The case for the “Vehicle Station Gateway” and the “Unified Gateway Protocol”

Equipment and Tool Institute

VSG-UGP-ITS_ETI-ToolTech_2014-04-30.pptx

Vehicle Data Access

- Many people are looking to access vehicle network data for a wide variety of use-cases
- Most all are utilizing the SAE J1962 connector to access the in-vehicle network
- OEM's will tell you, and they are correct, that the SAE J1962 connector was made for a diagnostic tool to access the network and CARB/EPA mandated its existence.

From SAE J1962 Document

- “On-Board Diagnostic (OBD) regulations require passenger cars, and light and medium duty trucks, to be equipped with a standardized connector for purposes of access to on-board diagnostic information by “generic” test equipment. This document describes the requirements for the physical connection and associated pin usage to allow for standard access to the OBD data.”

The “OBD” SAE J1962 connector

- Certain pins on the SAE J1962 connector were specified in the standard to be used for specific OBD protocols and functions like power and grounds
- Other pins were un-regulated and left to the manufacturer to use as they desired.
- Most OEM’s used these “Discretionary” pins for their manufacturer specific (Non-OBD) protocols
- Nearly all OEM’s use the SAE J1962 connector for their “proprietary” or “Dealer” tool connections as well as the regulated use.

The “OBD” SAE J1962 connector

- Recently this connector has been utilized by aftermarket companies and insurance companies for many other uses outside the scope of the original use-case for “Generic Test Equipment” or Scan Tool access.
- Some of these use cases now have a device permanently or semi-permanently attached to the SAE J1962 connector
- These other use-cases have brought concern to the OEM’s worried about A) whether these use-cases will act responsibly on their networks and not interfere with the network in a negative way that could impede its ability to perform its intended functions impacting driver safety and/or proper vehicle operation.
- …. and B) some believe that the OEM’s do not want aftermarket offerings to compete with their factory installed features.
- C) Data privacy regulations

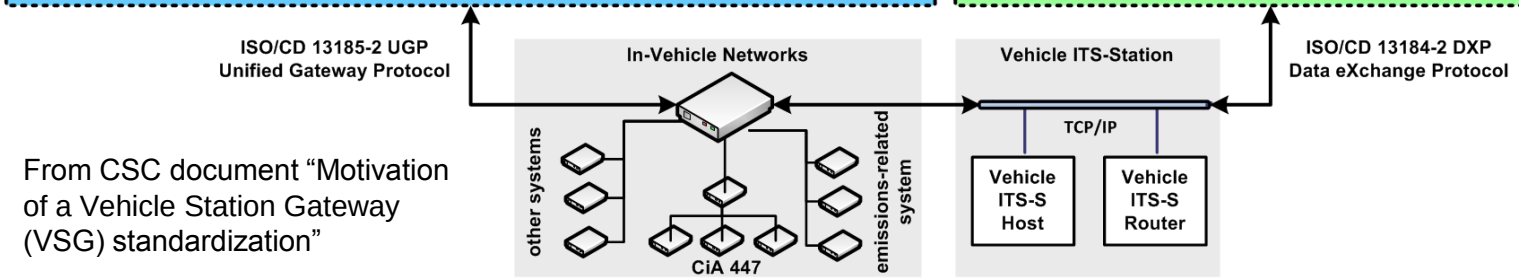
Use-Case (and cluster) examples

ISO/SAE Diagnostic, Flash Programming, Assistance, Repair

Legislated OBD/WWH-OBD Data Access	Support <= 240 ECUs ISO 15765-4 DoCAN Amd.	Tachograph ISO 16844
EC Mandate M/421 ISO 18541 Std. Access to RMI	ECU Software Update: SAE J 2534, ISO 22900-2	CiA 447 Special Vehicle Add-On Devices
Enhanced Diagnostics (Vehicle Repair)	IVI (In-Vehicle Infotainment)	Tbd.
EDR (Event Data Recorder)	Non ITS-compliant Smartphone Apps	Multiple Client (Tester) Handling
In-vehicle Data Acquisition	Safety-related Systems PTI (Germany HU §29)	Future Diagnostic Connector ISO 13400-4 DoIP
End Of Life Pyrotechnical Devices ISO 26021	PTI (Periodic Technical Inspection) 2009/40/EC	@Vehicle Roadside Assistance

ISO/CEN/ETSI/SAE Intelligent Transport

ISO ITS, CEN, ETSI Use Cases	
Remote Roadside Assistance	Tbd.
EC Mandate M/457 TPG & TPMS	(CARB) Continuous Remote OBD I/M SAE J3005
EC Mandate M/468 EV V2G-CI ISO 15118 ???	EC Mandate M/453 Cooperative Intelligent Transport
Roadworthiness EC DG MOVE 2000/30/EC	Remote Diagnostics
Remote Tachograph (fraud protection)	FMS (Fleet Management Standard)



From CSC document "Motivation of a Vehicle Station Gateway (VSG) standardization"

From the “Motivation” presentation for the Vehicle Station Gateway

- Known Issues related to in-vehicle data access via diagnostic connector
  - ISO 15031-5/SAE J1979 emissions-related OBD protocol does not prohibit “back to back” tester data request and vehicle ECU(s) response(s)
  - This causes uncontrolled network bandwidth problems which may impact the functional vehicle safety when driving on the road
  - Each device compatible to the ISO 15031-3/SAE J1962 diagnostic connector is authorized to access vehicle data according to legislation
  - Today’s vehicle architecture implementations are not able to provide asynchronous communication between external test equipment connected via the diagnostic connector and the in-vehicle network ECUs
  - Today’s diagnostic protocols are not designed to support multiple client (test devices) implementation support for vehicle ECUs
  - An update or redesign of the most common diagnostic protocol (ISO 14229 UDS) is unrealistic because of non-backward compatibility to existing release
  - Unauthorized access may cause violation of data privacy regulations

From the “Motivation” presentation for the Vehicle Station Gateway

- At this point the only standardization effort in play to address the issue of open access and common protocol for communicating with in-vehicle networks is the ISO effort for the VSG.
- This effort addresses:
  - Common wireless communication protocol
  - Independent from wireless (or wired) comm. layers
  - Common vehicle access points (with ITS)
  - Multi-client access

Current (and near term) use-case clusters

CURRENT

Vehicle Data Access Occasionally Causes In-Vehicle Network Disturbance

Roadside Assistance

IAM & VM Diagnostics

Scan Tool Supplier

Periodic Technical Inspection

IAM & VM Repair

Assistance, Inspection, Diagnostics, Repair, ...

CONCERNS

Vehicle Tracker Adapters

Smartphone Adapters

Insurance Aftermarket eCall

Data Acquisition Adapters

ITS Adapters ?

Insurance (Pay-As-You-Drive), Smartphone Apps, Car-PC, etc.

THE FUTURE

Vehicle Data Access Required in the Near Future

ITS V2I

ITS SAE J2735

Central ITS-S

Public Trans-port

ITS I2V

Roadside ITS-S

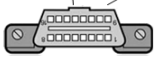
ITS EFC

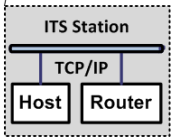
ITS ...

eCall

Intelligent Transport Systems (ITS)

Protocols, Connector, ...:
approx. 95 ISO Standards
approx. 33 SAE Standards





ITS:
approx. 150 Standards developed to satisfy EC mandate M/453

From CSC document “Motivation of a Vehicle Station Gateway (VSG) standardization”

Why VSG and UGP?

- VSG is a physical Gateway through which all outside communications to the in-vehicle networks flow.
- Install asynchronous communication between diagnostic connector and in-vehicle networks.
- Install use case-specific authentication to access use case-specific in-vehicle network data with externally connected test equipment.
- Install encrypted message transfer between externally connected test equipment and in-vehicle network ECUs.
- Utilize standardized data format and high abstract protocol defined according to ASN.1 (ISO 8825-2, X.691-0207_ASN.1 - PER)
  - **Diagnostics**, etc. (**blue use cases**) with ISO 13185 (UGP = Unified Gateway Protocol)
  - Smartphones, insurance adapters, etc. (red use cases) with ISO 13185 UGP
  - **ITS** (**green use cases**) with Data eXchange Message (defines PDU only); Other ITS standards specify communication services, etc.
- Support of a wireless interface between vehicle and connected test device.
- VSG is a future design and not currently developed to be backward compatible

Combined Technology Chain for ITS and “other” use-cases

Use Case Clusters

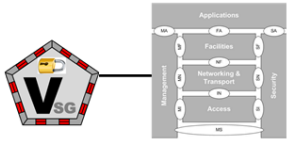
ISO/SAE Diagnostic, Flash Programming, Assistance, Repair

Use Case Clusters

ISO/CEN/IEEE/ETSI/SAE Intelligent Transport

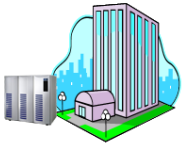
Services next to Vehicle

WI-FI Direct Peer-to-Peer (HotSpot, DHCP)



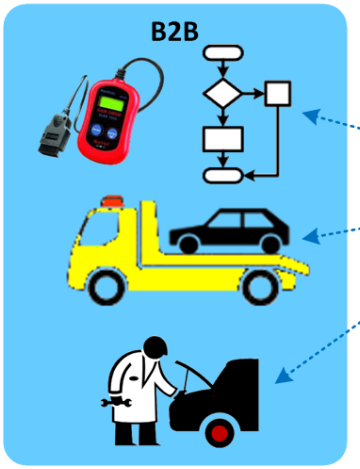
Vehicle ITS-Station

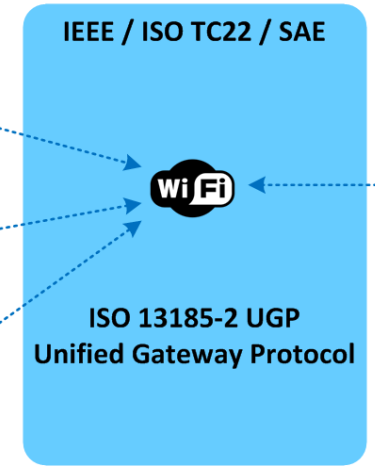
Roadside ITS-Station

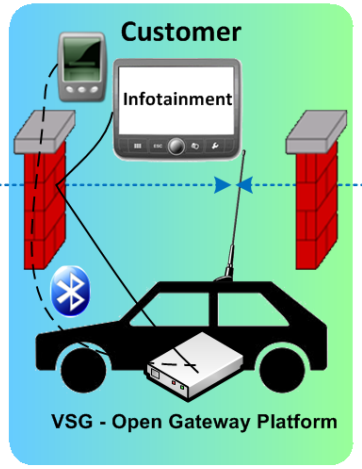


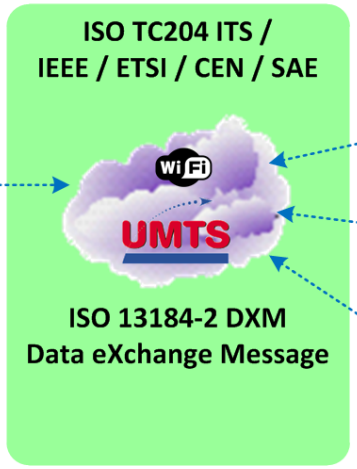
Central ITS-Station

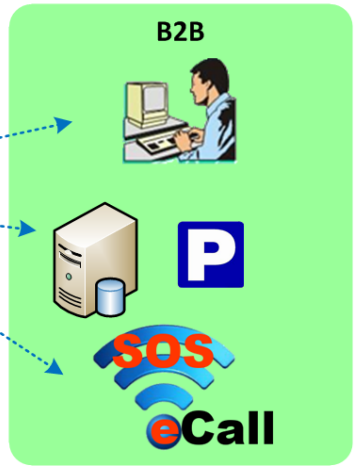
Direct Repair Technology	Direct Communication Technology	Vehicle Technology	Cooperative Communication Technology	Cooperative Services Technology











From CSC document “Motivation of a Vehicle Station Gateway (VSG) standardization”

Why VSG and UGP?

- Currently for a device to get information from a vehicle network, the device needs to understand and implement the following
  - How do I request the information?
    - How do I ask the question (diagnostic services)
  - What physical layer network do I talk to?
    - CAN?, SWCAN?, ISO9141?, FlexRay?, Baud rate?, P2/P3 timing, ..
  - What protocol (Language) do I speak?
    - UDS?, KWP2000?, SAE J1979?, SAE J1939?, OEM proprietary, …
  - How do I interpret the data?
    - Decode the “Message” to make sense to the requester. i.e., “Vehicle Speed is 25 MPH”
  - Is the data I want supported and available?

Why VSG and UGP?

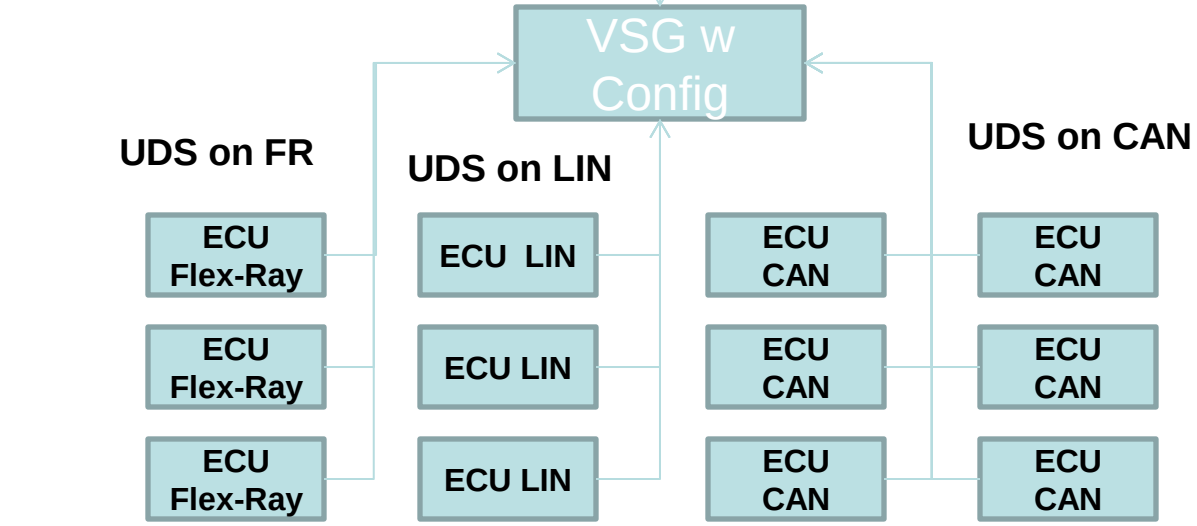
- Most all of this will be the responsibility of the VSG
- All the “Nomadic Device” needs to do is ask the question using the UGP and the message definitions (VDIF) defined in it
- …and de-construct the returning message
- It does not need to know anything about the vehicle or how it communicates internally.

Why VSG and UGP?

Ask “give me vehicle speed”
Don’t care what protocol to talk, what physical layer to connect to, what module has the information, etc.

UGP

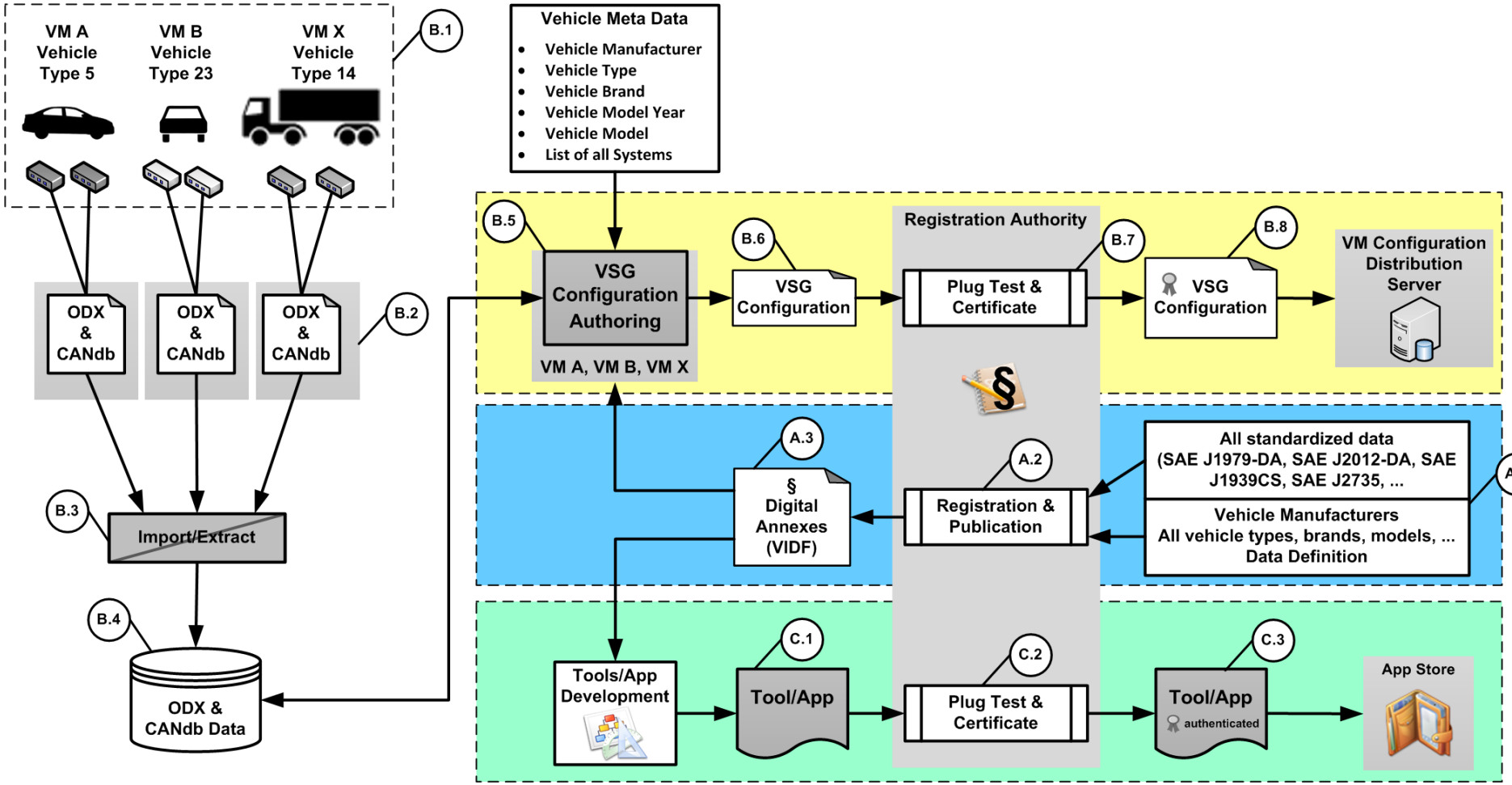
Config File maps UGP to the UDS



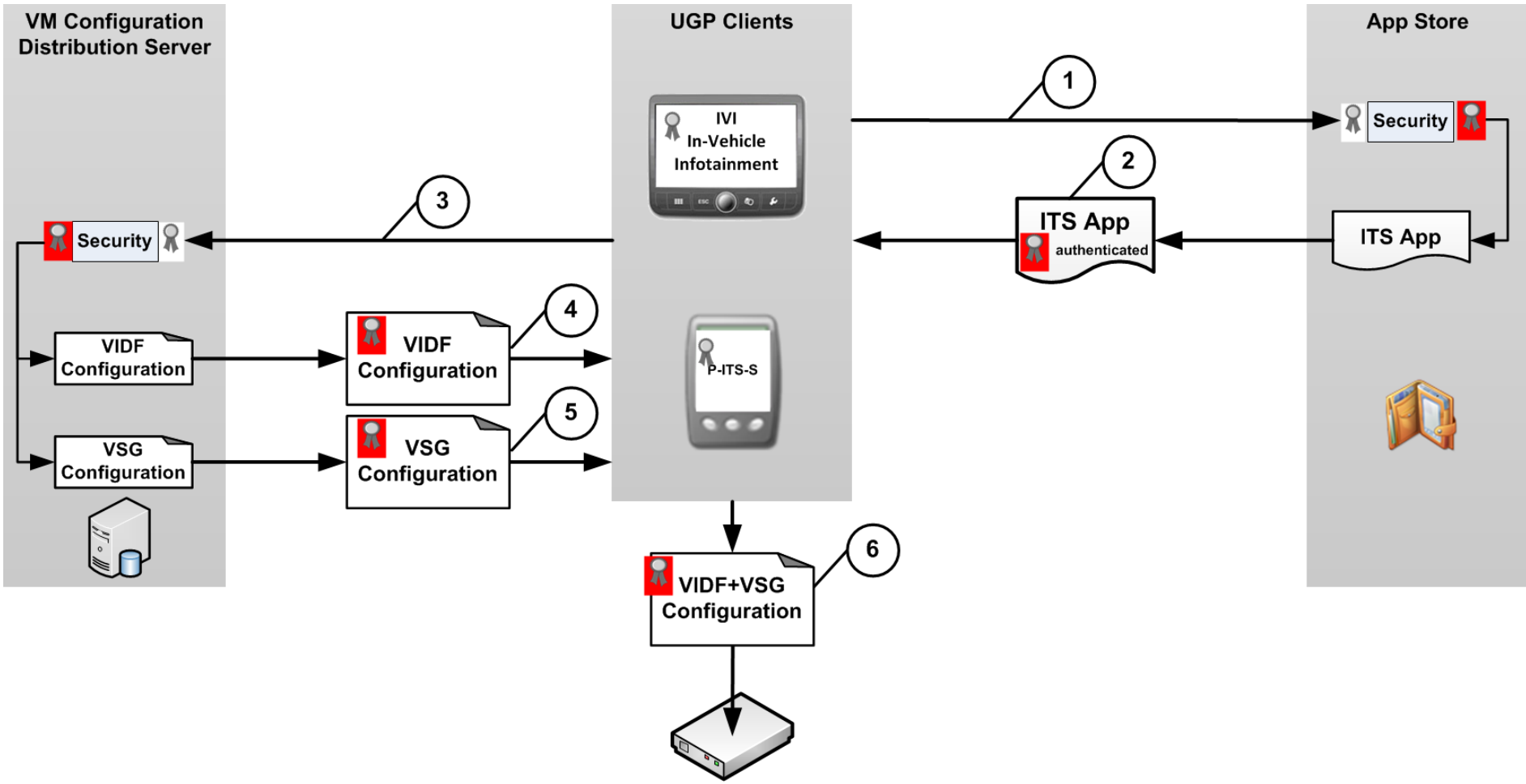
Notes on the current J3005

- Does not address the multi-client
- Only regulates how a device connected to a SAE J1962 connector communicates with the in-vehicle network
- No outside vehicle wireless standardization
- Very little input from Insurance and PEMS sectors.

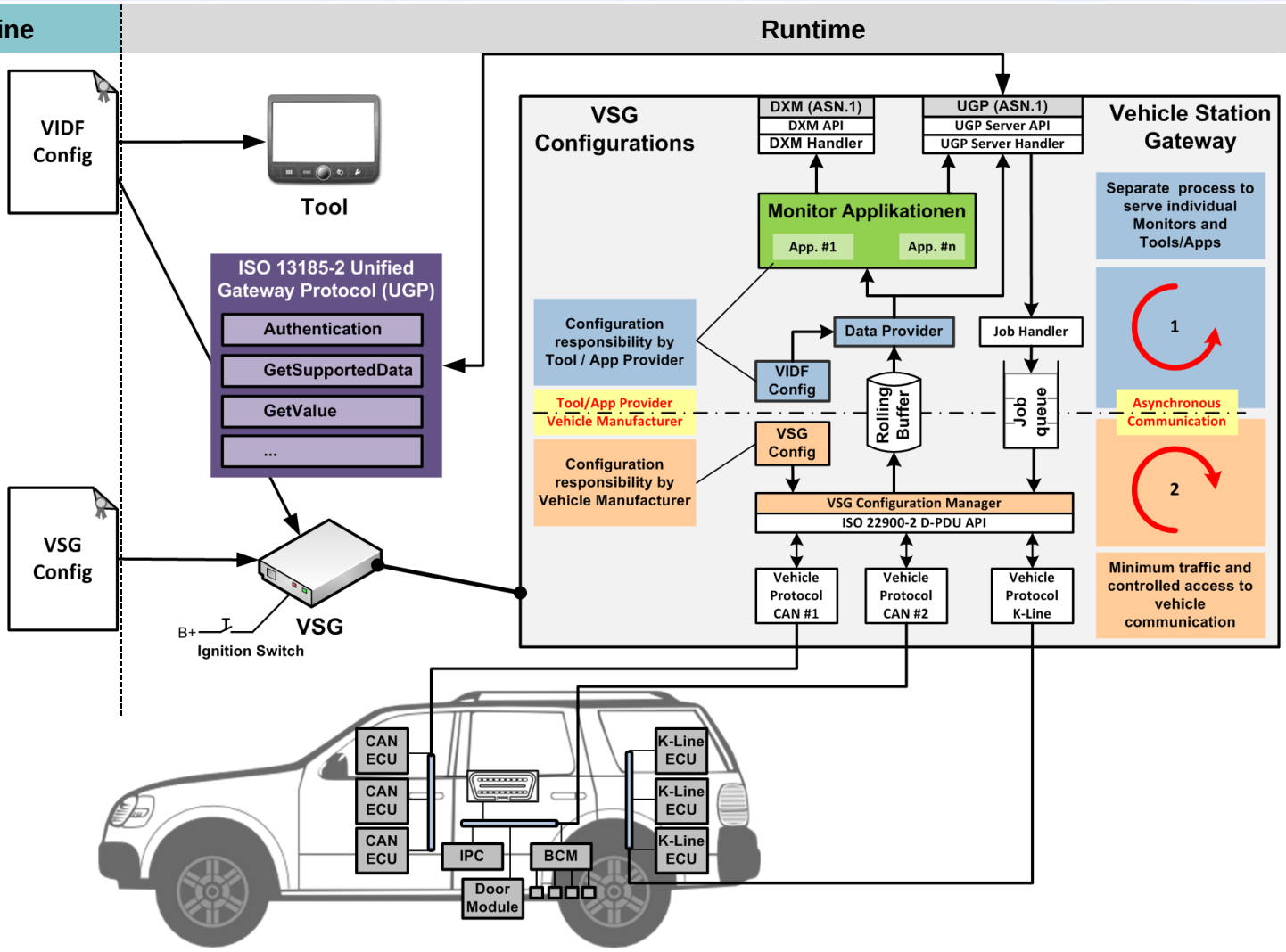
Configuration and App Development Process



App and VSG Installation Process



VSG and UGP overview



VIDF and VSG configuration

VIDF (Vehicle Interface Data Format) Configuration

Ecu	
ecuId	*longname*
0	Vehicle Station Gateway
32	ABS Control Module
51	Emissions OBD system

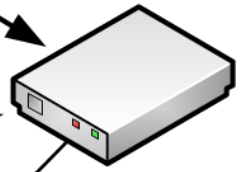
Data Parameter	
rvId	*longname*
1002	Engine Control module voltage
7368	Second.Air Syst.Mon.Ready
2341	Engine Coolant Temperature
1123	Hardware Part Number
461	Vehicle Identification Number
10042	Engine Speed
492	Front Left Wheel Speed
499	Ignition Switch Status
10050	Continuous Remote OBD I/M

Data Types		
dtId	*type*	*type values*
331	numeric	value = v / 100; 8.8 – 15.6 V
332	enum	0 "no", 1 "yes"
333	numeric	value = v / 10; - 120.0 °C
334	string	
360	string	"A..HJ..NPR..Z0..9"; len=17
338	numeric	value = v/4; 0-16768 rpm
7	numeric	value = v/100; m/s
374	enum	0 "off", 1 "on"
337	structure	1002, 7368, 2341, 10042

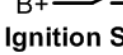
Tool

VSG Configuration

Scheduler & Vehicle Protocol Mapping		
rvId	*schedule*	*service mapping definition*
1002	1000	UDS REQ=0x22, DID=..., RESP...
7368	1000	...
2341	500	
1123	oneshot	
461	oneshot	
10042	500	
499	500	
10050	1209600000	



VSG



Ignition Switch

VIDF Configuration detail 1


```
vidfConfigExample VIDFConfig ::= { configName "VIDFConfig example",
    unitType {
        { unitTypeId 7, name { textId 9007, longname "temperature" } },
        { unitTypeId 10, name { textId 9010, longname "electrical potential" } },
        { unitTypeId 21, name { textId 9021, longname "frequency" } },
        { unitTypeId 28, name { textId 9028, longname "velocity / speed" } }
    }, unit {
        { unitTypeId 7, unitId 11, name { textId 9111, shortname "°C",
            longname "degree celsius" }, formula 0, c0 1, c1 0, c2 0 },
        { unitTypeId 10, unitId 17, name { textId 9117, shortname "V",
            longname "volt" }, formula 0, c0 1, c1 0, c2 0 },
        { unitTypeId 21, unitId 23, name { textId 9123, shortname "rpm",
            longname "revolutions per minute" }, formula 0, c0 1, c1 0, c2 0 },
        { unitTypeId 28, unitId 24, name { textId 9124, shortname "kph",
            longname "kilometers per hour" }, formula 9, c0 10, c1 36, c2 0 },
        { unitTypeId 28, unitId 60, name { textId 9160, shortname "m/s",
            longname "meter per second" }, formula 0, c0 1, c1 0, c2 0 }
    }, provider {
        { providerId 0, name { textId 9200, longname "UNKNOWN" } },
        { providerId 1, name { textId 9201, shortname "ANY", longname "Any supplier" } }
    }, ecu {
        { ecuId 0, name { textId 13000, shortname "VSG", longname "Vehicle Station Gateway" }, providerId 1 },
        { ecuId 32, name { textId 13007, shortname "ABS", longname "Anti-Lock Brake System" }, providerId 0 },
        { ecuId 51, name { textId 13003, shortname "OBDII", longname "Emissions OBD II System" }, providerId 0 }
    },
```

formulaId	formula
0	y = c0 * x + c1
1	y = c0 * (x + c1)
2	y = c0 / (x + c1) + c2
3	y = x / c0 + c1
5	y = (x + c0) / c1 + c2
9	y = x * c0 / c1
10	y = x + c0 / c1
11	y = (x + c0) * c1 / c2

see ISO 14229-1 UDS
Table C.6 formulaIdentifier encoding

VIDF Configuration detail 2


```
dataType {
    { dataTypeId 331, name { textId 10180, longname "Voltage in 1/100 V" }, type numeric: {
      decimalPlaces 2, unitId 17, factor 1, quotient 100, addend 0, min 880, max 1560 } },
    { dataTypeId 332, name { textId 10005, longname "Answer (no, yes)" }, type enumString: {
      { value 0, name { textId 10006, longname "no" } },
      { value 1, name { textId 10007, longname "yes" } } } },
    { dataTypeId 333, name { textId 10181, shortname "", longname "Temperature in 1/10 °C" }, type numeric: {
      decimalPlaces 1, unitId 11, factor 1, quotient 10, addend 0 } },
    { dataTypeId 334, name { textId 10000, longname "Unlimited string" }, type string: {} },
    { dataTypeId 360, name { textId 10183, longname "VIN" }, type string: {
      allowedCharacters "A..HJ..NPR..Z0..9", minLen 17, maxLen 17 } },
    { dataTypeId 338, name { textId 10182, longname "RPM" }, type lnumeric: {
      unitId 23, factor 1, quotient 4, addend 0, min 0, max 65535 } },
    { dataTypeId 7, name { textId 10030, longname "Speed in cm/s" }, type numeric: {
      decimalPlaces 2, unitId 60, factor 1, quotient 100, addend 0, min -32765, max 32765 } },
    { dataTypeId 374, name { textId 10008, longname "Switch (off, on)" }, type enumString: {
      { value 0, name { textId 10009, longname "off" } },
      { value 1, name { textId 10010, longname "on" } } } },
    { dataTypeId 337, name { textId 10230, longname "Continuous Remote OBD I/M {}" }, type structure: {
      { 1002, 7368, 2341, 10042 }, convention mandatory } }
},
```

Definition of formula for type numeric, lnumeric

```
realValue = value * factor / quotient + addend
```

DataTypeId 331

```
realValue = value * 1 / 100 + 0 = value / 100
=> min = 8.8 ; max = 15.6
```

VIDF Configuration detail 3


```
dataParam {
    { rvId 1002, name { textId 11152, shortname "ECMB+", longname "Engine Control Module Voltage" },
      dataTypeId 331, accessType '10000'B, dataParamProperty sensor },
    { rvId 7368, name { textId 10134, shortname "AIR_RDY", longname "Secondary Air System Monitoring Ready" },
      dataTypeId 332, accessType '10000'B, dataParamProperty ecu-internal-monitor },
    { rvId 2341, name { textId 11157, shortname "ECT", longname "Engine Coolant Temperature" },
      dataTypeId 333, accessType '10000'B, dataParamProperty sensor },
    { rvId 1123, name { textId 10131, shortname "HWPNO", longname "Hardware Part Number" },
      dataTypeId 334, accessType '10000'B, dataParamProperty ecu-internal-signal },
    { rvId 461, name { textId 11104, shortname "VIN", longname "Vehicle Identification Number" },
      dataTypeId 360, accessType '10000'B, dataParamProperty ecu-internal-signal },
    { rvId 10042, name { textId 11158, shortname "RPM", longname "Engine RPM" },
      dataTypeId 338, accessType '10000'B, dataParamProperty sensor },
    { rvId 492, name { textId 11112, shortname "FLWS", longname "Front Left Wheel Speed" },
      dataTypeId 360, accessType '10000'B, dataParamProperty ecu-internal-signal },
    { rvId 499, name { textId 11113, shortname "IGSWST", longname "Ignition Switch Status" },
      dataTypeId 374, accessType '10000'B, dataParamProperty sensor },
    { rvId 10050, name { textId 11170, shortname "CROBDIM", longname "Continuous Remote OBD I/M" },
      dataTypeId 337, accessType '10000'B, dataParamProperty collection },
},
```

accessTypes

```
r (0) = read,
w (1) = write,
x (2) = execute,
i (3) = internal
u (4) = user
```

```
'10000'B
= read only
```

dataParamProperty

```
ecu-supported-info,
sensor,
actuator,
ecu-internal-signal,
ecu-internal-monitor,
collection,
routine,
fix,
other
```

VIDF Configuration detail 4


```
dataParamMapping {
    { rvId 1002, ecuId 51 },
    { rvId 7368, ecuId 51 },
    { rvId 2341, ecuId 51 },
    { rvId 1123, ecuId 51 },
    { rvId 461, ecuId 51 },
    { rvId 10042, ecuId 51 },
    { rvId 492, ecuId 32 },
    { rvId 499, ecuId 0 },
    { rvId 10050, ecuId 0 }
}, fixedValue {
    { rvId 20001, ecuId 0, value string: "passcar_manufacturer_model_year" },
    { rvId 20002, ecuId 0, value enumString: 4 },
    { rvId 20003, ecuId 0, value enumString: 8 },
    { rvId 20004, ecuId 0, value displayName: "Manufacturer" },
    { rvId 20005, ecuId 0, value displayName: "Model" }
},
```

Vehicle Key

Vehicle Type = passcar

Vehicle Class = passenger vehicle

Vehicle Brand

Vehicle Model

VIDF Configuration detail 5

From SAE J2012-DA

```
dtcBase {
    { rDtcBaseId 1, ecuId 51, name { textId 20001, longname "Fuel Vol.Regul.Control Circuit/Open" } },
    { rDtcBaseId 2, ecuId 51, name { textId 20002, longname "Fuel Vol.Regul.Control Circuit Range/Performance" } },
    { rDtcBaseId 3, ecuId 51, name { textId 20003, longname "Fuel Vol.Regul.Control Circuit Low" } },
    { rDtcBaseId 4, ecuId 51, name { textId 20004, longname "Fuel Vol.Regul.Control Circuit High" } },
    { rDtcBaseId 5, ecuId 51, name { textId 20005, longname "Fuel Shutoff Valve 'A' Control Circuit/Open" },
        dataParamMapping {
            { rvId 1002, ecuId 17 },
            { rvId 2341, ecuId 17 }
    } },
    ...
```

Environment data

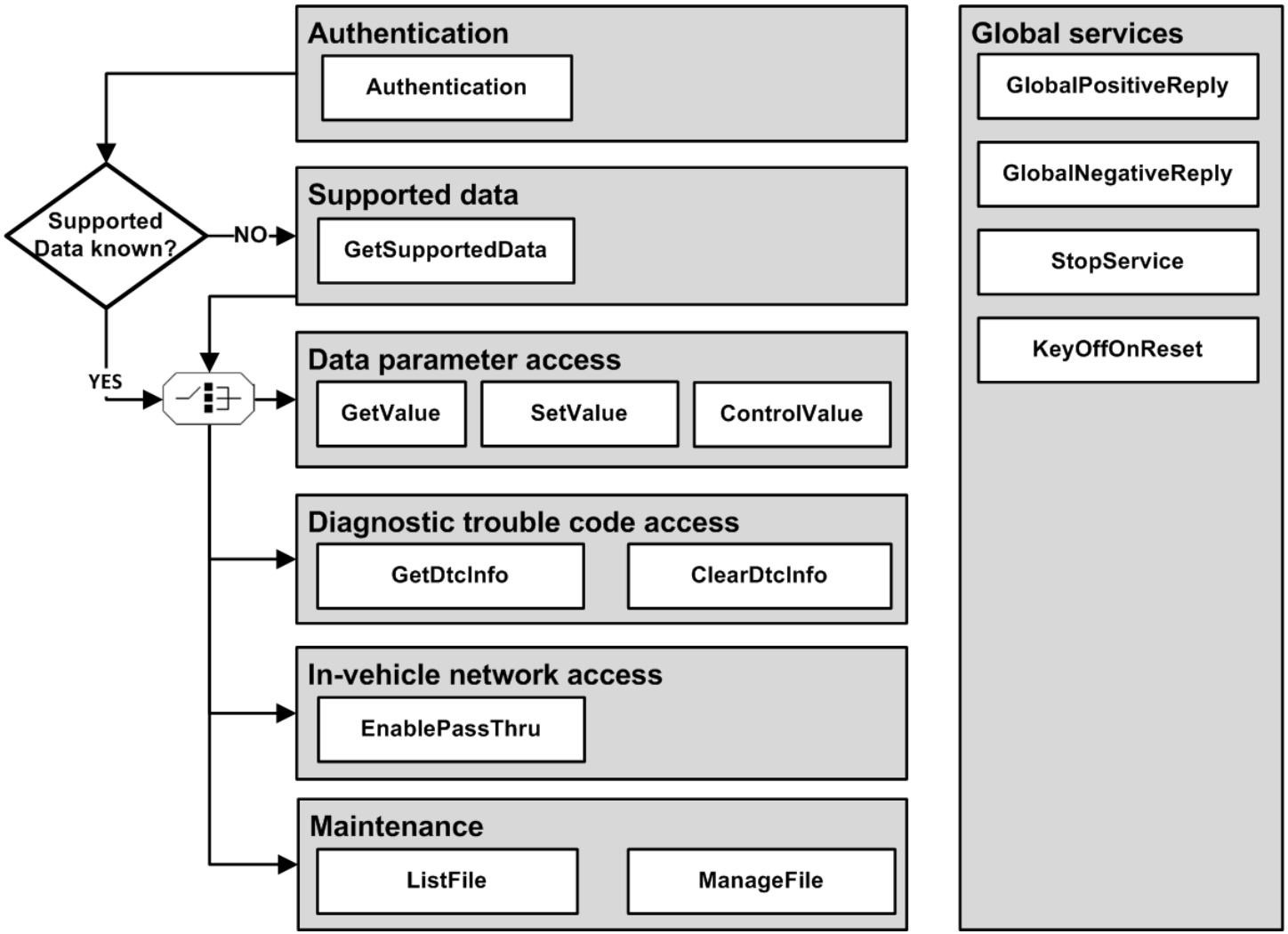
Examples

```
    { rDtcBaseId 295, ecuId 51, name { textId 20006, longname "Intake Air Temperature Too High" } },
    ...
    { rDtcBaseId 16433, ecuId 51, name { textId 20007, longname "Left Front Wheel Speed Sensor" } },
    { rDtcBaseId 32825, ecuId 51, name { textId 20008, longname "Second Row Right Front.Stage 1 Deploym.Control" } },
    { rDtcBaseId 49280, ecuId 51, name { textId 20009, longname "Vehicle Communication Bus 'F'" } },
    ...
},
```

From SAE J2012-DA

```
dtcSymptom {
    { rDtcSymptomId 0, name { textId 21000, longname "No Sub Type Information" } },
    { rDtcSymptomId 1, name { textId 21001, longname "General Electrical Failure" } } },
    { rDtcSymptomId 2, name { textId 21002, longname "General Signal Failure" } },
    { rDtcSymptomId 3, name { textId 21003, longname "FM (Frequency Modulated) / PWM (Pulse Width Module)" } },
    { rDtcSymptomId 4, name { textId 21004, longname "System Internal Failure" } },
    { rDtcSymptomId 5, name { textId 21005, longname "System Programming Failure" } },
    { rDtcSymptomId 6, name { textId 21006, longname "Algorithm Based Failure" } },
    { rDtcSymptomId 7, name { textId 21007, longname "Mechanical Failure" } },
    ...
    { rDtcSymptomId 255, name { textId 21255, longname "Manufacturer Defined" } }
}
}
```

UGP services



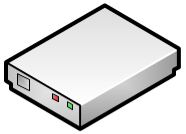
Tool talks to VSG via UGP

Tool

Use

ISO 13185-2 UGP (ASN.1)

to obtain vehicle specific data from VSG.

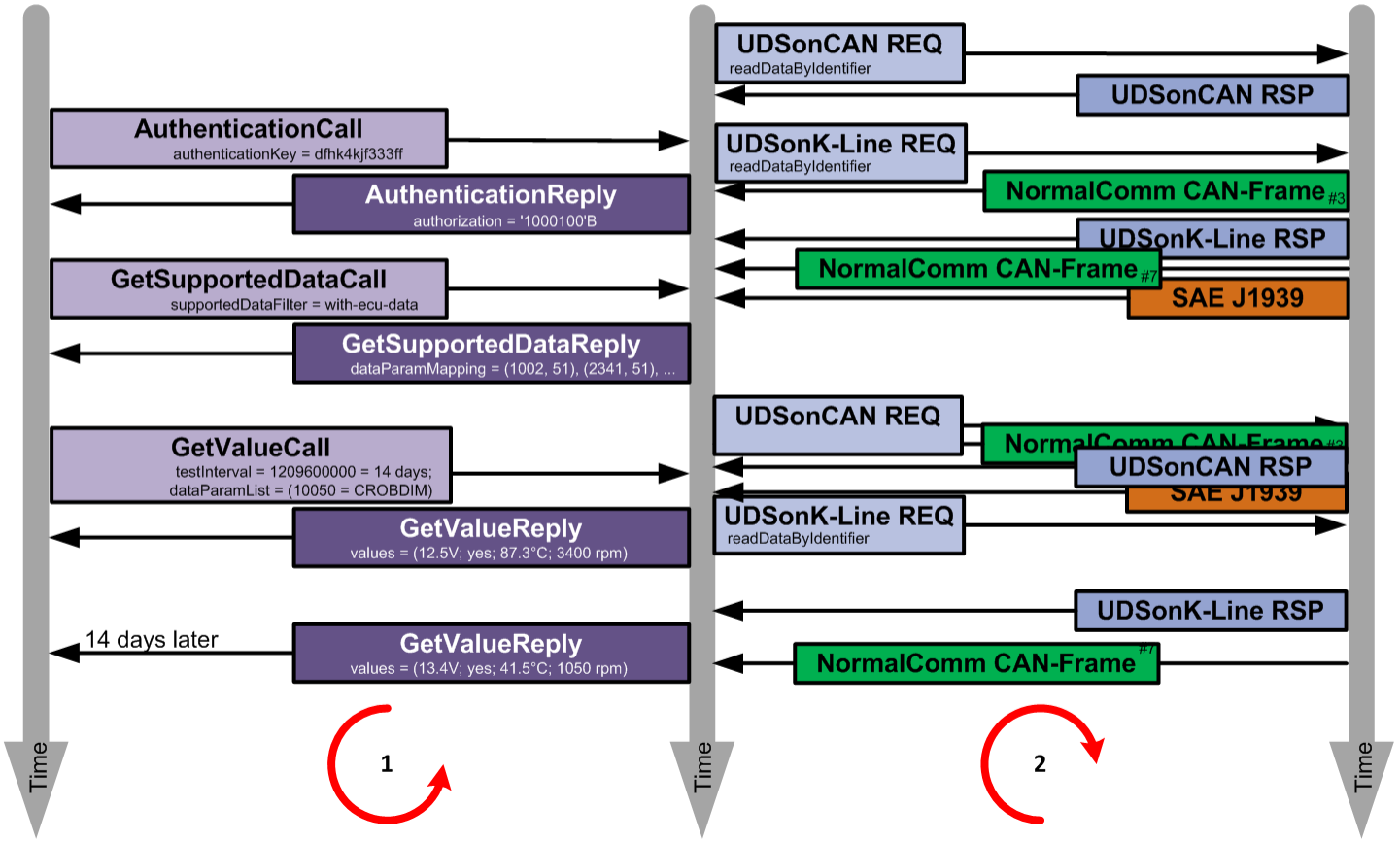


VSG

Interprete VSG configuration and schedule services using

ISO 22900-1,-2 MVCI

Vehicle



UGP Authentication example


```
authenticationCall UGPMessage ::= { callSequenceNumber 1,
    choiceUGP authenticationCall: {
        authenticationKey "0vrmYdc6ziscBdMhGphiT6m0a0D2dNLCnETRS1g"
} }
```

```
authenticationPosReply UGPMessage ::= { callSequenceNumber 1,
    choiceUGP authenticationReply: {
        authorization '1000100'B
} }
```

```
AuthorizationBits ::= BIT STRING {
   ✓ get-value-extended-access   (0),
     set-value-access            (1),
     control-value-access        (2),
     enable-pass-thru-access     (3),
   ✓ file-download-access        (4),
     file-upload-access          (5),
     file-delete-access          (6)
} (SIZE(7, ...))
```

UGP GetSupportedData example


```
getSupportedDataCallWithEcuData UGPMessage ::= { callSequenceNumber 113,
    choiceUGP getSupportedDataCall: { supportedDataFilter with-ecu-data
} }
```

```
SupportedDataFilter ::= ENUMERATED { vehicle-info-only, with-ecu-data,
   without-ecu-data, ...
}
```

```
getSupportedDataReplyWithEcuData UGPMessage ::= { callSequenceNumber 113,
    choiceUGP getSupportedDataReply: { dataParamMapping {
        { rvId 1002, ecuId 17 }, { rvId 2341, ecuId 51 },
        { rvId 10020, ecuId 21 }, { rvId 10021, ecuId 21 },
        { rvId 10022, ecuId 21 }, { rvId 10023, ecuId 21 },
        { rvId 10024, ecuId 21 }, { rvId 10042, ecuId 51 },
        { rvId 10050, ecuId 51 }
} } }
```

UGP GetValueCall definition

Msg	GetValueCall		Request data parameter values	
Attributes	**Name**		**Description**	**Cvt**
	test-Interval	condition	specifies the interpretation of the combined attributes `testInterval` and `condition`	O
	= 0	not set	reply message is sent only once	
		set	test the condition and if test condition is true then send one reply message	
	> 0	not set	reply message is sent each time the testInterval [ms] has expired	
		set	test the condition each time the testInterval [ms] has expired and if test condition is true then send a reply message	
	`dataParamList {}`		List of ITS-registered value identifiers to retrieve	O1[a]
	`dataParamMapping {}`		List of mappings between data parameter and ECU	O2[a]
	`condition`		A `ComplexCondition` element (see A.20). If the condition is set, the data parameter values should be responded only if the condition is true.	O
[a]:	Either O1 or O2 must be defined			

ASN.1

```
GetValueCall ::= SEQUENCE {
    testInterval        SNUM32,
    dataParamList       SEQUENCE OF Identifier                  OPTIONAL,
    dataParamMapping    SEQUENCE OF DataParamMapping            OPTIONAL,
    condition           ComplexCondition                        OPTIONAL,
    ...
}
```

UGP GetValueCall example 1


```
getValueCallInList UGPMessage ::= { callSequenceNumber 192,
    choiceUGP getValueCall: { dataParamList { 1002, 7368, 2341, 10042 }
} }
```

```
getValueReplyInList UGPMessage ::= { callSequenceNumber 192,
    choiceUGP getValueReply: { valueTS {
        { value numeric: 1250 }, { value enumString: 1 }, { value numeric: 873 },
        { value numeric: 6800 }
} } }
```

Engine Control Module Voltage = 12.5 V

Secondary Air System Monitor Ready = yes

Engine Coolant Temperature = 87.3 °C

Engine Speed = 3400 rpm

Example: Continuous Remote OBD I/M

```
getValueCallCROBDIM UGPMessage ::= { callSequenceNumber 196,
    choiceUGP getValueCall: { testInterval 1209600000, dataParamList { 10050 }
} }
```

UGP GetValueCall example 2


```
getValueCallCond UGPMessage ::= { callSequenceNumber 193,
    choiceUGP getValueCall: { testInterval 5000, dataParamMapping {
        { rvId 1002, ecuId 17 },
        { rvId 7368, ecuId 17 },
        { rvId 2341, ecuId 51 }
    }, condition {
        paramCond: { rvId 2341, operator gt, value numeric: 1100 }
} }
```

```
getValueReplyCond1 UGPMessage ::= { callSequenceNumber 193,
    choiceUGP getValueReply: { valueTS {
} } }
```

Condition false

```
getValueReplyCondN UGPMessage ::= { callSequenceNumber 193,
    choiceUGP getValueReply: { valueTS {
        { value numeric: 1250 }, { value enumString: 1 }, { value numeric: 1101 }
} } }
```

Condition true

UGP GetDtcInfo example


```
getDtcInfoCall1 UGPMessage ::= { callSequenceNumber 257,
    choiceUGP getDtcInfoCall: { testInterval 10000
} }
```

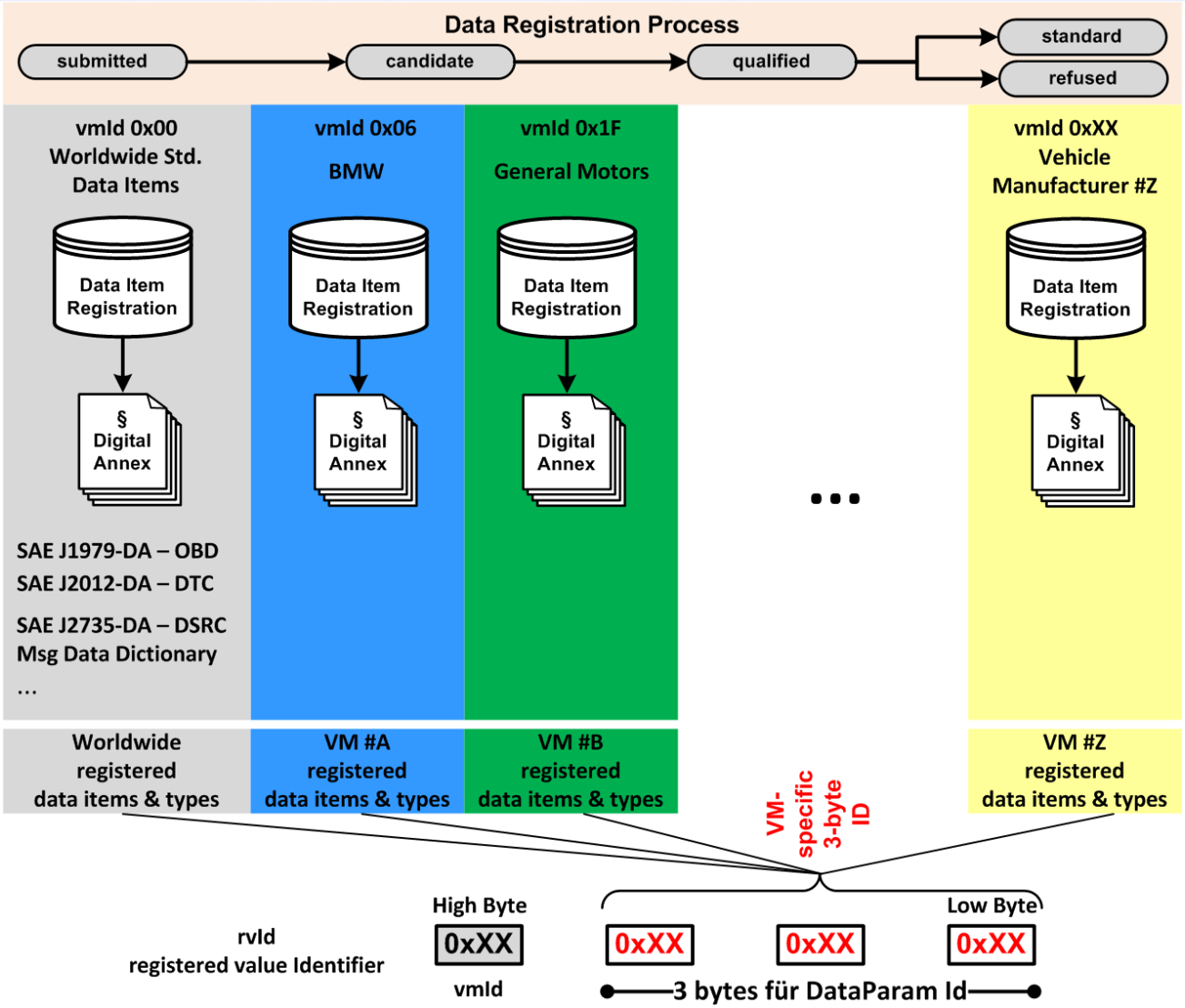
```
getDtcInfoReply1 UGPMessage ::= { callSequenceNumber 257,
    choiceUGP getDtcInfoReply: { dtcInfo {
        { rDtcBaseId 5, rDtcSymptomId 1, ecuId 21,
          complementary '0010000000000000'B },
        { rDtcBaseId 256, rDtcSymptomId 1, ecuId 17,
          complementary '0000000100000000'B },
        { rDtcBaseId 157, rDtcSymptomId 1, ecuId 17,
          complementary '0010000000100000'B }
} } }
```

```
DtcComplementary ::= BIT STRING {
   testFailed                          (0),
   testFailedThisOperationCycle        (1),
✓  pendingDTC                          (2),
   confirmedDTC                        (3),
   testNotCompletedSinceLastClear      (4),
   testFailedSinceLastClear            (5),
   testNotCompletedThisOperationCycle  (6),
   warningIndicatorRequested           (7),
   permanent                           (8),
   maintenanceOnly                     (9),
   checkAtNextHalt                    (10),
   checkImmediately                   (11),
   dTCClass0                          (12),
   dTCClass1                          (13),
   dTCClass2                          (14),
   dTCClass3                          (15),
   dTCClass4                          (16)
} (SIZE(17, ...))
```

```
getDtcInfoCall2 UGPMessage ::= { callSequenceNumber 258,
    choiceUGP getDtcInfoCall: { ecuList { 21 }, withEnvData TRUE
} }
```

```
getDtcInfoReply2 UGPMessage ::= { callSequenceNumber 258,
    choiceUGP getDtcInfoReply: { dtcInfo {
        { rDtcBaseId 5, rDtcSymptomId 1, ecuId 21,
          complementary '0010000000000000'B, envData {
            { value numeric: 1250 }, { value numeric: 910 } } }
} } }
```

Registered Value Identifiers (rvIds)



vmId (vehicle manufacturer Id) and WMI (World Manufacturer Id) mapping

High Byte of rvId

0xXX

vmId	Vehicle manufacturer
00	World-Wide-Standards
01	Alexander Dennis
02	Alfa Romeo
03	Aprilia
04	Aston
05	Audi
06	BMW
07	Brilliance Zhonghua
08	Buick
09	Cadillac
0A	Cagiva SpA
0B	CAMI
0C	Carrocerias Ayats
0D	Chery
0E	Chevrolet
0F	Chrysler
10	Citroën
11	Daewoo General Motors
12	DaimlerChrysler
13	DeLorean Motor Cars N.
14	Dodge
15	Dong Feng (DFM)
16	Ducati Motor Holdings SpA
⋮	⋮

WMI	Manufacturer
⋮	⋮
WAU	Audi
93U	Audi Brazil
TRU	Audi Hungary
WBA	BMW
WBS	BMW M
4US	BMW USA
LSY	Brilliance Zhonghua
2G4	Buick Canada
1G4	Buick USA
1G6	Cadillac USA
ZCG	Cagiva SpA
2CN	CAMI
VS9	Carrocerias Ayats (Spain)
LVV	Chery, China
8AG	Chevrolet Argentina
9BG	Chevrolet Brazil
2G1	Chevrolet Canada
8GG	Chevrolet Chile
MMM	Chevrolet Thailand
1GC	Chevrolet Truck USA
1G1	Chevrolet USA
1C3	Chrysler
1C6	Chrysler
⋮	⋮

rvId (registered value Identifier) Examples

vmId | standard | extension | standardized

Type	rvId			Short Name	Long Name	Comment
PID	00	00	F4 05	ECT	Engine Coolant Temperature	SAE J1979DA
Monitor	00	00	F6 01	...	Exhaust Gas Sensor Monitor Bank 1 – Sensor 1	SAE J1979DA
InfoTypeId	00	00	F8 02	VIN	Vehicle Identification Number	SAE J1979DA
...						

vmId | VMx

Type	rvId			Short Name	Long Name	Comment
DID	1F	00	01 EC	FLWSPD	Front Left Wheel Speed	VM-xc DA
LID	1F	00	23 9A	...	...	VM-xc DA
...						

rdtcBaseId (registered dtc Base Identifier) Examples

vmId | standard | extension | standardized

Type	rdtcBaseId	Short Name	Long Name	Comment
rdtcBaseId	00 00 00 13	P0013	"B" Camshaft Position Actuator Control Circuit/Open Bank 1	SAE J2012DA
rdtcBaseId	00 00 80 38	B0038	Second Row Side Airbag Deployment Control (Subfault)	SAE J2012DA
rdtcBaseId	00 00 04 B8	1208	Pre-Filter Oil Pressure	ISO 27145-2(J1939)
...				

VMx

Type	rdtcBaseId	Short Name	Long Name	Comment
rdtcBaseId	1F 00 01 01	...	Manufacturer Controlled DTC Name - General Electrical Failure	VMx DA
...				

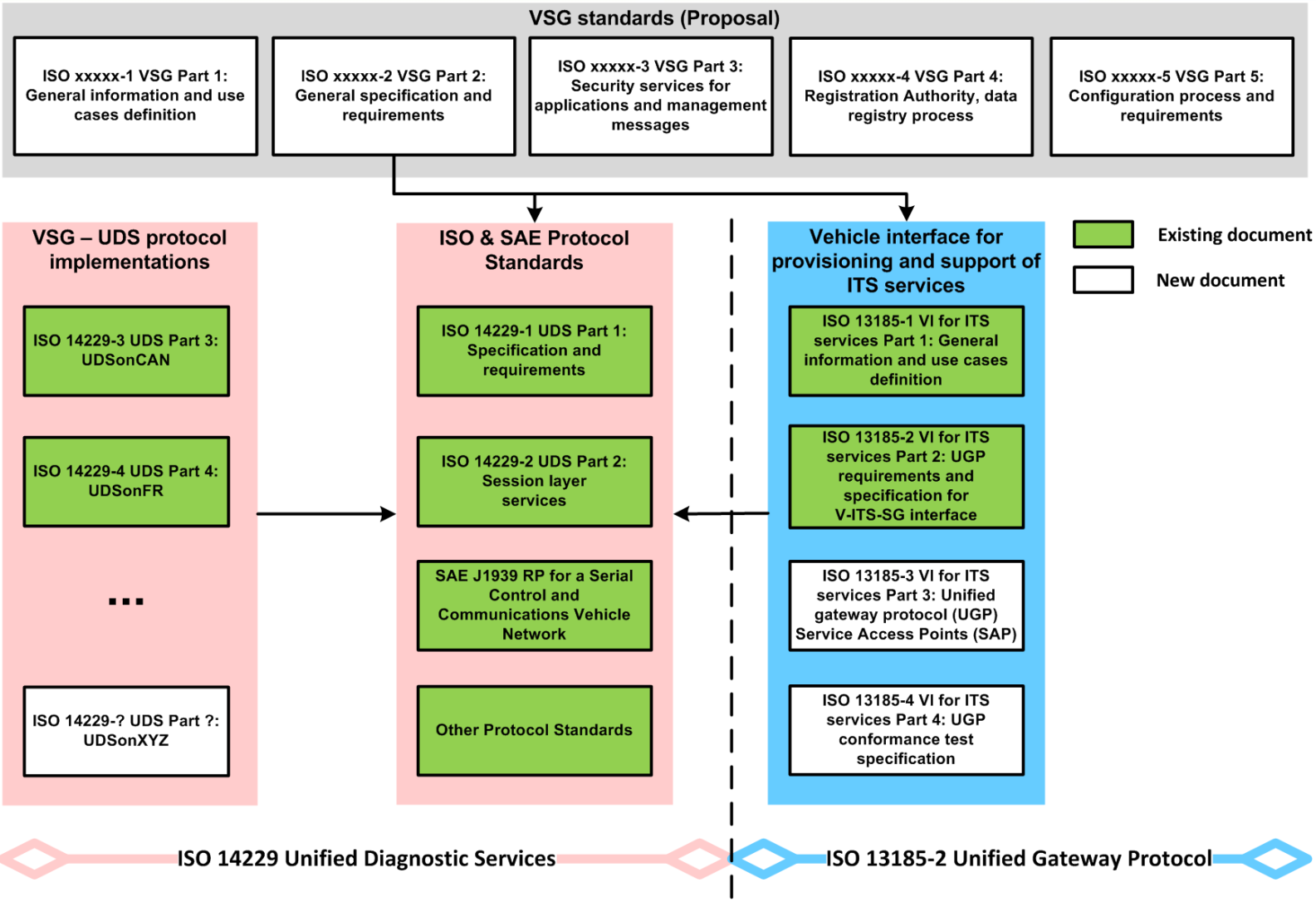
rdtcSymptomId (registered dtc Symptom Identifier) Examples

vmId | standard | extension | Std. identifier | standardized

Type	rdtcSymptomId				Short Name	Long Name	Comment
rdtcSymptomId	00	00	00	01	...	General Electrical Failure	SAE J2012DA
rdtcSymptomId	00	00	00	02	...	General Signal Failure	SAE J2012DA
rdtcSymptomId	00	00	00	11	...	Circuit Short To Ground	SAE J2012DA
rdtcSymptomId	00	00	00	21	...	Signal Amplitude < Minimum	SAE J2012DA
rdtcSymptomId	00	01	00	00	...	no fault symptom available for this DTC	ISO 14230-3
rdtcSymptomId	00	01	00	01	...	above maximum threshold	ISO 14230-3
rdtcSymptomId	00	01	00	02	...	below minimum threshold	ISO 14230-3
rdtcSymptomId	00	01	00	04	...	no signal	ISO 14230-3
rdtcSymptomId	00	01	00	08	...	invalid signal	ISO 14230-3
rdtcSymptomId	1F	01	00	xx	...	...	VMx DA/ISO 14230-3
rdtcSymptomId	00	02	00	01	...	Data Valid But Below Normal Operational Range	SAE J1939 FMI
rdtcSymptomId	00	02	00	02	...	Data Erratic, Intermittent or Incorrect	SAE J1939 FMI
rdtcSymptomId	00	02	00	03	...	Voltage above normal or shorted to High Source	SAE J1939 FMI
rdtcSymptomId	00	02	00	xx	...	...	SAE J1939 FMI
...							

VMx

VSG, UGP and UDS document structure (Proposal)



Summary – UGP and VSG

- Motivation factors for the VSG Concept
- Why VSG and UGP? (asynchronous comm., Firewall, ...)
- The Vehicle Interface Data Format (VIDF) Configuration Concept
- The VSG Configuration Concept (Mapping between in-vehicle network ECUs and VIDF
- The Authoring & Configuration Process
- The App and VSG Installation Process
- The Standards & Vehicle Manufacturer Data Identifier Mapping Concept (Migration of current data & DTC)
- Proposed VSG Document Structure

Intelligent Transportation Systems

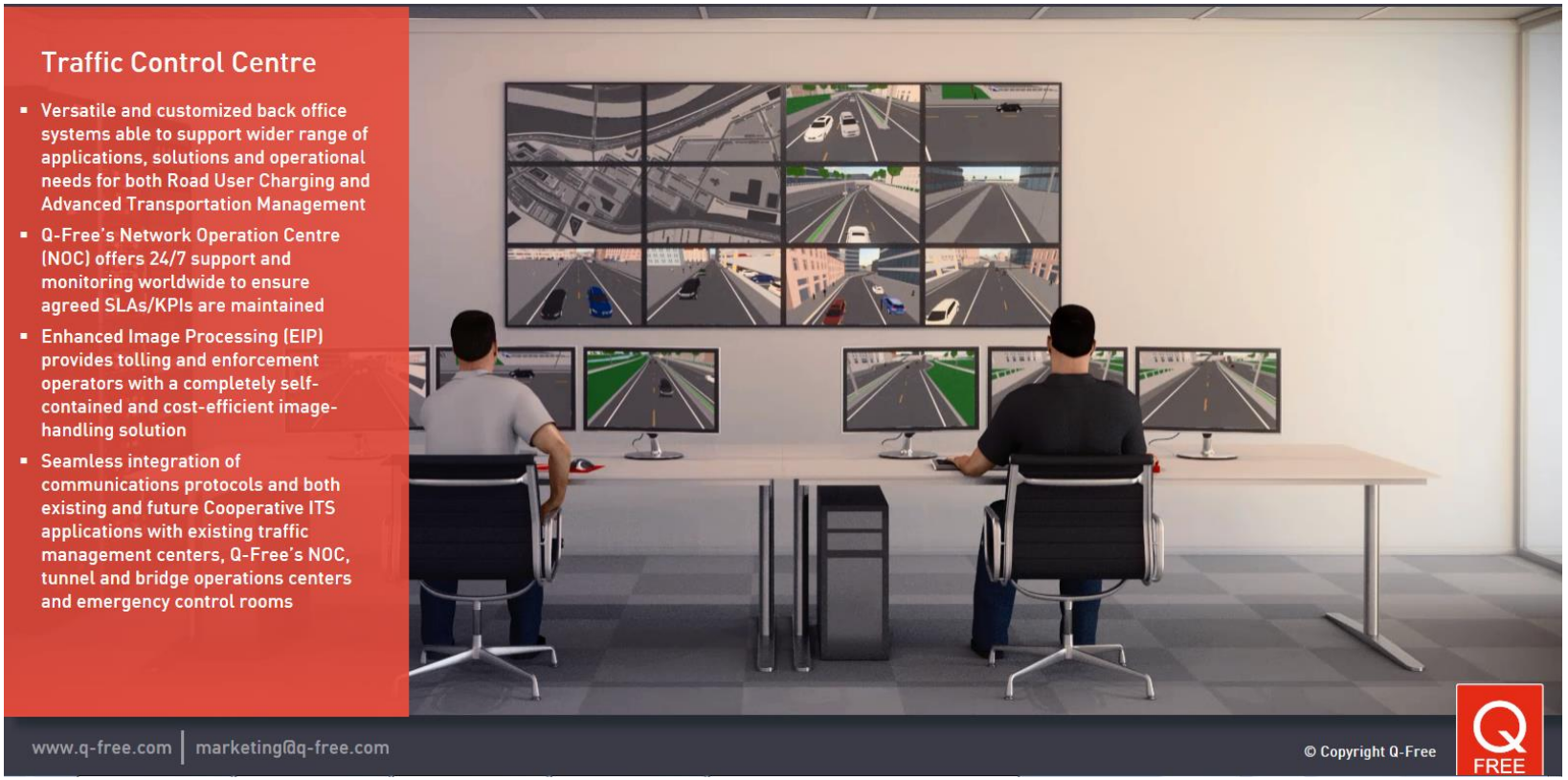
- ITS and in-vehicle networks
- Cooperative ITS (C-ITS) and trust
- ITS stations/subsystems
- Vehicular ITS stations/subsystems and the ITS-S gateway
- General IVN data and application requirements (security!)

Intelligent Transportation Systems

- ITS involves exchange of information between a variety of communication units throughout the transportation network

Intelligent Transportation Systems

- Control centers integrate incoming information and provide value-added services

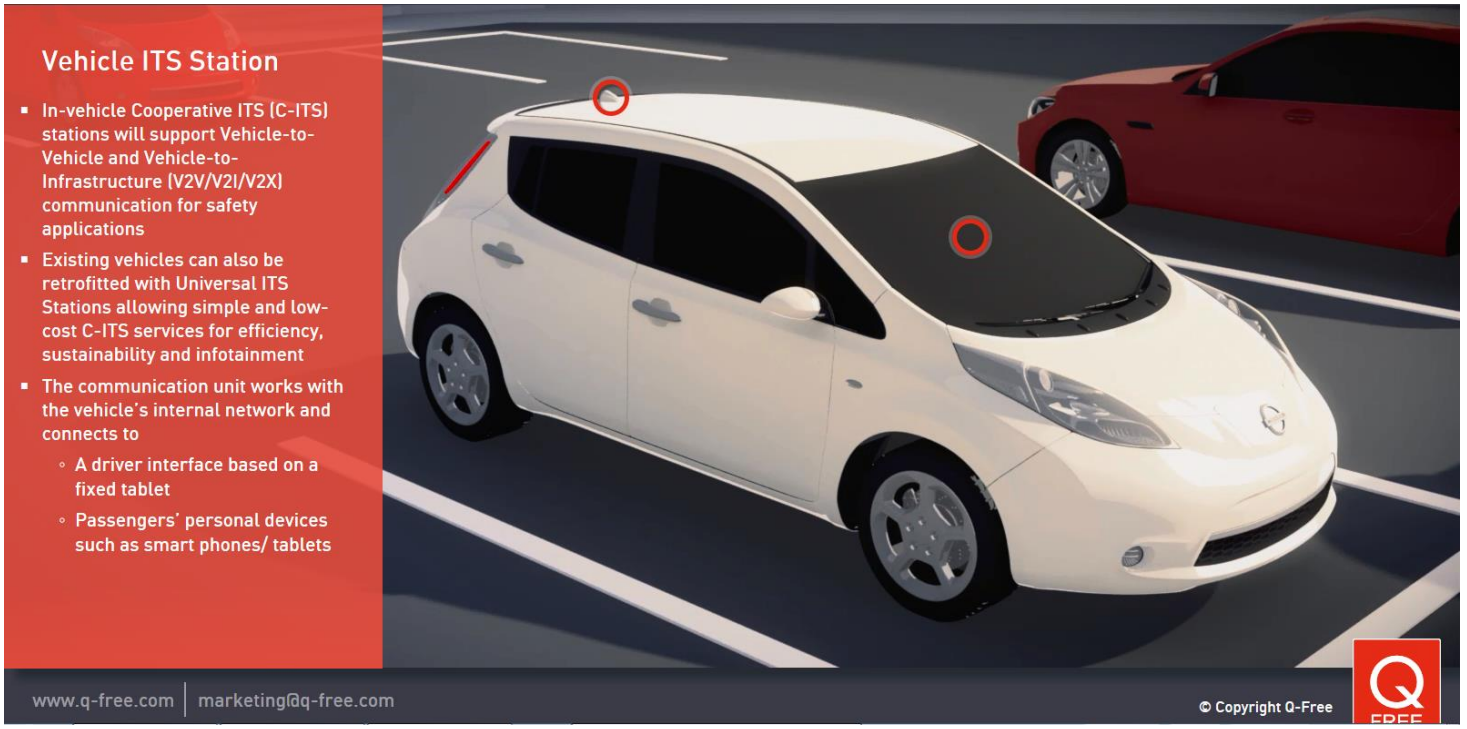


Intelligent Transportation Systems

- Instrumented intersections provide valuable information to vehicles and pedestrians for collision avoidance and other applications

Intelligent Transportation Systems

- Vehicular units provide V2V and V2I communications and allow external devices to access in-vehicle networks through a secure gateway
- In-vehicle data is valuable for real-time as well as remote diagnostic and maintenance applications … RMI/OBD is part of ITS!



EU Law - ITS Directive

- In 2010, the EU passed a law:
  - "DIRECTIVE 2010/40/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 7 July 2010 on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport" (aka the ITS Directive)

EU Law - ITS Directive

- **Priority area IV: Linking the vehicle with the transport infrastructure** .

  — 1.2. The definition of necessary measures to further progress the development and implementation of cooperative (vehicle-vehicle, vehicle-infrastructure, infrastructure-infrastructure) systems, based on:

  - the facilitation of the exchange of data or information between vehicles, infrastructures and between vehicle and infrastructure,
  - the availability of the relevant data or information to be exchanged to the respective vehicle or road infrastructure parties,
  - the use of a standardised message format for the exchange of data or information between the vehicle and the infrastructure,
  - the definition of a communication infrastructure for data or information exchange between vehicles, infrastructures and between vehicle and infrastructure,
  - the use of standardisation processes to adopt the respective architectures.

US – NHTSA Decision

- On 3 February 2014, NHTSA announced its decision to move forward with V2V communications technology for light vehicles as a result of successful V2V (and V2I) tests conducted in the US (e.g. SafetyPilot)
- “NHTSA is also considering future actions on active safety technologies that rely on on-board sensors, […] technologies [that] are eventually expected to blend with the V2V technology.”

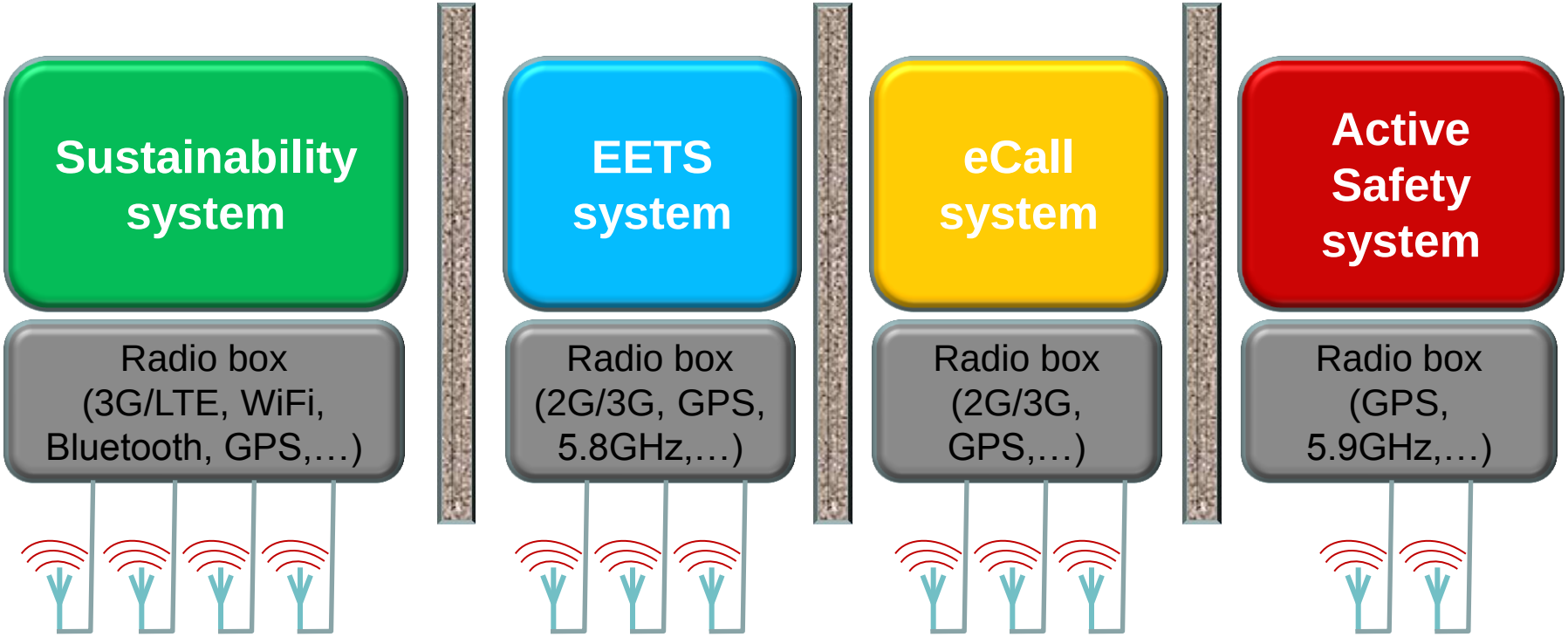
Cooperative ITS

- Cooperative-ITS is a collaboration among ITS stations, interacting with each other through communication networks to provide ITS services, where ITS-S applications in these ITS stations engage in “cooperative ITS activities” (cf. ISO 17465-1 Definition of C-ITS)
- Collaboration in a cooperative ITS activity involves the sharing of information between and among ITS-S applications in ITS stations (cf. ISO 17429 - Generic Message Handling Service n / Data publication and subscription).

Cooperative ITS

- This sharing of information leads to the deployment of new and improved ITS services for increasing safety, sustainability, efficiency and comfort of transportation systems.
- Examples of a Cooperative-ITS activities include:
  - Vehicular ITS-SUs broadcasting their status to nearby ITS-SUs for collision avoidance and eco-driving
  - Exchange of real-time status information between ITS-SUs in public transport vehicles and ITS-SUs located in traffic signal controllers, to grant priority access to vehicles at intersections.
  - Exchange of vehicle status information with a service provider to notify the owner of recommended or mandatory maintenance

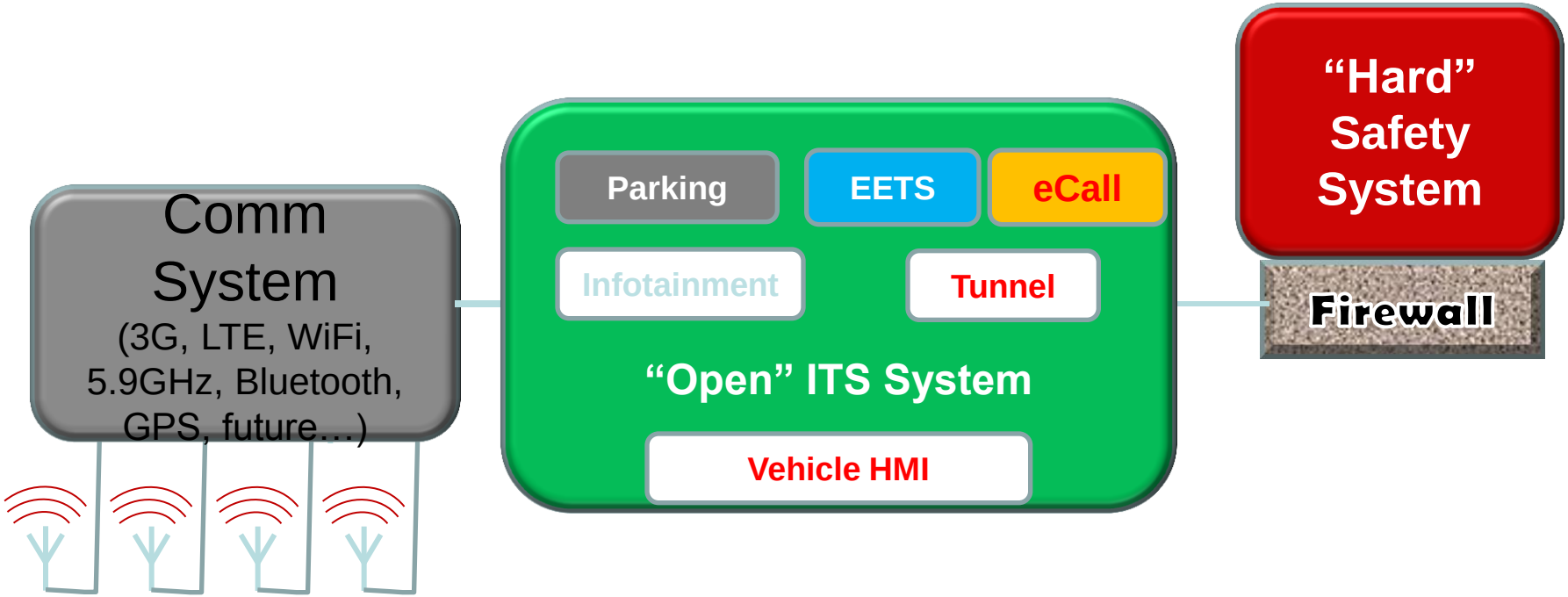
“Silo” Approach to Multiple Applications



Mandated and value-added applications implemented in "silos“ … duplication of equipment and no sharing of data and resources.

More flexible approach to multiple applicaitons

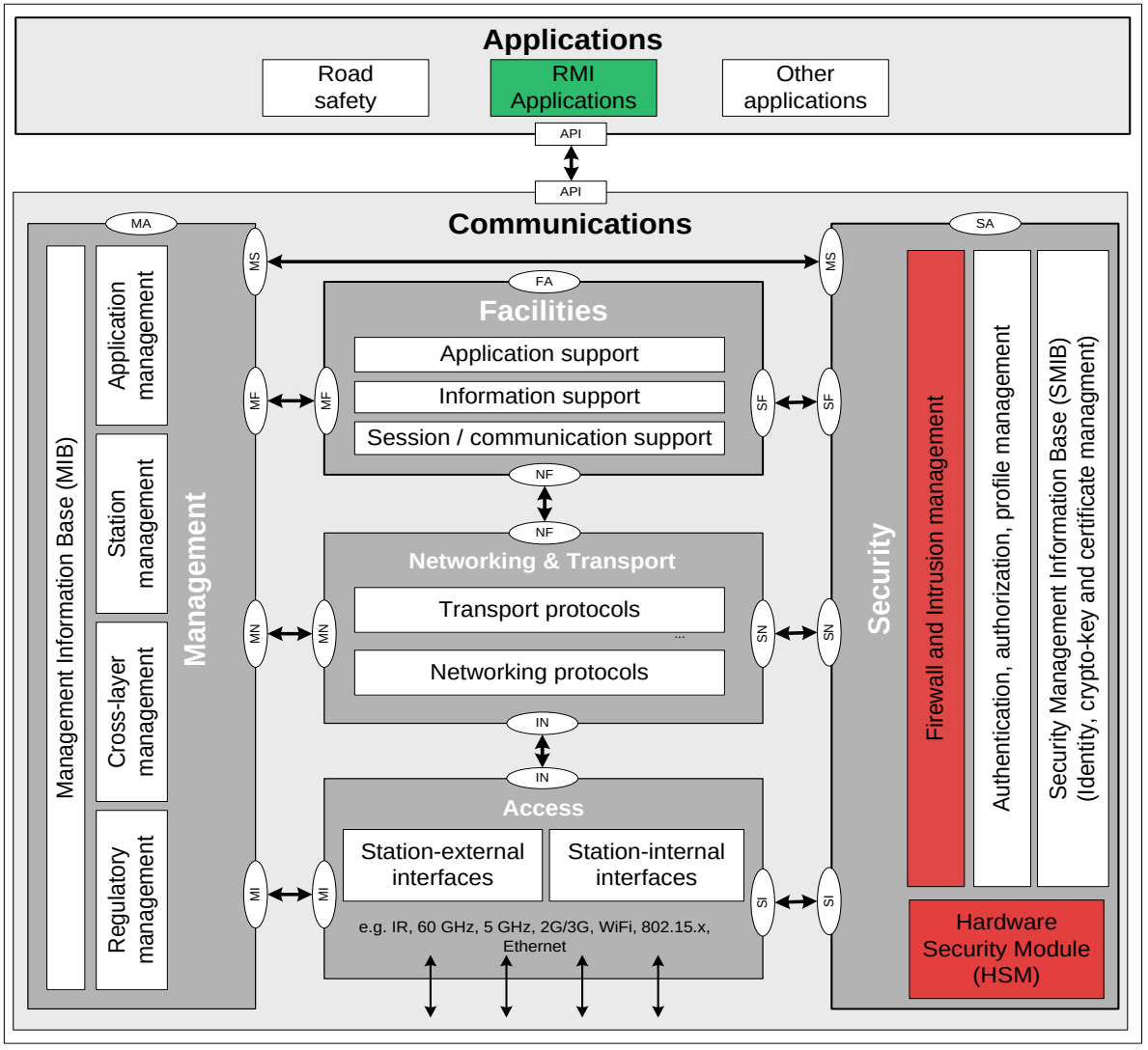
- Sharing of equipment and data!
- IVN firewall protected!



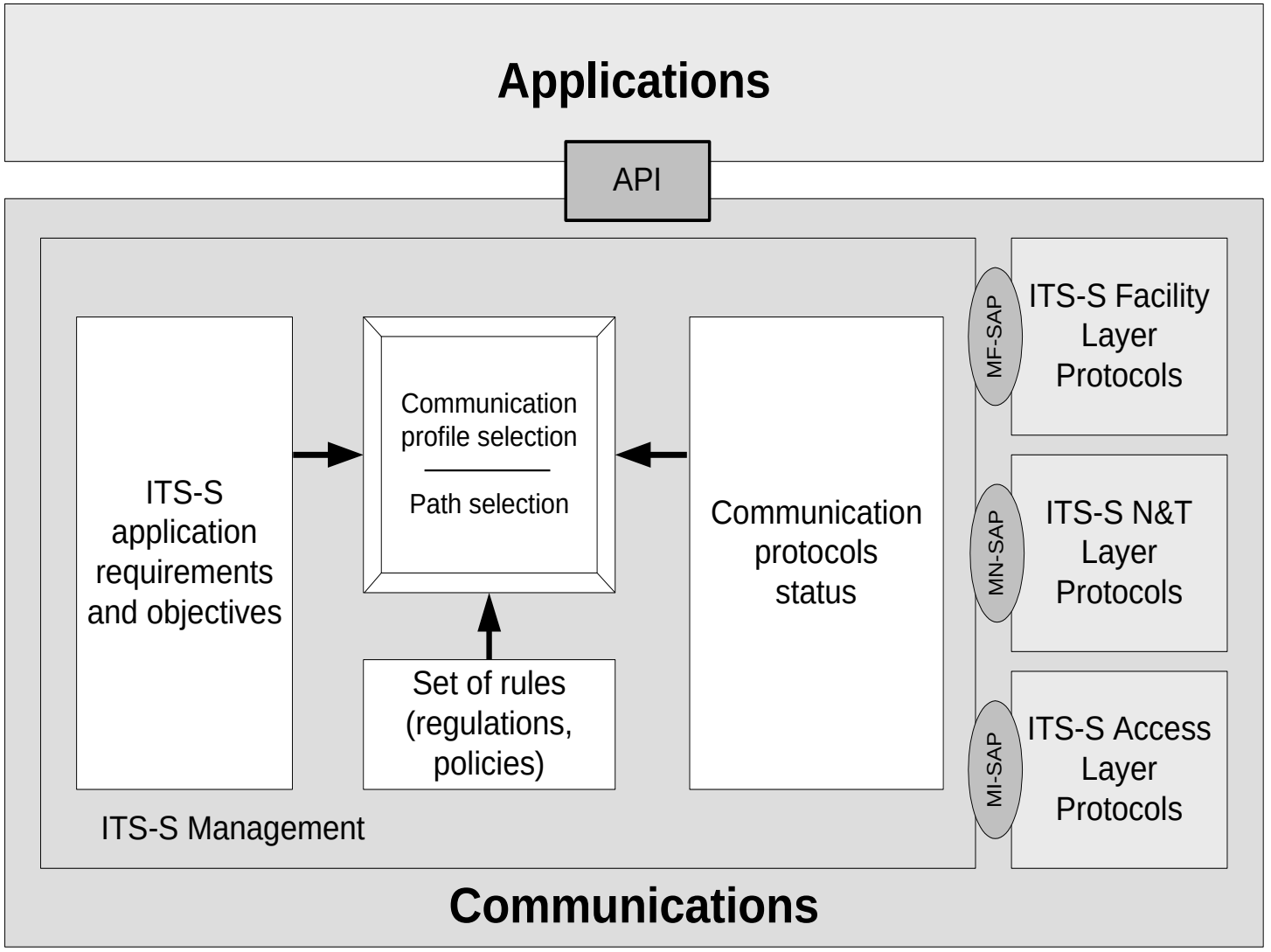
ITS stations

- A “Trusted” ITS station (ITS-S) is a functional entity consisting of ITS-S application processes running inside a bounded, SECURED, managed domain (cf. ISO 21217).
- SECURED means “ITS trusted/trustable”
- MANAGED means “someone is watching the store”
- The ITS-S architecture is designed around the abstraction of ITS-S application processes from the communication means used to connect to peer entities.
- Instantiations of ITS-Ss are called ITS station units (ITS-SUs).
- ITS-SUs communicate with peer entities using the “optimal” means available (cf. ISO 17419/17423/ 17429/24102-6).

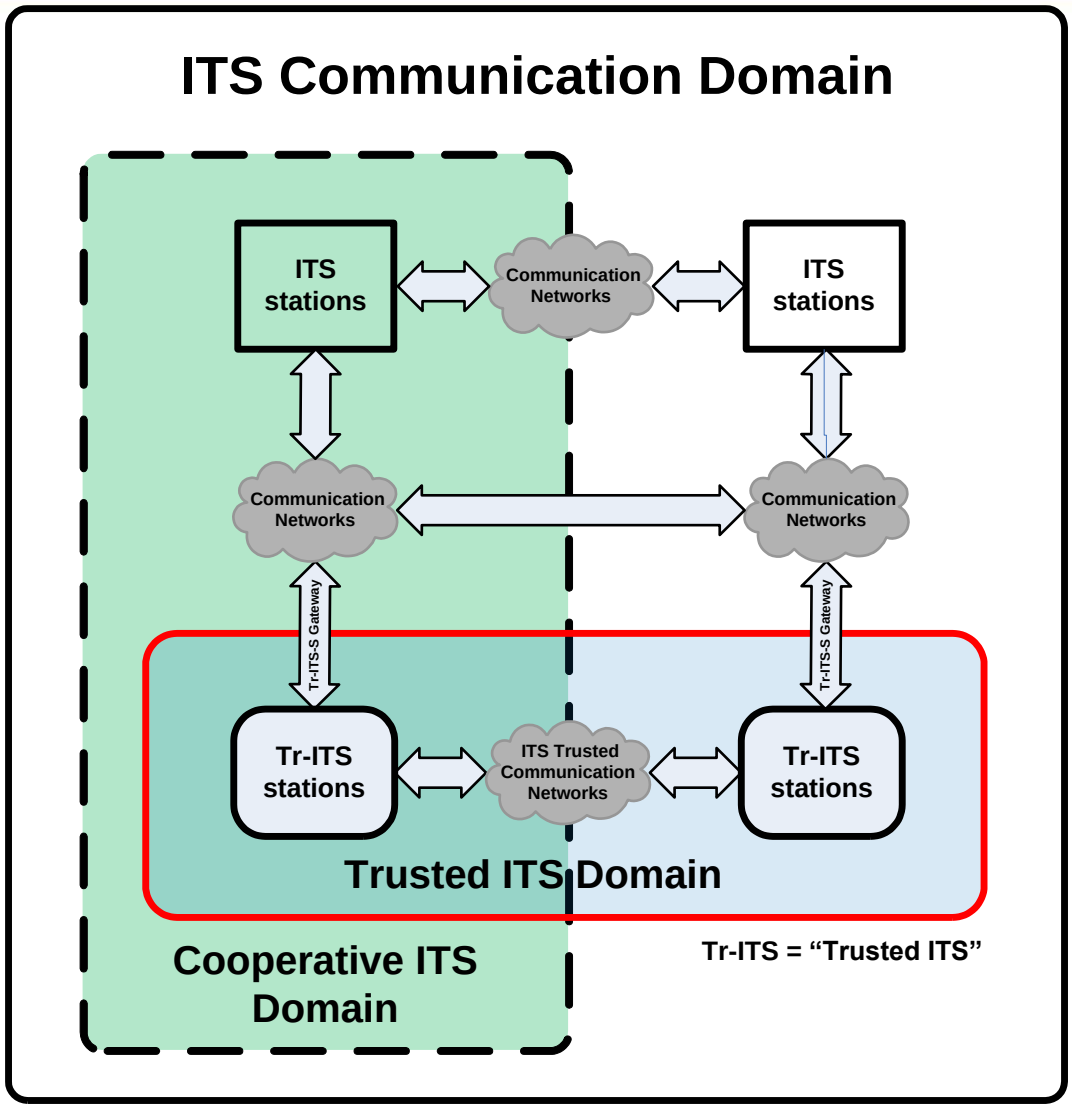
ITS stations



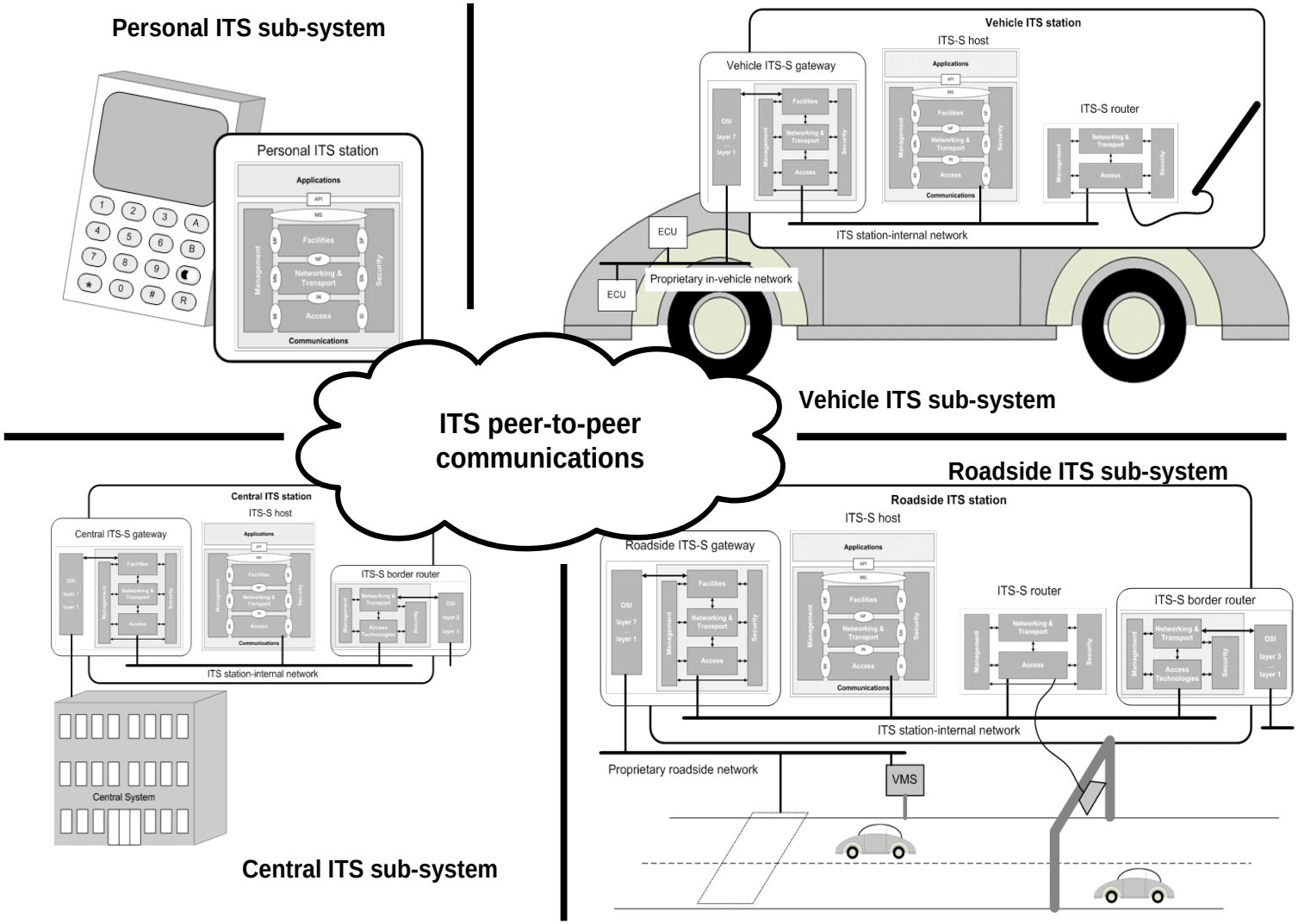
ITS stations



Cooperative and Trusted ITS Domains



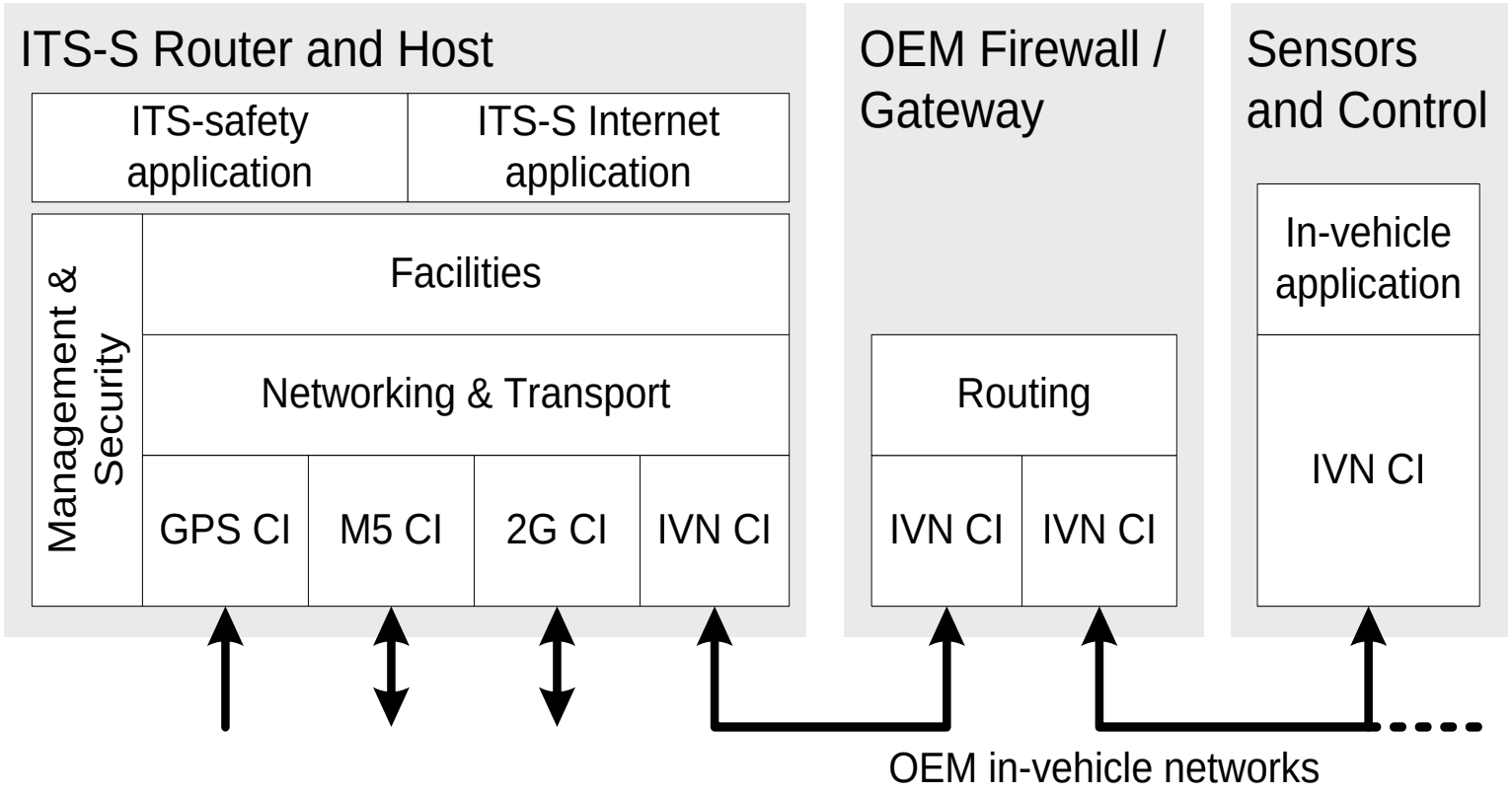
ITS Subsystems



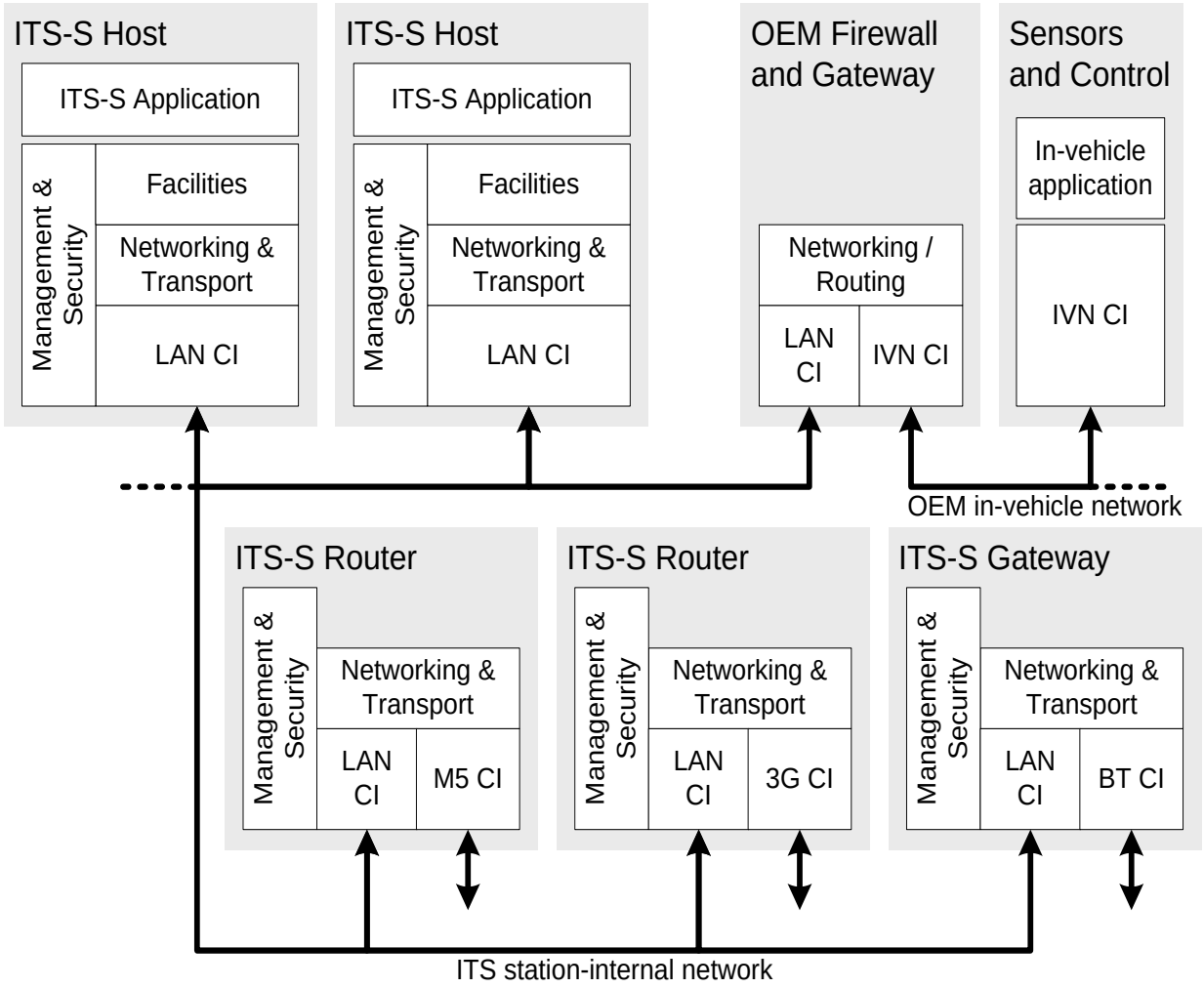
Advantages and considerations

- Open architecture (“plug-and-play”)
  - ITS communications is more than just V2V and V2I!
- ITS-S application developers can instantiate services easily including RMI/OBD
- System security designed in from the start
  - VLAN establishment possible
- NB: Many implementations are possible as long as there is conformance to specifications at open interfaces! (“OEMs can take on as much of the job as they choose.”)

Simple Vehicular ITS-S Implementation



Complex Vehicular ITS-S Implementation



In-vehicle gateways

- Two gateways into Vehicular ITS-SUs:
  - In-vehicle gateway provides access between the ITS-SU internal network and the in-vehicle network (e.g. CAN bus) … the VSG!
  - ITS-SU egress/ingress gateway provides access between the ITS-SU and a personal device (e.g. smart phone)
- Both provide secure and managed access to vehicular ITS-SU internal communication services and application processes

Implementations to date

- Connected Vehicle Infrastructure System (CVIS) project in Europe (several 100's of OBUs and 10's RSUs with a large number of applications)
- US DoT sponsored SafetyPilot (several 1000's of OBUs and 10's of RSUs)

General IVN Data & VSG Requirements

- Security and Authentication
  - Data sources (cf. IEEE 1609.2)
  - "Comprehensive Experimental Analyses of Automotive Attack Surfaces" (20th USENIX Conference, San Diego, CA)

    Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage
    University of California, San Diego
    Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno
    University of Washington
- Unambiguous specification/definition of information elements/messages
  - e.g. Accelerometer outputs
- Openly available data dictionaries / message set specifications (cf. ISO 17419 / SAE J2735)
- Provide services that allow for differential access to information/data
- Manage IVN access to ensure control system integrity and safety

General IVN Application Requirements

- IVN apps (could) use ISO 17419 functions to implement secure and verifiable ECU software upgrades (in the same manner that ITS-Ss manage software upgrades)
- IVN apps (could) use ISO 17423 functions to express any run time requirements (if any)
- IVN apps (could) use ISO 17429 publish mechanism to provide “in-vehicle” data to ITS-S applications in ITS-SUs

General Security Considerations

- Two main security-related concerns:
  - Verify the “person(s)” you are communicating with is “who he claims to be” (authentication)
  - Communicate without anyone else being able to “listen in” (secure session)
- Authentication with and without pre-shared information (secrets)
  - Public Key Infrastructure (PKI)
- Secure sessions (asymmetric/symmetric keys)
- Hardware security modules (tamper-proof secure storage for security material)

Some Relevant ISO/IEEE/SAE Standards (not exhaustive!)

- ISO 21217 - CALM Architecture
- ISO 24102-x ITS station management
  - (ISO 24102-5 Fast Service Advertisement Protocol)
- ISO 29281-1 - Fast Networking Protocol
- ISO 17419 – ITS-S application identifiers, registries and certification (Message sets and IDs)
- ISO 17423 - ITS-S application management
- ISO 17429 - Generic Message Handling Services
- ISO 18750 - Local Dynamic Map
- IEEE 1609.2 - WAVE Security Services
- SAE J2735 – DSRC Message Set Dictionary (incl. probe vehicle data)
- IEEE 802.11 / 1609.3 / 1609.4 (5.9 GHz safety communications)
- ISO 21215 / EN 302 665 (5.9 GHz safety communications)

Contact Information

DG Technology
Greg Potter
+1 (734) 891-3816
GPotter@dgtech.com

Concepts & Services Consulting
Gangolf Feiter
+49 (2452) 96 484 - 11
gangolf.feiter@csc-online.eu

SRA, Inc.
Dr. Richard Roy
+1 (650) 861 - 3351
dickroy@alum.mit.edu