|  |  |
| --- | --- |
| **(Confidential)** | |
| Scope of disclosure | [Company] for EPH only |
| Period of confidentiality | 7 year after issued date |
| Head of Information Owner | Head of engineer department |
| Handling restriction | NA |

**Software Architecture Design Specification**

**EPH**

**Toshiba Software Development (Vietnam) Co., Ltd.**

|  |
| --- |
| Document ID: TSDV-21A-EPH-SAD |
| Total: Page No. 1 |

Revision History

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rev. No.  (X.YY) | Date  (YYYY-MM-DD) | Section No. Changed | Change Description | Author | Reviewed by | Approved by |
| 0.01 | 2021-09-13 | All | Initialize documents | HoaDV | ThanhBX |  |
| 0.02 | 2021-09-15 | 5.2 | Update for review | HoaDV |  |  |
| 1.00 | 2021-09-17 | 5 | Update for comments from customer | HoaDV | ThanhBX | BinhNK |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Content

[1. Introduction 5](#_Toc82766875)

[*2.* References 5](#_Toc82766876)

[3. Definitions and Acronyms 5](#_Toc82766877)

[*4.* System Overview 6](#_Toc82766878)

[5. SDK Architecture 7](#_Toc82766879)

[5.1. DDS Deployment Modes 7](#_Toc82766880)

[5.2. SDK Worker Thread 8](#_Toc82766881)

[*6.* Exception 9](#_Toc82766882)

List of Figures

Figure 4‑1: Target system overview 6

Figure 5‑1: OpenSplice shared memory (left) & single process deployment (right) 7

Figure 5‑2: SDK position in application software stack for Pi (left) and EPH (right) 7

Figure 5‑3: Job request handling of Pi&EPH SDK 8

# Introduction

This document is about software architecture design for EPH project. Base on requirements in SRS, this document defines the direction of design for software detailed design later.

# References

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document ID** | **Document/Standards Name/Title** | **Source** | **Version No. /Release or Publication date** | **Brief Description/Section Reference** |
| TSDV-EPH-SRS | TSDV-21A-EPH-SRS.docx | - | 0.06 | Software requirement specification of EPH project |
| Vortex OpenSplice Architecture | https://material.adlinktech.com/ADLinkFile/Publication/1326/Vortex\_OSPL\_Architecture.pdf | - | - | Architecture of Vertex OpenSplice |
| OpenSplice Evaluation & Benchmarking Guide | http://download.ist.adlinktech.com/docs/Vortex/html/ospl/EvaluationGuide/index.html | - | - | Brief guideline to deploy Vertex OpenSplice for DDS network application |
| The Vortex OpenSplice Deployment Guide | http://download.prismtech.com/docs/Vortex/html/ospl/DeploymentGuide/index.html | - | - | Reference on how to configure the OpenSplice service |
| QnA list | TSDV-EPH-QA\_20210820.xls | - | 20210817 | Question and answer between TSDV and TTEC |
| - | - | Debug\_HPF\_evaluation\_dummy\_Reliable.zip | 20210813 | Sample source code from TTEC about using DDS with EPH board |

# Definitions and Acronyms

|  |  |  |
| --- | --- | --- |
| No | Acronyms | Definition |
| 1 | SRS | Software Requirement Specification |
| 2 | DDS | Data Distribution Service |
| 3 | EPH | EndPoint Hub (also name of the project) |
| 4 | SDK | Software Development Kit |
| 5 | API | Application Programming Interface |
| 6 | TTEC | Toshiba Tec Corporation |
| 7 | DNN | Deep Neural Network |

# System Overview

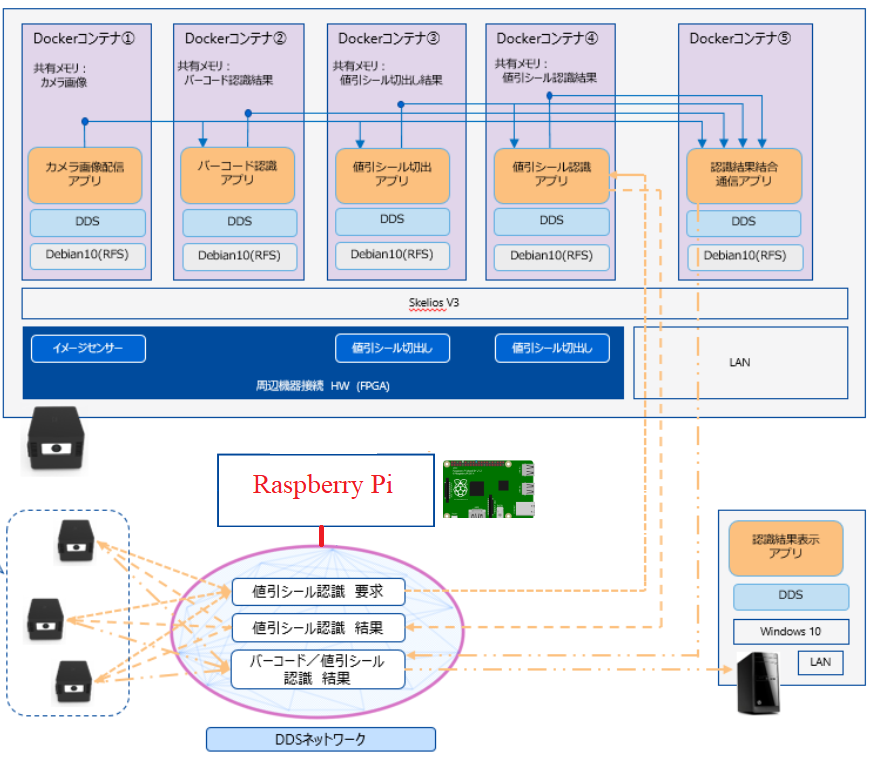


Figure 4‑1: Target system overview

Figure 4‑1 describes target system that SDK will be installed and run on. Target device are Raspberry Pi (for short, Pi is used) and EndPoint Hub (EPH). Even, PC (with Windows 10) is in the figure, but it is not target device for development in 2021A.

Both Pi and EPH devices are ARM-based architecture. The SDK will be implemented as C/C++ library.

# SDK Architecture

This section describes the architecture design of SDK with 2 point of view.

* The first view is about DDS deployment modes. Since SDK itself does not contains DDS network service but depends on Vertex OpenSplice. Then, DDS deployment mode is important aspect to design SDK.
* The second point of view is how SDK serves application requests. This view describes how SDK handles it roles in application.

## DDS Deployment Modes

From the application point of view, the SDK will handle communication between computing node in network to share data. However, the SDK itself does not contains features to implement DDS service but depends on Vertex OpenSplice.

From Vertex, there are 2 deployment modes with OpenSplice as described in Figure 5‑1.

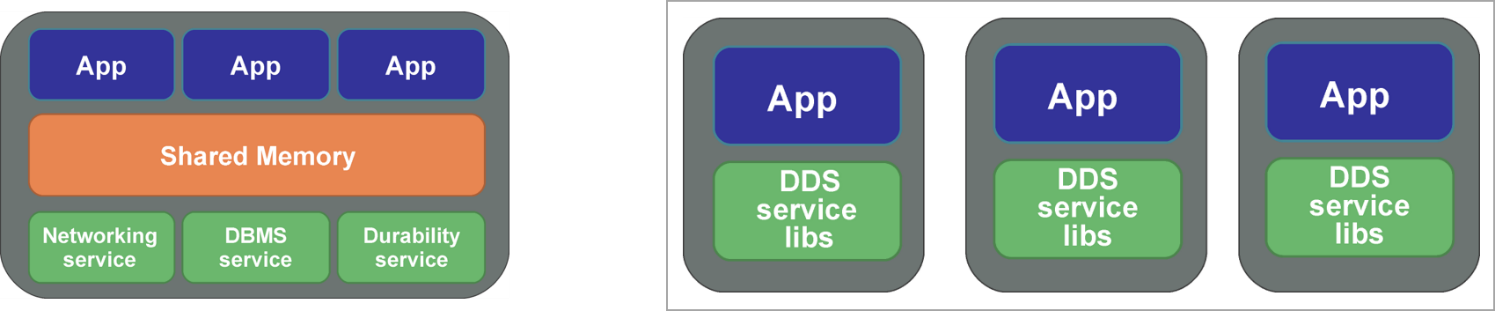


Figure 5‑1: OpenSplice shared memory (left) & single process deployment (right)

In 2021A development, target development is prototyping with small demonstration on Pi and EPH devices. To reduce complexity of using SDK and fast development, **single process deployment is used**.

From application point of view, Figure 5‑2 describes the position of SDK in application stacks.

* Since OpenSplice is used with standalone mode, from application point of view, it does not need to do anything excepts directly calling SDK API to use DDS services.
* SDK will not touch directly to DDS service libs, but via a communication library, called **com\_middle**. This library is a wrapper of DDS service libs to make DDS accessing become simply. **com\_middle** is provided by TTEC, and already complied into library file for ARM architecture.
* DNN service library is a library for image recognition, also will be provided by TTEC.

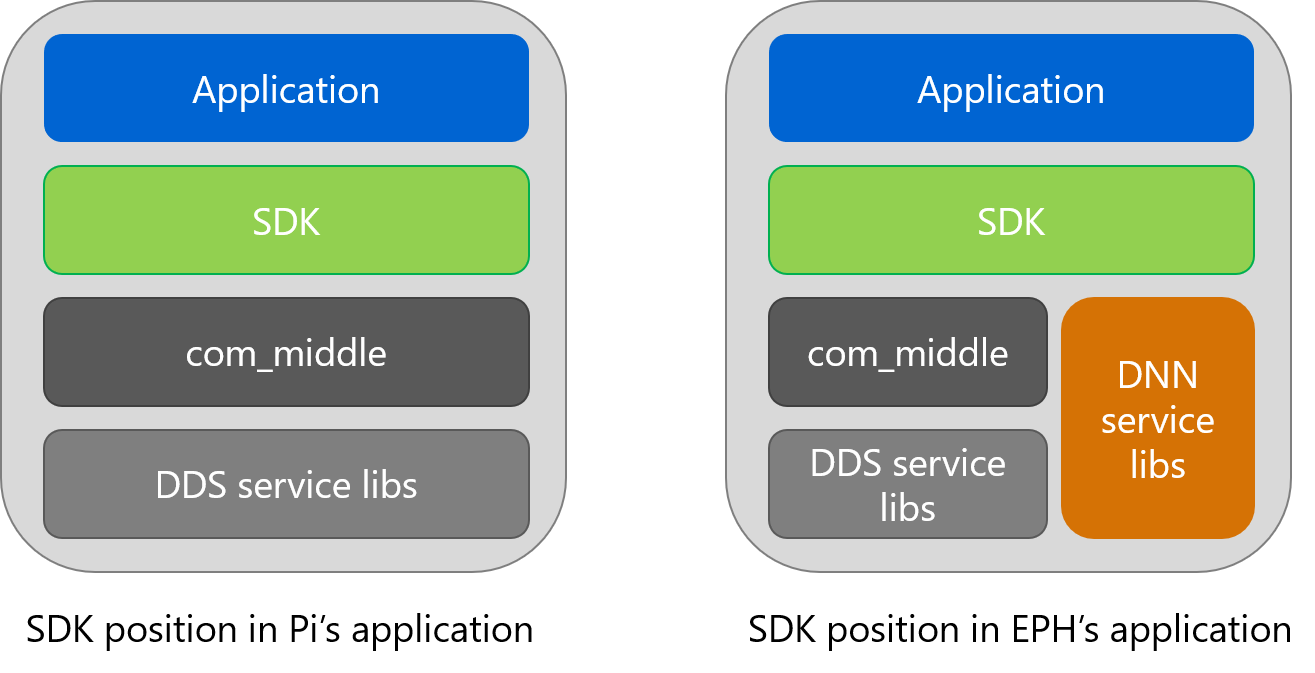


Figure 5‑2: SDK position in application software stack for Pi (left) and EPH (right)

## SDK Worker Thread

This section describes how internal SDK handles requests from application.

The SDK is about working with communication application, then almost requests activities are asynchronous. To be easy to handle asynchronous activities, event-based pattern is used to design SDK. Design idea is shown in Figure 5‑3.

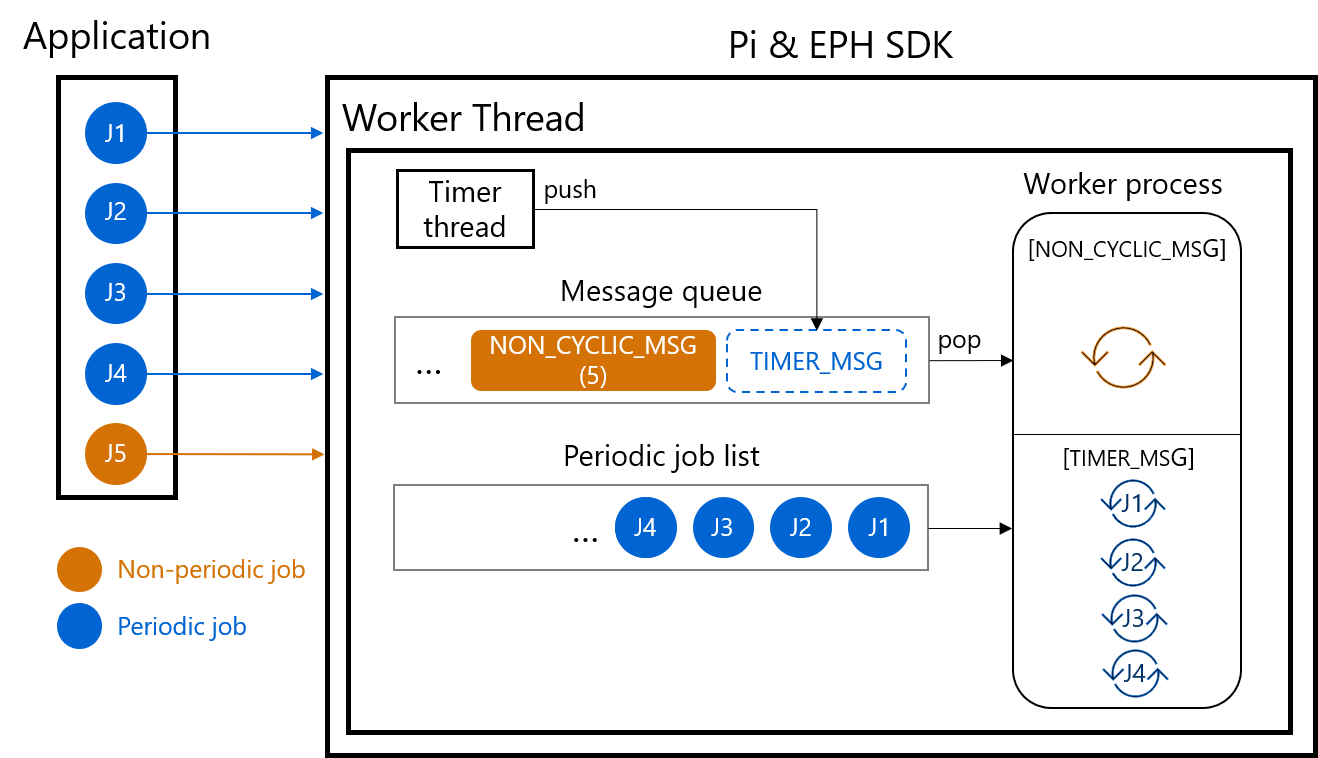


Figure 5‑3: Job request handling of Pi&EPH SDK

When application issues jobs requests, the SDK receives request and stores request information to its internal buffers. If requesting job is periodic one, the job is added to internal periodic job list. Otherwise, none-periodic job is pushed into internal message queue.

SDK itself creates its own a Timer Thread to generate events cyclically for periodic jobs. When timer expires, Timer Thread issues a timer message which is pushed to message queue.

Worker Thread is always active, it pops messages from message queue and executes corresponding job as shown in Figure 5‑3.

* When message is a timer message, SDK processes all jobs in periodic job lists.
* In contrast, when message is for non-cyclic message, SDK processes related non-periodic job.

For 2021A development, all functionalities of SDK are arranged into 4 periodic jobs as below.

|  |  |  |
| --- | --- | --- |
| **Job ID** | **Job Name** | **Target Functionality** |
| J1 | Save captured image job | Application requests this job to receive images over DDS network |
| J2 | Capture image job | Application requests this job to capture, then send images over DDS network |
| J3 | Get image recognition result job | Application requests this job to send image, then get back recognition result over DDS network |
| J4 | Recognize image job | Application requests this job to serve for image recognition over DDS network |

J5 in Figure 5‑3 is to demonstrate that SDK can handle non-periodic job also. It means that it is very easy to add more job to SDK without impact to existed jobs.

# Exception

When execute job requests from application, there exists cases that an error occurs. In such situation, SDK should not itself terminates threads, system or removing all jobs from queue. But a callback should be issued to application. Purpose of the callback is to notify to application that error occurred, then application can decide corresponding actions that suitable for the error.

Template's Revision History

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Rev. No.  (X.YY) | Date  (YYYY-MM-DD) | Section No. Changed | Change Description | Author | Reviewed by | Approved by |
| 1.01 | 2011-04-26 | 9 | Delete traceability matrix part | GiangNT | ThuyTTP  TrinhNTT  Kojima-san |  |
| 2.00 | 2011-07-28 |  | Approved by managers | GiangNT | GD.Tabe  PM.AnhPT | GD.Tabe  PM.AnhPT |
| 2.01 | 2012-05-08 | Revision History | Fix date format | ToanDT |  |  |
| 3.00 | 2012-06-06 | Template's RH | Approved | ToanDT |  | AnhPT |
| 3.01 | 2018-11-06 | All | Change document into new template/logo | HienNN | [QAM] T.Ninomiya | [GD] K.Shigenaka  [PM] TrungDV, ThinhND |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |