CDL Project Status Overview
CDL Project Status Overview

• Registered as P2-PC in Miranda release (11.0)
• CDL concept demo is integrated into GDP12

• Focusing on implementing proof of concept for AC
CDL Project Status Overview
Integrate CDL Into GDP12

- CDL concept demo code is integrated into GDP12

- Source code is available:
Integrate CDL Into GDP12

- CDL concept demo is designed to run on 3 GDP devices
  - It is available run with:
    - Desktop (X86, All units together)
    - 2 GDP devices (CDL/Cluster, AV) + Desktop (Vehicle Data Generator)
Details of PoC Implementation
Goal

• Is to integrate as many GENIVI components as possible into CDL

• PoC of CDL is implemented using:
  – Vehicle Simulator
  – VSS / VSI
  – RVI_Core
  – CommonAPI DBus/SomeIP
Architecture

PC (Windows10)

GENIVI Vehicle Simulator

Vehicle Data Source

Raspberry Pi3 (GDP11)

Cluster HMI Application

Vehicle Data Consumer

Provide

CDL Daemon

Collect

Store

Transfer

VSI

VSS Data

CAN Router Simulator

File (JSON)

RVI

TCC8935 (arm-linux)

AV Application

Vehicle Data Consumer

Provide

PC (Ubuntu)

Vehicle Data Viewer (Historical Data)

Big Data Server (Vehicle Data)

RVI

Ethernet

socket

SOME/IP

JSON-RPC
GENIVI Vehicle Simulator

• Vehicle Simulator as a vehicle data source

• Data Example
  
  "EMS\rSetSpeed, 0.0000, 29.36198\rEngineSpeed, 800.0001, 29.36198\rGearPosActual, 1.00, 29.36198\rGearPosTarget, 1.00, 29.36198\rAcceleratorPedalPos, 0.0000, 29.36198\rDeceleratorPedalPos, 0.0000, 29.36198\rRollRate, 0.0693, 29.36198\rSteeringWheelAngle, 0.0000, 29.36198\rVehicleSpeed, 0.0038, 29.36198\rVehicleSpeedOverGnd, 0.0038, 29.36198\rWheelSpeedFrL, 12.9269, 29.36198\rWheelSpeedFrR, 11.8869, 29.36198\rWheelSpeedReL, -1.4984, 29.36198\rWheelSpeedReR, -2.5402, 29.36198\rYawRate, 0.0014, 29.36198\r"
VSS (Vehicle Signal Specification)

• Specifies vehicle signals

```
"Speed": {
  "description": "Vehicle speed, as sensed by the gearbox.",
  "min": -250,
  "max": 250,
  "type": "Int16",
  "id": 63,
  "unit": "km/h"
}
```

• CDL daemon collects vehicle data and validates collected data referring VSS information (min/max value ranges)
VSI (Vehicle Signal Interface)

- Framework for data sharing using shared memory
Configuration File for Data Collection

• Specify data collection cycle or by event

```json
{
  "Cycle":
  {
    "200":
    [
      "Signal.Drivetrain.Transmission.Speed"
    ],
    "100":
    [
      "Signal.Drivetrain.InternalCombustionEngine.RPM"
    ]
  },
  "Event":
  [
    "Signal.Cabin.HVAC.IsAirConditioningActive",
    "Signal.Cabin.HVAC.IsFrontDefrosterActive",
    "Signal.Cabin.HVAC.Row1.LeftTemperature",
    "Signal.Cabin.HVAC.Row1.RightTemperature",
    "Signal.Cabin.HVAC.IsDualModeActive",
    "Signal.Cabin.HVAC.Row1.FanSpeed"
  ]
}```
Store Data to File

• It can be stored various format
• For this PoC, because of ease of debugging, stores collected data to JSON format
Configuration File for Data Store

- Data store location
- Maximum file size
- Storage usage size
- Storage management period
- Transaction size
Provide Data to On-board Components

• Component that wants to use vehicle data can receive desired data using CDL Client API
• CDL provides vehicle data through DBus or SOME/IP using CommonAPI
• The client can register the data that it wants to receive so that it can receive data when the value is updated or changed.
```cpp
/**
 * @description : register and authenticate client
 */
method registerClient {
  in {
    /**
     * @description : key for authentication. key could be private ssh key, password, ...
     */
    String key
  } out {
    /**
     * @description : handle for client. the value of handle is 0, when registration and authentication failed
     */
    ClientAPITypes.Handle handle
    /**
     * @description : registration result authentication
     */
    ClientAPITypes.ResultCode result
  }
}
```
CDL Client API (FIDL)

```fidl
/**
 * @description: set id list for listen. only specified data will notified to client
 */
method setListenData {
  in {
    /**
     * @description: handle obtained when registering client
     */
    ClientAPITypes.Handle handle
    /**
     * @description: signal name list to listen
     */
    String [] signalNameList
    /**
     * @description: updated = notify data when data is updated.
     *               changed = notify data when data is changed only
     */
    ClientAPITypes.NotifyType type
  }
  out {
    /**
     * @description: result for request
     */
    ClientAPITypes.ResultCode result
  }
}
```
CDL Client API (FIDL)

```fidl
/**
 * @description: get single data instantly
 */
method getData {
    in {
        /**
         * @description: handle obtained when registering client
         */
        ClientAPITypes.Handle handle
    }
    /**
     * @description: name of data
     */
    String signalName
}
out {
    /**
     * @description: result of request
     * data will be notified to client via broadcasting
     */
    ClientAPITypes.ResultCode result
}
```
CDL Client API (FIDL)

```cpp
/**
@description : notify data to client
***/
broadcast notifyData selective {
  out {
    /**
      @description : name of data
    ***/
    String signalName
    /**
      @description : type of data
    ***/
    ClientAPITypes.Types type
    /**
      @description : unit of data
    ***/
    String unit
    /**
      @description : value of data
    ***/
    ClientAPITypes.CDLValue value
    /**
      @description : timestamp of data
    ***/
    UInt64 timeStamp
  }
}
```
Provide Data to Off-board Server

- Using RVI_Core, transfers stored data to off-board server
Further Implementation
To complete PoC for AC

- Collect event data using VSI
- Regular expression on data collect configuration
- Client authentication of CDL Client API
- Transfer data using RVI_Lib (written in C)
- Data encryption
- Improve performance
Roadmap
CDL Project Roadmap

CDL

Implementing PoC for AC

PoC Done

AC Done

GENIVI

CDL

2017

May

May

Genivi

16th AMM

17th AMM

18th AMM

2018

Oct

2018
CDL Showcase for 2017 Spring AMM
Use Cases

Cluster Application
- Vehicle Speed
- Engine Speed

AV Application
Driving Restriction

Off-board Server
- Transfer Data to Server
- Historical View

Vehicle Simulator
Running Status
- Vehicle Speed
- Engine Speed

CDL
Collect → Store → Provide

On-board

Off-board
Thank you!

Visit GENIVI at http://www.genivi.org or http://projects.genivi.org
Contact us: help@genivi.org