




Automotive - UDS

UDS - ISO 14229 An Implementation Perspective

UDS - ISO 14229 An Implementation Perspective

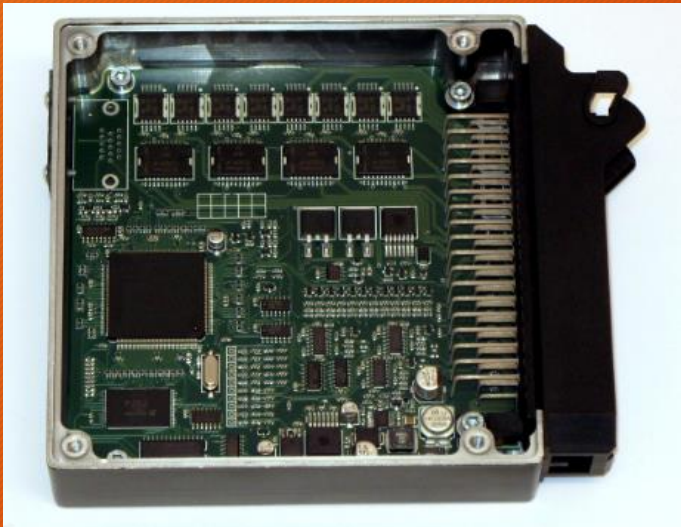
- ☐ What is UDS?
 - ☐ What is the Goal of UDS?
 - ☐ What are the Different Services UDS has?
 - ☐ How can we use the services in UDS?
 - ☐ What is NRC?
 - ☐ Where is the Application of UDS?
- 

Introduction of ECU

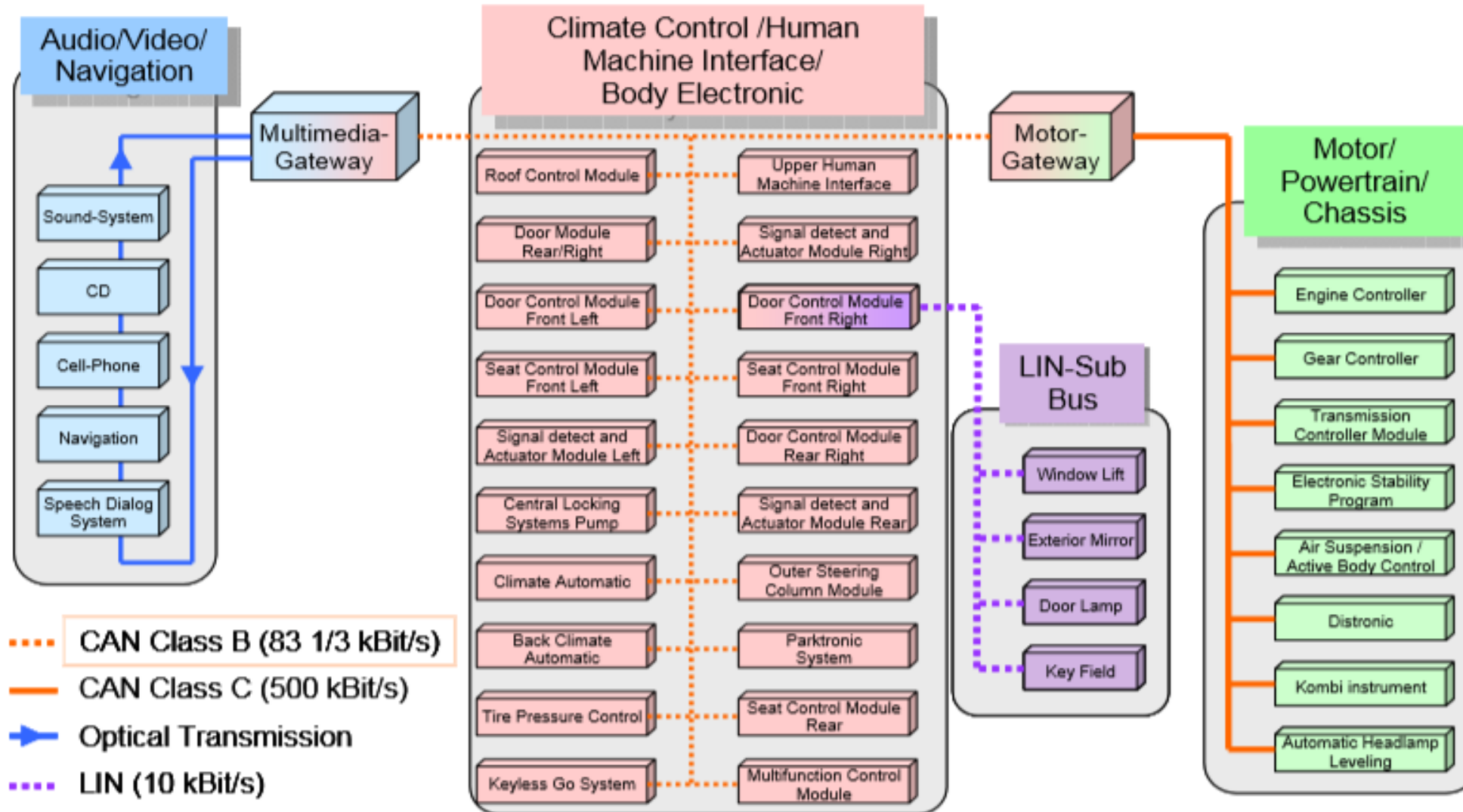
- ❑ A computer system that is embedded into an piece of equipment or a machine to control it.
- ❑ Embedded systems are applied to most electric/ electronic equipment, recently.
- ❑ ECU is any embedded system that controls one or more of the electrical system or subsystems in a transport vehicle.
- ❑ The first electronics pieces were used to control engine functions and were referred to as Engine Control Units (ECU).
- ❑ A modern car may have up to 100 ECUs and a commercial vehicle up to 40.

ECU consists of:

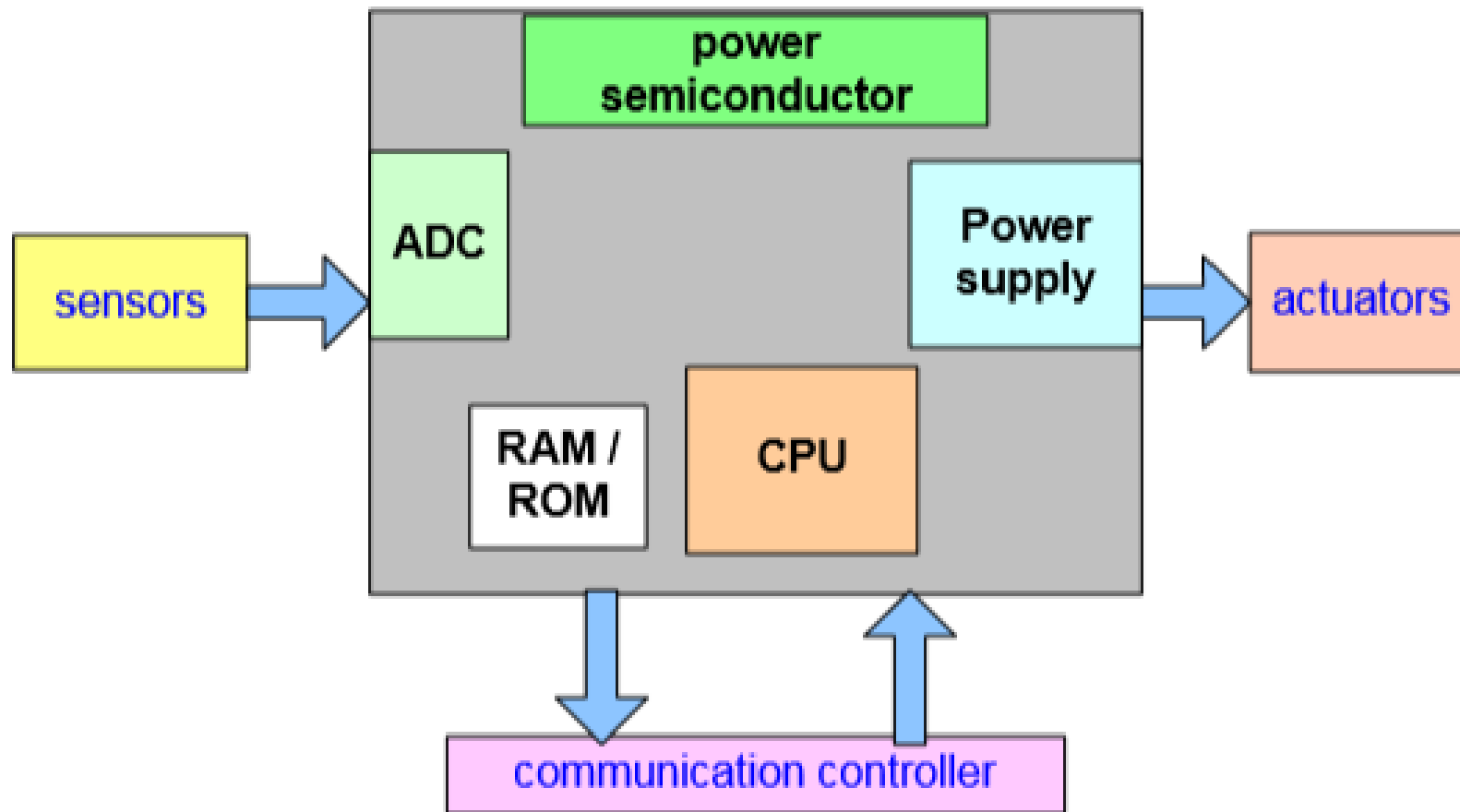
- Microcontrollers
- Sensors
- Power switches
- Actuators
- Voltage regulators
- Controls the functionalities of vehicle as per the inputs from sensors placed at different places inside vehicle.
- ECUs are used for energy saving, low emission, safety, convenience, entertainment etc..



Overview of Electronic System of a Modern Vehicle



Automotive - ECU



Unified Diagnostic Services UDS

□ What is UDS?

The Unified Diagnostic Services (UDS) Protocol is a communication protocol in the automotive field, developed based on the idea of Keyword Protocol (KWP2000 [3]) to fulfill common requirements for diagnostic systems on CAN buses. The commands in UDS are divided into six groups according to their functionalities described in the following sections.

UDS - Functional Units

□ UDS Protocol

The Unified Diagnostic Services (UDS) Protocol is a communication protocol in the automotive field, developed based on the idea of Keyword Protocol (KWP2000 [3]) to fulfill common requirements for diagnostic systems on CAN buses. The commands in UDS are divided into six groups according to their functionalities described in the following sections.

1. Diagnostic & Communication Management Functional Unit(DCM)

It specifies the ECU functionality, manages the diagnostic system, defines the user accessibility, controls the transmission of data etc

2. Data Transmission Functional Unit (DT)

It has the capability of accessing current data. Data must be read or written immediately after each request

3. Stored Data Transmission Functional Unit,

This group of functional unit allows Diagnostic tools to access the server's memory. The diagnostic tools can clear or read diagnostic information defined by the server's database inside memory.

UDS Functional Unit Contd.

❑ Input Output Control Functional Unit

It controls actuators, sensors, devices and equipment attached to the server. All of those inputs and outputs must be defined as Data Identifiers by OEM, Vehicle Manufacturer, etc., following the ISO standard.

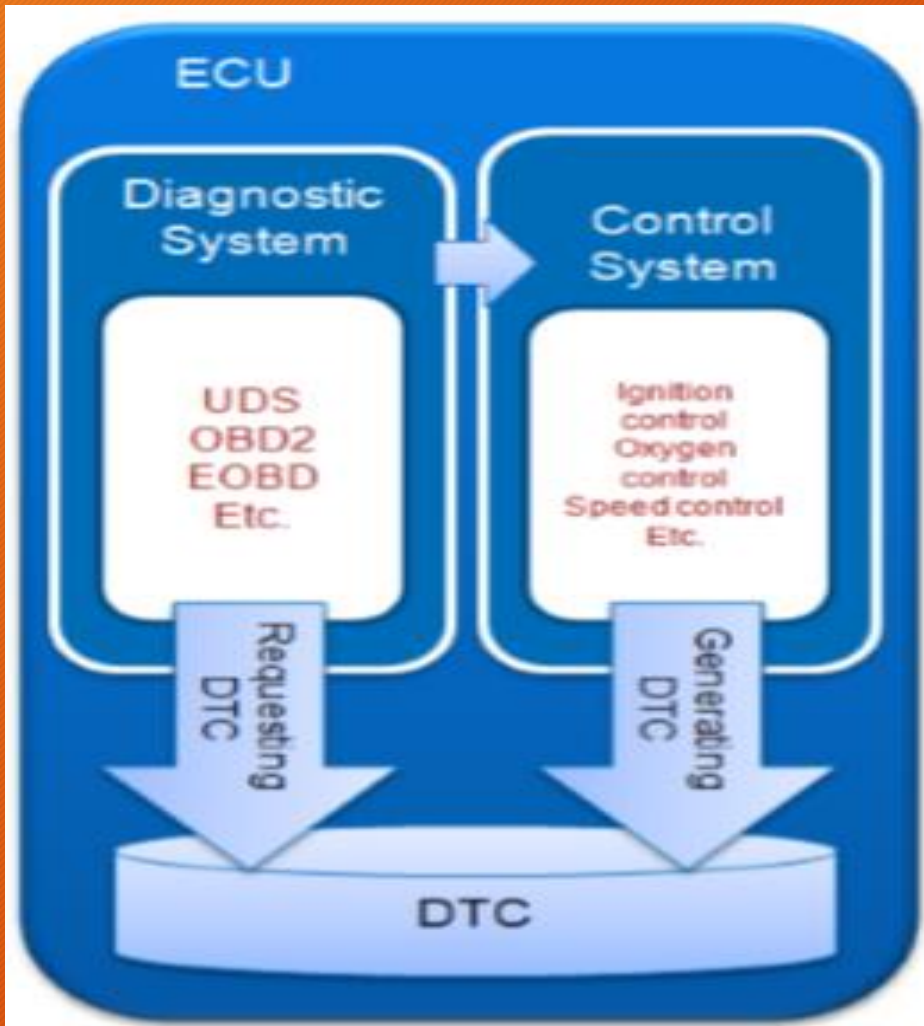
❑ Remote Activation of Routines Functional Unit

It is having set of service that influences an ECU's behavior. In order to use this service, the routine identifier must be defined first by the vehicle manufacturer, the ISO standard, system supplier, etc.

❑ Upload Download Functional Unit

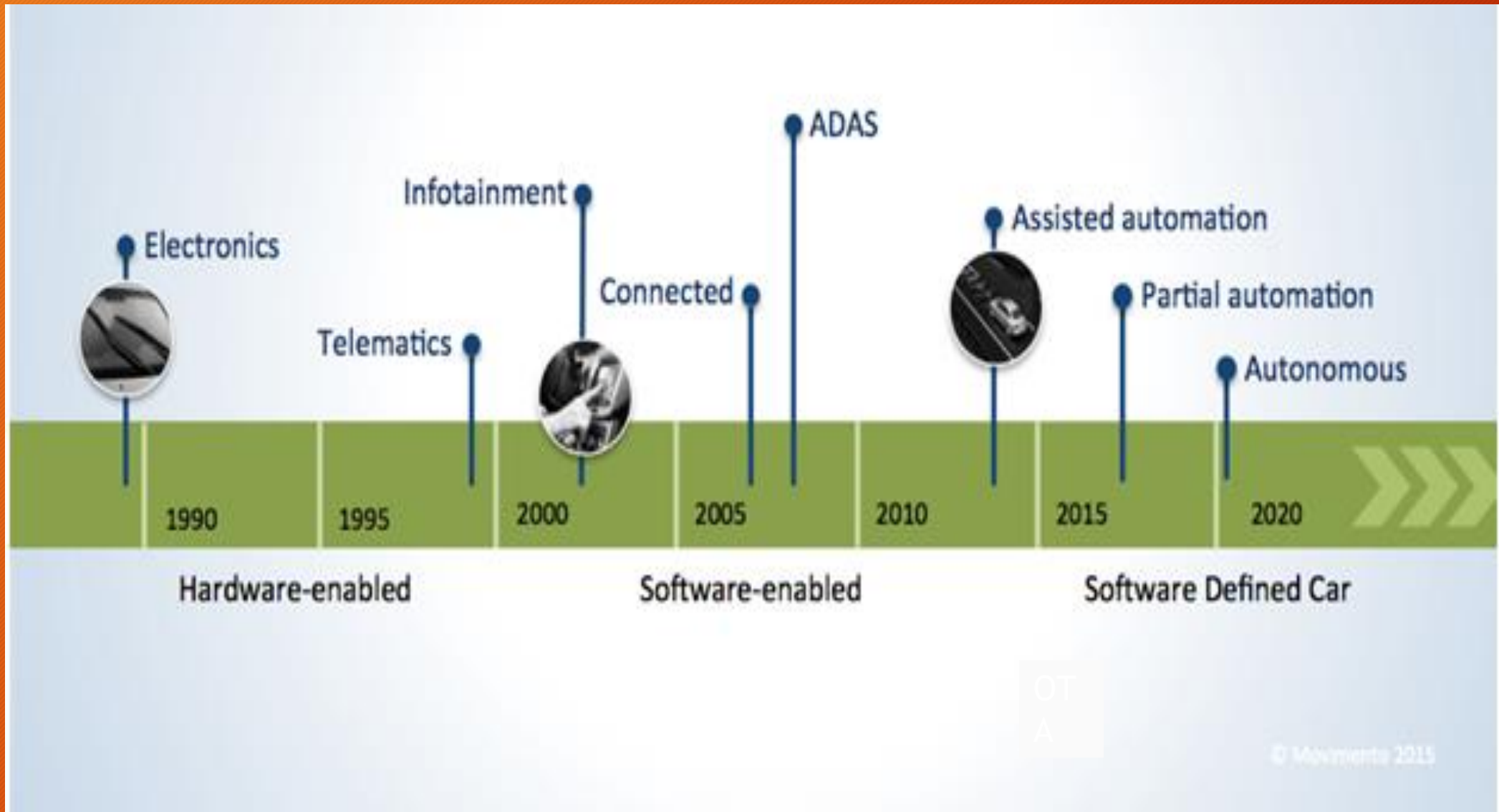
These command are used for requesting the negotiation of data transfer between a client and a server. This functional group contains services for requesting to start the transmission to manipulate data in an ECU's memory, for transferring data and for requesting to terminate a data transmission.

UDS - An example



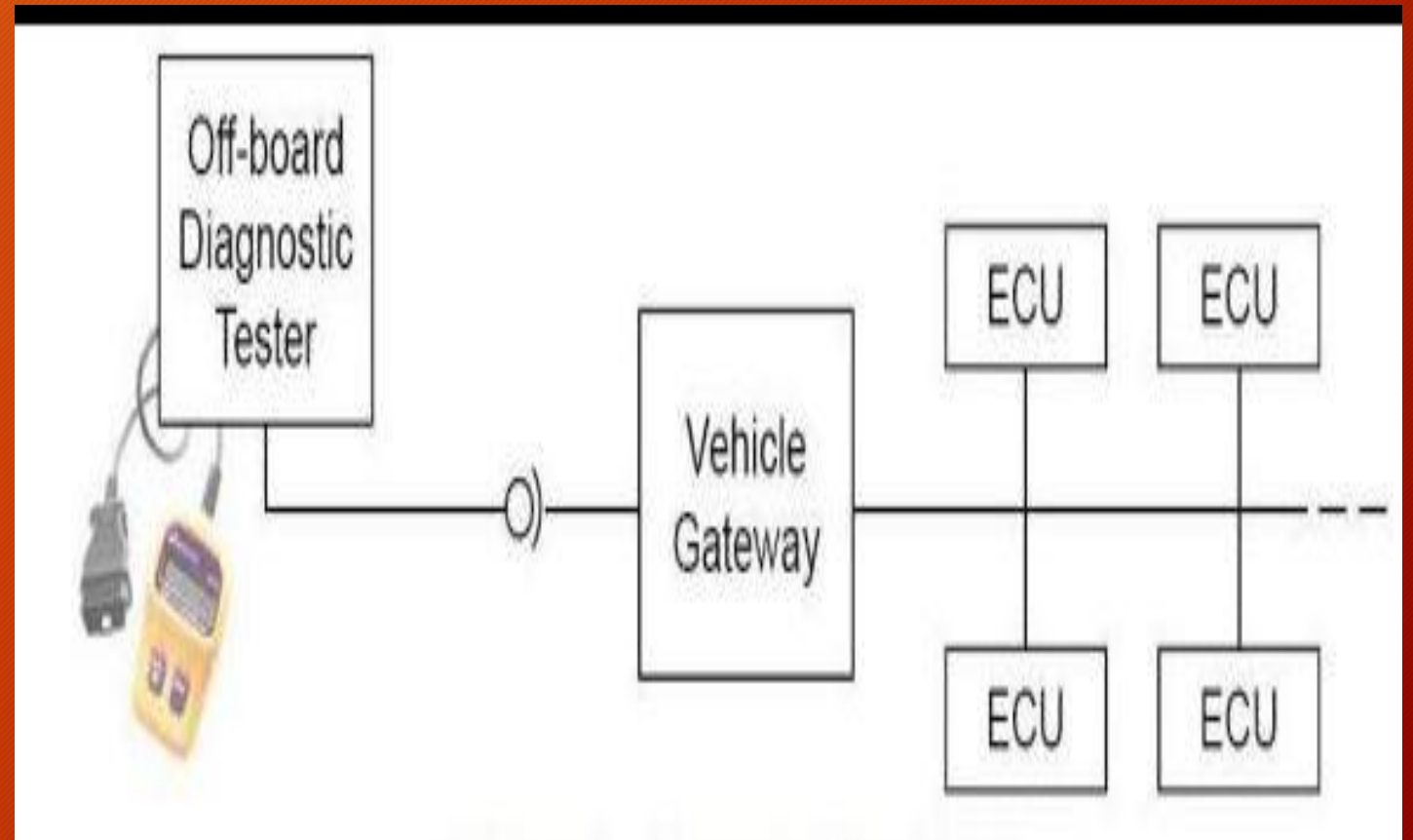
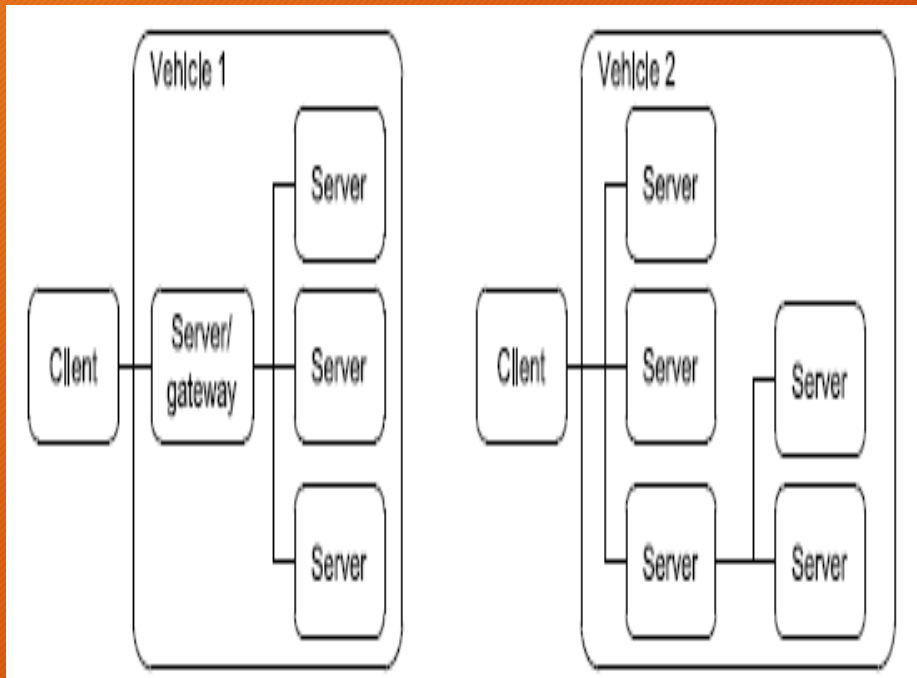
The control system is created for controlling a combustion engine. It has a capability to generate Diagnostic Trouble Codes (DTCs) if there was detected a malfunction. These DTCs provide status information of an engine. The DTCs are used by repairers or technicians for maintaining and diagnosing an engine. However, the repairer and technician must not have the capability to access the control system itself. Therefore, the diagnostic system is created to limit the access rights. The diagnostic system must provide complete information of a DTC and some part of the control system for calibrating purpose

Automotive Software-What Next for us



UDS - Network

Communication across the ECUs takes place as it happens in normal network. ECU network gate highways

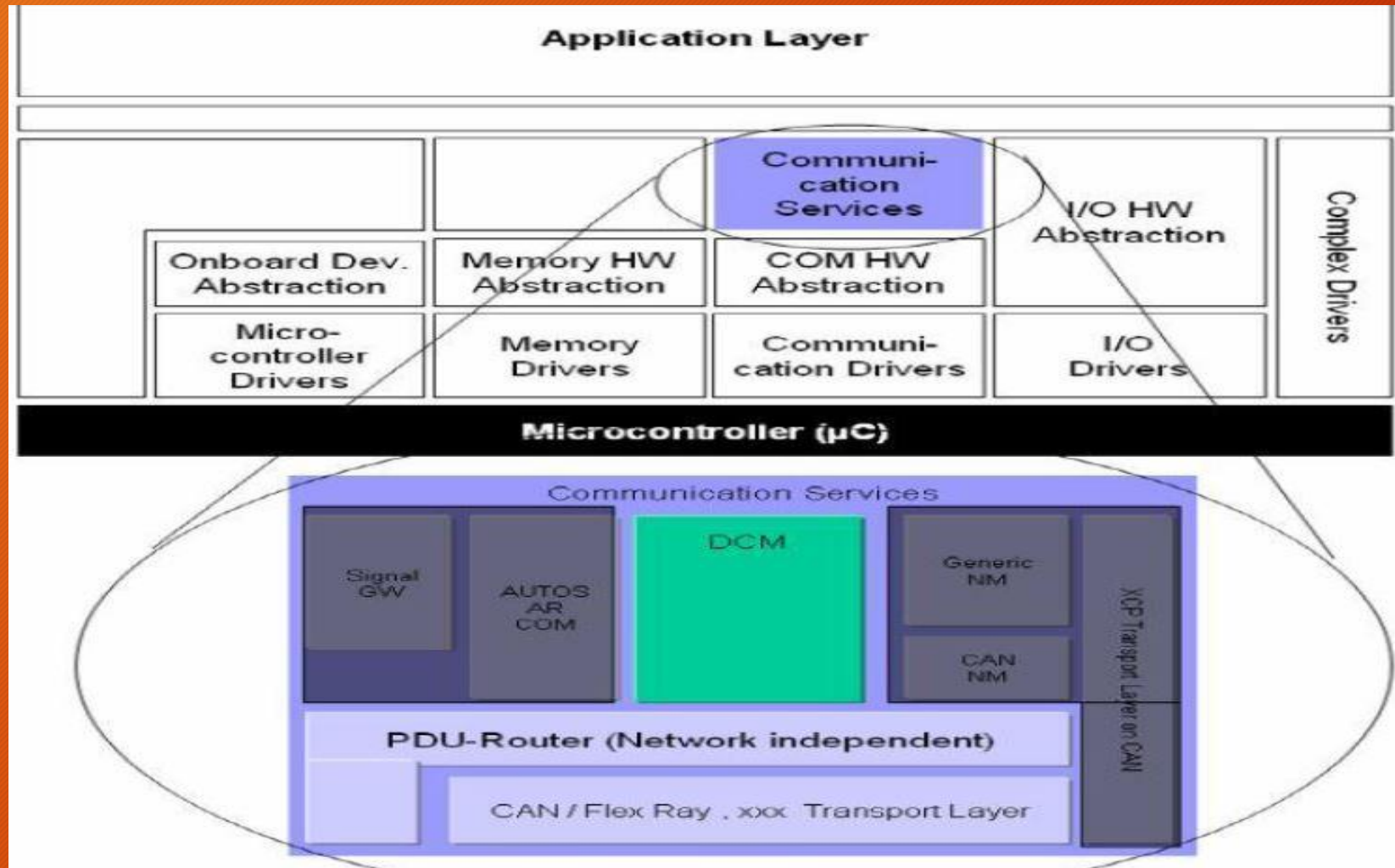


UDS - Transport Protocols

Communication across the ECUs takes place as it happens in normal network following the OSI layer protocols. There is proper ECU defined network

Application Layer	XCP	CCP	UDS	OBD2
Presentation Layer	N/A (Not Available)		N/A	
Session Layer	N/A		N/A	
Transport Layer	N/A		Diagnostic on CAN – Transport Layer (ISOTP)	
Network Layer	User defined		Diagnostic on CAN – Network Layer (ISOTP)	
Data Link Layer	CAN			
Physical Layer	CAN bus			

Relationship between UDS and AUTOSAR



What is the Goal of UDS?

- ❑ Automotive diagnostic service is the way to discover, what's wrong with the vehicle if it is not running / behaving properly.
- ❑ Diagnosing a vehicles problem in order to fix or repair a car can be done through a professional automobile diagnostic center. Diagnostic Service uses the read command for retrieving the information/data from ECU .

Diagnostic Session?

☐ Sessions:

Diagnostic session is the basis of communication between the ECU and the diagnostic tool.

- ☐ Default Session
- ☐ Programming Session
- ☐ Extended Diagnostic Session

What is the Service ?

❑ Service:

Information exchange initiated by a client in order to require diagnostic information from a server or/and to modify its behaviour for diagnostic purpose

which allow a diagnostic tester (client) to control diagnostic functions in an on-vehicle Electronic Control Unit (ECU, server).

Example : electronic fuel injection, automatic gear box, anti-lock braking system, etc.

What is Diagnostic Data?

❑ Diagnostic Data:

Diagnostic data that is located in the memory of an electronic control unit (ECU) which may be inspected or modified by the tester.

Example of Diagnostic Data: vehicle speed, throttle angle, mirror position, system status, etc

❑ Three types of values are defined for diagnostic data:

❑ Current Value

The value currently used by the normal operation of the Electronic Control Unit (ECU).

❑ The Stored Value

An internal copy of the current value, this copy is made under the control of the electronic control (ECU) unit.

❑ Static Value: e.g. VIN.

What is NRC?

❑ Negative Response Code :

0x11 ServiceNotSupported

0x12 Sub-functionNotSupported

0x13 IncorrectMessageLengthorInvalidFormat

0x14 ResponseTooLong

0x22 ConditionNotCorrect

0x31 requestOutOfRange

0x33 SecurityAccessDenied

Diagnostic Services

❑ Deagnostic Services

- ❑Diagnostic Session Control - 0x10
- ❑ECUReset - 0x11
- ❑Security Access - 0x27
- ❑Communication Control - 0x28
- ❑TesterPresent - 0x3E
- ❑ControlDTCSetting - 0x85
- ❑ReadDataByIdentifier - 0x22
- ❑WriteDataByIdentifier - 0x2E

Data Byte No.	Parameter Name	Data Value
1	Diagnostic Session Control	10
	Service Id	
2	sub Function	
	Default Session	01
	Programming Session	02
	Extended Session	03

Diagnostic Session Control

□ Diagnostic Session Control

Used to enable different diagnostic sessions in one ECU or a group of ECUs.

Data Byte No.	Parameter Name	Data Value
1	Diagnostic Session Control Service Id	10
2	sub Function	
	Default Session	01
	Programming Session	02
	Extended Session	03

Diagnostic Session Control - Responses

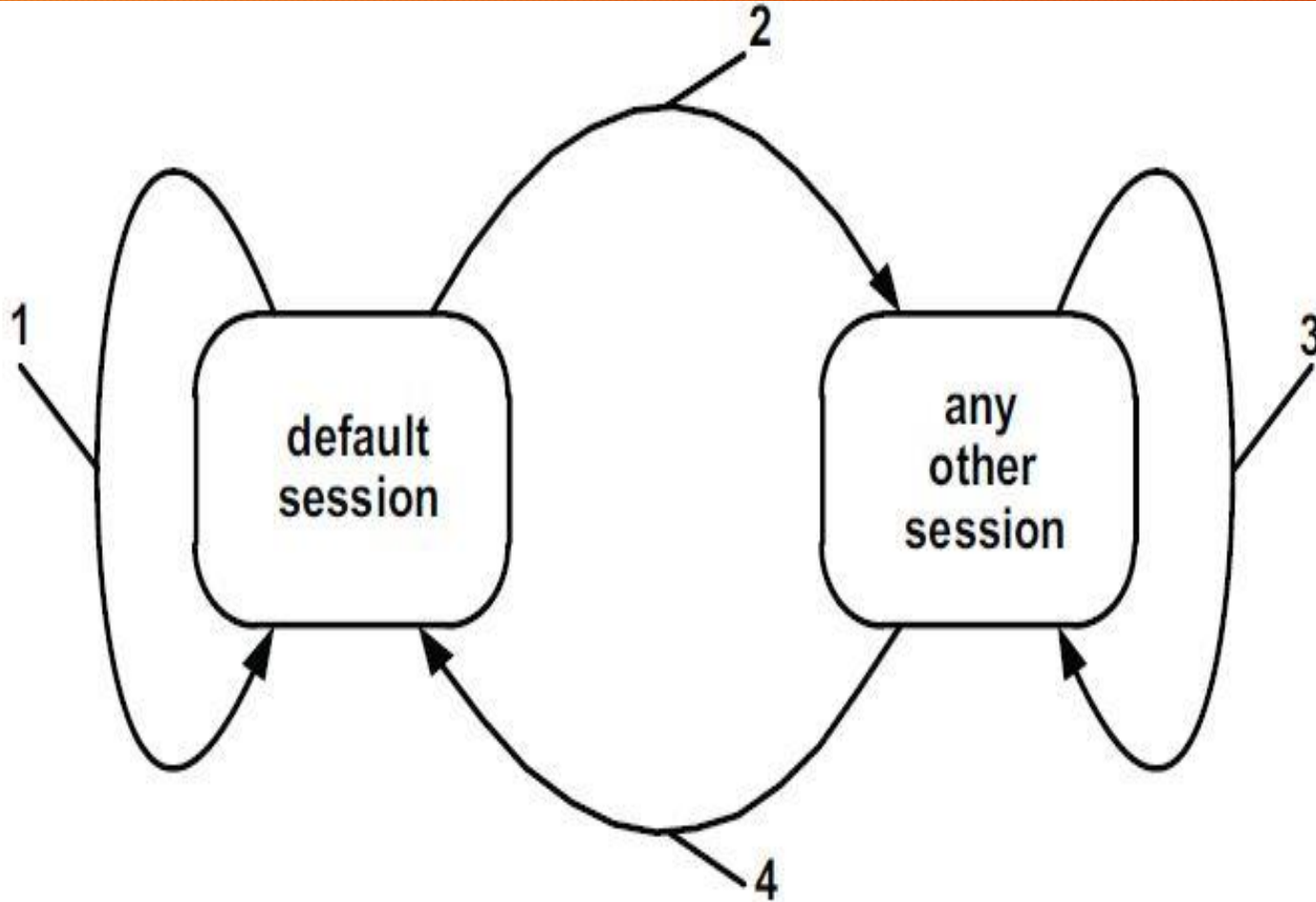
❑ Diagnostic Session Control Positive Response message frame:

Data Byte No.	Parameter Name	Data Value
1	Diagnostics Session Control Response Service Id	50
2	sub Function	00-FF
3	Session Parameter Record	00-FF

❑ Diagnostic Session Control Negative Response message frame:

Data Byte No.	Parameter Name	Data Value
1	Negative Response	7F
2	Diagnostics Session Control	10
3	sub function	00-FF
	Sub function Not supported	12
	Incorrect Message Length or Invalid Format	13
	Conditions are not correct	22

Diagnostic Session Control - State Transition at Servier



- 1. Default session
- 2 Other session
- 3 Same or other session
- 4 Default session

ECU Reset Service - Request Message Definition

Data Byte No.	Parameter Name	Data Value
1	ECU Reset Request Service Id	11
2	sub Function	
	Hard Reset – Positive response required	01
	Key On-Off Reset – Positive Response Required	02
	Soft Reset – Positive Response Required	03

ECU Reset- Responses

❑ ECU Reset Positive Response message frame

Data Byte No.	Parameter Name	Data Value
1	ECU Reset Response Service Id	51
2	sub Function	00-FF
3	Power Down Time	00-FF

❑ ECU Reset Negative Response message frame

Data Byte No.	Parameter Name	Data Value
1	Negative Response	7F
2	ECU Reset	11
3	sub Function	00-FF
	Sub function Not supported	12
	Incorrect Message Length or Invalid Format	13
	Conditions are not correct	22

Security Access

Data Byte No.	Parameter Name	Data Value
1	Security Access Request Service ID	27
2	sub Function = [Security Access type]	00-FF
Request Seed – Positive Response Required	01	
Send Key – Positive Response Required	02	
3	Security key[1]	00-FF

Security Access

Data Byte No.	Parameter Name	Data Value
1	Security Access Request Service ID	27
2	sub Function = [Security Access type]	00-FF
Request Seed – Positive Response Required	01	
Send Key – Positive Response Required	02	
3	Security key[1]	00-FF

Security Access - Responses

❑ Security Access Negative Response message frame:

Data Byte No.	Parameter Name	Data Value
1	Security Access Response Service Id	67
2	Security Access Type	00-FF
3	Security Seed [2]	00-FF

Data Byte No.	Parameter Name	Data Value
1	Negative Response	7F
2	SecurityAccess	27
3	sub-function	00-FF
Sub-Function Not Supported	12	
Incorrect Message Length	13	
Conditions Not Correct	22	
Request Sequence Error	24	
Invalid Key	35	
Exceeded Number Of Attempts	36	
Required Time Delay Not Expired	37	

Example : server is in a “locked” state but want to communicate with Client

❑ Security Access request message :

SecurityAccess Request SID	0x27
SecurityAccessType = requestSeed,	0x01

❑ Security Access positive response:

SecurityAccess Response SID	0x67
securityAccessType = requestSeed	0x01
securitySeed [byte#1] = seed#1 (high byte)	0x36
securitySeed [byte#2] = seed#2 (low byte)	0x57

Example : server is in a “locked” state but want to communicate with Client

Security Access request message :

SecurityAccess Request SID	0x27
securityAccessType = sendKey, suppressPosRspMsgIndicationBit = FALSE	0x02
securityKey [byte#1] = key#1 (high byte)	0xC9
securityKey [byte#2] = key#2 (low byte)	0xA9

Security Access positive response message :

SecurityAccess Response SID	0x67
securityAccessType = sendKey	0x02

Communication Messages

❑ Communication Control request message :

CommunicationControl Request SID	0x28
controlType = enableRxAndDisableTx, suppressPosRspMsgIndicationBit = FALSE	0x01
communicationType = network management	0x02

❑ Communication Control positive response message :

CommunicationControl Response SID	0x68
ControlType	0x01

Communication Messages

❑ Communication Control Negative Response message Flow

Data Byte No.	Parameter Name	Data Value
1	Negative Response	7F
2	Communication Control	28
3	sub-function	00-FF
Sub-Function Not Supported	12	
Incorrect Message Length	13	
Conditions Not Correct	22	
Request Sequence Error	31	

Tester Present

❑ Tester Present request message flow

TesterPresent Request SID	0x3E
zeroSubFunction, suppressPosRspMsgIndicationBit = FALSE	0x00

❑ Tester Present positive response message flow

TesterPresent Response SID	0x7E
zeroSubFunction, suppressPosRspMsgIndicationBit = FALSE	0x00

Tester Present

❑ Tester Present Negative Response message Flow

Data Byte No.	Parameter Name	Data Value
1	Negative Response	7F
2	Tester Present	3E
3	sub-function	00-FF
Sub Function Not Supported	12	
Incorrect Message Length – Invalid Format	13	

ControlDTCSetting

❑ ControlDTCSetting request message flow :

ControlDTCSetting Request SID	0x85
DTCSettingType = off, suppressPosRspMsgIndicationBit = FALSE	0x02

❑ ControlDTCSetting positive response message flow :

ControlDTCSetting Response SID	0xC5
DTCSettingType = off	0x02

ControlDTCSetting

ReadDataByIdentifier :

ReadDataByIdentifier Request SID	0x22
dataIdentifier [byte#1] (MSB)	0xF1
dataIdentifier [byte#2]	0x90

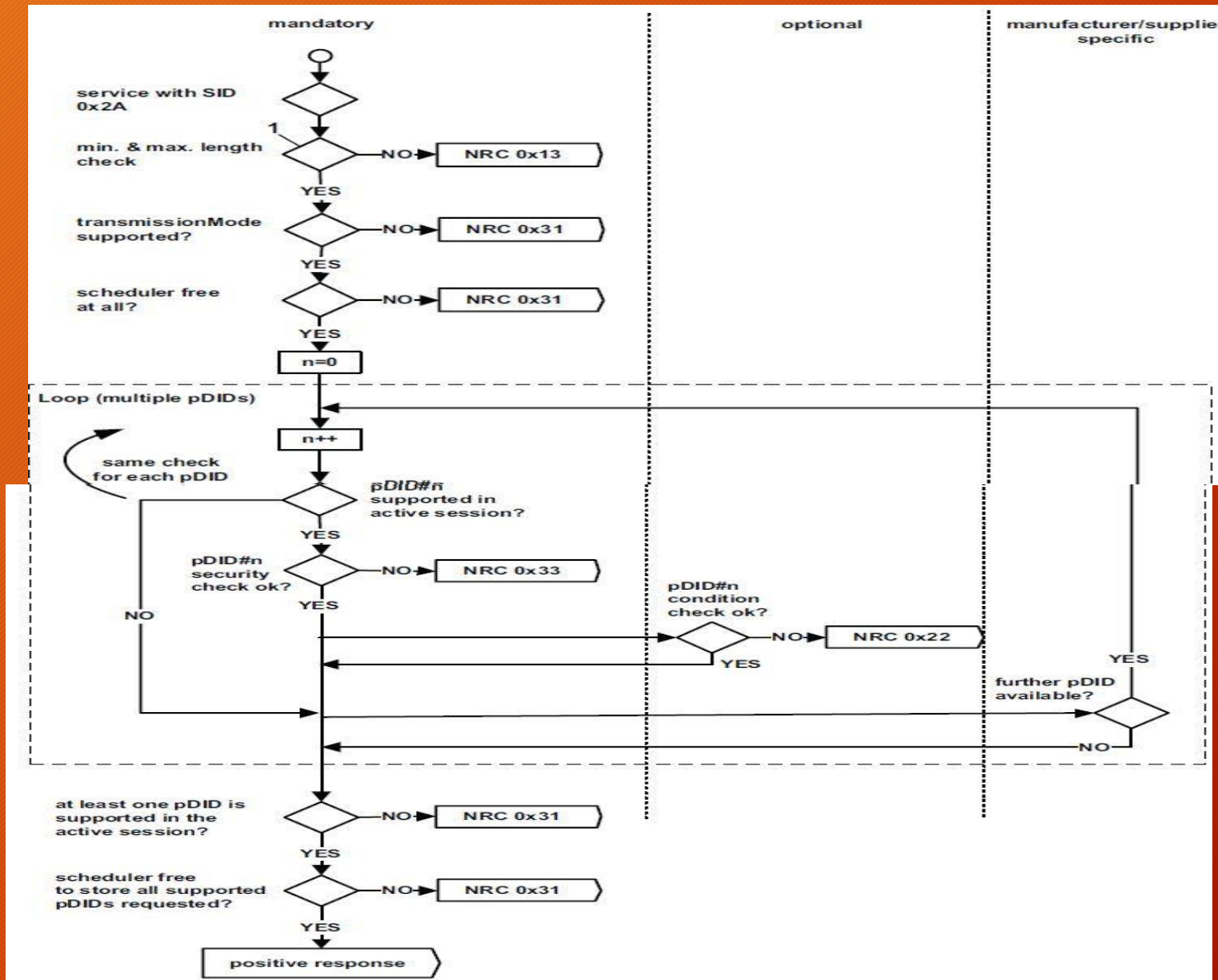
ReadDataByIdentifier positive response message flow :

ReadDataByIdentifier Response SID	0x62
dataIdentifier [byte#1] (MSB)	0xF1
dataIdentifier [byte#2]	0x90
dataRecord [data#1] = VIN Digit 1 = "W"	0x57
dataRecord [data#2] = VIN Digit 2 = "0"	0x30
dataRecord [data#3] = VIN Digit 3 = "L"	0x4C
dataRecord [data#4] = VIN Digit 4 = "0"	0x30
dataRecord [data#5] = VIN Digit 5 = "0"	0x30
dataRecord [data#6] = VIN Digit 6 = "0"	0x30
dataRecord [data#7] = VIN Digit 7 = "0"	0x30

Read Data By Identifier Negative Response message Flow

Data Byte No.	Parameter Name	Data Value
1	Negative Response	7F
2	Read Data By Identifier	22
3	sub-function	00-FF
	Incorrect Message Length- Invalid Format	13
	Conditions Not Correct	22
	Request Out of Range	31
	Security Access Denied	33

Flow



WriteDataByIdentifier

WriteDataByIdentifier request message flow

WriteDataByIdentifier Request SID	0x2E
dataIdentifier [byte#1] (MSB)	0xF1
dataIdentifier [byte#2]	0x90
dataRecord [data#1] = VIN Digit 1= "W"	0x57
dataRecord [data#2] = VIN Digit 2= "0"	0x30
dataRecord [data#3] = VIN Digit 3= "L"	0x4C
dataRecord [data#4] = VIN Digit 4= "0"	0x30
dataRecord [data#5] = VIN Digit 5= "0"	0x30
dataRecord [data#6] = VIN Digit 6= "0"	0x30
dataRecord [data#7] = VIN Digit 7= "0"	0x30
dataRecord [data#8] = VIN Digit 8= "4"	0x34
dataRecord [data#9] = VIN Digit 9= "3"	0x33
dataRecord [data#10] = VIN Digit 10 = "M"	0x4D
dataRecord [data#11] = VIN Digit 11 = "B"	0x42

WriteDataByIdentifier

WriteDataByIdentifier positive response message flow :

WriteDataByIdentifier Response SID	0x6E
dataIdentifier [byte#1] (MSB)	0xF1
dataIdentifier [byte#2] (LSB)	0x90

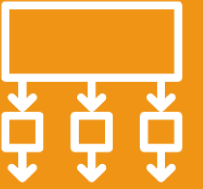
WriteDataByIdentifier negative response message flow :

Data Byte No.	Parameter Name	Data Value
1	Negative Response	7F
2	Read Data By Identifier	22
3	sub-function	00-FF
	Incorrect Message Length- Invalid Format	13
	Conditions Not Correct	22
	Request Out of Range	31
	Security Access Denied	33

Thank You



What are your next steps?



- UDS Design document



- UDS Sample coding



- Unit Testing & Unit Testing Tools



- SCM, Design Pattern & other tools

