AUTOSAR Diagnostic Overview

Terms and Definitions

Diagnostic service

information exchange initiated by a client in order to require diagnostic information from a server and/or to modify its behavior for diagnostic purposes

Server

function that is part of an electronic control unit and that provides the diagnostic services

Tester

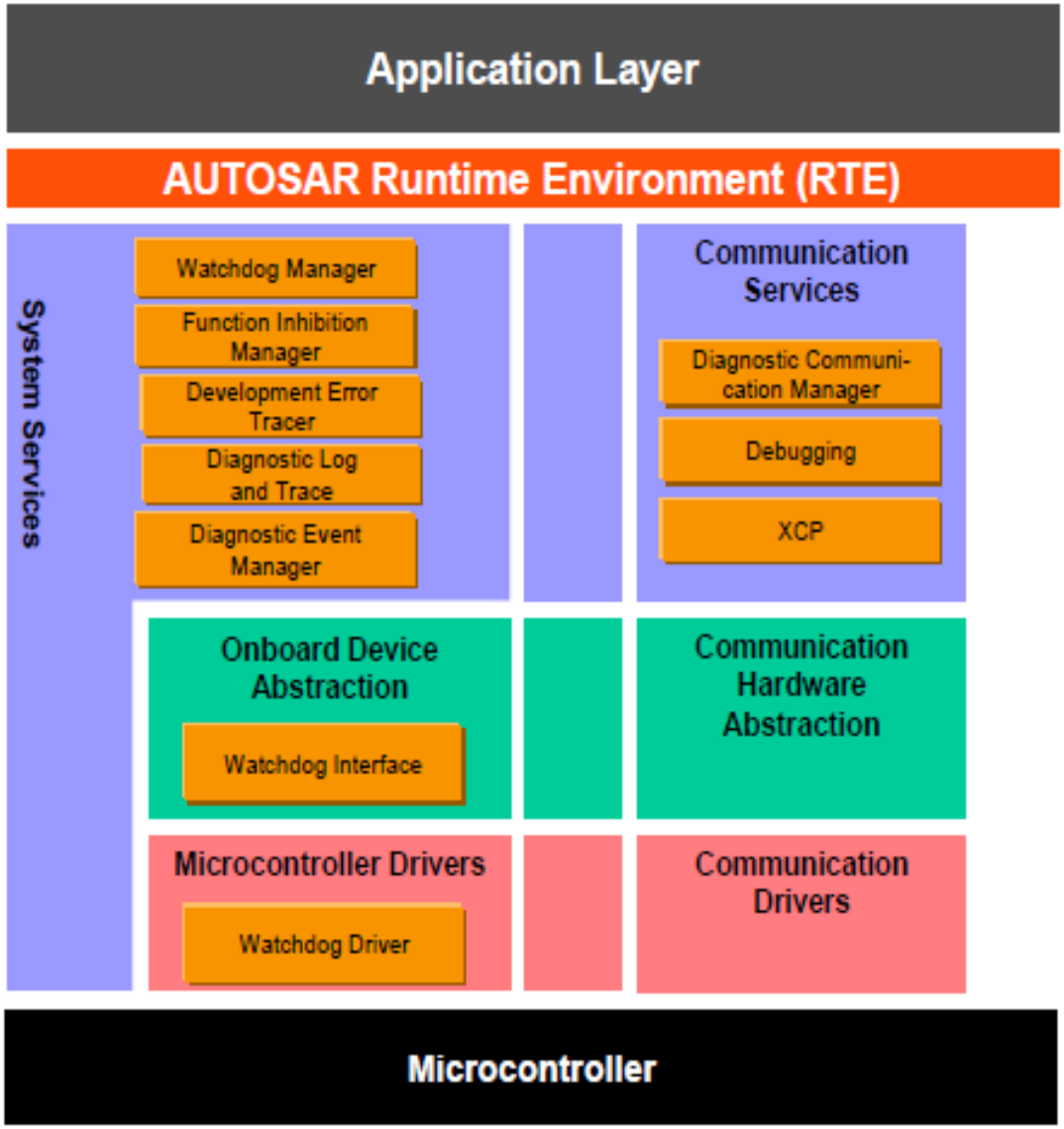
system that controls functions such as test, inspection, monitoring or diagnosis of an on-vehicle electronic control unit and which may be dedicated to a specific type of operator (e.g. a scan tool dedicated to garage mechanics or a test tool dedicated to assembly plant agents)

Diagnostic data

data that is located in the memory of an electronic control unit which may be inspected and/or possibly modified by the tester (diagnostic data includes analogue inputs and outputs, digital inputs and outputs, intermediate values and various status information)

Examples of diagnostic data include vehicle speed, throttle angle, mirror position, system status, etc.

Architecture – Contents of Software Layer

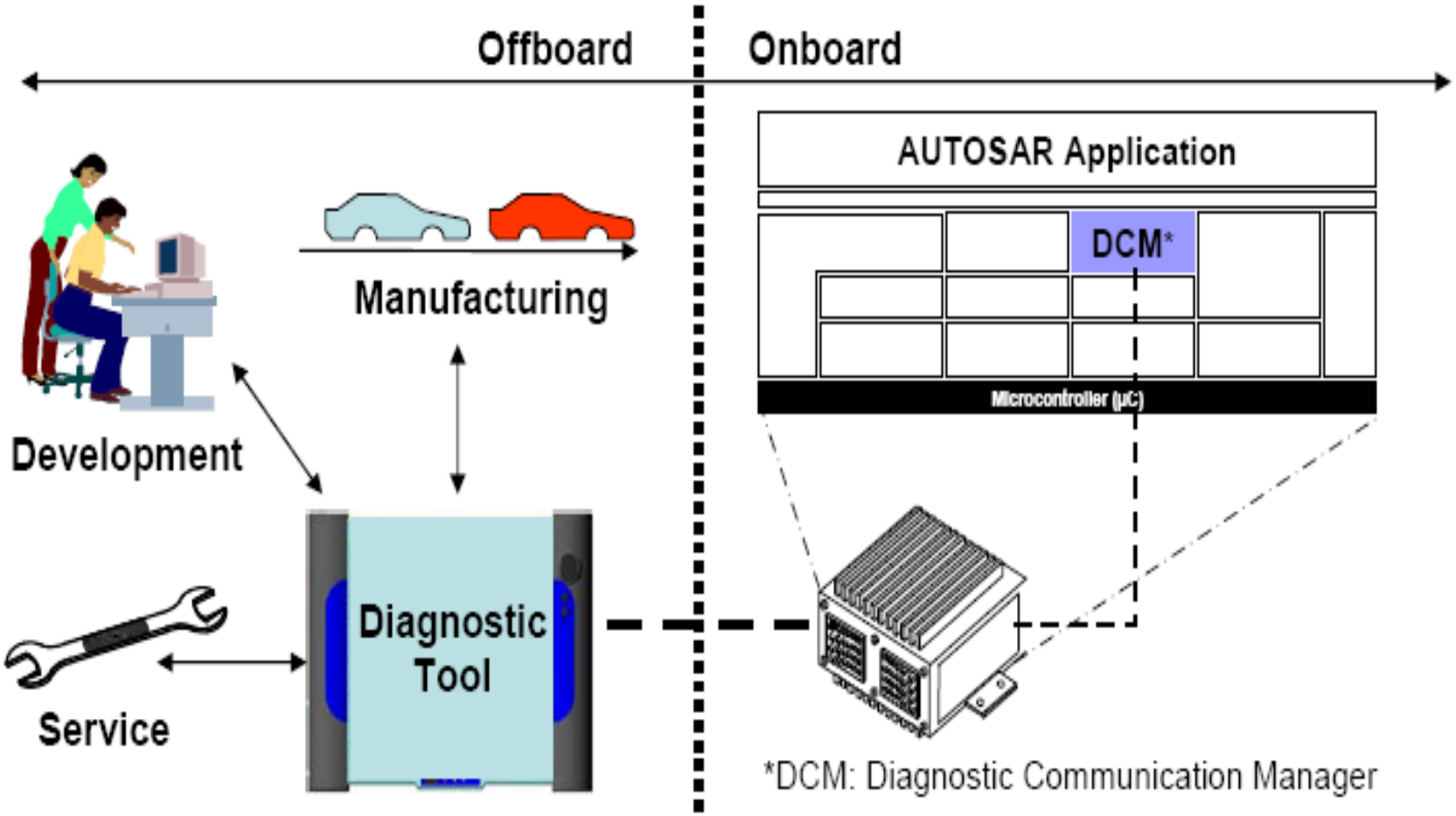


- The **Debugging** module supports debugging of the AUTOSAR BSW. It interfaces to ECU internal modules and to an external host system via communication .
- The **Diagnostic Event Manager** is responsible for processing and storing diagnostic events (errors) and associated FreezeFrame data.
- The module **Diagnostic Log and Trace** supports logging and tracing of applications. It collects user defined log messages and converts them into a standardized format.
- The **Diagnostic Communication Manager** provides a common API for diagnostic services
- The **Function Inhibition Manager** is responsible for providing a control mechanism for software components and the functionality
- The **Network layer** part of diagnostics are handled in respective communication stack (i.e. CAN TP, LIN TP, FR TP)

Why DCM?

- ✓ The functionality of the DCM module is used by external diagnostic tools during the development, manufacturing or service
- ✓ The DCM module ensures diagnostic data flow and manages the diagnostic states, especially diagnostic sessions and security states
- ✓ The DCM module checks if the diagnostic service request is supported and if the service may be executed in the current session according to the diagnostic states.

Overview of the communication between the external diagnostic tools and the onboard AUTOSAR Application



DCM Overview with AUTOSAR

- ✓ In the AUTOSAR architecture, the Diagnostic Communication Manager module is located in a 'Communication services' (Service Layer).
- ✓ The Dcm Interfaces with communication modules to transfer information from an ECU to an off-board Tester and vice-versa.
- ✓ UDS Service
  - ➢ This refers to a UDS Service as defined in ISO14229-1
  - ➢ Note: All UDS Service defined in ISO14229-1 is not supported by the Dcm module
- ✓ OBD Service
  - ➢ This refers to an OBD Service as defined in ISO15031-5
  - ➢ Note: All UDS OBD Service as defined in ISO15031-5 is not supported by the Dcm module

Diagnostics Communication Manager

- ✓ The functionality of Dcm module is used by external diagnostic tools e.g. in the development, manufacturing or service
- ✓ Diagnostic services include functions such as test, inspection, monitoring or diagnosis of on-board vehicle servers (ECU).
- ✓ The information transferred might consist of ECU fault information, parameter reporting, control routines, calibration values and other types of data used in performing ECU diagnostics.
- ✓ The Dcm module is network independent. All the network specific functionality (the Specifics of networks like CAN, LIN, FlexRay or MOST) is handled outside of the Dcm.
- ✓ The PDU Router (PduR) module provides a network-independent interface to the DCM module.
- ✓ The Dcm requests data or functional state from SW-Cs via port interfaces or from other BSW modules through direct function calls if any of the data elements or functional states cannot be provided by the Dcm module itself

Unified diagnostic services (UDS) – Supported (R4.0.3)

The DSP provides the following diagnostic UDS services:

- DiagnosticSessionControl (0x10)
- ECUReset (0x11)
- SecurityAccess (0x27)
- CommunicationControl(0x28)
- TesterPresent (0x3E)
- ControlDTCSetting (0x85)
- ResponseOnEvent (0x86)
- LinkControl (0x87)
- ReadDataByIdentifier (0x22)
- ReadMemoryByAddress (0x23)
- ReadScalingDataByIdentifier (0x24)
- ReadDataByPeriodicIdentifier (0x2A)
- DynamicallyDefineDataIdentifier (0x2C)
- WriteDataByIdentifier (0x2E)
- ClearDiagnosticInformation (0x14)
- ReadDTCInformation (0x19)
- InputOutputControlByIdentifier (0x2F)
- RoutineControl (0x31)
- RequestDownload (0x34 hex)
- RequestUpload (0x35)
- TransferData (0x36 hex)
- RequestTransferExit (0x37)

OBD service supported – Supported (R4.0.3)

The DSP provides the following diagnostic OBD services:

- Relevant OBD Service Identifier
- Request current powertrain diagnostic data ($01)
- Request current powertrain freeze frame data ($02)
- Request confirmed emission related DTCs ($03)
- Clear/Reset emission related diagnostic information ($04)
- Request monitoring test results for specific monitored systems ($06)
- Request pending emission related DTCs ($07)
- Request control of on-board system, test or component ($08)
- Request vehicle information and IUMPR ($09)
- Request emission-related DTCs with permanent status ($0A)

Diagnostics Event Manager

- ✓ The Diagnostic Event Manager (Dem) handles and stores the events detected by diagnostic monitors in both Software Components (SW-Cs) and Basic software (BSW) modules
- ✓ The stored event information is available via an interface to other BSW modules or SW-Cs
- ✓ The Dem module uses the EventId to manage the status of the ´Diagnostic Event´ of a system and performs the required actions for individual test results
- ✓ E.g. stores the freeze frame(If qualified as Failed)
- ✓ The Dem module shall represent each Diagnostic Event by an EventId and the related EventName.
- ✓ All monitors and BSW modules use the EventId as a symbolic EventName.
- ✓ The EventId and the related EventName shall be unique per Dem module represented by the ECU configuration

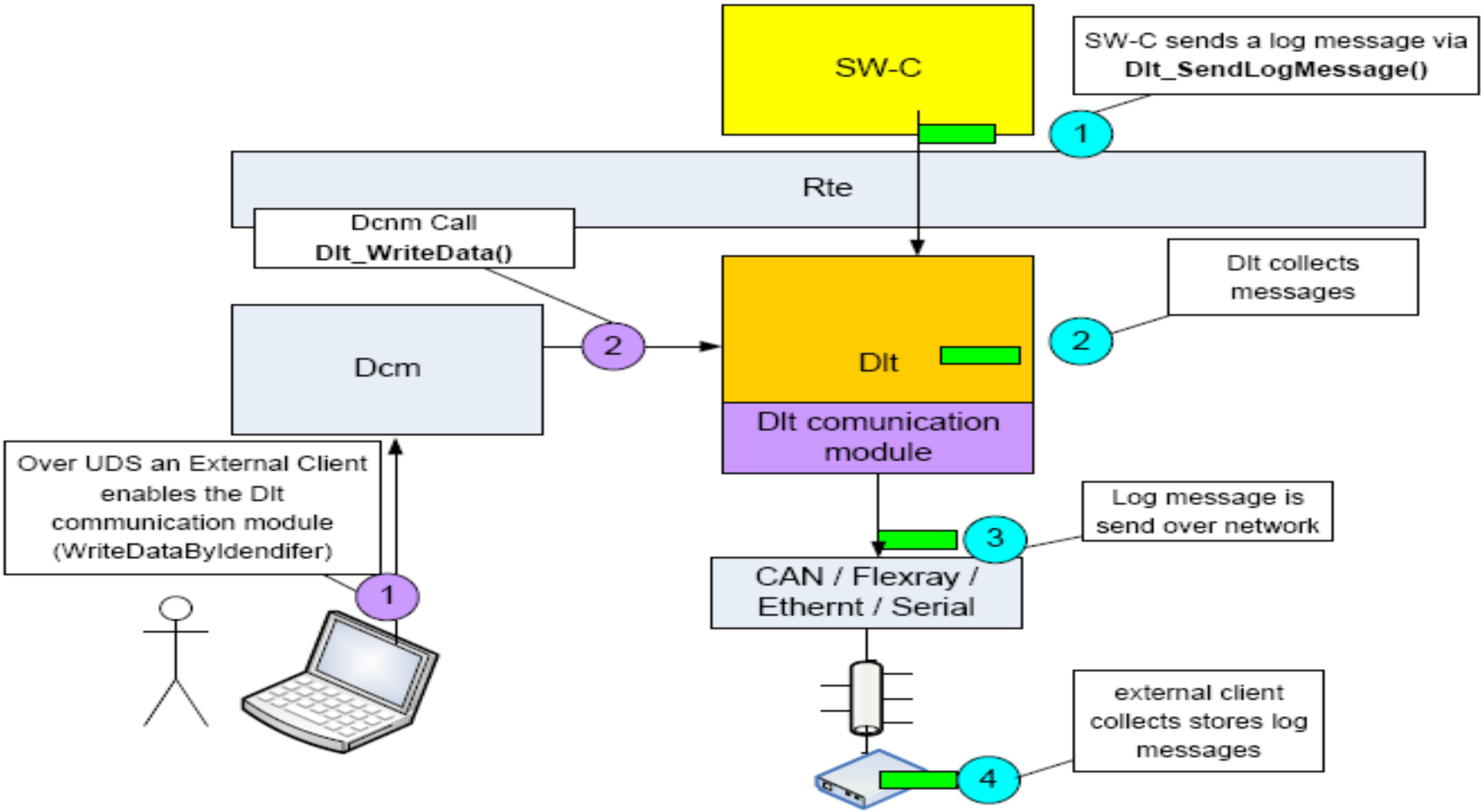
Diagnostic Log and Trace

- Each application software component (SW-C) can report errors to the Dlt

- Collects log messages and converts them into a standardized format. The Dlt forwards the data to the Dcm or a CDD which uses a serial interface for example

- Logging
  - ✓ logging of errors, warnings and info messages from AUTOSAR SW-Cs, providing a standardized AUTOSAR interface
  - ✓ gathering all log and trace messages from all AUTOSAR SW-Cs in a centralized AUTOSAR service component (Dlt) in the BSW
  - ✓ logging of messages from Det and
  - ✓ logging of messages from Dem
- Tracing
  - ✓ RTE activities
- Control
  - ✓ individual log and trace messages can be enabled/disabled and
  - ✓ Log levels can be controlled individually by back channel
- Generic
  - ✓ Dlt is available during development and production phase
  - ✓ access over standard diagnosis or platform specific test interface is possible and
  - ✓ security mechanisms to prevent misuse in production phase are provided

Diagnostic Log and Trace



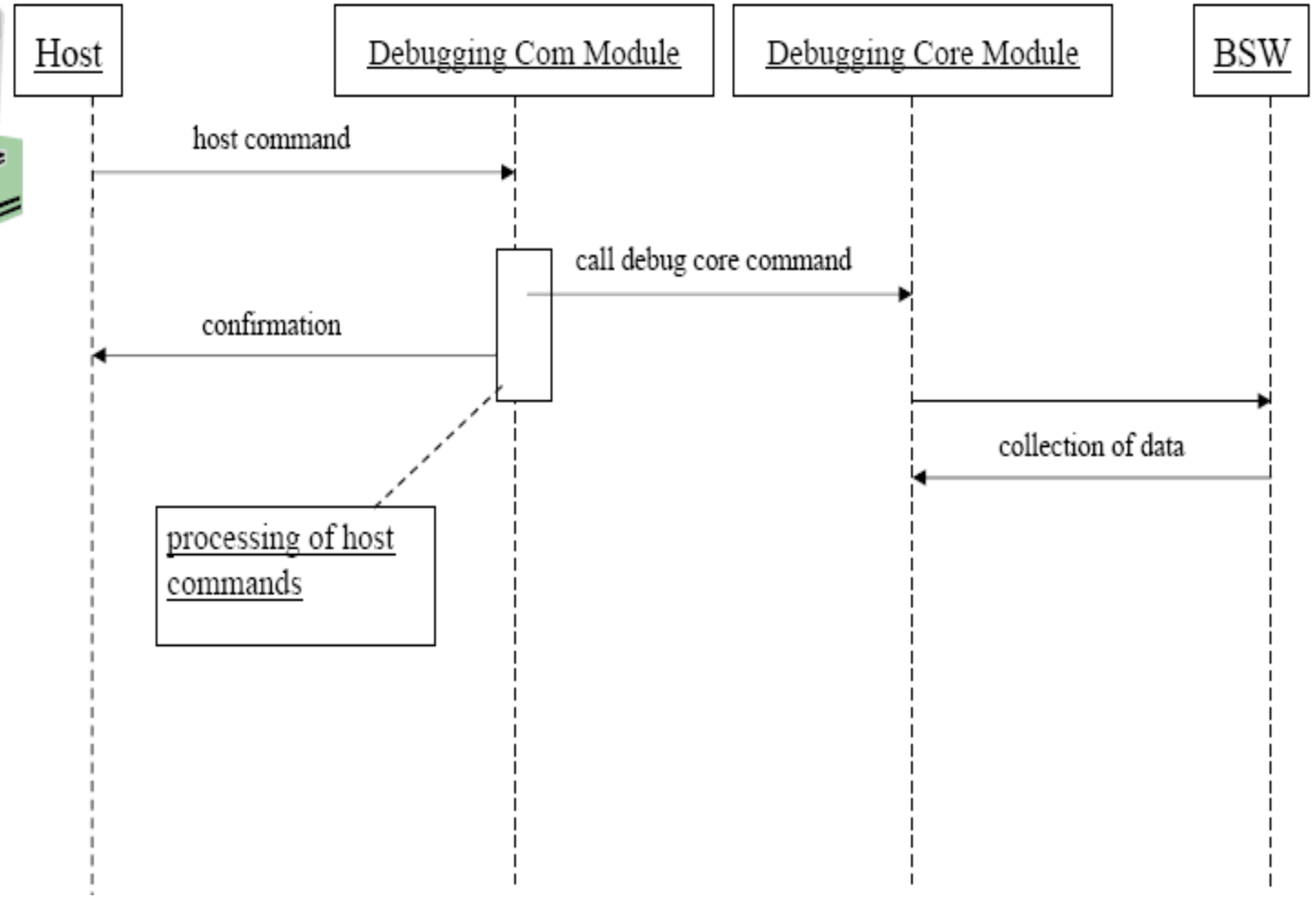
Debugging Module

- To support a user (system integrator or BSW developer) during development, in case the basic software does not behave as expected
- Collects as much information as possible about the runtime behavior of the systems without halting the processor
- Data is transmitted to an external host system via communication, to enable the user to identify the source of a problem
- An internal buffer is provided to decouple data collection from data transmission
- Main tasks of the Debugging Module are to
  - ✓ Collect and store data for tracing purposes
  - ✓ Collect and immediately transmit data to host
  - ✓ Modify data in target memory on host request
  - ✓ Transmit stored data to host
  - ✓ Accept commands to change the behavior of the Debugging Module

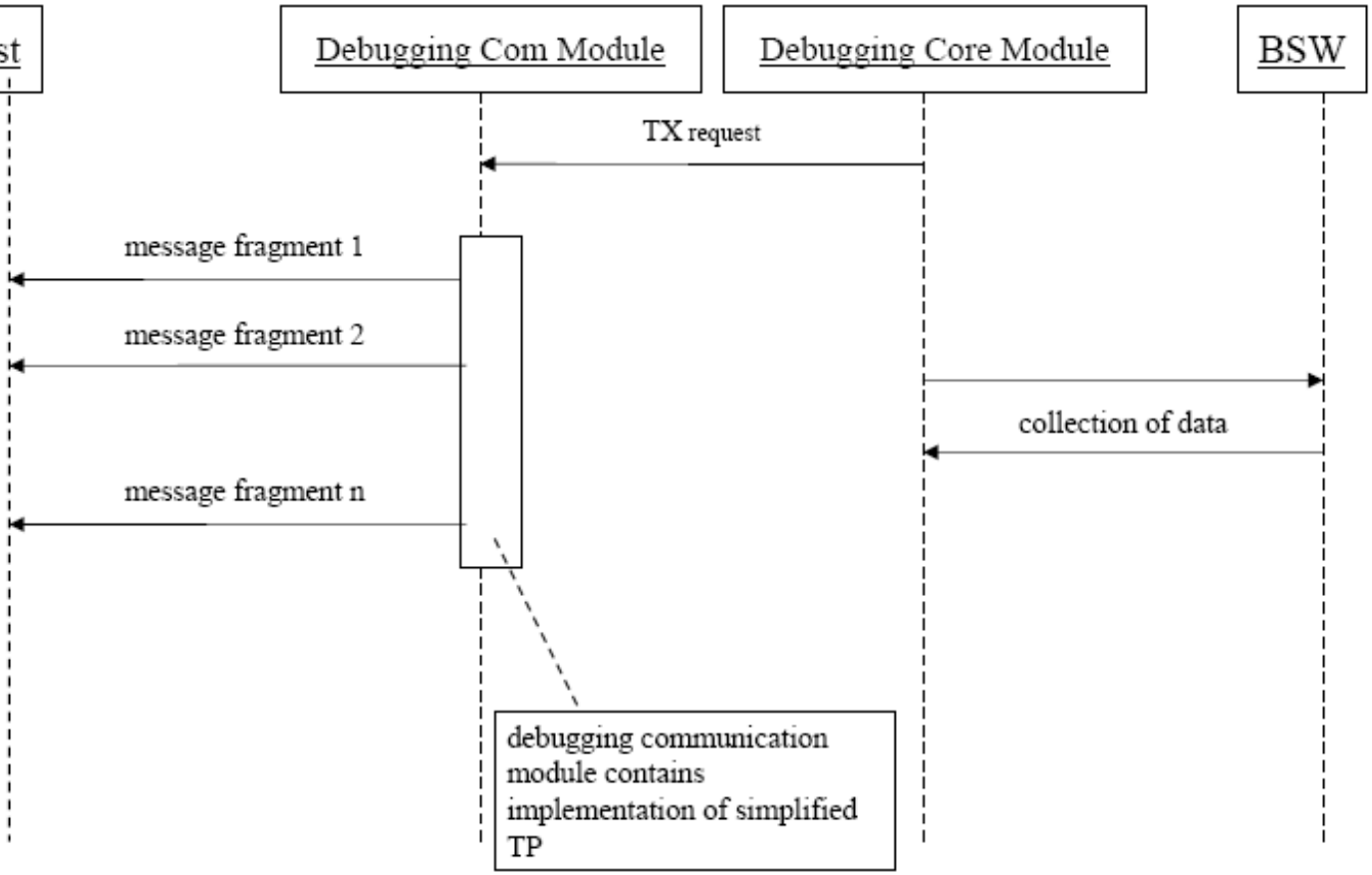
Debugging Module use cases

- Tracing Principles
  - ✓ Tracing means collecting information from running software. To perform tracing, the software is not halted
- Tracing of variables
  - ✓ Cyclically
  - ✓ on event
- Tracing of functions
  - ✓ In order to implement tracing of functions, the code has to be instrumented with calls to Collect Function Tracing Information (function entry) API when the function is entered and calls to Collect Function Tracing Information (function exit) API before leaving the function
- Tracing of SW-C
  - ✓ The debugging module offers functions to the RTE to trace information from the following RTE events
    - ➢ signal transmission
    - ➢ signal reception
    - ➢ COM callback
    - ➢ start runnable
    - ➢ terminate runnable

Debugging - Data Flow



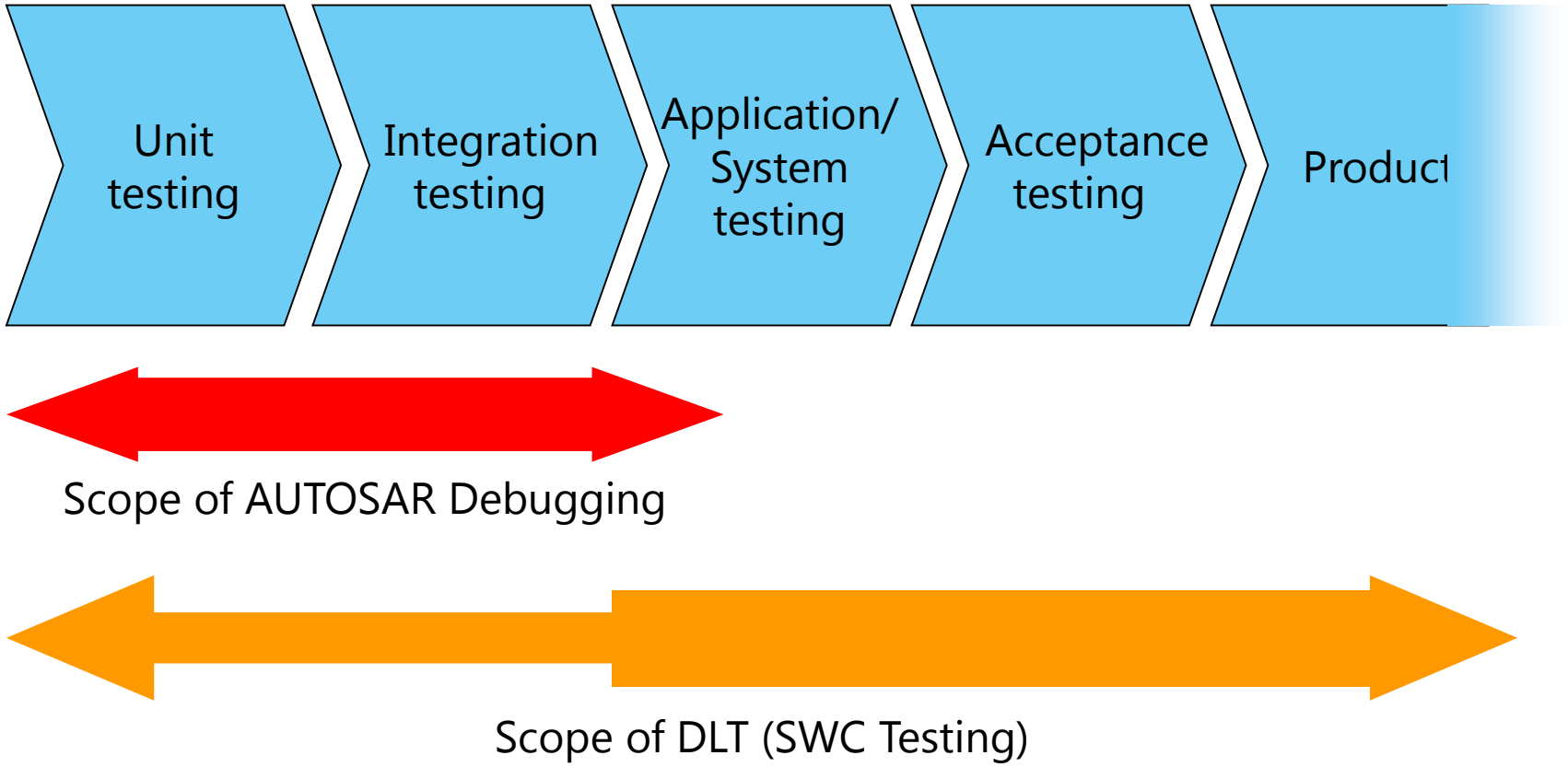
Debugging - Data Flow



Strategy to identify data

- The debugging core module identifies data by debugging identifiers (DIDs)
- Debugging identifiers are statically configured and need to be known to the debugging module and the host
- Two kinds of DIDs which can be used:
  - Standard debugging identifiers
  - Debugging identifiers without static address information

DLT and Debugging modules life span



Thank you

www.kpit.com

 /kpit  /kpit  @kpit  /kpit