TEST CONTROL INTERFACE

OVERVIEW

Nov 15, 2016

GOALS / SCOPE

mation

ed for testing

plement TCI

USDOT CERTIFICATION PROGRAM

- The USDOT competitively selected three certification service providers (7Layers, Danlaw, and Southwest Research Institute (OmniAir))
- All three work together through the Certification Operating Council (COC) to support certification testing for CV Pilots
- Vision is to help the industry to organize and run a self-sustaining certification program supporting deployment of DSRC-based services.

Danlaw provides connected vehicle telematics solutions and embedded electronics to OEMs and their Tier-1 supply chain.

OmniAir Cerfication Services (OCS) is a non-profit organization founded by OmniAir to execute independent certification for the intelligent transportation industry.

7Layers is an international group of engineering & test centers having a core competence in wireless technologies.

TEST SPECIFICATIONS FOR CERTIFICATION

V2X Test Specifications supporting device certification available on http://www.github.com

Test Specifications	Title / Scope
WAVE802.11-TSS&TP	Test Suite Structure and Test Purposes for IEEE 802.11 (scope of DSRC)
WAVEMCO-TSS&TP	Test Suite Structure and Test Purposes for Multi Channel Operation (IEEE 1609.4)
WAVENS-TSS&TP	Test Suite Structure and Test Purposes for Network Services (IEEE 1609.3)
WAVESEC-TSS&TP	Test Suite Structure and Test Purposes for Security Services (IEEE 1609.2)
J2945.1-TSS&TP	Test Suite Structure and Test Purposes for J2945.1
Test Control Interface	Test Control Interface protocol specification
Device Profiles	Test requirements supporting different device types

TECHNICAL STANDARDS

Data Messages

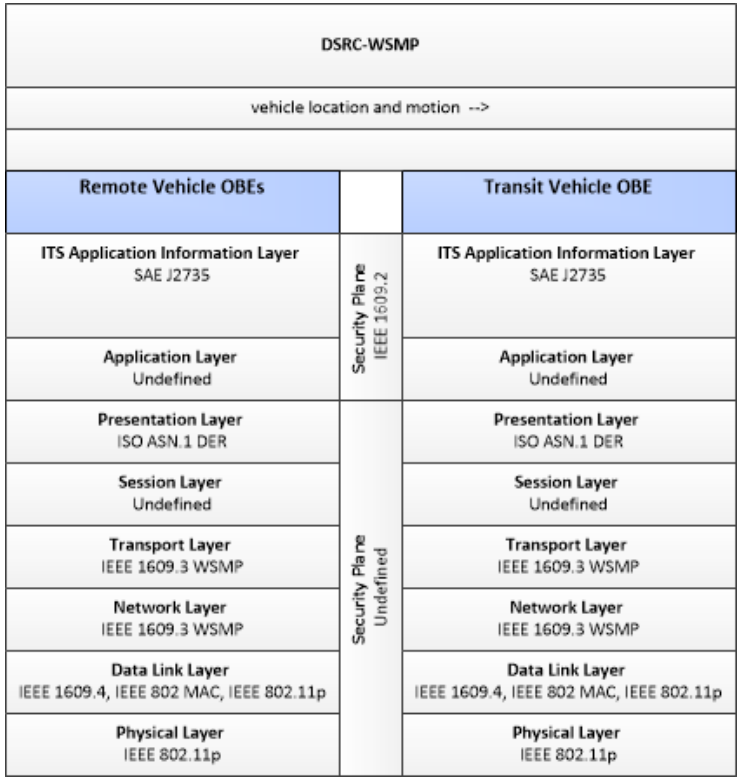
- SAE J2945/1 Requirements for V2V
- Encoding (ISO ASN.1 UPER)
- Process Information (SAEJ2735)

Data Transmission

- Transport (IEEE1609.3 WSMP, IPv6)
- Security (IEEE 1609.2)
- Link (IEEE 1609.4)
- Physical (IEEE 802.11)

Device Profiles

- **V2V** per SAE J2945/1
- **V2V + SCMS** per SAE J2945/1
- **CV Pilot OBU**
- **CV Pilot RSU** per USDOT RSU 4.1 spec

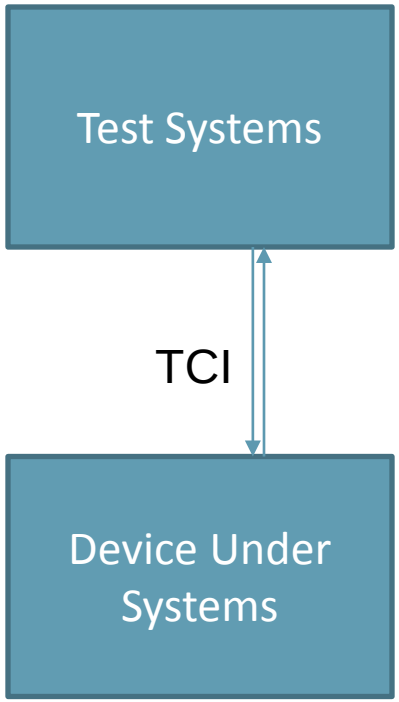


WHAT IS TEST CONTROL INTERFACE

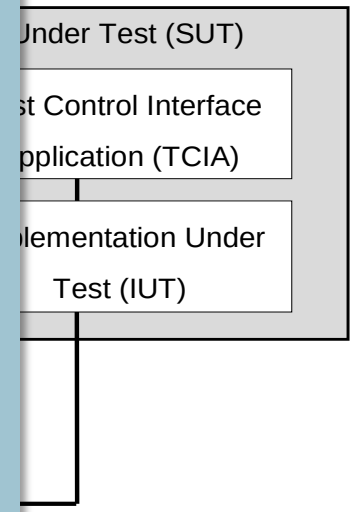
- Defines protocol and messages exchanged between a Test System (TS) and a Device Under Test (DUT)

- Developed by the Certification Operating Council (COC)
- Public and open specification; published on Github.com https://github.com/certificationoperatingcouncil/TCI_ASN1

- Defined in the TCI specification using ASN.1 notation
- Independent of the TS or DUT implementation details
- Supported the COC Test Specifications
  - Based on IEEE 802.11, IEEE 1609, SAE J2735
  - Not part of the IEEE or SAE standards



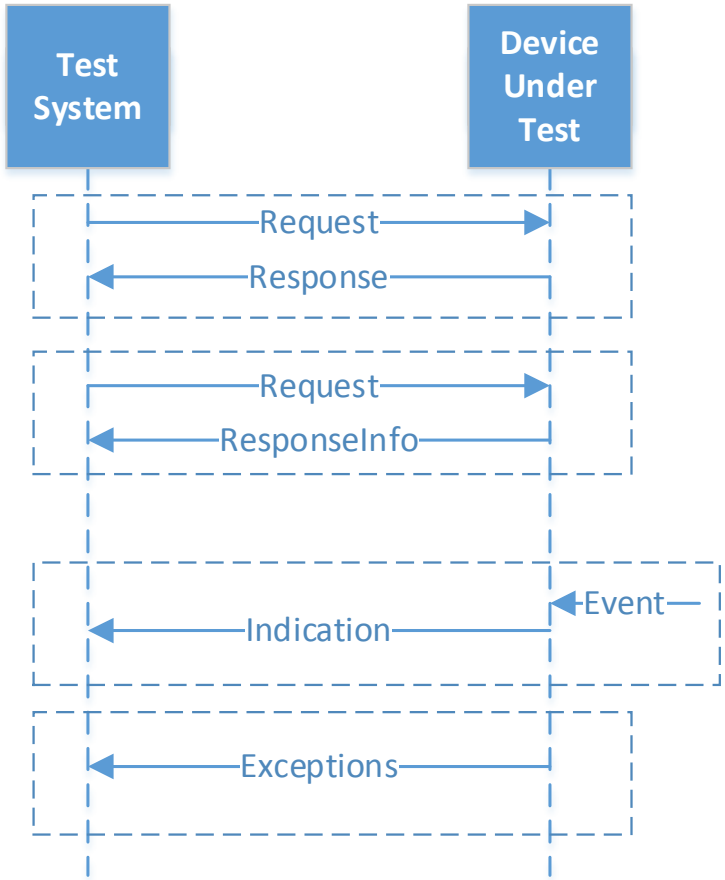
TEST SYSTEM ARCHITECTURE



s
13001
om which *SetInitialState*

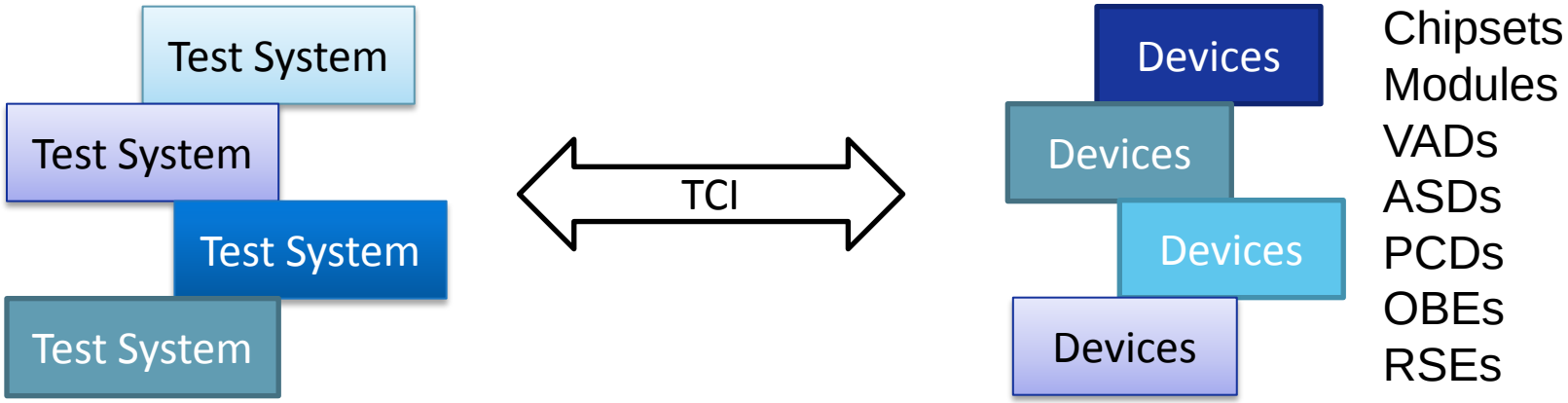
Test System vendors supporting Test Control Interface

TEST CONTROL INTERFACE



WHY TCI

??? WHAT IS THE ALTERNATIVE ???

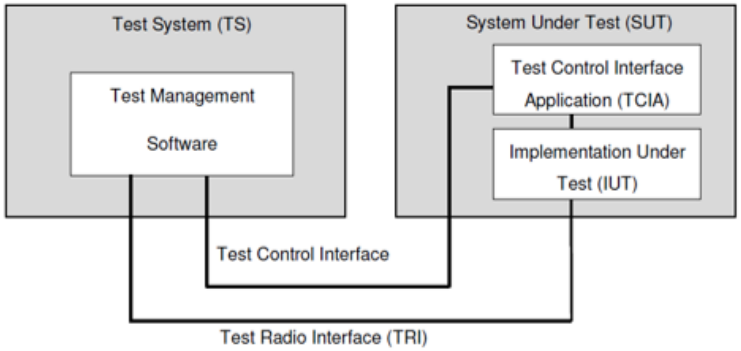


- Take advantage of automated testing
  - Low costs of testing
- Utilize common protocol between test tools and devices
- Exercise parameters for “non-normal” conditions
- Scalable as devices and applications multiply
- Focused on module level testing, (rather than “system level” testing)

MODULE VS VEHICLE LEVEL TESTING

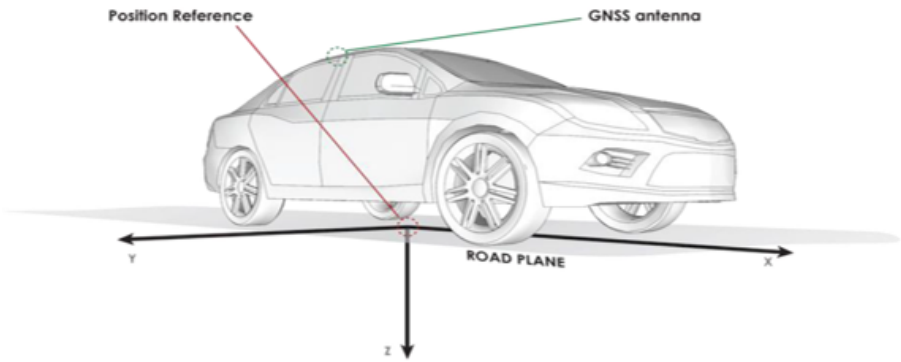
Module Level Testing

- Component Suppliers
- Fully exercise lower layers and individual data elements
- Controlled / Repeatable Tests
- Automated = Cost Effective



Vehicle Level Testing

- OEMs / Regulating Agencies
- Conformance testing at "system level"
- Positioning system accuracy
- Tx Performance Reqs
- Test Track / Rotation Table



Source: SwRI/OmniAir presentation at 2016 ITS World Congress

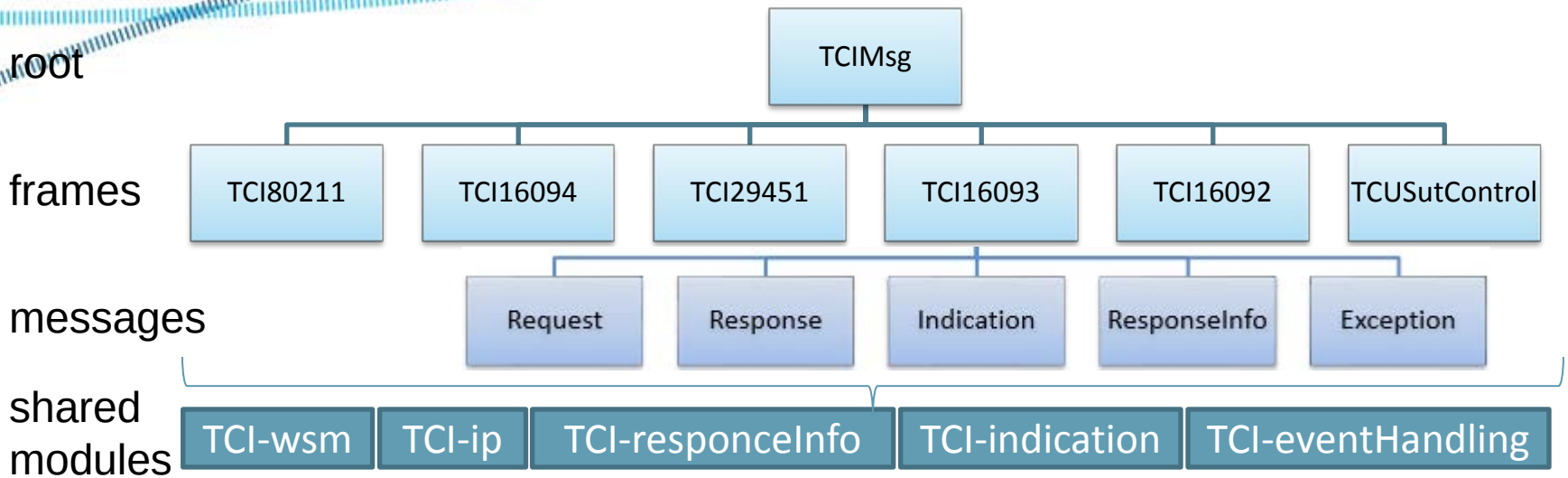
TCI EXPRESSED IN ASN.1

TCIDispatcher.asn

```
--<OSS.PDU TCI-MsgDispatcher.TCIMessage>--
TCI-Dispatcher
DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

IMPORTS

TCI16093            FROM TCI-16093
TCI16094            FROM TCI-16094
TCI16092            FROM TCI-16092
TCI80211            FROM TCI-80211
TCI29451            FROM TCI-29451
TCISutControl       FROM TCI-SutControl
Time64              FROM TCI-CommonTypes

;

TCIMsg ::= SEQUENCE{
    version     INTEGER(1), -- protocol version 1
    time        Time64,             -- UTC timestamp with microsecond, GPS syncronized
    frame       Frame,              -- protocol layer
    ...
    }

Frame ::=CHOICE {
    d16093 [1] TCI16093,
    d16092 [2] TCI16092,
    d80211 [3] TCI80211,
    d16094 [4] TCI16094,
    d29451 [5] TCI29451,
    sutCtrl [6] TCISutControl,
    ...
}

END
```

- Abstract Syntax Notation 1
- Specifies data structures at high level of abstraction
- Standardized and vendor independent
- Supported by encoding rules to represent values in data structures when they are transferred over computer networks
- TCI messages are OER encoded
  - OER (octet encoding rules) – one of ISO standardized ASN.1 encoding rules
  - Also used by IEEE1609.2

TCI MESSAGES ORGANIZATION



- All TCI messages have common root (TCIMsg)
- TCI message comprised of message frames (one for each test specification)
- All frames share the same message types: Request, Response, etc.
- Shared modules specify parameters for each type of message

TCIMSG/TCI16093/REQUEST

TCI-dot3-req-Dot3SetWsmTxInfo.txt

PDU Name/Identifier	Value	Typereference	Built-in Type	Default Value
TCIMsg		TCIMsg	SEQUENCE	
version	1		INTEGER	
time	1234123412341234	Time64	INTEGER	
frame	d16093	Frame	CHOICE	
d16093	request	TCI16093	CHOICE	
request		Request	SEQUENCE	
messageId	2	MsgID	INTEGER <MES...	
value	Dot3SetWsmTxInfo		MESSAGE-ID-A...	
Dot3SetWsmTxInfo		Dot3SetWsmTxI...	SEQUENCE	
psid	content	Psid	CHOICE	
radio		RadioInterface	SEQUENCE	
security		SecurityContext	SEQUENCE	
channelIdentifier	172	ChannelNumbe...	INTEGER	
timeslot	continuous	TimeSlot	ENUMERATED	
dataRate	6	DataRate80211	INTEGER	
transmitPowerLevel	15	TXpower80211	INTEGER	
infoElementsIncluded	'000000000000000000000000'B	WaveElementsI...	BIT STRING	'0000000000000...
userPriority	4	UserPriority	INTEGER	
destinationMACAddr	'FFFFFFFFFFFF'H	MACaddress	OCTET STRING	'FFFFFFFFFFFF'H
expiryTime		Time64	INTEGER	
channelLoad		Opaque	OCTET STRING	
repeatRate		RepeatRate	INTEGER	
payload		Opaque	OCTET STRING	

- All TCI messages have version and timestamp
- TCI reuses data structures from 1609, J2735
  - Minimizes data conversion

ASN1 Tools can auto-generate C/C++/Java code for encoding/decoding messages

Encoding Viewer

OER

```
00000000  00 01 00 04 62 6D BF 9A 01 F2 81 80 00 02 17 60  ....bm.........`
00000010  80 20 80 00 03 00 03 01 AC 03 06 0F 00 00 00 04  . ..............
00000020  FF FF FF FF FF FF                                ......
```

ASN1 Tools can aid in interpreting encoded messages

MESSAGE AMONG TCI FRAMES

TCI80211	TCI1609.4	TCI16093	TCI16092	TCI29451	TCISutControl
WSM Tx/Rx group: SetInitialState, SetWsmTxInfo, StartWsmTx, StopWsmTx, StartWsmRx, StopWsmRx	WSM Tx/RX group, IP group: GetIPv6InterfaceInfo, StartIPv6Tx, StopIPv6Tx, StartIPv6Rx, StopIPv6Rx, SetIPv6Address, StartIPv6Ping, StopIPv6Ping	WSM Tx/RX group, IP group, WSA Tx group: StartWsaTxPerdiodic, StopWsaTxPeriodic	WSM Tx/Rx group, WSA Tx group	WSM Tx/Rx group, VehSensor group	SutControl group

TYING TEST CASES & TCI

Identifier	TP-16093-WSM-MST-BV-01
Summary	To verify that the IUT will transmit a WSM with the correct version number and EtherType.
Test Configuration	TC1
IUT	IUT
Reference:	
PICS Selection	

Pre-test conditions

- The IUT is in the initial state

Test Sequence

Step	Type	Description	Verdict
1	Configure	The IUT is configured to transmit **WSM_without_nExt** in Table 7-1.	
2	Stimulus	The IUT transmits WSM	
3	Verify	The IUT transmitted WSM	Pass/Fail
4	Verify	WSM is included in 802.11 frame, containing Logical-Link Control section, containing 'Type' field indicating EtherType value 0x88DC.	Pass/Fail
4	Verify	WSM N-Header contains 'Subtype' (bits 4-7) indicating '0'	Pass / Fail
5	Verify	WSM N-Header contains 'WSMP-N Header Option Indicator' (bit 3) indicating '0' for **WSM_without_nExt,** or indicating '1' for **WSM_nExt.**	Pass / Fail
6	Verify	WSM-N-Header containing 'WSMP Version' indicating '3'	Pass/Fail
7	Configure	The IUT is configured to transmit **WSM_nExt** in Table 7-2.	
8	Procedure	Repeat steps 2-6	

All messages in this example are prefixed with TCIMsg.TCI1609.*

TS->SUT: request.SetInitialState
SUT->TS: response

TS->SUT: request.SetWsmTxInfo
SUT->TS: response

TS->SUT: request.StartWsmTx
SUT->TS: response

Test System validation of received messages

TS->SUT: request.StopWsmTx
SUT->TS: response
TS->SUT: request.SetWsmTxInfo
SUT->TS: response
Repeat from step 2

TCI SPEC USE CASES AND MESSAGE SEQUENCES

- TCI doesn't explain how execute test cases
- Rather, TCI provides use cases and corresponding message sequences

Table 30 Use cases supported by TCI16093

UC #	Use case objective	Flow Direction	Message Sequence
1	Reset the SUT to the Initial state	TS -> SUT SUT -> TS	request.SetInitialState response
2	The SUT transmits a single or periodic WSMs	TS -> SUT SUT -> TS TS -> SUT SUT -> TS	request.SetWsmTxInfo response request.StartWsmTx response
3	The SUT stops transmitting periodic WSMs	TS -> SUT SUT -> TS	request.StopWsmTx response
4	The SUT receives WSMs and sends event indications to the TS	TS -> SUT SUT -> TS SUT -> TS	request.StartRx response indication
5	The SUT stops receiving WSMs	TS -> SUT SUT -> TS	request.StopRx response

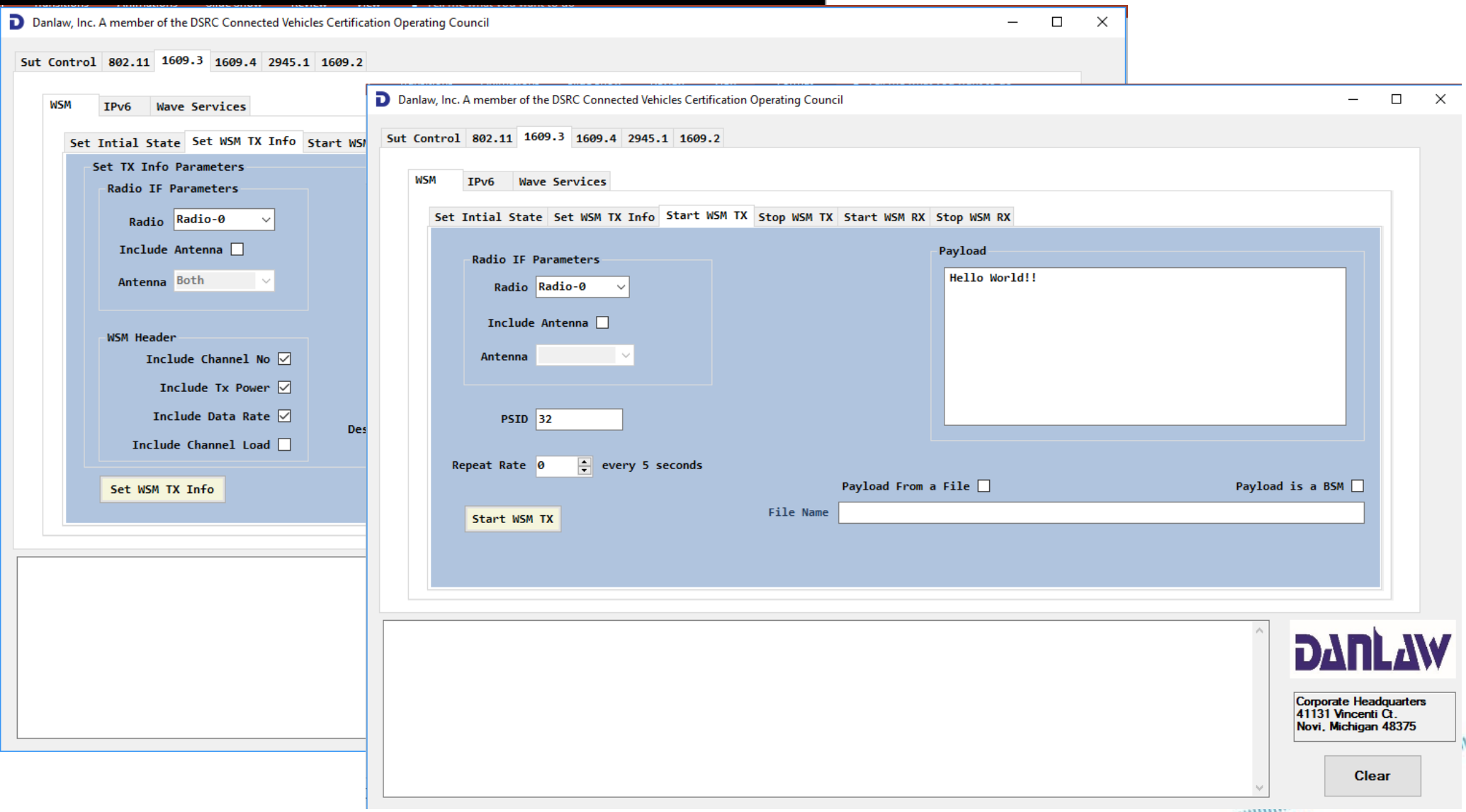
WHAT DUT VENDOR MUST DO

- DUT must support IPv4-based link (Ethernet, BroadR, USB, WiFi, etc)
- Vendor must implement TCI Application (TCIA)
  - DUT is under control of TCI Application
    - Generates BSM, reads security credentials, reboots DUT
  - TCIA is not present in the product offered for sale

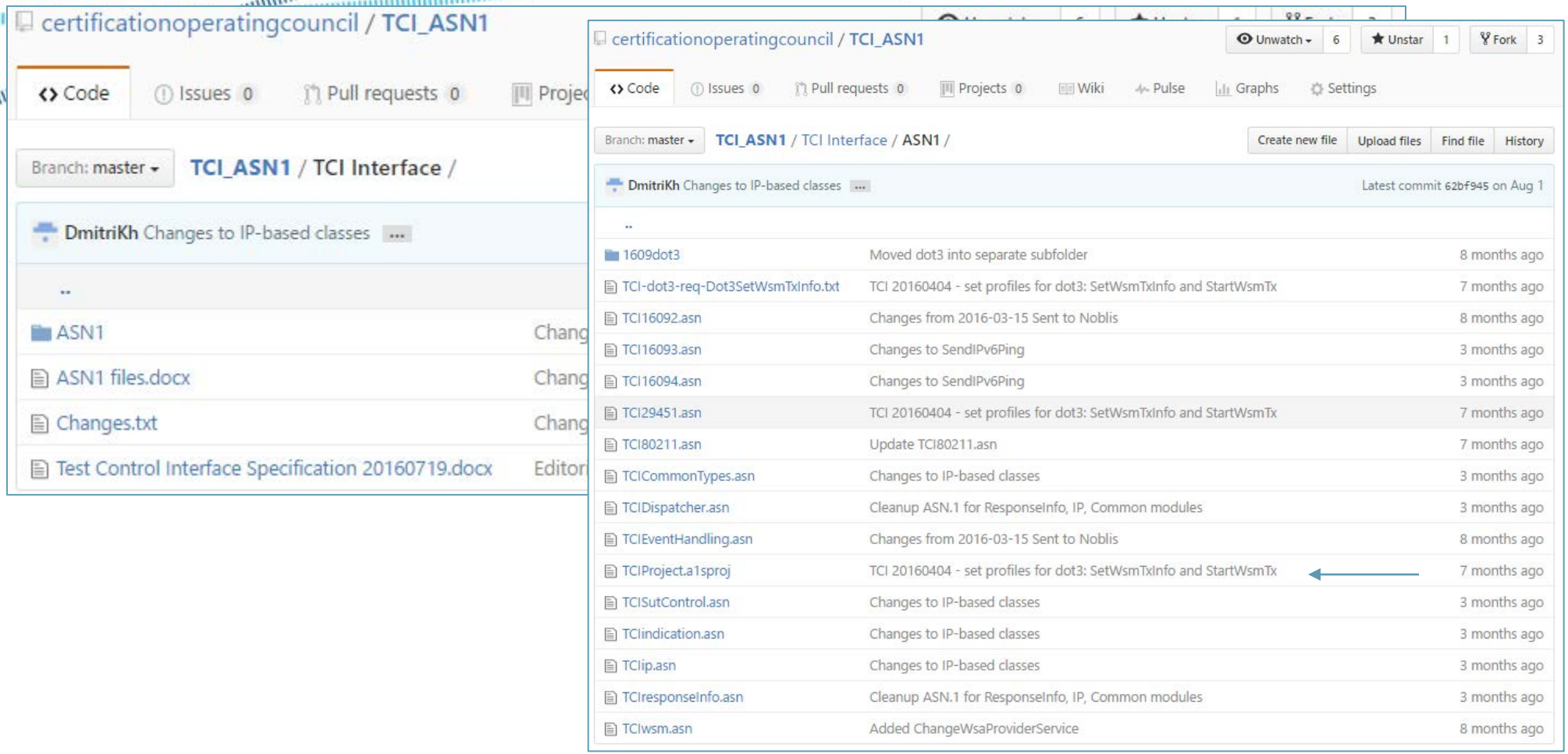
Simulates a test system by generating TCI messages

Download from:
https://github.com/certificationoperatingcouncil/TCI_Tools

TCI MESSAGE GENERATOR



STARTING POINT - GITHUB



https://github.com/certificationoperatingcouncil/TCI_ASN1/releases

Current release V0.3.0

WHERE TO GET MORE INFORMATION

Where to get support for TCI - Certification Operating Council:
certificationoperatingcouncil@gmail.com

Certification Test specifications:
https://github.com/certificationoperatingcouncil/COC_TestSpecs

Test Control Interface Specification:
https://github.com/certificationoperatingcouncil/TCI_ASN1

Test Control Interface Tools:
https://github.com/certificationoperatingcouncil/TCI_Tools

DSRC Proxy:
https://github.com/certificationoperatingcouncil/DSRCproxy

CONTACTS

Dmitri Khijniak

dmitri.khijniak@7layers.com

949-732-8022

Automotive Technology Services Development Manager

7layers - Irvine, California

7layers is the wireless/smart world expert of Bureau Veritas

- founded in 1999
- worlds' leading wireless group
- located around the world with a strong footprint in Asia, Europe &North America.
- with more than 300 employees
- Independent test and validation services laboratory accredited to ISO 17025

NORTH AMERICA

USA

Irvine, CA

Phone +1.949.716.6512

E-Mail info.us@7layers.com

Sunnyvale, CA

Phone +1.669.600.5293

E-Mail info.us@7layers.com

EUROPE

Germany

Ratingen

Phone +49.2102.749.0

E-Mail info@7layers.com

Agency Spain

Bilbao

Phone +34.634.507.296

E-Mail info@7layers.com

Agency France

Paris

Phone +33.612.717.783

E-Mail info.fr@7layers.com

ASIA

P.R. of China

Beijing

Phone +86.10.6805.0368

E-Mail info.cn@7layers.com

Shenzhen

Phone +86.755.865.23100

E-Mail info.cn@7layers.com

Japan

Yokohama

Phone +81.45.949.6020

E-Mail info.jp@7layers.com

South Korea

Suwon

Phone +82.70.8853.2301

E-Mail info.kr@7layers.com

Agency Taiwan

New Taipei City

Phone +886.2.2696.2828.237

E-Mail info.tw@7layers.com