual authentication.

The OBF applications are required to:

- be OBF-aware, and be able to indicate to the device the protocol and keying material required to connect to the application;
- implement Diameter / HTTP proxy functionality to act as a proxy towards the OBF realm in which the user is bootstrapped;
- be able to locate the user's OBF realm and communicate securely with the OBF functions;
- acquire the user's security parameters from the OBF realm; and
- implement the security parameters in its security protocol used for creating secure associations between the device and the application.

8 OBF Reference model

Based on the concept and the requirements described above, a reference model is required to address the followingA reference model has been provided which defines :

identify thethe elements required within the entities ; and

define the the requisite –trust and service interactionsreference points required between for the interactions between the elements to meet the requirements stated in the clause above. The reference model is described in the diagram below. entities.

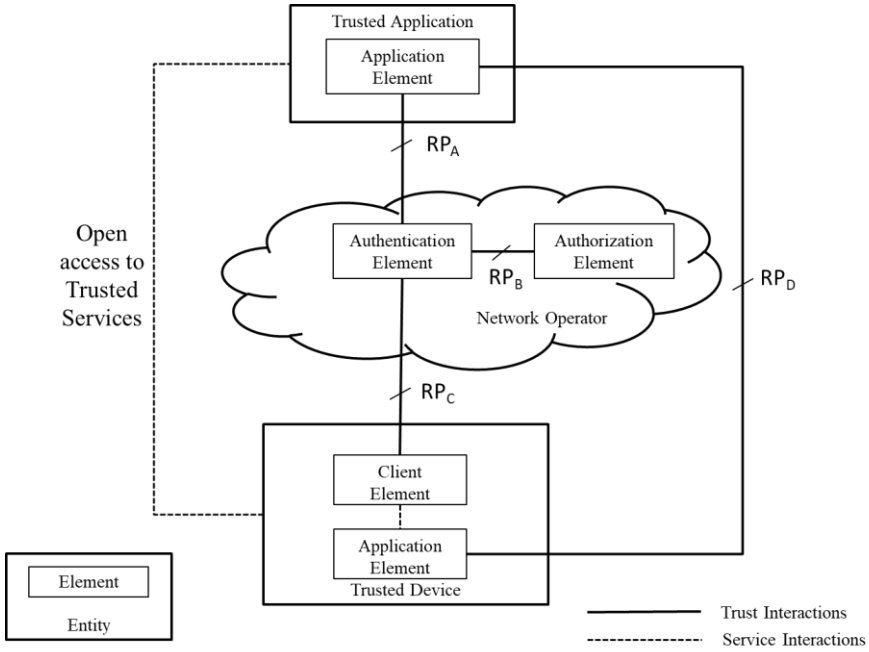
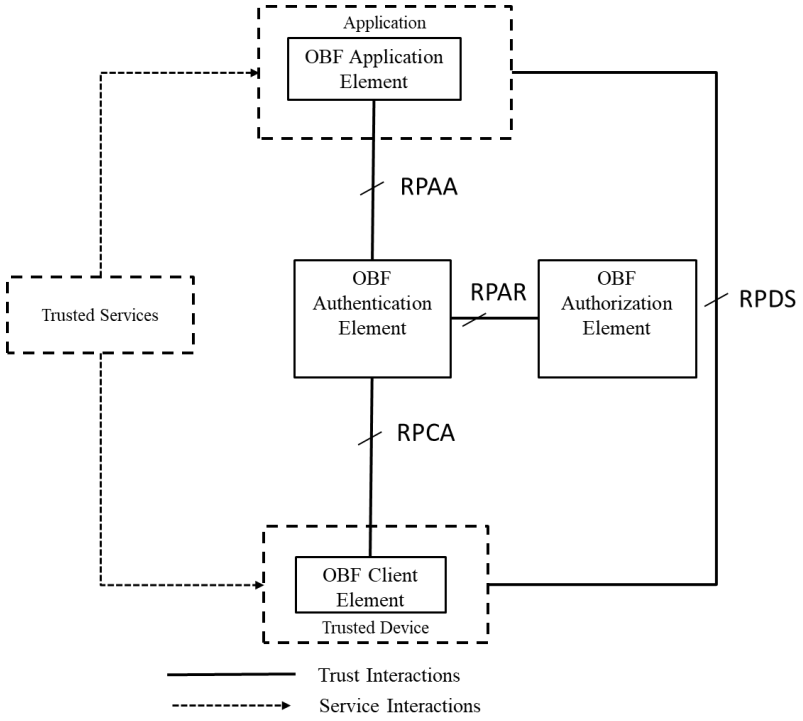


Figure 8-1: ~~OBF~~ Reference model

~~The OBF reference model describes the key elements and the reference points over which the functions interact with each other. The trusted device and the application are also shown in the diagram as these are the beneficiaries of the trust framework.~~

~~The OBF reference model with the required elements~~

~~and reference points to support the service and trust interactions between the entities is shown in the diagram below:~~



8.1 Elements of the trusted device entity ~~OBF Bootstrapping elements~~

The trusted device hosts a client element and an application element for ~~OBF bootstrapping elements shown in the reference model provide the capability for~~ supporting the trust and service interactions, respectively. ~~establishing the required trust and functionality for the user to securely access the trusted application.~~ These elements are ~~enable two types of interaction between the device and the application. The trust interactions establish the required security between the user of the connected device and the application. The Service interactions allow the user to benefit from the use the application which required the secure association.~~

~~The elements of the OBF enable theses interactions, each of which is~~ described below.

8.1.1 ~~OBF~~ Client element

The ~~OBF~~ client element is an application resident in the trusted device, or optionally in its associated connectivity element (e.g. the SIM or the authentication element), that provides the ~~bootstrapping~~ keying material and the ~~bootstrapping~~ authentication mechanism ~~application and the keying material on the device~~ for ~~the~~ bootstrapping ~~of~~ the ~~device~~trusted device to the network operator for purposes of secure access to trusted services. ~~The OBF client element is specified and provisioned by the network operator that is providing the OBF network realm and the associated trust services.~~

8.1.2 Application element of the trusted device entity

The ~~application~~ connection element sets up the secure connection between the trusted device and application using the security enablement provided by the client element.

~~**8.1.2 OBF Authorization element**~~

~~**The OBF authorization element carries out the key management and provides the keying material as per standard security protocols.**~~ **8.2 Elements of the network operator entity**

The network operator adds two important elements, namely i) authentication element and ii) authorization element to address the capabilities of on-boarding ASPs and the trusted applications, and further to allow controlled access to the trusted services from the trusted devices of the subscribers of its network. These elements are described below.

8.2~~1~~.1~~3~~ ~~OBF~~ Authentication element

The ~~OBF~~ authentication element identifies and authenticates the ~~OBF~~ client element of the trusted device using ~~the~~ ~~industry standard keying material~~ authentication protocols (e.g., XXX) and security parameters (e.g., XXX) ~~from the OBF authorization element~~.

8.2~~1~~.2 Authorization element

The authorization element carries out the key and certificate management functions required to support the cryptographic processes for on-boarding trusted devices and applications. It also provides the keying material, support for industry standard protocols (e.g., XXX) and the mapping of the access controls between the trusted devices and applications.

8.3 Application element of the trusted application entity

For ASPs to benefit from the bootstrapping capabilities exposed by the network operator, its trusted applications have an application element that comply to industry standard protocols for bootstrapping, access control and session management.

~~**8.3.1.4 OBF Application element**~~

The ~~OBF~~ application element of the trusted application entity sets up the secure connections between the trusted devices and ~~the~~ applications using the network operator specified industry standard protocols and security parameters~~the security enablement from the other OBF elements~~. The application element is deployed in each trusted application.

8.4~~2~~ ~~OBF~~ Reference points

The reference points are a very important part of the reference model as they make the interactions between the five elements secure, standardised, interoperable and transferable. It is because of the reference points that the bootstrapping capabilities are openly accessible by trusted devices and applications without constraints of network technology or network operator domain.

The four reference points are described below:

~~bootstrapping capabilities require the following OBF specifies four reference points:~~

(a) ~~, namely,~~ RP_{A}~~A~~ - the reference point between the authentication ~~function~~ element of the network operator and the application element ~~hosted in the~~of the trusted application;

(b) ~~,~~ RP_{ARB} - the reference point between ~~OBF~~ authentication ~~function~~ element ~~-~~and the ~~OBF~~ authorization ~~function~~element belonging to the network operator;

(c) ~~,~~ RP_{C}~~A~~ - the reference point between the ~~OBF~~ client element~~function~~ hosted in the trusted device and the ~~OBF~~ authentication ~~function~~element of the network operator;~~,~~ and

(d) RP_{DS} - the reference point between ~~the trusted device and the application~~the application connection element of the trusted device and the application element of the trusted application ~~element~~.

The functionality required to support the features and the flow of information for the service and trust interactions are is described in the clauses below.

9 ~~OBF~~ Functional architecture

A functional architecture is provided to make it possible for the entities to implement

- the required functionality and the interfaces within the network, the trusted devices and applications; and
- the required information and transaction flows that are necessary for enabling the bootstrapping capabilities.
- NOTE - An implementation of the bootstrapping functional architecture by a network operator is referred to as a realm. The instantiated functions within the realm are referred to as nodes. As an example, an authentication element, when instantiated in the network by the network operator entity, will be referred to as the authentication node in the realm of that network operator entity.

~~The elements of the entities identified in the reference model are required to have functions that can meet the requirements listed in section 7. The A functional architecture diagram below describes the functionalities of the entities and their elementsOBF. The OBF Elements are further detailed into various functions along with the specifications of the reference points.~~ The functional architecture

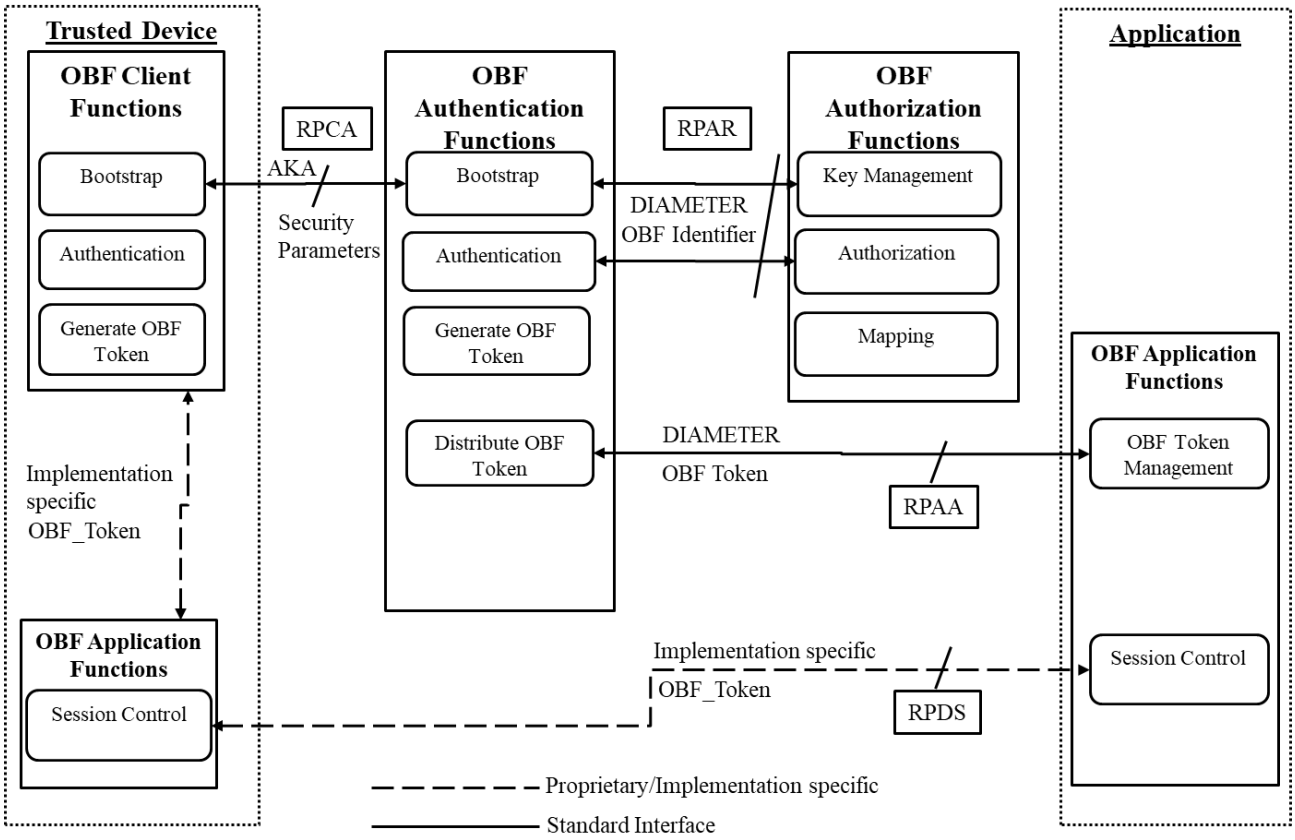
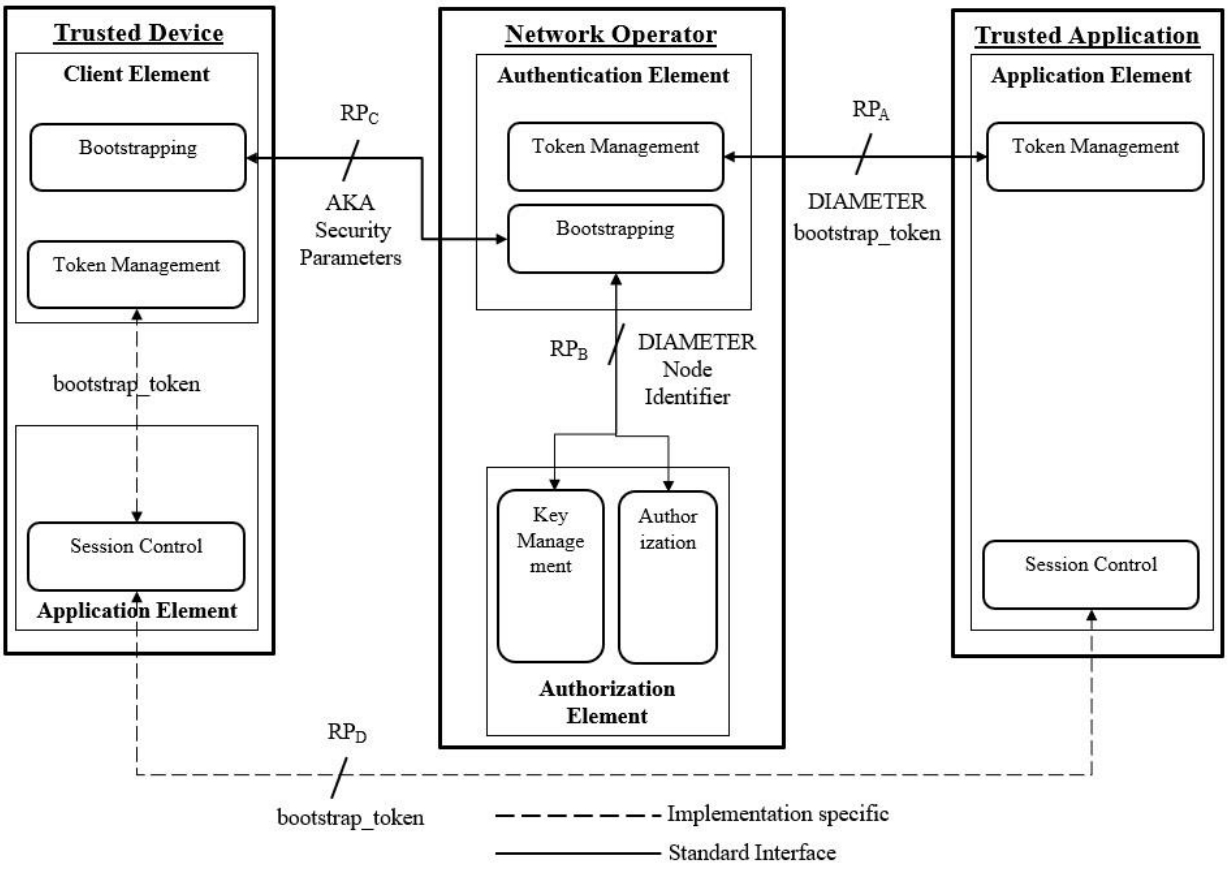


diagram shown in Figure 9-1~~below~~ describes the following:

- the required functions within the elements;
- the reference points required for the interfaces between the functions; and

- the security parameters that are used by the functions over the reference points to enable bootstrapping capabilities.

Figure 9-1: ~~OBF~~ Functional architecture


Trusted Device
Client Element
Bootstrapping
Token Management
bootstrap_token
Session Control
Application Element
RPC
AKA
Security
Parameters
Network Operator
Authentication Element
Token Management
Bootstrapping
RPB
DIAMETER
Node
Identifier
Key
Manage
ment
Author
ization
Authorization
Element
RPA
DIAMETER
bootstrap_token
Trusted Application
Application Element
Token Management
Session Control
RPD
bootstrap_token
Implementation specific
Standard Interface

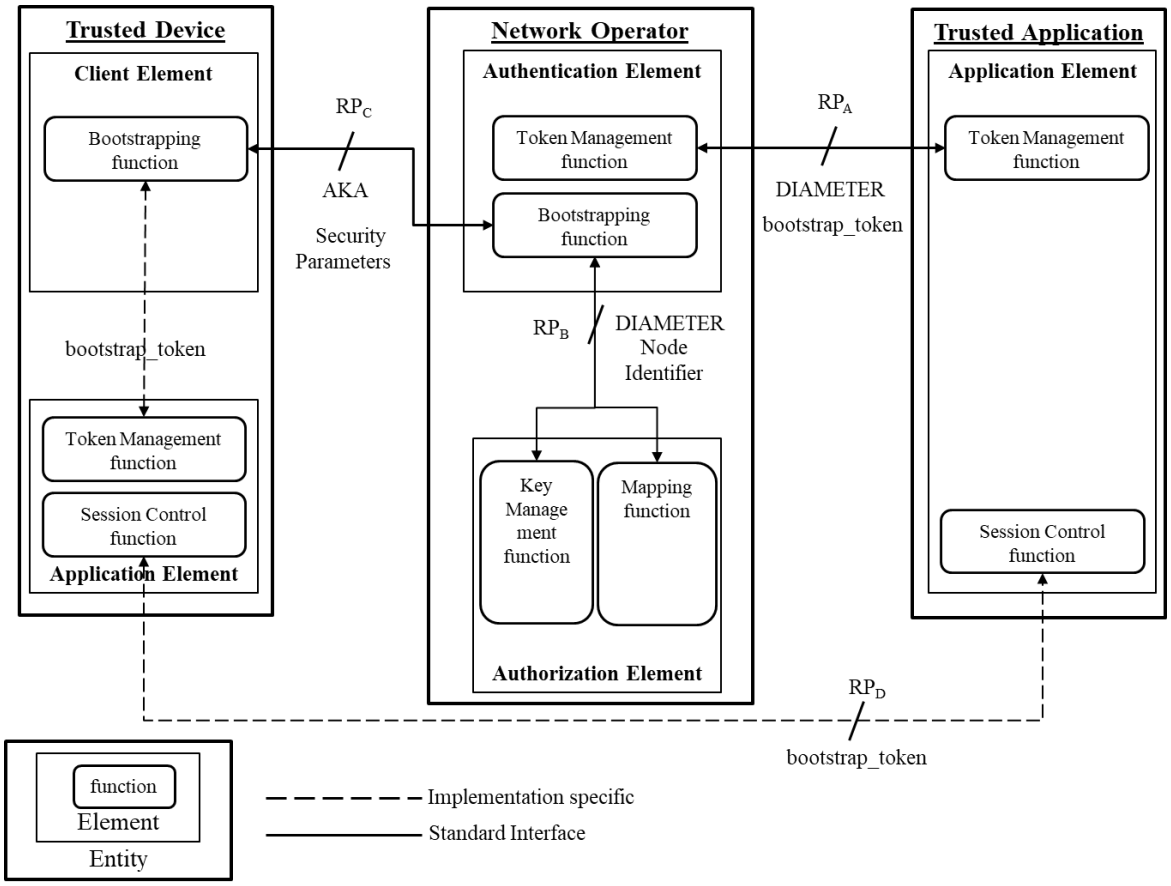


Figure 9-1: ~~OBF~~ Functional architecture

~~The functional architecture consists of the following:~~

- ~~the OBF functionselements;~~
- ~~the reference points; and~~
- ~~the security parameters used within the OBF elements.~~

~~The functionselements, reference points and security parameters are described below.~~

~~9.1 Functions~~

~~9.1.1 Bootstrapping Function~~

~~The Bootstrapping function is implemented in both, the client element of the trusted device entity and also the Application~~

~~9.1.1 Bootstrapping Function~~

~~OBF Functions~~

~~NOTE – When the OBF bootstrapping is deployed in the network, that context implementation is referred to as a realm. The instantiated functions within the realm are referred as nodes. As an example, an Authentication function, when instantiated in the network, will be called the authentication node in the realm of that OBF deployment.~~

~~The following functionalities are supported by all the OBF functions~~functions~~:~~

- ~~identification and numbering of trusted devices and network elements as per the network technology layer;~~
- ~~identification and numbering of trusted applications as per industry standards applicable to the distributed ecosystem to which the trusted application belongs;~~
- ~~security parameters for mutual authentication using an identifier from each of the network, the trusted device and the trusted application domain;~~
- ~~transferability of bootstrapped trusted devices and applications between network operators~~
- ~~industry standard authentication and authorization protocols; and~~
- ~~industry standard application protocols for service interactions.~~
- ~~the OBF numbering;~~
- ~~identification and authentication of each other within the OBF realm(s);~~
- ~~identification and authentication of OBF clients; and~~
- ~~transferability between network operators.~~

The~~Each of the~~ functions ~~and their specific capabilities within the elements~~ are described below.

9.1 ~~.1 OBF A~~Functions of a~~uthentication functions~~element

The ~~OBF Authentication functions~~authentication element ~~are a group of four~~has two functions that enable the bootstrapping of the trusted device. Each of the functions are described below.

~~(a)~~ 9.1.1 ~~Authentication~~ Bootstrapping function

This function provides the functionality for a new registration of a trusted device by ~~way of~~ establishment of new long-term secret key(s) for secure communication. In addition, t~~T~~his function mutually authenticates the ~~OBF client~~ client element and the authentication ~~node~~element, as an enabling step in the process towards generation of long-term keying material within the bootstrapping function. The function is executed over the reference point RP_{C}~~A~~.

The ~~authentication function~~bootstrapping function provides the following functionalities:

~~maintains the list of users, authorized applications and the related subscription parameters;~~

- registers the users and devices that have requested and been authenticated
- protects the use of the network subscriber identity against discovery and misuse;
- supports AKA protocols such that it can support the one used by the underlying network technology layer;
- manages the lifecycle of keys as per the agreed AKA protocol;
- configures and communicates the format of the ~~OBF~~ bootstrapping identifier to the ~~OBF client functions~~client element; ~~and~~
- fetches the data from the authorization element; and
- configures the bootstrapping ~~OBF~~ security parameters in conjunction with the ~~OBF authorization functions~~authorization element and communicates that to the ~~OBF client functions~~client element.

9.1.2 Token management function (aAuthentication element)

~~**(b) Bootstrapping function**~~

~~**This function provides the functionality for a new registration of a trusted device by way of establishment of new long-term secret key(s) for secure communication.**~~

~~**(c) Generate OBF_Token** bootstrap_token function~~

This function provides the functionality for generatinges the ~~OBF_Token~~bootstrap_token by using the agreed security parameters as well as transferring the bootstrap_token to the trusted application, so it can be used by the session functions in the trusted application. ~~, after bootstrapping has successfully been completed, by using the agreed OBF security parameters. The OBF_Token bootstrap_token is specific to the subscription information and the application for which it is generated.~~

This function also securely transfers the bootstrap_token to the trusted application, so it can be used by the session functions in the trusted application.

NOTE – The bootstrap_token is specific to the client element ~~subscription information~~ and the trusted application for which it is generated. The lifetime of the ~~OBF_Token~~ bootstrap_token may vary significantly across various use cases. When the application ~~client~~ element of the trusted device ~~function~~ is invoked, or required to initiate the interaction, ~~with~~ by ~~the~~ a trusted application, the ~~OBF_Token~~ bootstrap_token ~~may be~~is validated to ensure the lifetime of the token has not expired. If the lifetime has expired or if no current ~~OBF_Token~~ bootstrap_token is available or when indicated by the trusted application, the ~~application~~ client ~~function~~ element will use the ~~generate OBF_Token~~ token management function to obtain a new ~~OBF~~bootstrap_t~~T~~oken.

~~**(d) Distribute OBF_Token bootstrap_token function**~~

~~This function securely transfers the OBF_Token bootstrap_token to the application, so it can be used by the session functions in the application.~~

9.~~1.~~2 Functions of ~~OBF authorization functions~~authorization element

~~The OBF Authorization functions~~authorization element ~~has~~are a group of ~~two~~three functions ~~that work together to ensure that applications can be mapped to devices and the security parameters can be agreed between~~ trusted ~~devices and the trusted applications.~~

The authorization ~~functions~~ element has the capability to ~~are the repository~~ store the security parameters for the verified users and the trusted devices belonging to the subscribers of the network operator. It maintains the identity of the ASPs on-boarded by the network operator. It maintains the mapping of the trusted devices that have been authorized to access the trusted applications, and keeps the updated access control list.

The ~~of the UIDs of ASPs that are authorized to provide services. It maintains the list of users, authorized applications and the related subscription parameters and~~ holds the mapping information ~~between applications registered by ASPs and the access rights provided to the users as a list of OBF client function identifiers.~~

~~The authorization function~~ element ~~provides the mechanisms for the network operator to authorize ASPs to offer certain services and users to access the authorized services of the ASP.~~ The authorization element has two functions ~~that work together to ensure that applications can be mapped to devices and the security parameters can be agreed between trusted devices and the trusted applications. Each of the functions~~that are described below.

~~(a)~~ 9.2.1 Key management function

This function provides the management and association of keys and algorithms between the ~~authorization~~ mapping function and the ~~OBF client function~~bootstrapping function of the client element~~.~~ It stores the pre-shared keys or certificates corresponding to the trusted devices and manages the keys and lifecycle of the keying material as per the agreed AKA protocol.

9.2.2 Mapping function

~~(b) Authorization function~~

This function validates if the ~~device~~trusted device can access the trusted application based on the ~~OBF_Token~~ bootstrap_token sent in the authentication request. The function hosts the repository of ~~registered~~ authorized trusted applications that can be permitted for use by the trusted device, and also the mapping of the specific trusted applications that are allowed to be used by ~~OBF client functions~~client element of a trusted device.

The ~~authorization~~ mapping function provides the following functionalities:

- supports the protocols required over the reference point RP_{A}~~A~~;
- provisions the users and trusted applications with the required security parameters;~~ and~~
- responds to the ~~authentication function~~bootstrapping function over the reference points RP_{A}~~A~~ with the authentication vector and user's security parameters such as the key lifetime and user identities;
- addition / deletion of authorized ~~device~~trusted devices / users through standardized API or user interfaces;
- delegation / revocation of access control rights to authorized client element through standardized API or user interfaces;
- addition / deletion of authorized application providers / trusted applications through standardized API or user interfaces and enables provisioning; and
- de-provisioning of authorized users of trusted application through standardized API or user interfaces.

~~(c) Mapping function~~

~~The mapping function is an administrative function to map users, trusted devices and permitted applications. This can be done on an individual level, or based on the agreement between the user and the OBF network operator providing bootstrapping capabilitiesprovider.~~

~~The mapping function provides the following functionalities:~~

- ~~addition / deletion of authorized devices / users through standardized API or user interfaces;~~
- ~~delegation / revocation of access control rights to authorized OBF client functionsclient element through standardized API or user interfaces;~~
- ~~addition / deletion of authorized application providers / applications through standardized API or user interfaces and enables provisioning; and~~
- ~~de-provisioning of authorized users of application through standardized API or user interfaces.~~

9.~~1~~.3 Bootstrapping function of the client element

The bootstrapping function of the client element corresponds to the bootstrapping function of the authentication element and has the same features as described in clause 9.1.1.

The bootstrap function of the client implements the following functionality:

- interact with the secure element of the trusted device;
- support the required AKA protocol;
- store the keying material and select from one amongst several keys for security enablement;
- generate the bootstrap_token as per security parameters negotiated during the bootstrapping process; and
- select from one amongst the several available bootstrap_token corresponding to multiple network operator realms, allowing only one bootstrap_token to be active at a given point in time~~;~~.

~~**OBF application functions**~~

~~The OBF Application functions are deployed in the device of the user and the applications of the ASP. This group of functions enable the session security between the device and the application, each of which is described below.~~

~~**(a) 9.3.1 Session control function**~~

~~This function is application specific. It utilizes the OBF_Token to initiate and maintain a secure session towards the application. The function is implemented within an industry standard session control such as TLS, PSK-TLS, Kerberos, IPSec.~~

~~**(b) 9.3.2 OBF_Token Token management function (Application element)management function**~~

~~The OBF_Token management function receives and stores the OBF_Token within the application element to be used for for securing the future sessions between the device and the application.~~

~~NOTE – The process of storing and use of the token for future session management is implementation specific and out of scope of this Recommendation.~~

9.1.4 ~~OBF client functions~~Functions of the application element

The functions of the application element are deployed in the trusted device and the trusted application. These functions enable establishment and maintenance of the session and session security between the trusted device and application.

The two functions of the~~hree OBF client functions~~application element are ~~, namely bootstrapping function, authentication function and~~ i) ~~OBF_Token generation~~ token management function and ii) session control function the functionality of which is described below.

9.4.1 Token management function of the application element

The token management function of the application element exists in both the trusted device and the trusted application. It corresponds to the token management function of the authentication element. It provides the storage and lifecycle management of the bootstrap_token. In the case of the trusted device, it is responsible for using the secure element for storage of the bootstrap_token. In case of the trusted application, it is responsible for using the storage as per the storage resource provided by the trusted application.

This function also securely transfers the bootstrap_token to the trusted application, so it can be used by the session functions in the trusted application.

~~, correspond to the OBF Authentication Functions with the same functionality as described in section 9.1.1.~~

9.4.2 Session control function of application element

The session control function of the application element exists in both the trusted device and the trusted application. It is application specific. It utilizes the bootstrap_token to initiate and maintain a secure session between the application element of the trusted device and that of the trusted application. The function is implemented within an industry standard session control such as TLS, PSK-TLS, Kerberos, IPSec. It protects the use of the network subscriber identity against discovery and misuse. It supports the application protocol in the reference point RP_D and initiates the request for bootstrap_token when indicated by the trusted application.

~~Together, the three functions enable the OBF client to:~~

- ~~interact with the secure element, of or the trusted device or the connectivity elementexecution environment;~~
- ~~support the required AKA protocol;~~
- ~~store the keying material and select from one amongst several keys for security enablement;~~
- ~~select from one amongst several available authentication functions, allowing services of only one authentication function at a given point in time;~~
- ~~generate and / or retrieve the OBF identifier as per the selected authentication function;~~
- ~~securely store the security parameters including identifiers, subscription information and the OBF_Token;~~
- ~~generate the OBF_Token as per security parameters negotiated during the bootstrapping process;~~
- ~~protect the use of the network subscriber identity against discovery and misuse; and~~
- ~~support the application protocol in the reference point RPDS and initiate the bootstrapping process if indicated by the application.~~

9.~~2~~5 Specifications of ~~OBF~~ reference points

The functionality of the ~~OBF specifies~~ four reference points ~~for use of the bootstrapping capabilities are~~ is ~~, each of which is~~ described below:

9.~~2~~5.1 Reference point RP_{A}~~A~~

~~The reference point RP_{AA} is used to fetch communicate application specific subscription information of the user from the authentication function if requested.~~

The reference point RP_{A} ~~A~~ provides the following functionalities:

- enables secure communication between the authentication element and the application element;
- allows the transfer of the ~~user's~~ subscription information related to the trusted device to enforce access control policies between trusted devices and ~~the~~ applications;
- supports the DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol;
- ~~enables secure communication between the authentication function and the application;~~
- allows the application to send its address (e.g. FQDN), public entity identity (e.g., UID), basic key material (e.g., a shared secret or a public-key certificate), entity service permission flag, supported authentication mechanisms and the authentication inquiring and key generation mechanism to the ~~authentication function~~bootstrapping function;

- allows the ~~authentication function~~token management function of the authentication element to ~~verify that~~transfer the bootstrap token to the token management function of the application element of the trusted application ~~is authorized to obtain the identifiers, key material and subscription information for a user~~;
- allows the token management function of the application element to indicate to the ~~authentication function~~token management function the authentication element the eligibility of the bootstrap token for a single or multiple application ~~or several applications for which it requires user identity and security parameters~~;.
- ~~allows the application to obtain a selected set of application specific user security parameters;~~
- ~~allows the transfer of the OBF~~bootstrap token from the authentication function to the application.~~; and~~
- ~~allows the application to indicate to the authentication function the protocol identifier of the RP$_D$S security protocol for which it requires the keying material.~~

9.~~2~~5.2 ~~RPAR~~RP$_B$

The reference point ~~RPAR~~ RP$_B$ enables the mutual authentication between the bootstrapping function of the authentication element and the functions of the authorization element. It supports the DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol.

It provides the subscription information regarding the ~~OBF~~ client ~~functions~~ elements when ~~users~~ trusted devices request access~~attempt to access~~ to trusted ~~certain ASP~~ applications. The reference point also provides the keying material for the ~~OBF~~ client ~~functions~~ element ~~during~~ for the bootstrapping ~~mechanism~~~~process~~information flow.

~~The reference point RPAR RP$_B$ provides the following functionalities of :~~

- ~~identification~~ and validation that a client is It maintains the permissions ~~tted to use the bootstrapping to~~for the client element to access certain trusted applications. ~~ASPs and as well as mutual authentication between the authentication function and authorization function on supported DIAMETER [b-RFC 6733] and [b-RFC 7155] protocol.;~~
  - ~~the transfer of security parameter required for bootstrapping;~~
  - ~~the transfer of subscription information to establish the access control policies between trusted devices and the applications;~~
  - ~~the authentication function~~bootstrapping function ~~to request bootstrapping information for specific users; and~~
  - ~~the authorization function to send the user's security parameters to the authentication function~~bootstrapping function~~.~~

9.~~2~~5.3 RP$_{CA}$

The reference point RP$_C$~~A~~ provides the interfaces for the bootstrapping of the ~~OBF~~ client ~~functions~~ element to the ~~OBF~~ authentication ~~functions~~element.

- The reference point RP$_C$~~A~~ provides the following functionalities:
  - supports the HTTP Digest protocol [b-RFC7616] and may ~~, it may~~ optionall~~ly~~ support other industry standard protocols ~~as well~~;
  - uses the agreed AKA for authentication between authentication element and the client element; and
  - ~~establishes the identity of the OBF client function~~ element ~~of a trusted device to the authentication function~~element~~;~~

- ~~supports the HTTP Digest protocol [b-RFC7616], it may optionally support other protocols as well;~~
- ~~uses the agreed AKA for authentication between authentication function element and the OBF client functionelement;~~
- ~~transfers the identification of the OBF client function element using the OBF identifier;~~
- ~~supports the bootstrapping process between the OBF client function element and the authentication functionelement;~~
- ~~identifies and mutually authenticates the trusted device and the application using the OBF client function element and the authentication functionelement; and~~
- establishes the ~~agreement for use of~~ security parameters and ~~algorithms~~AKA ~~-~~for generation of the ~~OBF~~bootstrap_token ~~between the authentication function element and the OBF client functionelement~~.

9.~~2~~5.4 RP$_{D}$~~S~~

The reference point RP$_{D}$~~S~~ supports the ~~protocol required~~interfaces for the secure interaction between the ~~device and the application~~trusted device and application.

The reference point RP$_{D}$~~S~~ provides the following functionalities:

- supports the application-specific protocol between ~~the trusted device and the application~~the trusted device and application;
- sends the indication from the trusted application to the trusted device that a valid or new ~~OBF~~bootstrap_token is required prior to connecting to the trusted application;
- supports the use of the ~~OBF~~bootstrap_token for creating the secure association between ~~the trusted device and the application~~the trusted device and application; and
- ~~allows the application to indicate to the application client function, the invalid OBF_Token for the required authentication;~~
- ~~enables the negotiation and selection of the key between the client function and the application;~~
- ~~uses a security protocol identifier as required by the underlying network technology layer;~~
- allows the ~~application~~application element to signal to the ~~application~~ client element ~~function~~ regarding lifecycle management of keys; ~~and~~
- ~~enables the use of the OBFbootstrap_token for securing the association between the application client function and the application.~~

9.~~3~~6 Security parameters

The security parameters include identifiers, subscription information and the keying material which together create the~~i.e.~~ ~~OBF~~bootstrap_token. The purpose of the identifiers is to uniquely identi~~f~~y and address the trusted devices and the ~~OBF nodes~~~~elements~~nodes in a~~n~~ network operator~~OBF implementation~~~~network operator~~ realm. The purpose of the subscription information is to authenticate and authorize the secure interactions between ~~users~~ trusted devices and applications~~and ASPs via the network operator~~.

The security parameters are implementation specific, and can change significantly from one deployment to another. They are determined by several factors, including but not limited to, the ~~OBF~~ deployment model, the underlying network technology, the AKA protocol, the numbering/ identification mechanism of the network and internet layer, the service type and the security degree required for the use case, etc.

9.~~3~~6.1 Identifiers

The ~~OBF~~ identifiers uniquely identify a~~n~~ ~~OBF~~ client ~~function~~element in~~,~~ a ~~bootstrapped~~ trusted device to an authentication ~~function~~ element and the application element. The ~~OBF provides for the~~ following identifiers are relevant:

a. ~~OBF~~ N~~n~~ode identifier;
b. ~~OBF~~ Trusted device~~client~~ identifier;
~~b.~~c. Trusted application identifier; and~~and~~
~~c.~~d. ~~OBF~~ security protocol identifier.

The description of the various identifiers ~~is provided~~are as below.

(a) **~~OBF~~ Node identifier**:

The ~~OBF~~ node identifier comprises such minimum connection and security attributes that can uniquely address and fully support the ~~OBF~~ authentication ~~function~~ element from one of many in multiple technology domains. As an example, an authentication element ~~function~~ will require the node's FQDN~~,~~ ~~and~~ the Global Title Address and the associated AKA to fully qualify the requirement of the ~~OBF~~ node identifier, when such a node is deployed in a GSM network. The ~~OBF~~ node identifier provides an implementation dependent~~-~~ address, connection and security ~~information~~ details of the ~~authentication function~~elements deployed in a network operator realm.

(b) **~~OBF~~ Client identifier**:
It is an identifier of the ~~OBF~~ client ~~function~~ element or the trusted device, which includes at least a network technology identifier, ~~underlying~~ network ~~layer~~ identifier ~~of the device,~~ and IP layer identifier of the ~~device~~trusted device.

(c) **Trusted application identifier**:
It is an identifier of the trusted application that includes an FQDN and a unique identifier provided by the network operator or an application registry.

~~(c)~~(d) **~~OBF~~ Security protocol identifier**:
It is an identifier, which is associated with a security protocol over reference point RP_D~~S~~. The ~~OBF~~ security protocol identifier ~~is a string of five octets. The first octet denotes the organization, which specifies the security protocol. The remaining four octets denote a specific security protocol~~is defined by the network operator and it is ~~-~~network technology specific. NOTE ~~-~~ As an example, in case of 3GPP it is as per Annex-H of [b-3GPP TS 33.220] ~~within the responsibility of the organization~~.

9.~~3~~6.2 Subscription information

Subscription information [ITU-T X.1124 (11/2007)] between a user and its home network contains the user's private entity identifier (e.g., Mobile Station International Subscriber Directory Number (MSISDN)), the basic key material (e.g., a shared secret or a public-key certificate) and its lifetime, entity service permission flag (i.e., whether it is allowed to request a specific service), the supported authentication mechanism(s) (e.g., HTTP authentication and key agreement, Diffie-Hellman based authentication mechanisms, a biometric authentication mechanism, etc.), and the authentication inquiring and key generation mechanism (e.g., GBA, Kerberos, Mediation), etc. Subscription information between an ASP and a network operator contains the ASP's identity information and public entity identifier (e.g., UID) according to the service, optionally the basic key material (e.g., a shared secret or a public-key certificate) and its lifetime, entity service permission flag (e.g., whether it is allowed to provide a specific service), the supported authentication mechanisms (e.g., certificate-based TLS authentication mechanism, PSK-TLS, IPSec), and the authentication inquiring and key generation mechanism (e.g., GBA, Kerberos, Mediation), etc.

The subscription information related to the user and its authentication ~~function~~ is delivered to the ~~OBF~~ client element ~~function~~ from the authorization element ~~function~~ via the authentication element ~~function~~ during the bootstrapping process. The subscription information related to the trusted application (e.g. access to application allowed, type of certificates ~~which~~ that may be issued) is sent to the ~~OBF~~ client element~~function~~.

In addition, the subscription information contains a mechanism for key selection, which is used in the ~~OBF~~ client element ~~function~~ to mandate the usage of either the trusted device-based key or the external secure element-based key or both.

9.~~3~~6.3 ~~OBF~~Bootstrap_token~~b~~Bootstrap_token

The ~~OBF~~bootstrap_token binds the user's identity to the keying material in the reference points. The bootstrap_token ~~OBF_Token~~ is a session key, independently generated in the ~~OBF~~ client ~~function~~ element of the trusted device ~~/ user equipment (UE)~~ as well as in the authentication ~~function~~ element based on an agreed security schema between the client element ~~device~~ and the authentication ~~function~~element. The bootstrap_token ~~OBF_Token~~ is generated by using the security parameters negotiated as part of the bootstrapping process. It is used for establishing a secure session between the trusted device and ~~the~~ application. ~~The timestamp of the bootstrap_token OBF_Token is synchronized and controlled by the authentication functionelement.~~

The characteristics of the bootstrap_token ~~OBF_Token~~ are as follows:

(a) It binds the user identity to the keying material used in the reference points;

(b) It is the globally unique identifier of the realm of the ~~OBF~~ network operator in which it is issued;

~~(c) It supports any underlying network technology;~~

~~(d) It identifies the realm of the OBF network operator in which it is issued;~~

~~(e)~~(c) It serves as a temporary identifier of the ~~user~~trusted device to which it is issued; and

~~(f)~~(d) It ~~is a key~~ identifies the key used in the cryptographic processes~~r in protocols used in~~ over reference point RP_C~~A~~ and RP_D~~S~~;

~~(g) It enables the application to detect and address the authentication function element that has sponsored the bootstrap_tokenOBF_Token; and~~

~~(h) It has a format that is usable by the underlying network technology layer bootstrapping capabilities.~~

10 Information ~~work~~flows

This clause specifies procedures for ASPs to access bootstrapping capabilities exposed by network operators in accordance with the functional architecture identified in clause 9. It describes xxx major flows that enable trust and service interactions within the ecosystem entities, namely, i) Network operator bootstrapping capability exposure ii) ASP on-boarding flow iii) bootstrap_token generation flow iv) trusted device and application session flow iv) Mapping of trusted device and application v) Authentication and authorisation flow vi) Operator change flow

10.1 Network operator bootstrapping capability exposure

In order to allow its subscribers to access an ASP's trusted applications, the network operator must enhance its network with certain nodes that implement the bootstrapping functions described in the

clause 9 above. The network operator provides the information for users and ASPs to opt for the bootstrapping capability in the network.

The flow is described in the diagram below:

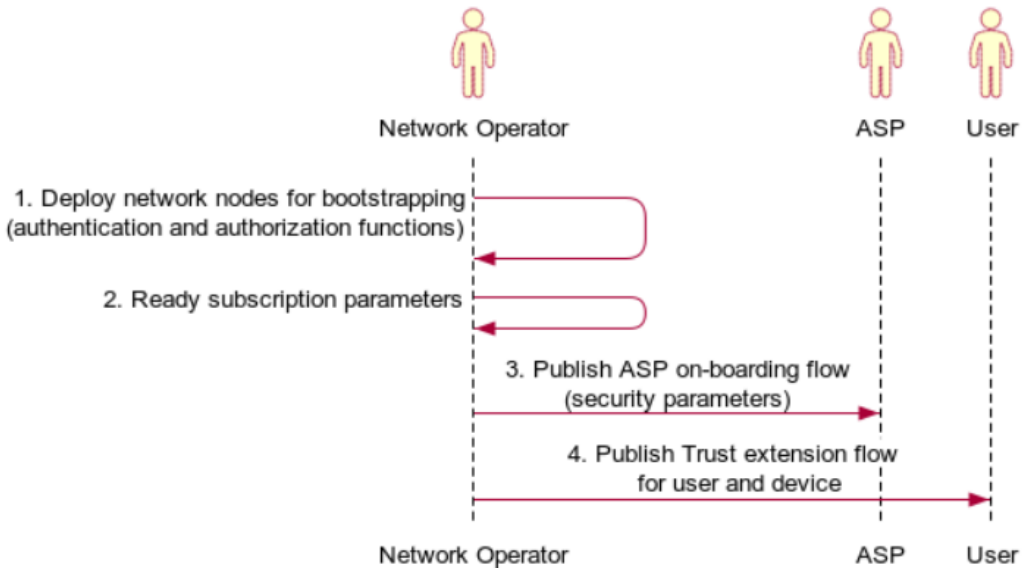


Figure 10-1: Network operator bootstrapping capability exposure

Step 1: Network operator deploys the authentication and authorisation functions in its network

Step2: Network operator defines and readies the security parameters as per industry standards

Step 3: Network operator publishes the ASP registration process with its security parameters. The ASP configures the trusted application with network operator node identifiers to uniquely identify and address the elements in the network operator realm, and complies to the bootstrap_token containing the subscription information to authenticate and authorize the secure interactions between trusted devices and applications via the network operator nodes.

Step 4: Network operator publishes the process for device bootstrapping for subscribers who wish to access ASP trusted applications.

~~The implementation of bootstrapping capabilities outlined above makes it possible for ASPs to offer trusted services to users of trusted devices by simply registering its trusted applications to the network operator providing the bootstrapping capabilities.~~

~~This clause specifies important~~ information flows ~~between the entities to support the functionality of the bootstrapping capabilities, procedures for the trust and service type interactions in accordance with the functional architecture outlined in the section 9. Four major~~ information flows ~~are described, two for bootstrapping and authentication, and another two for changing the OBF~~ network operator ~~realm whilst using symmetric or asymmetric keys.~~

~~**The details of the four information workflows are described in the sections below.**~~**10.2 ASP on-boarding flow**

The ASP on-boarding procedure enables ASPs to register themselves and their trusted applications on the network operator authentication and authorisation nodes. The flow readies the trusted applications for registration and controlled access by trusted devices and subscribers of the network operator.

The procedure for ASP on-boarding is shown in the diagram below:

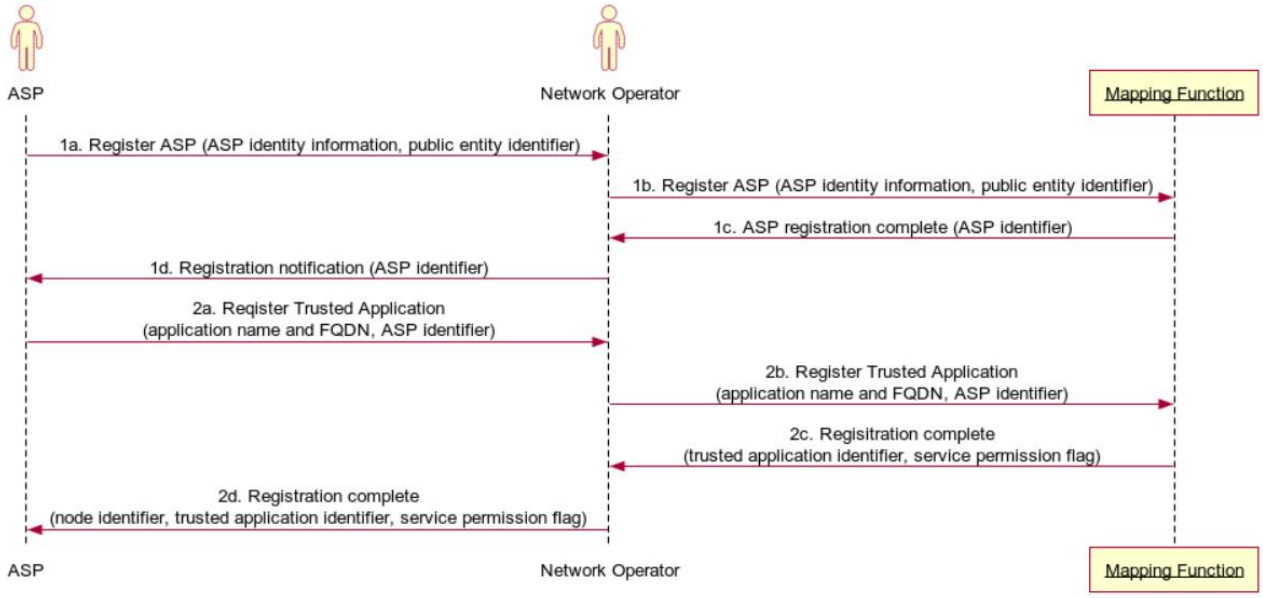


Figure 10-2: ASP on-boarding flow

Step 1a: The ASP initiates the registration with the network operator by providing its identity information, public entity identifier (e.g. UID).

Step 1b: The ASP identity information and public entity identifier are added to the mapping function of the network operator securely.

Step 1c: The mapping function generates a unique identifier for the ASP and sends a notification of successful registration.

Step 1d: The network operator sends ASP its unique identifier upon successful registration.

Step 2a: The ASP initiates the registration of its trusted application with the network operator by providing the application name and FQDN.

Step 2b: The ASP trusted application name and FQDN is added to the mapping function of the network operator securely.

Step 2c: The mapping function generates a unique trusted application identifier and sends a notification of successful registration

Step 2d: The network operator sends the node identifier, trusted application identifier, service permission flag corresponding to the trusted application to the ASP

10.3 Trust extension flow for user and device

The network operator and the ASP inform the network operator's subscribers about the ASP trusted applications. For users that express an interest in ASP's trusted application(s), the network operator checks the user's existing verification information and shares the network identifiers with the ASP if the user credentials merit access to the trusted application(s). The ASP can then assign appropriate permissions for the user access to the trusted application(s).

The process is shown in the diagram below:

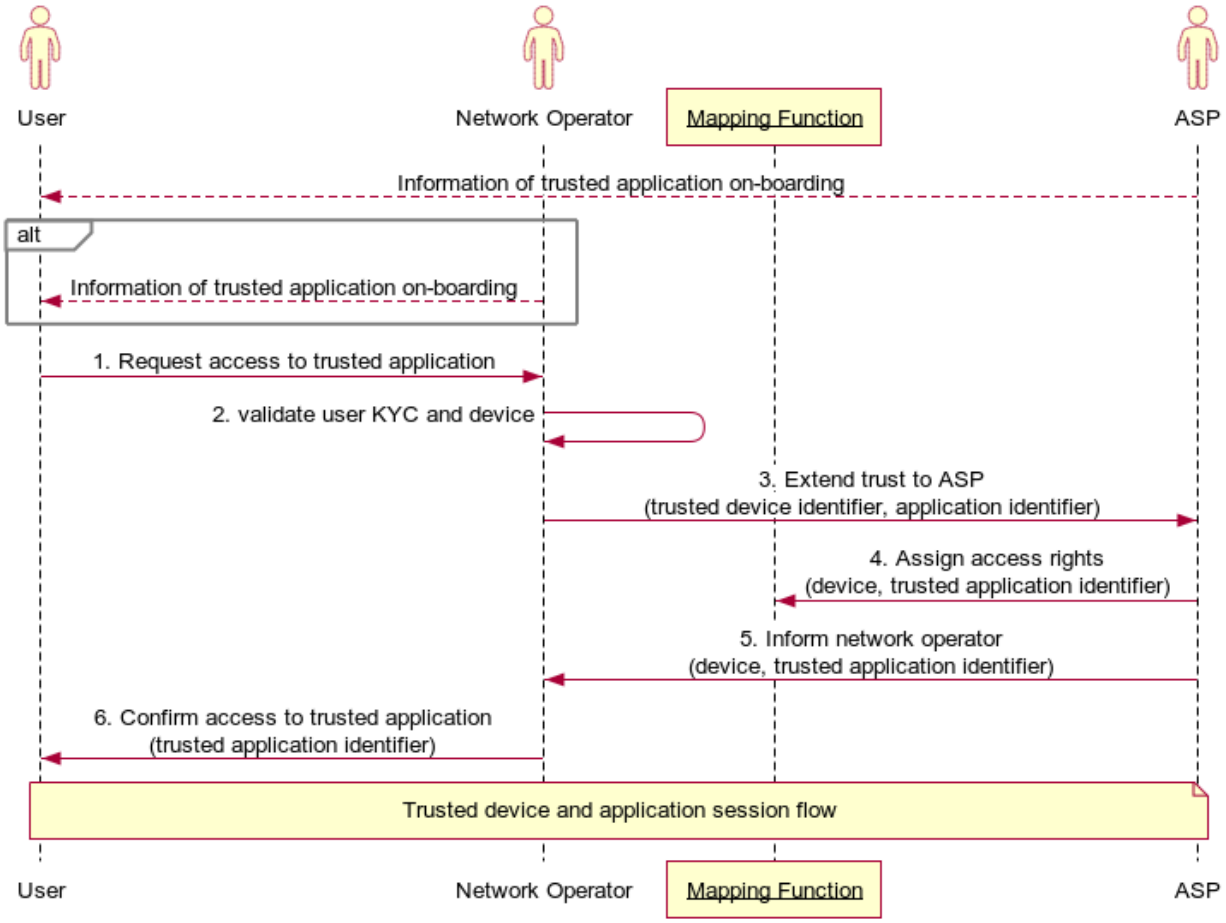


Figure 10-3: Trust extension flow for user and device

Step 1: User requests access to ASP trusted application.

Step 2: Network operator checks user's existing verification and device information.

Step 3: Network operator extends the interested user's trusted device information to the ASP.

Step 4: ASP provisions access rights to the trusted device identifier for the trusted application identifiers on the network operator's mapping function.

Step 5: ASP informs the network operator about the provisioning of access rights to the trusted device for the trusted application.

Step 6: Network operator confirms to the user regarding the access and the trusted application identifiers

After this stage, the trusted device can follow the trusted device and application session flow to initiate the service and trust interactions.

10.~~4~~1 Bootstrap_token generation flow

The bootstrap_token generation flow enables the generation of the bootstrap_token. It is invoked when a trusted device requests a session with a trusted application but the token management function does not find a valid bootstrap_token to use for the creation of a secure session.

The process is shown in the diagram below:

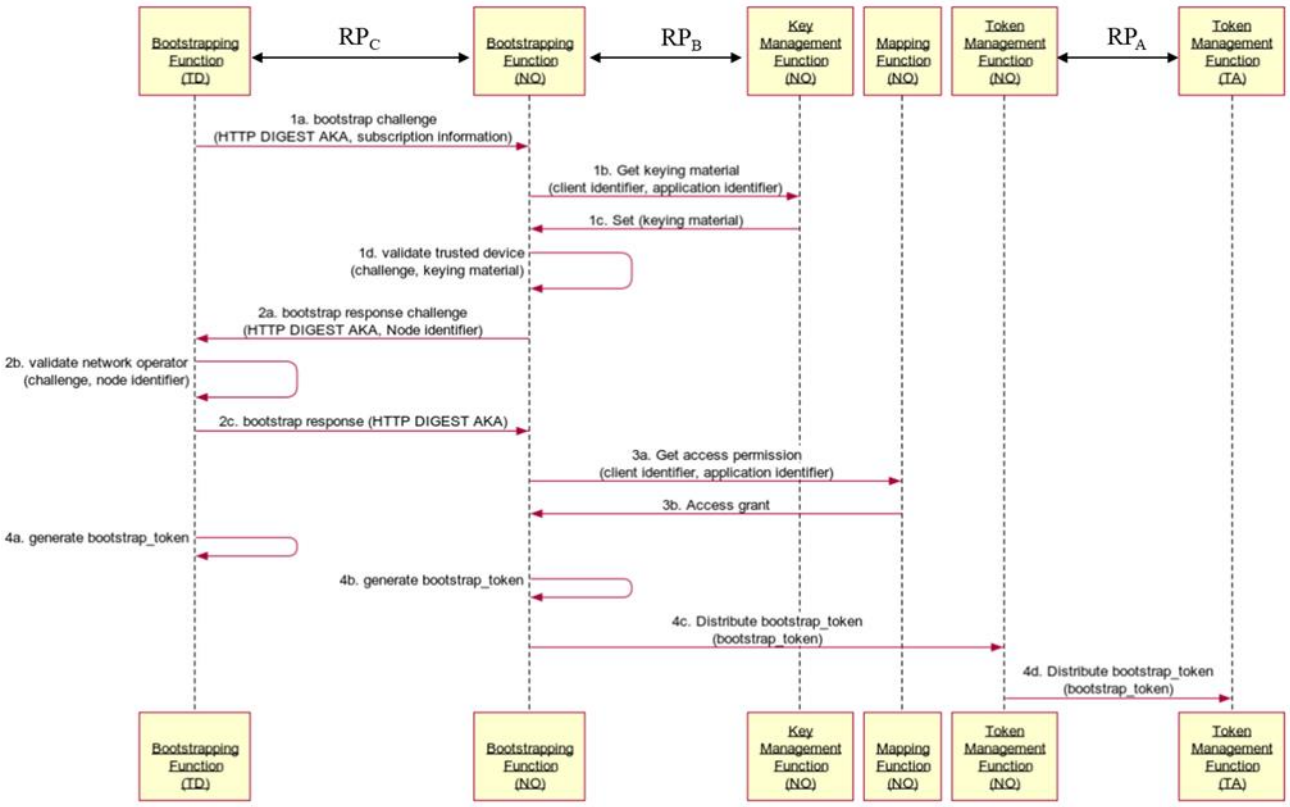


Figure 10-4: Bootstrap_token generation flow

Step 1a. At the start of the bootstrap_token generation process, the bootstrapping function of the trusted device uses the capabilities of the reference point RP_C to send a challenge to the authentication element using the identifiers of the trusted device and the subscription information of the trusted application.

Step 1b. The bootstrapping function of the network operator uses the capabilities of the reference point RP_B for requesting the key management function for the keying material corresponding to the client element and the application identifier

Step 1c. the key management function sets the keying material for the bootstrapping function of the network operator

Step 1d. The bootstrapping function of the network operator validates the credentials of the client element for based on the keying material set in step 1c above using the HTTP Digest/AKA;

Step 2a: The bootstrapping function of the network operator sends back a challenge to the client element using its node identifier as a part of the security challenge.

Step 2b. The bootstrapping function of the trusted device validates the challenge from the network operator.

Step 2c. The bootstrapping function of the trusted device generates a response based on the challenge and the HTTP Digest/AKA.

Upon the successful mutual authentication, the bootstrapping functions check if the given trusted device is authorized to use the bootstrapping services for ~~the~~ a given trusted application.

Step 3a. The bootstrapping function of the network operator requests the mapping function for access permissions by supplying the client identifier and the trusted application identifier information.

Step 3b. The mapping function approves the requested access if the permissions for the trusted device to access the trusted application are set by the ASP as part of the ASP registration process.

Step 4a. Upon successful confirmation in Step 3b, the bootstrapping function of the client element generates the bootstrap_token.

Step 4b. Upon successful confirmation in Step 3b, the bootstrapping function of the network operator generates the bootstrap_token.

Step 4c. The bootstrapping function of the network operator transfers the bootstrap_token to the token management function of the network operator using a proprietary interface.

Step 4d. The token management function of the network operator uses the capabilities of the reference point RP_A to transfer the bootstrap_token securely to the token management function of the trusted application.

At this stage, the token management functions in each of the client element, authentication element and the application element are updated with the newly generated bootstrap_token.

NOTE - The bootstrap_token generation flow shown above shows the use of symmetric keys for the establishment of secure connections; the flow with asymmetric keys is similar, with the exception that, in place of pre-shared keys the public keys are used for bootstrapping. That flow is not shown explicitly.

10.5 Trusted device and application session flow

The trusted device and application session ~~Bootstrapping & authentication workflows~~

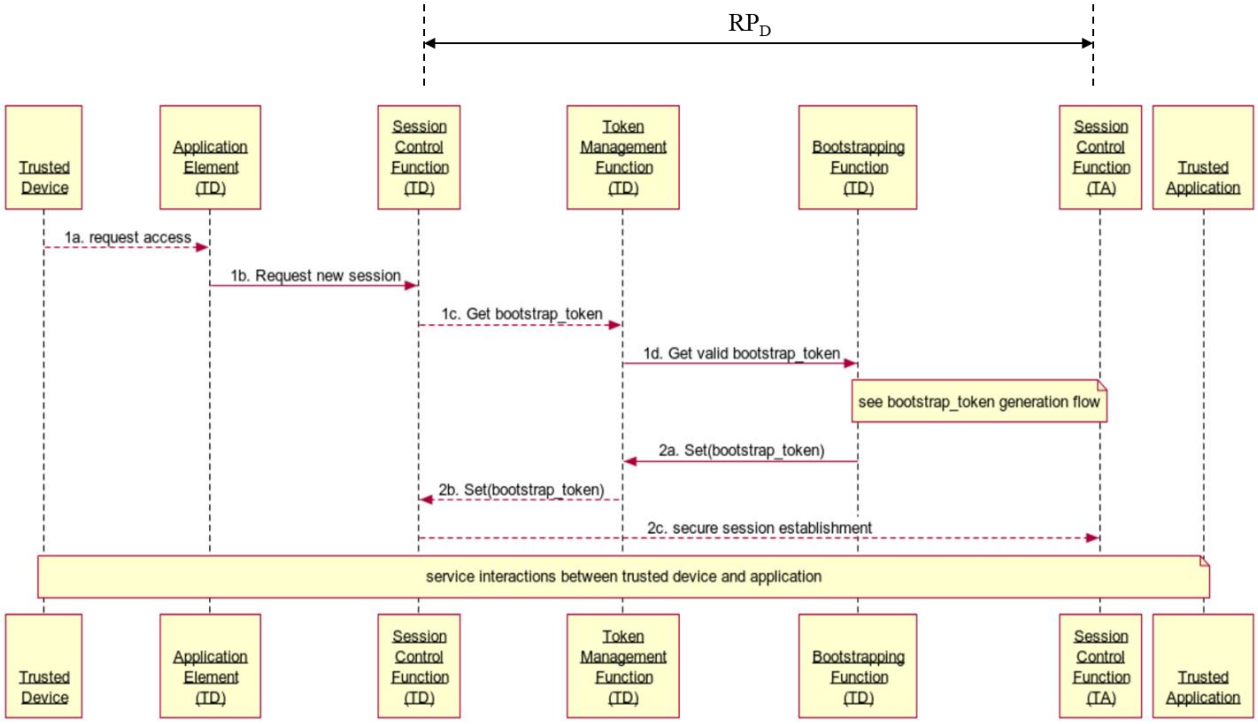


Figure 10-5: Trusted device and application session flow

flow establishes a secure session over which the service interactions can be carried out. The flow is described in the diagram below:

Step 1a: Trusted device requests access to a trusted application

Step 1b: The Application element of the trusted device requests a session

Step 1c: The session control function of the trusted device requests the token management function for a valid bootstrap token

Step 1d: The token management function either has a valid token, or requests the bootstrapping for a new bootstrap_token

At this stage, the bootstrap_token generation flow is called if a new bootstrap_token is required.

Step 2a: The token management function gets the bootstrap_token from the bootstrapping function

Step 2b: The token management function sets the bootstrap_token for session control function

Step 2c: The session control function establishes a secure session over the reference point RP_D

At this stage, the trusted device and application can initiate service interactions over the secure session.

~~The bootstrapping and authentication workflows are meant for bootstrapping a device to the OBF realm, and authorizing it for using a particular trusted application. Two types of information workflows are provided: (i) Bootstrapping workflow, and (ii) Authentication workflow.~~

~~10.1.1 Bootstrapping workflow~~

~~Prior to using the authentication services of the OBF, the OBF client functions of the device performs a bootstrapping workflow with the OBF authentication functions.~~

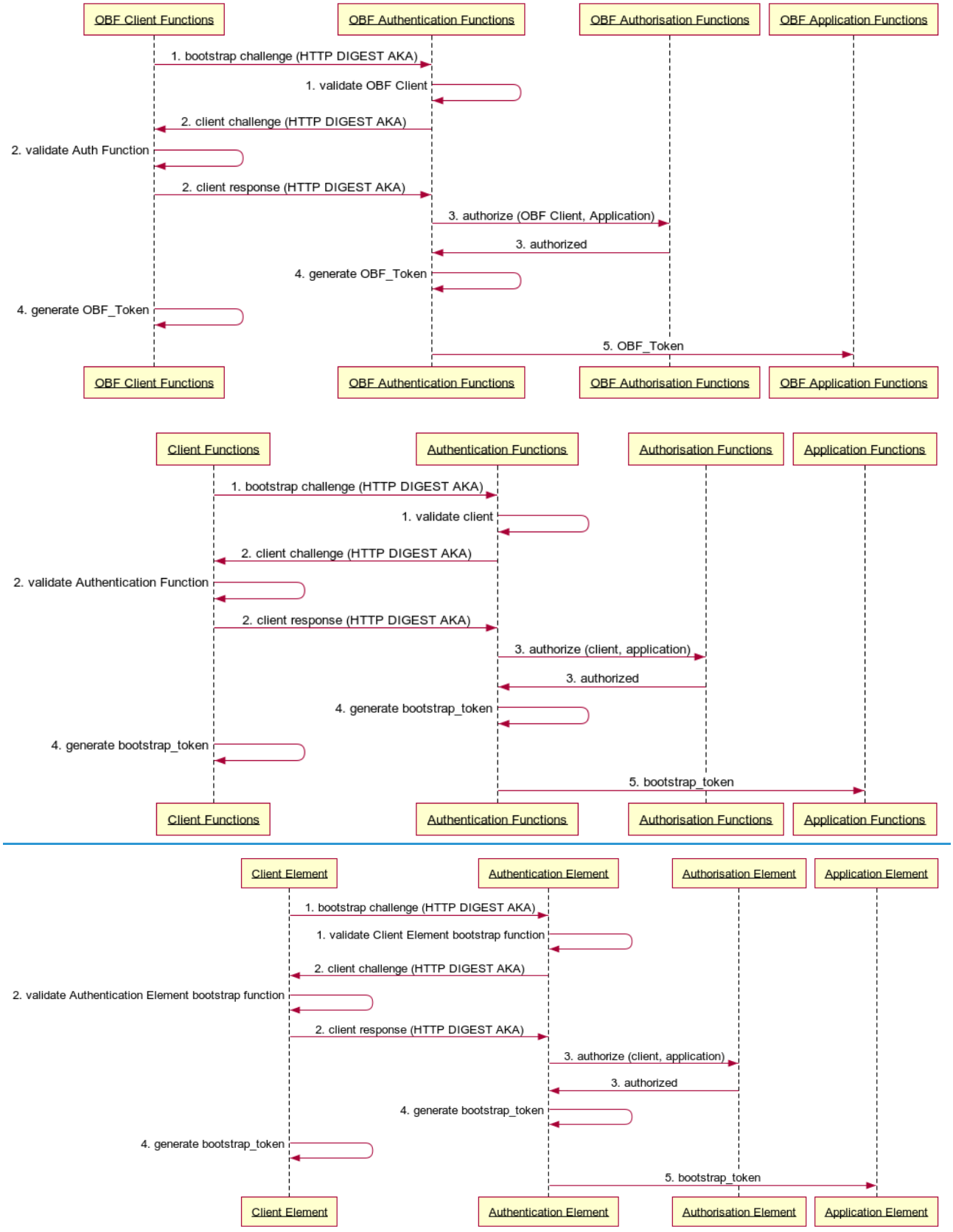
~~The bootstrapping function uses the symmetric (pre-shared) keys, which exist on, both, the secure element of the device and in the OBF authorization functions. These keys are used to mutually authenticate the OBF client function and the OBF authentication functions.~~

~~After the mutual authentication, the session keys are generated which are used for securing the communication between the trusted device and an application. This process is accomplished in the following steps:~~

1. ~~The OBF client functions using the bootstrap function will send a challenge request to the OBF Authentication functions. The OBF authentication function will validate the credentials of the OBF client based on the keys/algorithms used in the HTTP Digest/AKA;~~
2. ~~The OBF authentication function will send back a challenge back to the OBF client functions; The OBF client functions will validate the OBF Authentication functions based on the keys/algorithms used in the HTTP Digest/AKA;~~
3. ~~After the successful mutual authentication in steps 1 and 2, the OBF authentication functions will check if the given device is authorized to use OBF for trusted services for the given application;~~
4. ~~When the authorization has been approved, the OBF client functions and the OBF authentication functions generate an OBF_Token as per the agreed AKA protocol; and~~
5. ~~The OBF_Token is provided to the application element for use in subsequent security associations.~~

~~NOTE – The steps 1, 2, 3 are a part of the digest access authentication AKA.~~

~~The bootstrapping and the session key management process is described in the diagram below (Figure 10-1) in which the numbering of the steps in the diagram follows the numbering of steps in the paragraph above:~~



~~Figure 10-1: Bootstrapping workflow~~

NOTE: The workflow for bootstrapping using asymmetric keys is similar, with the exception that in place of pre-shared keys the public keys are used for authentication.

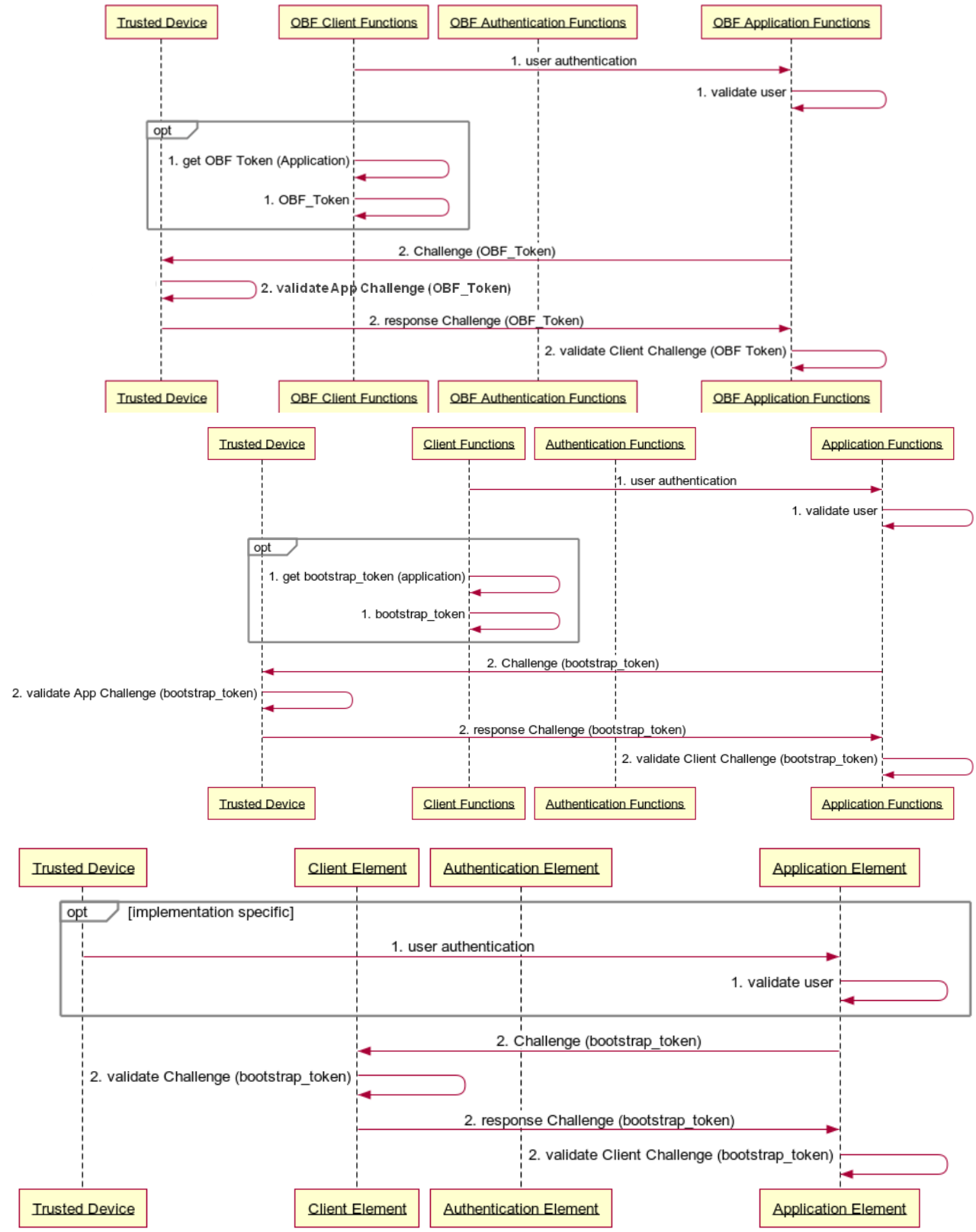
10.1.2 Authentication workflow

When a User requires to access an application from the trusted device, or the application requires to exchange data with the trusted device, it signals to the OBF client functions to use the bootstrap framework for authentication. This process is accomplished in the following steps, provided that the bootstrapping has been completed as per 10.1.1:

1. The user request towards the application is executed and the application uses a challenge-response mechanism to identify and authenticate the user and the user responds to the challenge-response mechanism used by the application; and optionally requests the OBF client functions to get a new OBF_Token if no previous is available, or has expired; and
2. The OBF application functions use the OBF_Token to send a challenge to the device. Upon success, the OBF_Token and the session control function are used to secure the data exchange between the device and the application.

NOTE – The mechanism to invoke the OBF client function for initiating the bootstrap procedure is left to the implementation and not covered in the scope of this recommendation.

The Authentication workflow is described in the diagram below:



~~Figure 10-2: Authentication workflow~~

10.62 ~~WorkFf~~low for change ofs ~~in OBF realm~~network operator

A user that is a beneficiary of the bootstrapping capabilities provided by a network operator may require to change the network operator, but may want to continue the use of trusted services which were supported by the network operator. A process will be required for the transfer of bootstrapping capabilities from the old network operator to the new network operator. The changing of the network operator bootstrapping realm is enabled by the process defined below

10.62.11 ~~OBF realm~~network operator flow (symmetric keys)

The process for change of network operator offering bootstrapping services is shown in the diagram below:

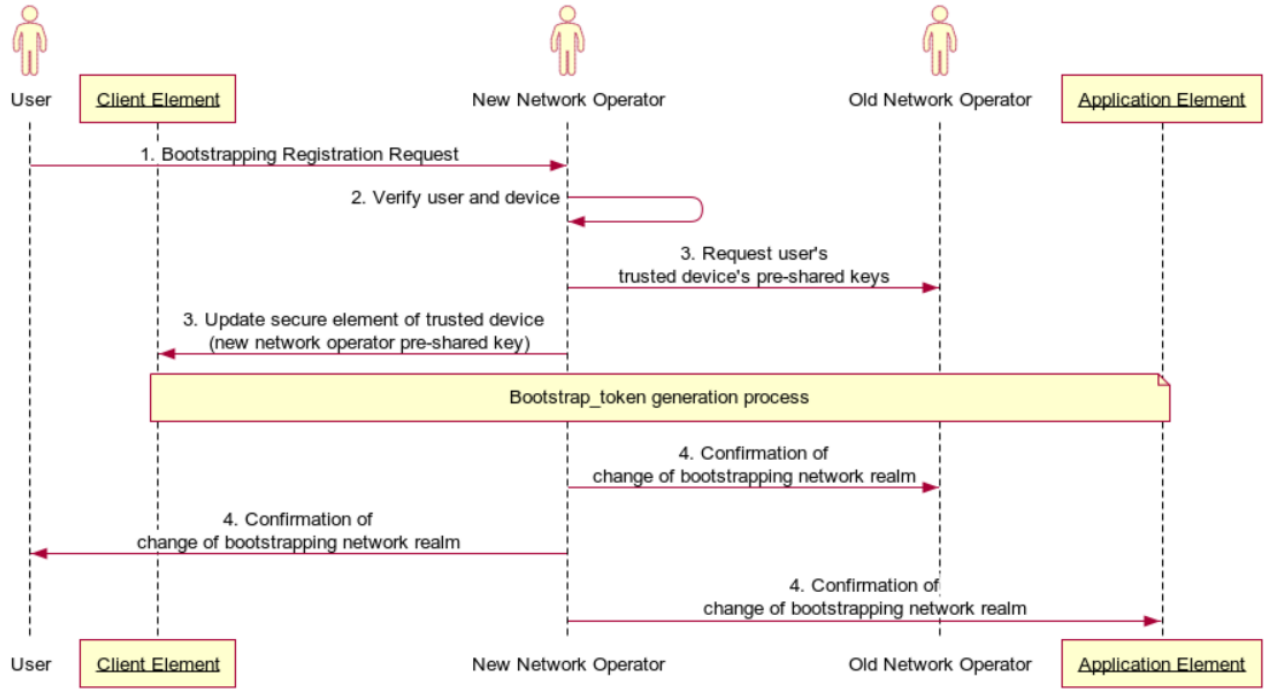


Figure 10-6: Change of network operator (symmetric keys)

Step 1: The user of the trusted application approaches the new network operator registration to the new network operator bootstrapping capabilities for access to trusted applications

Step 2: The new network operator undertakes the verification of the user and the trusted device (machine KYC) and upon successful verification, requests the old network operator for the user's pre-shared keys

Step 3: The new network operator updates the secure element of the user's trusted device with its own pre-shared key(s)

After this stage, the trusted device of the user is on-boarded to the new network operator as per the Bootstrap_token generation flow.

Step 4: upon success, the new network operator informs the user and the old network operator of the successful on-boarding of the user's trusted device to the new network operator

~~A user that is beneficiary of the OBF enabled trusted services provided by a network operator may require to change the network operator, but still may want to continue the use of trusted services, which were supported by the OBF authentication function.~~

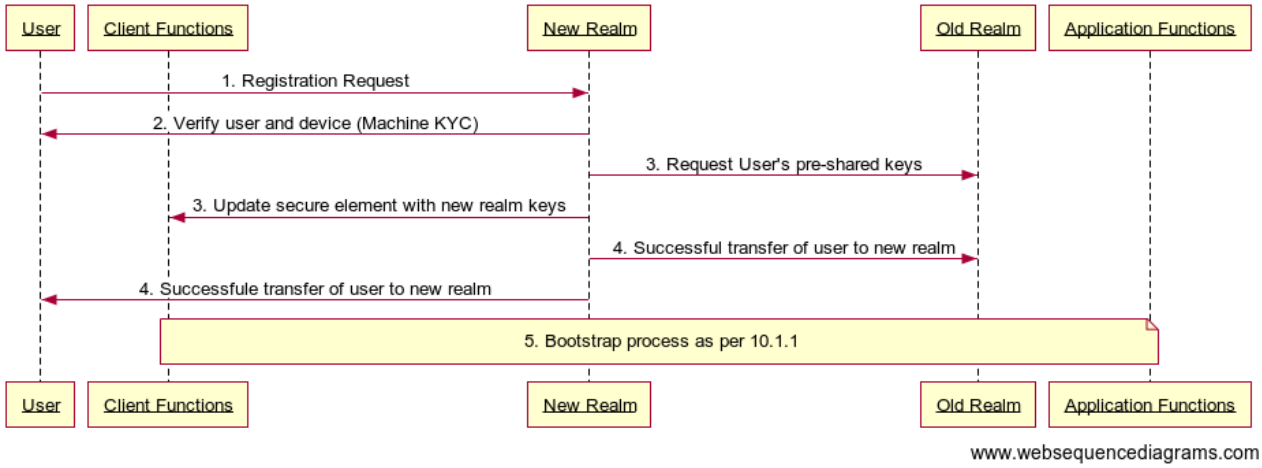
~~The changing of the OBF realm is enabled by the OBF mechanism as per the mechanism defined below.~~

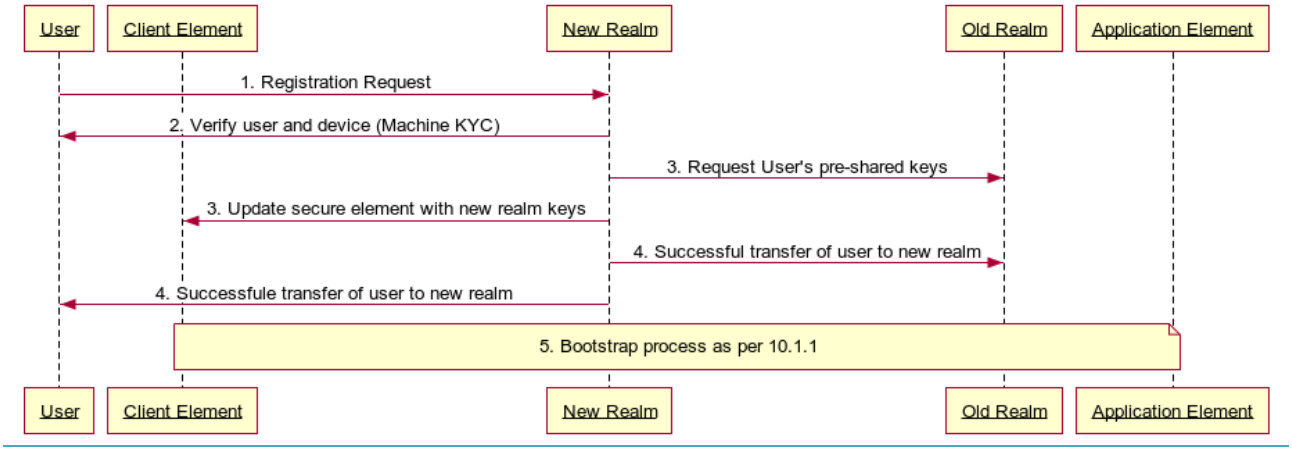
- ~~The user of the service has to approach the next network operator or IoT service provider, referred as the new OBF realm, for enabling the use of the trusted services for his device.~~
- ~~The steps for such a transfer of realm, in the case when symmetric keys are used for authentication, are described below:~~

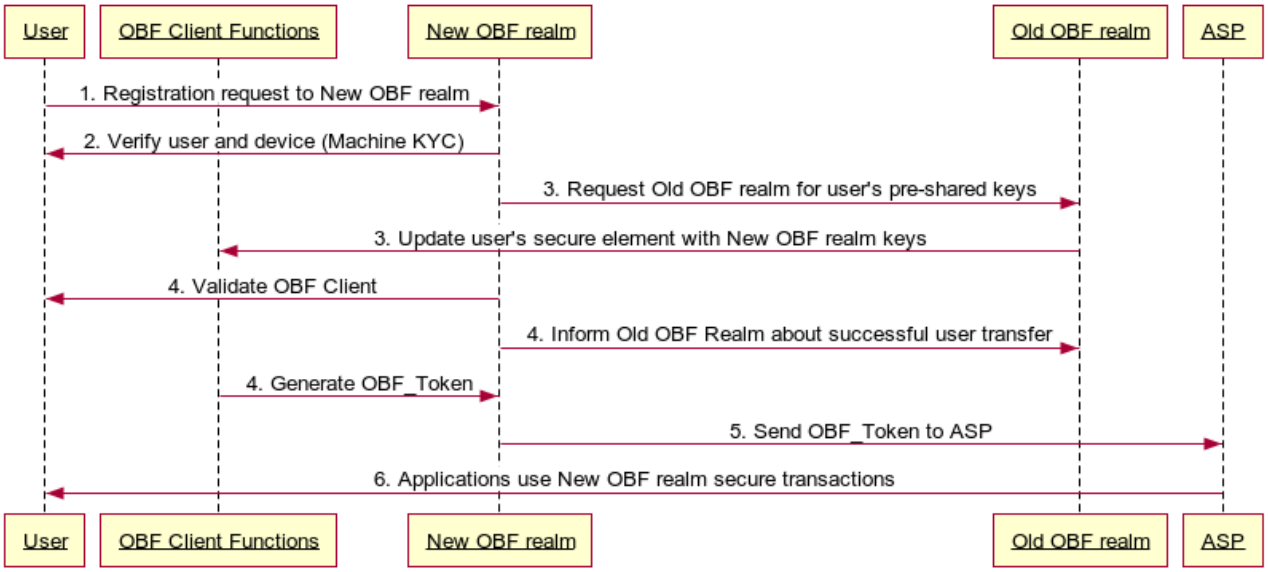
1. ~~User requests new OBF realm for its services;~~
2. ~~The new OBF realm undertakes the verification of the user and the device (machine KYC) and upon successful verification, requests the old OBF realm for the user's shared keys;~~
3. ~~The new OBF realm uses the old OBF realm's key(s) to update the secure element with the key(s) of the new OBF realm;~~
4. ~~The new OBF realm authenticates the OBF client functions using its keys, and upon success, informs the user and the old OBF realm of the successful transfer of the user to the new OBF realm; the new OBF realm and the user generate a new OBF_Token for use in the new OBF realm.~~
5. ~~The new OBF realm transfers the user's OBF_Token to the ASP; and~~
6. ~~The ASP uses the new OBF_Token to provide trusted services to the user.~~

NOTE - Machine KYC is the process of establishing a relationship between a machine and its ~~custodian~~user, usually accomplished by the network operator or IoT service provider by the use of physical or digital verification processes that establish the linkage between the identity of the ~~custodian~~ user and the identity of the ~~device~~trusted device owned by the ~~custodian~~user.

~~The process is described in the diagram below (Figure 10-3):~~







~~**Figure 10-3: Change of OBF realm (symmetric keys)**~~

10.~~6~~2.~~2~~2 Change of ~~OBF realm~~network operator flow (asymmetric keys)

In case~~It is possible that the new OBF realm is using~~ asymmetric keys are used for authentication~~, .~~ ~~T~~the ~~Steps~~ steps for ~~transfer~~ change of the ~~OBF realm~~network operator ~~, in the case when asymmetric keys are used for authentication,~~ are described in the diagram below:

~~1. User requests new OBF realm for its services;~~

~~2. The new OBF realm completes the machine KYC;~~

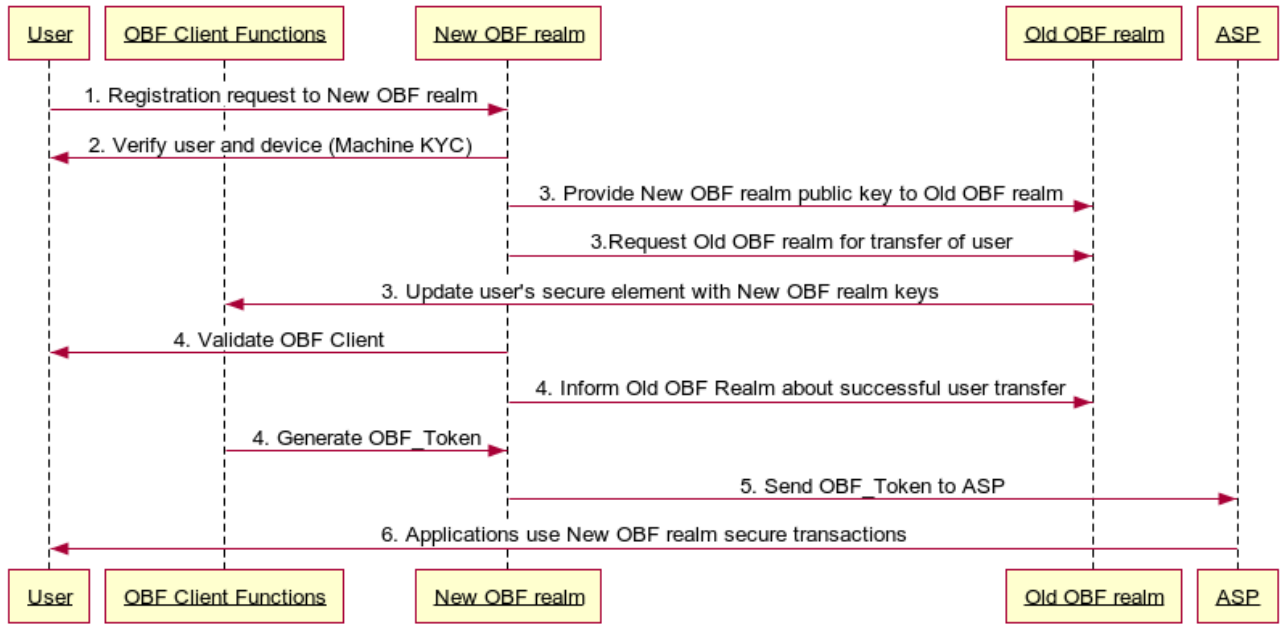
~~3. The new OBF realm provides its public key to the old OBF realm with a request to transfer the user's account to the new OBF realm;~~

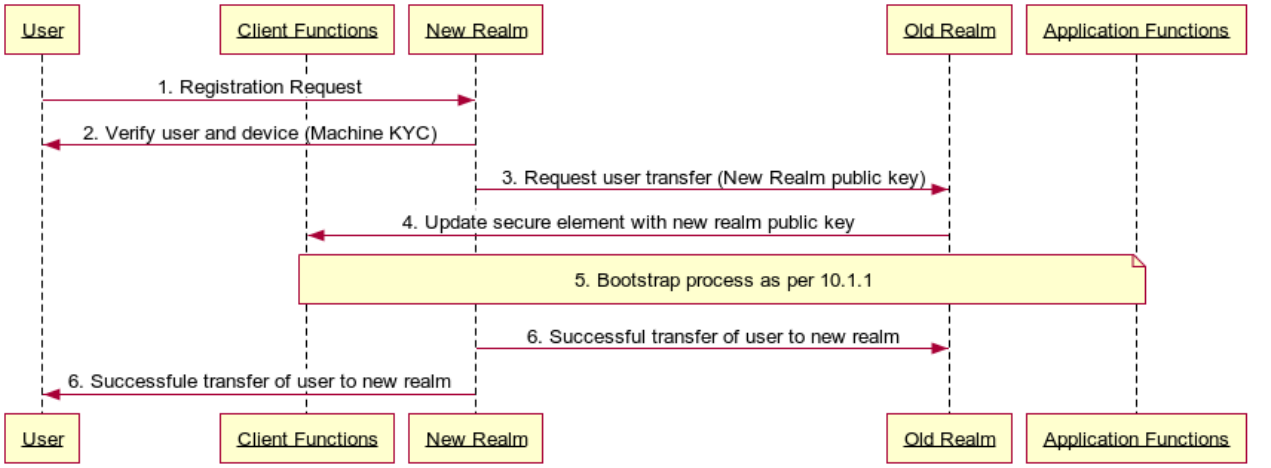
~~4. The old OBF realm uses its private key to update the secure element of the user with the public key of the new OBF realm;~~

~~5. Upon successful confirmation of the transfer the new OBF realm informs the ASP about the change in the OBF_Token for a user; and~~

~~6. The ASP uses the new OBF_Token to authenticate the user.~~

~~The Process is described in the diagram below (Figure 10-4):~~





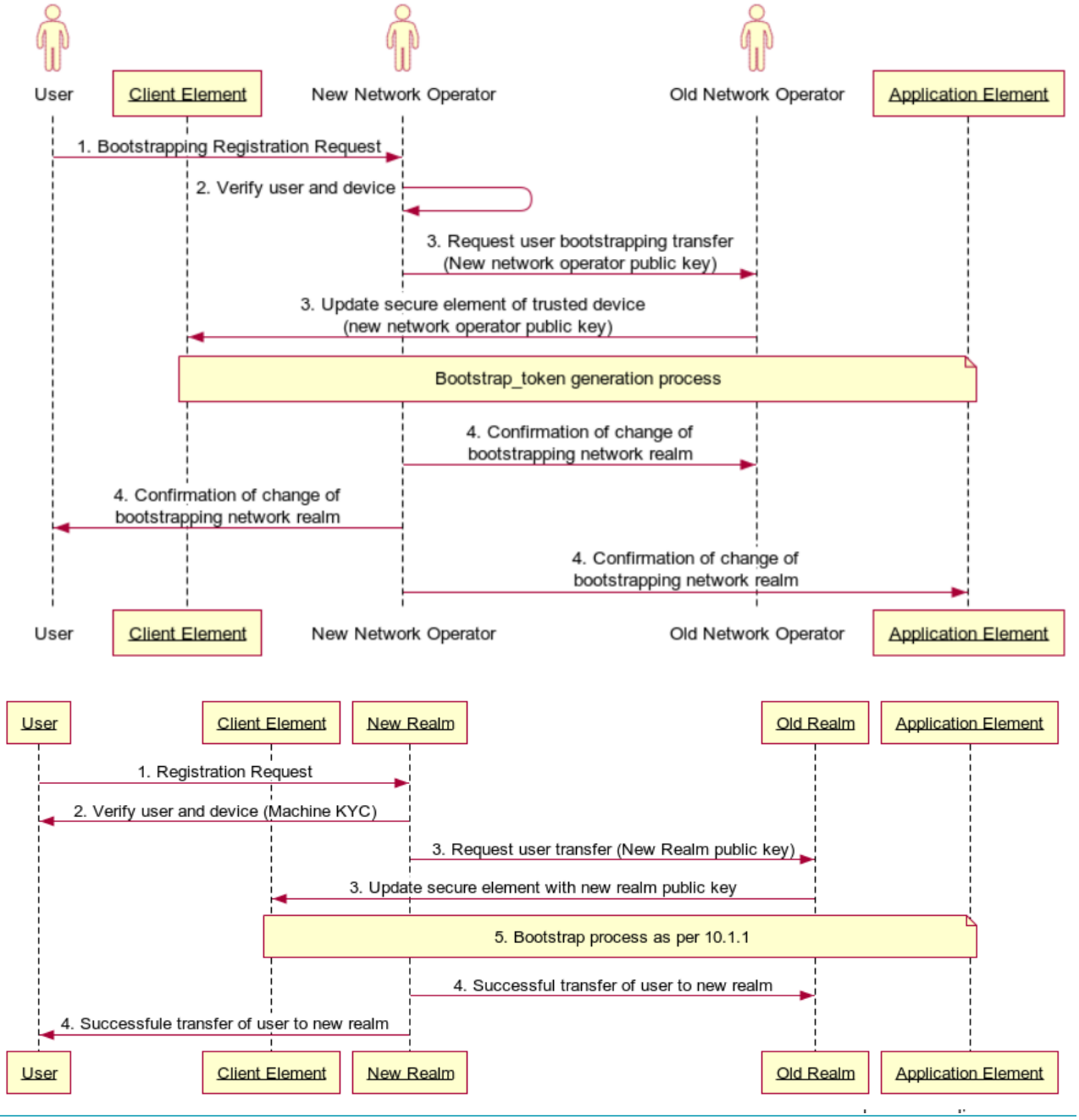


Figure 10-74: Change of ~~OBF realm~~network operator (asymmetric keys)

Step 1: The user of the trusted application approaches the new network operator registration to the new network operator bootstrapping capabilities for access to trusted applications.

Step 2: The new network operator undertakes the verification of the user and the trusted device (machine KYC) and upon successful verification, requests the old network operator to update the secure element of the user's trusted device by replacing the old network operator's public key(s) with those of the new network operator.

Step 3: The old network operator updates the secure element of the user's trusted device with the public key(s) of the new network operator.

After this stage, the trusted device of the user is on-boarded to the new network operator as per the Bootstrap_token generation flow.

Step 4: Upon success, the new network operator informs the user and the old network operator of the successful on-boarding of the user's trusted device to the new network operator.

Bibliography

[b-ITU-T X.1113] Recommendation ITU-T X.1113 (2007), *Guideline on user authentication mechanisms for home network services*

[b-ITU-T X.1124] Recommendation ITU-T X.1124 (2007), *Authentication architecture for mobile end-to-end communication*

[b-ITU-T X.1158] Recommendation ITU-T X.1158 (2014), *Multi-factor authentication mechanisms using a mobile device*

[b-ITU-T X.1311] Recommendation ITU-T X.1311 (2011), *Information technology - Security framework for ubiquitous sensor networks*

[b-ITU-R F.1399] Recommendation ITU-R F.1399 (2001), *Vocabulary of terms for wireless access*

[b-ITU-T Y.3052] Recommendation ITU-T Y.3052 (2017), *Overview of trust provisioning for information and communication technology infrastructures and services*

[b-RFC 6733] IETF, Request for Comments: 6733 (October 2012), *Diameter Base Protocol*

[b-RFC 7155] IETF, Request for Comments: 7155 (April 2014), *Diameter Network Access Server Application*

[b-RFC 7616] IETF, Request for Comments: 7616 (September 2015), *HTTP Digest Access Authentication*.

[b-3GPP TS 33.220] 3GPP TS 33.220 V16.0.0 (2019-09), *Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (Release 16).*

_____