Uniform platform for Trusted application – High Level Design

Submitted to

Alok Dubey

Prepared by	: Anusha ,Kamalesh,Pushpa
Reviewed by	: Alok
Date Submitted	: 6/09/2018

Version	: 0.1a
Last revised date	: 27/06/2018

This document is created by **Votary Tech Pvt. Ltd**. This document and any supporting appendices submitted with it are confidential and intended solely for the use of the business users and IT team working at Votarytech Soft Tech Solutions.

Revision History

Change Record

Date	Author	Version	Change reference
27/06/2018	Alok Dubey	0.1	Initial Version
07/09/2018	L. Anusha Kamalesh Patil. Pushpa Methekar	0.1a	Added Goals&objective,solution overview,system architecture,Code flow.

Reviewers

Name	Version approved	Date

Table of Contents

Abbreviations

NSW	Non Secure World
SW	Secure World
VSF	Votary Secure Framework
EE	Execution Environment
TEE	TrustZone Execution Environment
QEE	Qualcomm Execution Environment
XEE	X Execution Environment (X – Any)
VEE	Votary Execution Environment
EEC	Execution Environment Client

1.Introduction

"ARM® TrustZone® technology is a system- wide approach to security for a wide array of client and server computing platforms, including handsets, tablets, wearable devices and enterprise systems. Applications enabled by the technology are extremely varied but include payment protection technology, digital rights management, BYOD, and a host of secured enterprise solutions."

Trust zone is a set of security extensions added to ARM processors. It can run 2 operating systems. 1.Secure operating system 2.Normal operating system. Both operating system have the same capabilities and Operate in a separate memory space.

Enables a single physical processor core to execute from both the Normal world and the Secure world. Normal world components cannot access secure world resources and secure world can access normal world components.

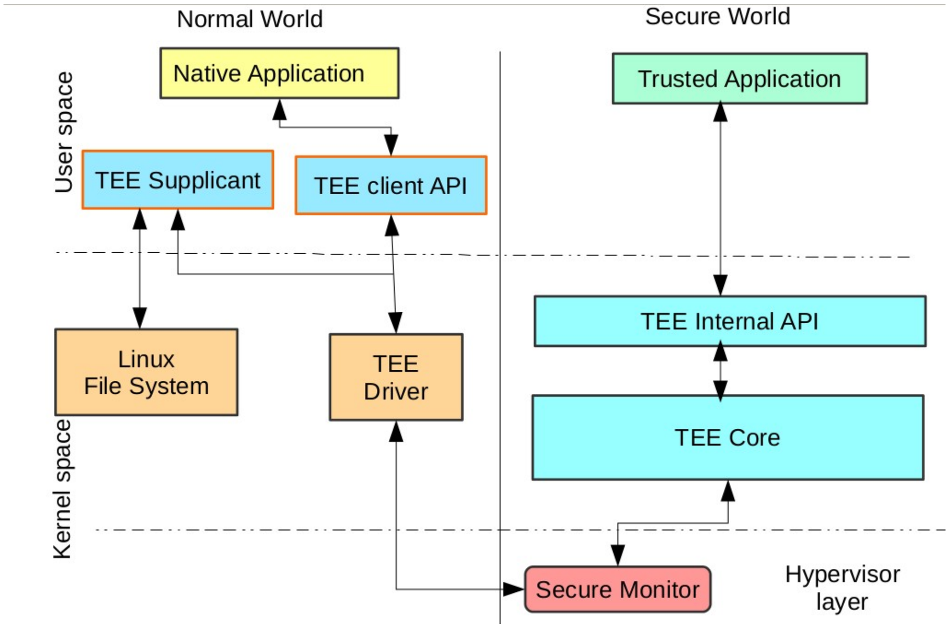

Fig1.a: Basic architecture of TrustZone

2.Goals and Objectives

To develop a VEE library for ARM trustzone using OPTEE which will support any platform like TrustZone Execution Environment(TEE), Qualcomm Execution Environment (QEE),X Execution Environment (X – Any) (XEE).

3.Solution Overview

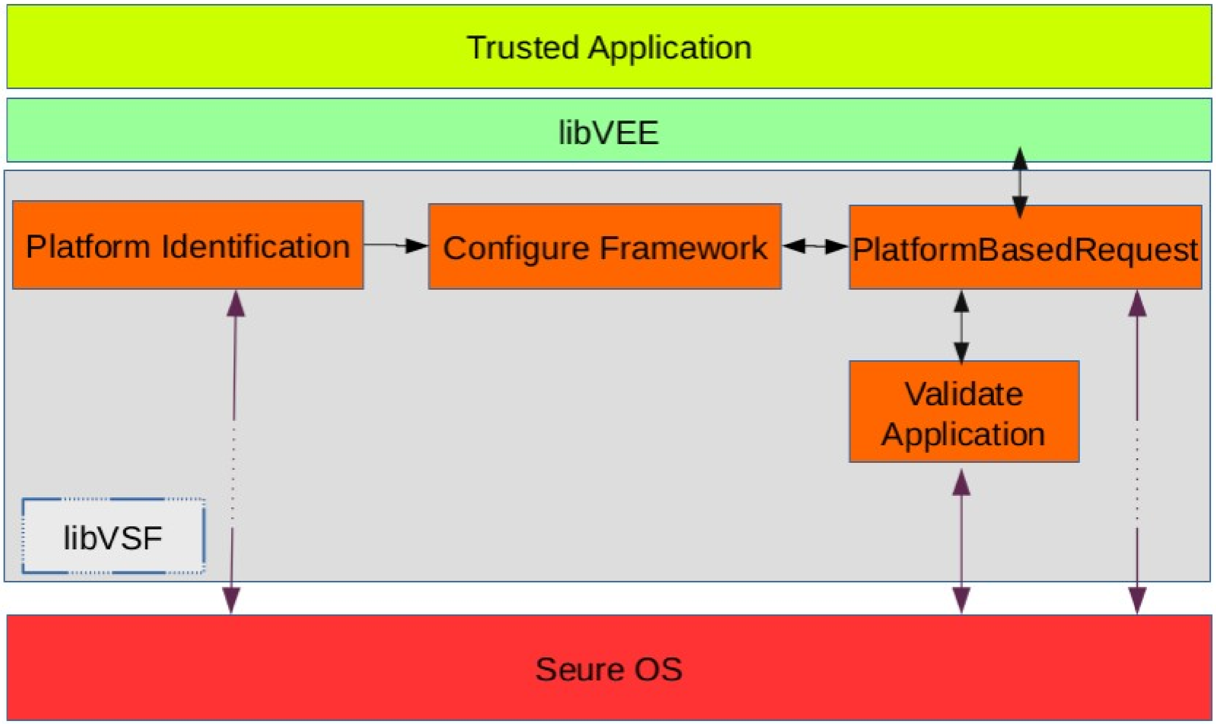

Fig 3.a VSF architecture

The main task of this VSF to develop libVEE that will identify a platform and configure framework. Framework will map VEE APIs with the related APIs in the trusted application and it also rise a request to connect to the related API. Based on that APIs trusted application is executed in secure OS.

4.System Architecture

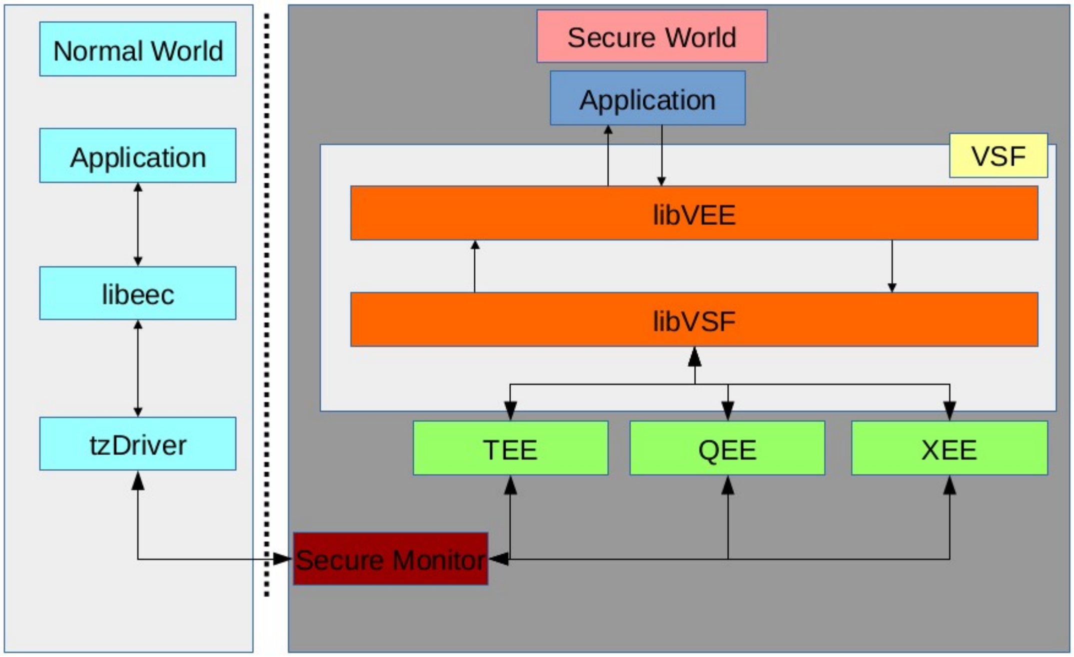

Fig1.b Platform independent system architecture

In this architecture secure world having VEElib which support any execution environment platform(TEE,QEE,XEE). Votary secure frame select appropriate execution environment platform to execute trusted application.

5.Code Flow

5.1 InitializeContext

The TEEC_initializecontext() call enters “TEE Core” via “TEE Driver”.”TEE Core” calls VSF_TA_CreateEntryPoint() via VSF. Control is returned back to hello_world in user.

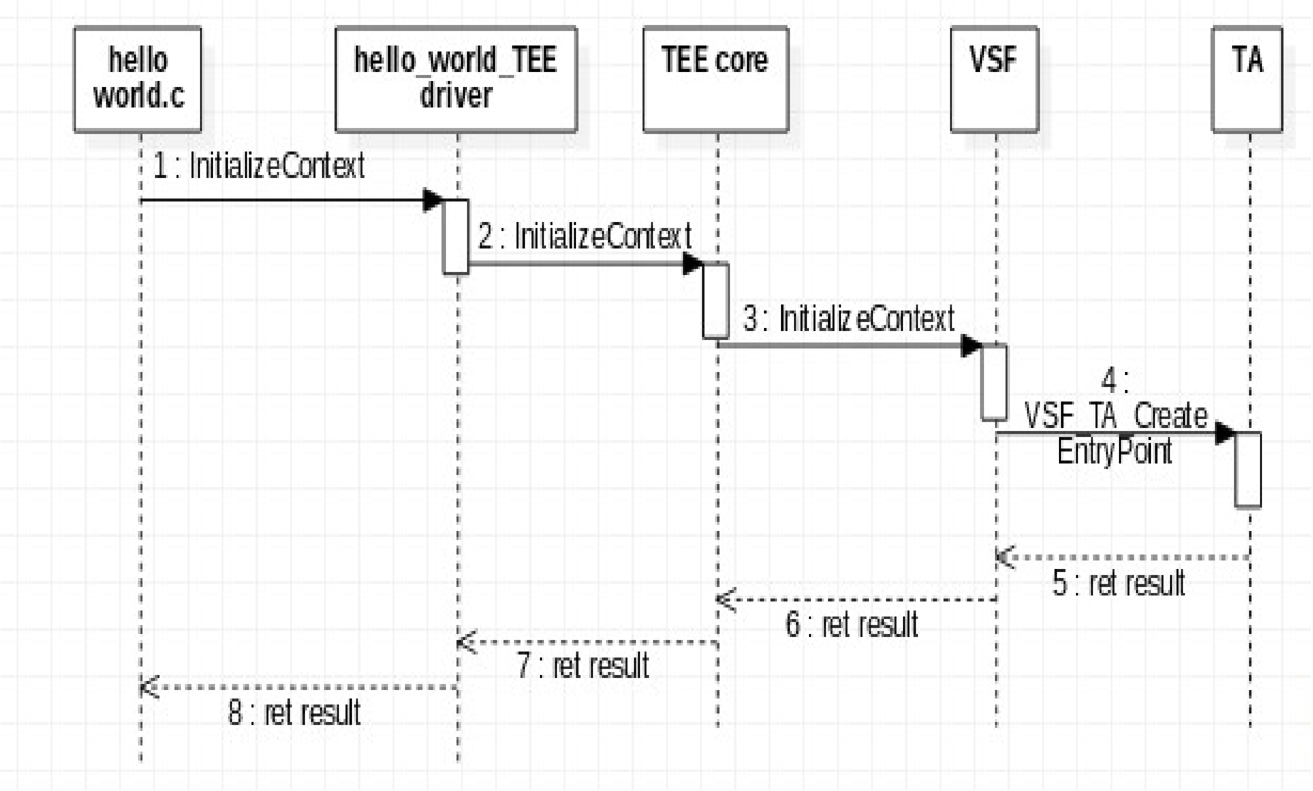

Fig 5.a:Sequence Diagram for InitializeContext

5.2 OpenSession

The TEEC_OpenSession() call enters “TEE Core” via “TEE Driver”. “TEECore” loads the TA binary with help of the Linux User space tee_supplicant. “TEE Core” cpoies the TA into secure RAM and call VSF_TA_OpenSessionEntryPoint() via VSF. Session is returned back to hello_world in user space.

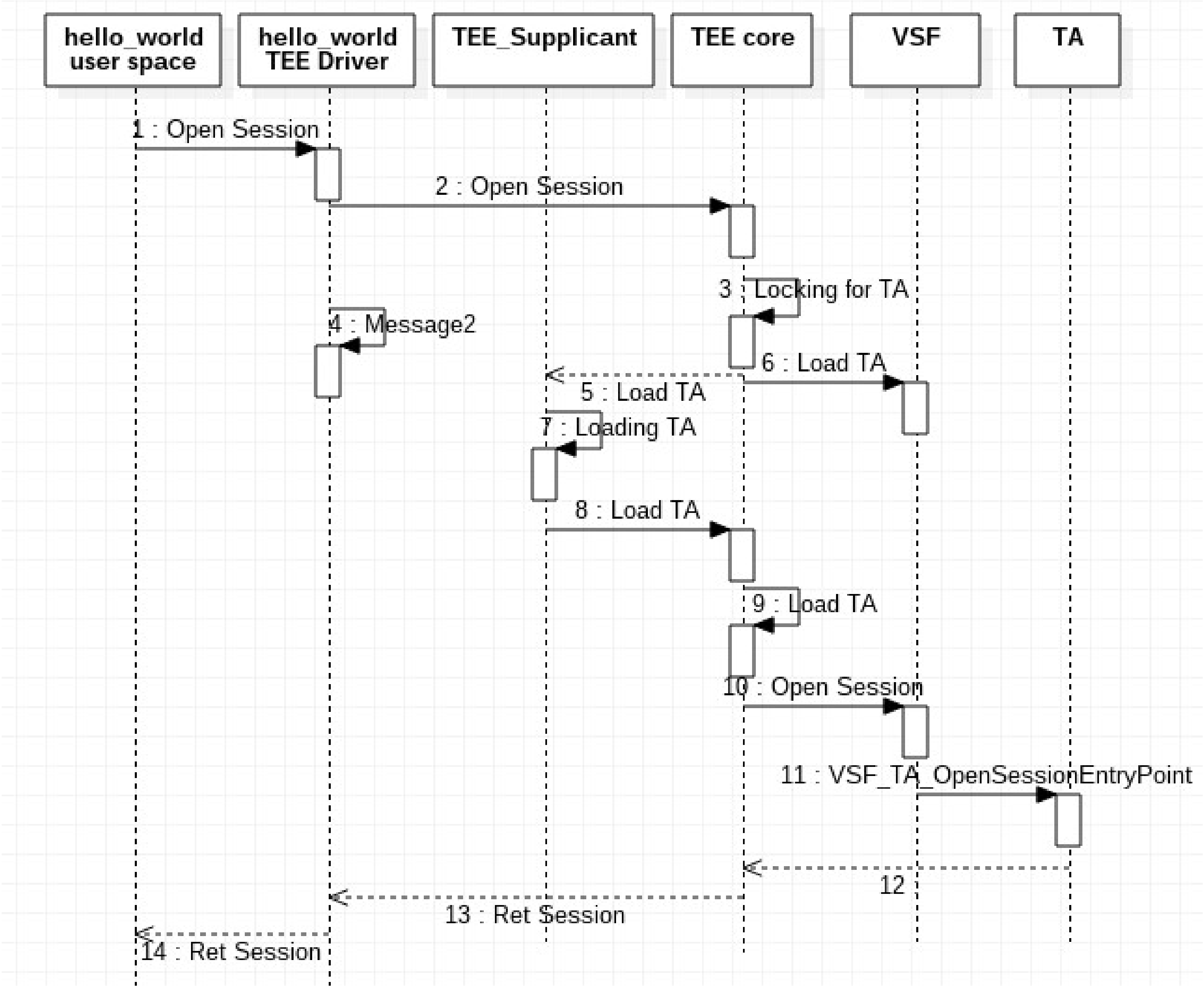

Fig 5.b:Sequence Diagram for OpenSession

5.3 InvokeCommand

The TEEC_InvokeCommand() call enters “TEE Core”via “TEE Driver”.“TEE Core” calls VSF_TA_InvokeCommandEntrypoint() via VSF. Result is returned back to hello_world in user space.

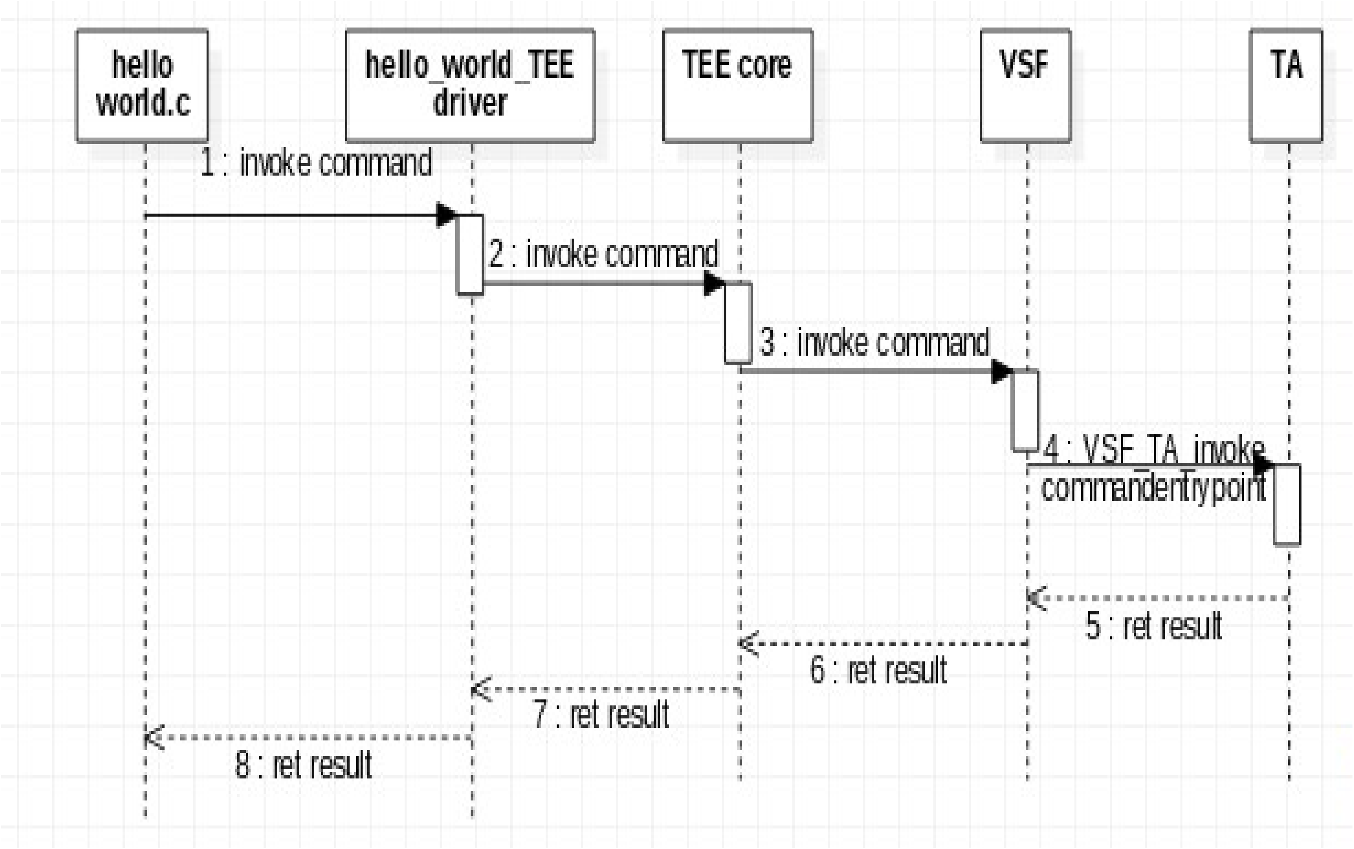

Fig5.c:Sequence Diagram for Invoke Command

5.4 CloseSession and FinalizeContext

The TEEC_CloseSession() call enters “TEE Core” via “TEE Driver”.”TEE Core” calls VSF_TA_CloseSessionEntryPoint() via VSF. Control is returned back to hello_world in user.

The TEEC_FinalizeContex() call enters “TEE Driver” Which cleans remaining resources. Control is returned back to hello_world in user.

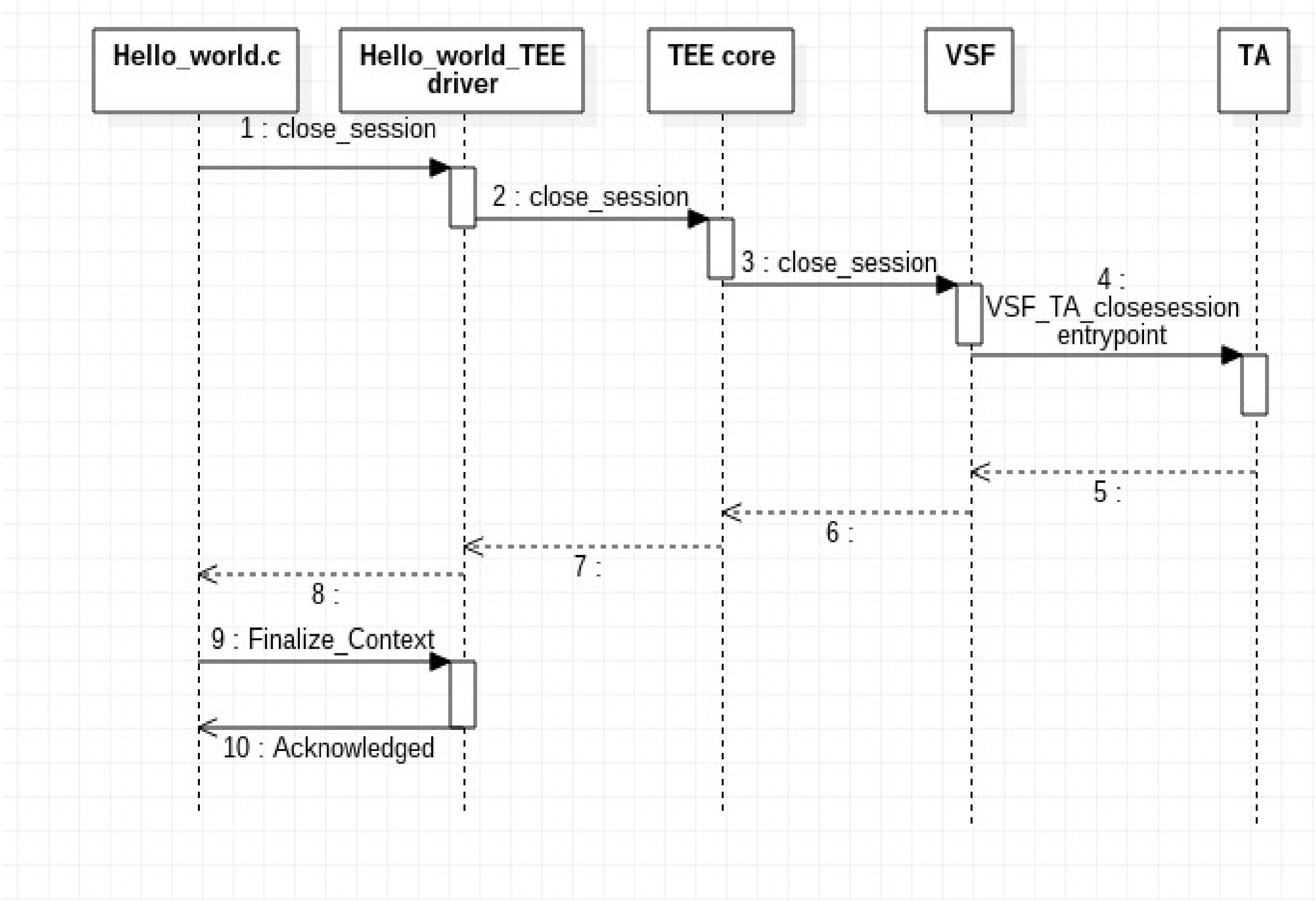

Fig 5.d:Sequence Diagram for Close Session and Finalize Context