

TUS SDK 010 Demonstration

Agenda

PROTECTED
関係者外秘

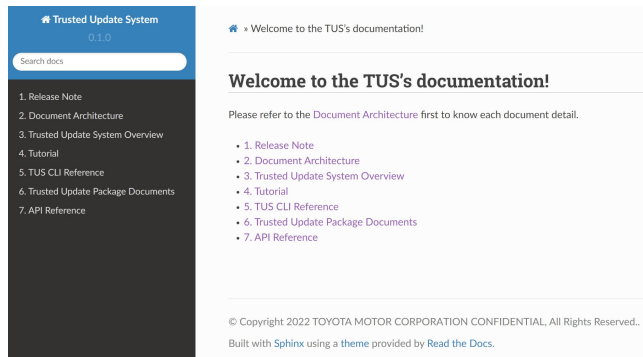


1. TUS SDK 010 概要
2. TUS の構成と動作モデル
3. Demonstration

TUS SDK 010 概要

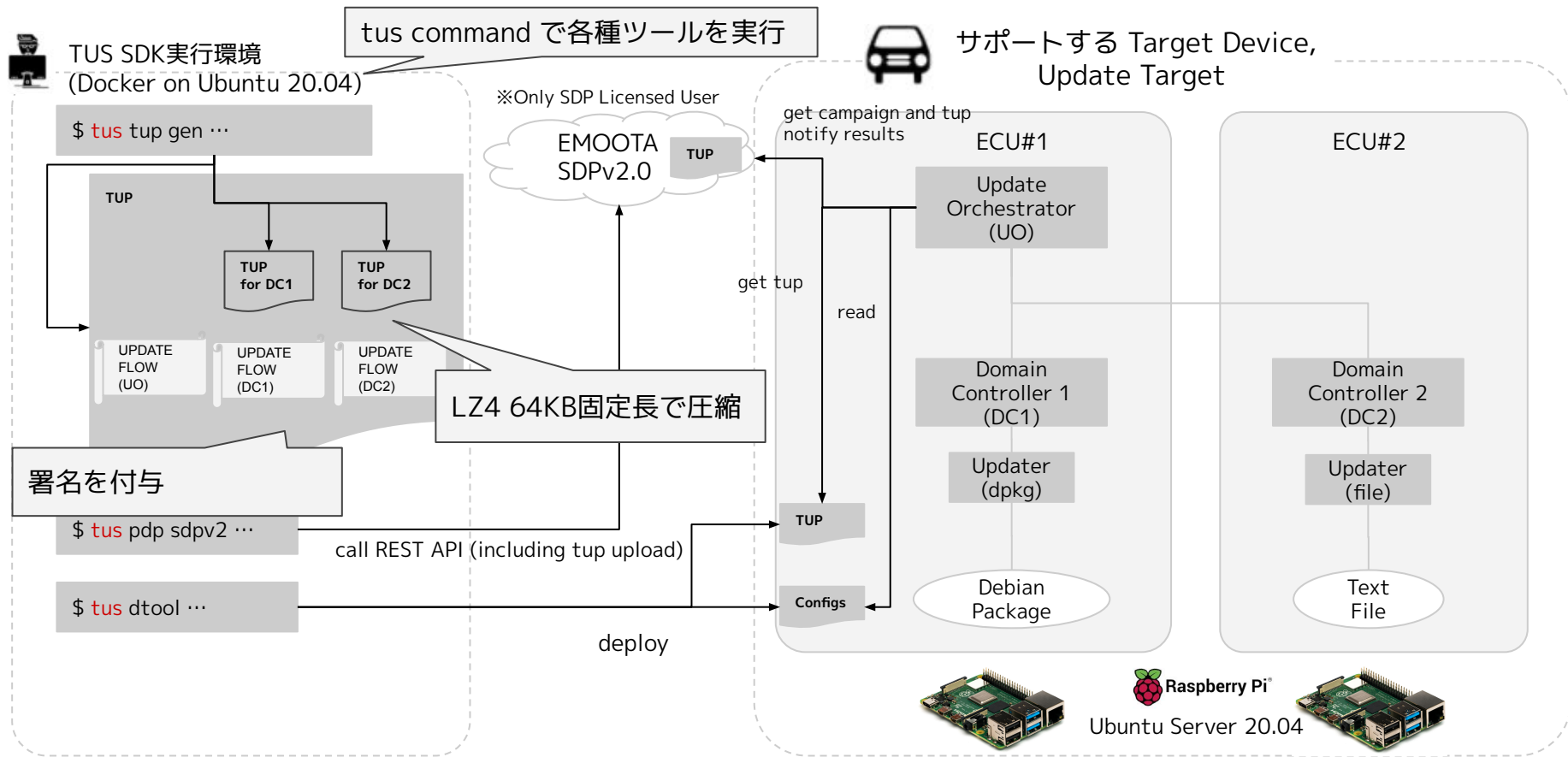
TUS SDK 010は下記から構成される

1. TargetDevice側で主に動作するソースコード群
 - TUS Runtime (Update Orchestrator(UO), Domain Controller(DC), Updater)
 - TUP Parser Library
2. 提供先の開発ステージ・プロセスに適応可能なツール群(Host-Tools)
 - tus command (各ツールを包括制御するcommand)
 - tup generator (TUP生成ツール)
 - deployment tool (Configuration, Packageをtargetへデプロイするツール)
 - EMOOTA SDPv2 operator (only licensed user)
3. 関連Documents
 - Release Notes
 - Tutorial
 - TUS Overview
 - TUP Design and Specification
 - API Document



TUS SDK 010でのサポート構成概要

PROTECTED
関係者外秘



※TUPの暗号化は非対応

© 2022 TOYOTA MOTOR CORPORATION. All Rights Reserved.

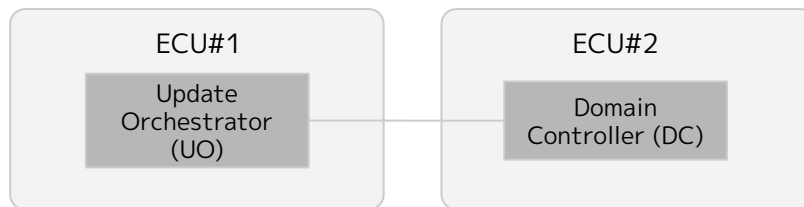
TUS の構成と動作モデル

TUS の構成と動作モデル - 前提

PROTECTED
関係者外秘



- 静的な構成図・及び具体的なシーケンス（起動処理、Campaignの問い合わせ、UO/DCのUPDATE FLOW起動、状態制御）を用いて、各種Moduleの関係を説明
- UO/DCの構成は下記とする（SDK010構成を説明用に簡略化したもの）



SoftwareModule構成

PROTECTED
関係者外秘

TOYOTA
INFO TECH

Lua
Extension

Pure Lua
Module

Process

Update Orchestrator(UO)

Domain Controller(DC)

UO/DC Main共にシステムに書き込まれている想定
※UO/DCMainは、libluaでlua実行環境を提供するNative実装

TUP

UPDATE
FLOW
(UO)

UPDATE
FLOW
(DC)

TUP
for DC

UO Main(main.lua)

providers domains tup_parser
gRPC protobuf

execute

UPDATE FLOW(UO) in TUP

providers domains tup_parser
fsm gRPC protobuf

expose

httpd

TUP

DC Main(lua)

gRPC
Server

Notification
Queue

fsm tup_parser
gRPC protobuf

switch
(LuaStateをDCMainと切り替え)

UPDATE FLOW(DC) in TUP

Updater(file) fsm tup_parser
Updater(dpkg) gRPC protobuf

Data Pass
(TUPやUpdateScript等の取得)

Control Pass
(DCの構成や状態取得、状態更新等)

UPDATEFLOWでは有限状態機械(FSM)でUpdate処理が進行
※詳細は後ほど説明

SoftwareModule構成

PROTECTED
関係者外秘

TOYOTA
INFO TECH

Lua
Extension

Pure Lua
Module

Process

Update Orchestrator(UO)

Domain Controller(DC)

providers
更新情報に関する機能(キャン
ペーン確認、結果通知)

domains
ドメイン管理機能(構成情報な
どの取得)

tup_parser
tup parser library

※domain構成の設定について
は後ほどご説明

fsm
UOのFSMの制御機能

domains
ドメイン管理機能、DCの更新制
御(DCの状態更新等)

httpd
DCからTUPをHTTP経由で取
得可能に。
(010ではシステムの httpdを利用し、TUP等をファイルシス
テムに書き出している)

UO Main(main.lua)

providers

domains

tup_parser

gRPC

protobuf

execute

UPDATE FLOW(UO) in TUP

providers

domains

tup_parser

fsm

gRPC

protobuf

expose

httpd

TUP

DC Main(lua)

gRPC
Server

Notification
Queue

fsm

tup_parser

gRPC

protobuf

switch
(LuaStateをDCMainと切り替え)

UPDATE FLOW(DC) in TUP

Updater(file)

fsm

tup_parser

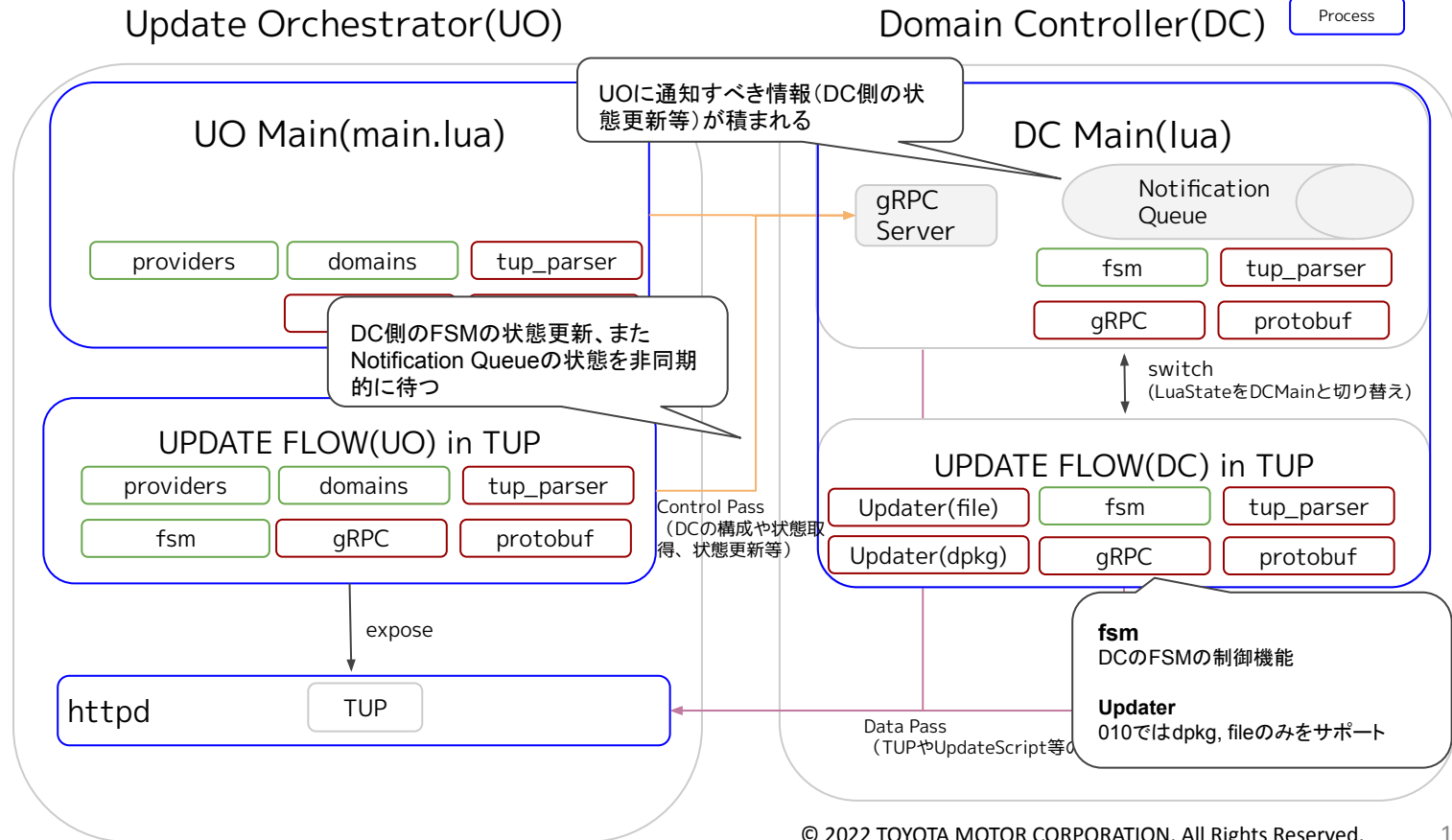
Updater(dpkg)

gRPC

protobuf

Data Pass
(TUPやUpdateScript等の取得)

Control Pass
(DCの構成や状態取
得、状態更新等)



Target Device上の設定ファイル

PROTECTED
関係者外秘



Update Orchestrator

ファイルパス※1	概要
config	rootのconfiguration file httpdの設定や下記ファイルのprovider_path, domain_pathが記載
<provider_path>/config	provider(TUPの保持環境)固有の設定ファイル (010ではEMOOTA SDPv2, LocalFileSystemのみサ ポート) 例: LocalFileSystemの場合、tupのファイルパスが記載
<domain_path>/config	各domain名及びdomainの設定ファイルへのパスが書 かれたファイル
<each_domain_config_p ath>	domain毎の情報が書かれたファイル 例: gRPC Serverの接続情報

※1. 010でファイルパス詳細はSDK Documentを参照
※UOの設定ファイルは、`tus dtool...`で書き込み可能

Domain Controller

ファイルパス※2	概要
version/<target_name>	UpdateTargetや UpdateTargetGroupのバージョン が格納されたファイル ※SDK010ではファイル形式のみ サポート

※2. 実行バイナリ(domain_controller)からの相対パス

UPDATE FLOW 概要

PROTECTED
関係者外秘

TOYOTA
INFOTECH

- TUPには、UO/DC向けのUPDATE FLOWが含まれる
 - TUP Generatorがmetadata(json)からUPDATE FLOWを自動生成
- Programming Modelとして有限状態機械（FSM）を採用
 - Update処理を、状態、遷移前・遷移時・遷移後処理、rollback処理（追加順と逆順に呼ばれる）で記述が可能



- FSMの状態は外部からのイベント（DC側FSMの状態変化、User操作等）で更新
- Update時のUpdateTargetの組み合わせ等により、FSMに関わる状態や遷移条件等を決める必要がある（O10では固定）

例

UOのFSM



DCのFSM



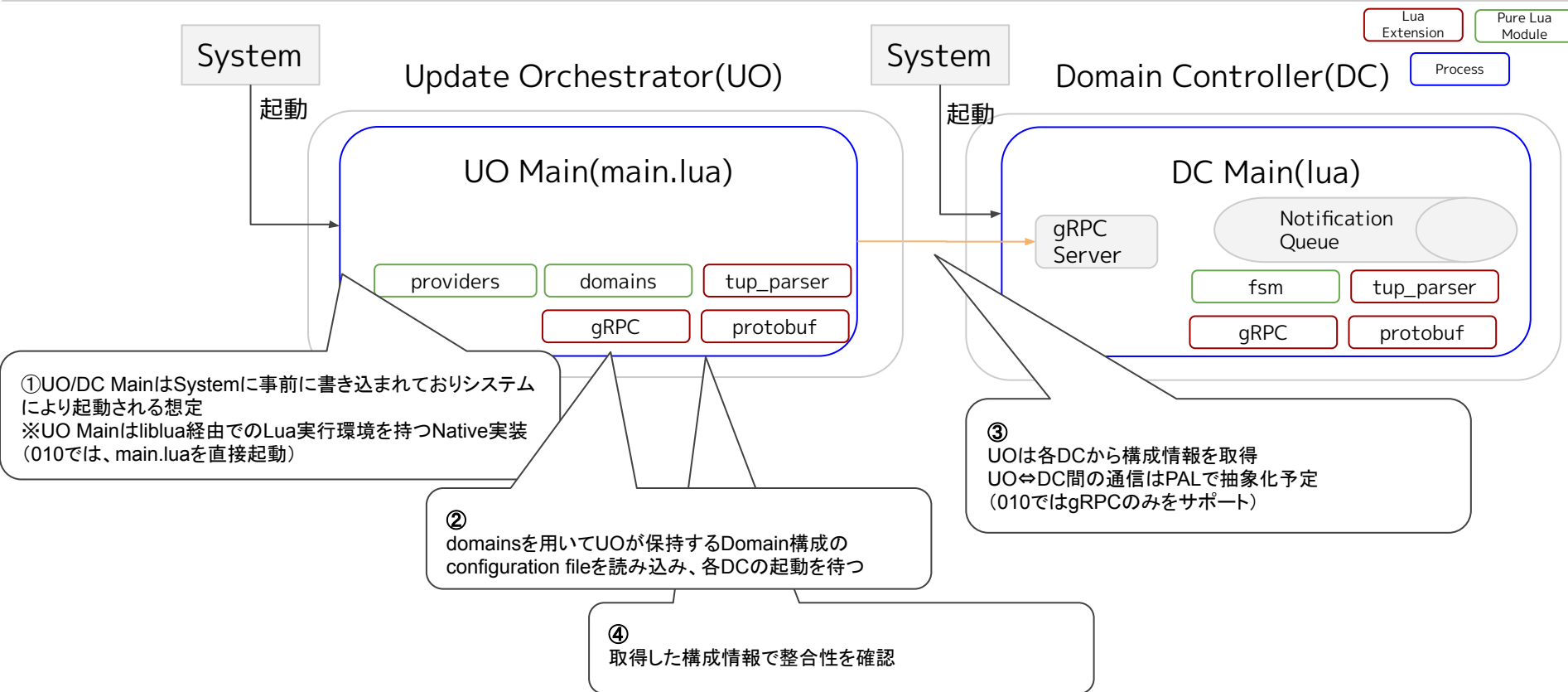
DC側FSMのUpdateScript（FSM部のみ一部抜粋）

```
dc.fsm:add_transition({
  from = nil, -- from initial
  to = "downloaded",
  action = download,
})
dc.fsm:add_transition({
  from = "downloaded",
  to = "updated",
  cond = check_dc_updated,
  action = update,
})
```

起動処理

PROTECTED
関係者外秘

TOYOTA
INFO TECH



Campaignの問い合わせ

PROTECTED
関係者外秘

TOYOTA
INFOTECH

Lua
Extension

Pure Lua
Module

Process

Update Orchestrator(UO)

Domain Controller(DC)

TUP
Provider

EMOOTA
SDPv2

Local
FileSystem

EMOOTA
SDPv3

UO Main(main.lua)

providers

domains

tup_parser

gRPC

protobuf

DC Main(lua)

gRPC
Server

Notification
Queue

fsm

tup_parser

gRPC

protobuf

①収集した構成情報を用いてCampaignの問い合わせ(providers)を実施

EMOOTA SDPv2

構成情報とサーバーの登録済み構成が一致するかを確認
構成情報から対応するCampaignがあればTUPをDownload

Local File System

config fileに書かれたファイルパスにTUPが存在するか確認

※ProviderとしてEMOOTA SDPv3やDummy Serverなどをサポート予定

UO用のUPDATE FLOWの起動

PROTECTED
関係者外秘

TOYOTA
INFOTECH

Lua
Extension

Pure Lua
Module

Process

- ①UO用のUPDATE FLOW起動に必要な箇所のみを**部分**ダウンロード
(010では、TUPを一括ダウンロード)
- ②正当性検証等を実施

Provider

EMOOTA
SDPv2

Local
FileSystem

EMOOTA
SDPv3

Update Orchestrator(UO)

UO Main(main.lua)

TUP

providers

domains

tup_parser

gRPC

protobuf

execute

UPDATE FLOW(UO) in TUP

providers

domains

tup_parser

fsm

gRPC

protobuf

Domain Controller(DC)

DC Main(lua)

gRPC
Server

Notification
Queue

fsm

tup_parser

gRPC

protobuf

- ③信頼性観点より別Processとして起動
liblua経由でのLua実行環境を持つNative実装

- ④DC側の状態を非同期的に取得するためworker threadを起動
- ⑤UPDATE FLOWを読み込み、FSM構築を実施(SDK010は固定)

開始

Update中

全DCが終了

終了

PROTECTED
関係者外秘

TOYOTA
InfoTECH

Lua
Extension

Pure Lua
Module



UOからDCへの状態更新設定

PROTECTED
関係者外秘

TOYOTA
INFOTECH

Lua
Extension

Pure Lua
Module

Process

Update Orchestrator(UO)

UO Main(main.lua)

providers

domains

tup_parser

gRPC

protobuf

UPDATE FLOW(UO) in TUP

providers

domains

tup_parser

fsm

gRPC

protobuf

httpd

TUP

Domain Controller(DC)

DC Main(lua)

gRPC
Server

Notification
Queue

fsm

tup_parser

gRPC

protobuf

switch
(LuaStateをDCMainと切り替え)

UPDATE FLOW(DC) in TUP

Updater(file)

fsm

tup_parser

Updater(dpkg)

gRPC

protobuf

①DCのFSMに対して状態を設定(010では下記)
dc.fsm:set_environment("url")

②DC側FSMが再評価
③Condを満たしたため終了状態を遷移し、UPDATE FLOWに記載の更新を実施

開始

InnerPkgのURLが
取得可能

終了
更新実施

DCからUOへの状態更新通知

PROTECTED
関係者外秘

TOYOTA
INFOTECH

Lua
Extension

Pure Lua
Module

Process

Update Orchestrator(UO)

UO Main(main.lua)

providers domains tup_parser
gRPC protobuf

UPDATE FLOW(UO) in TUP

providers domains tup_parser
fsm gRPC protobuf

④UO側のFSMが終了に遷移し、結果通知を行い終了

開始

Update中

終了

Domain Controller(DC)

DC Main(lua)

gRPC
Server

Notification
Queue

fsm tup_parser
gRPC protobuf

②DC側の状態更新
をキューに積む

UPDATE FLOW(DC) in TUP

Updater(File
) fsm tup_parser
gRPC protobuf

③UPDATE FLOWは、DC側の状態
変化を待ち再評価

①DC側FSMが終了へ遷移

開始

InnerPkgのURLが
取得可能

終了
更新実施

Demonstration

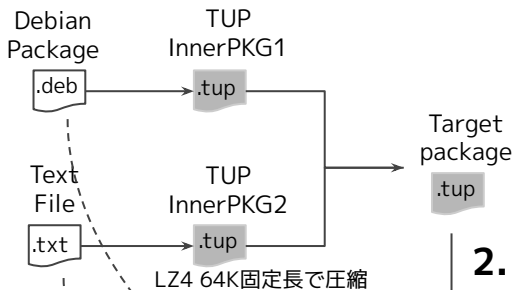
Demo デモンストレーション概要説明

PROTECTED
関係者外秘

TOYOTA
INFO TECH

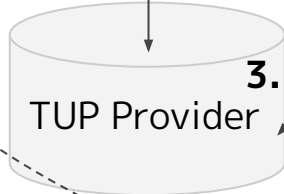


1. TUP生成



2. デプロイ

3. TUP取得



※target の構成詳細は pp.5 参照のこと

Target Device

ECU#1

ECU#2

UO

DC

DC

Demo-pkg.deb
v1.0

4. debパッケージ
更新

Demo-pkg.deb
v2.0

4. ファイル
インストール

"Alice in the
wonderland"

Demo #1
ローカル開発環境でのソフトウェアアップデート

Demo #2
OTAサーバーを経由した
ソフトウェアアップデート

Demo#1 ローカル開発環境でのソフトウェアアップデート

PROTECTED
関係者外秘



Target Device の Local FileSystem からソフトウェアアップデートを実施するシナリオ



TUS SDK実行環境
(tus command)

1. TUP生成

.tup

2. 設定情報の作成・デプロイ

configs

3. Local FS へ TUP デプロイ

4. Target Deviceでのプロセス(UO・DC等)の起動

※デモでは説明と処理の進行を合わせるために

手動で実行(本番環境ではシステムが起動する想定)

Target Device

ECU#1

ECU#2

Update

Install

Local
FS

Debian package

Text File

Demo#2 OTAサーバを経由したソフトウェアアップデート

PROTECTED
関係者外秘



※要EMOOTA SDPライセンス

