# SVENSK STANDARD
# SS-ISO 22900-2:2009

Fastställd/Approved: 2009-03-23
Publicerad/Published: 2009-04-23
Utgåva/Edition: 1
Språk/Language: engelska/English
ICS: 43.040.15

## Vägfordon – Modulärt gränssnitt för fordonskommunikation (MVCI) –
## Del 2: D-PDU API för diagnostik (ISO 22900-2:2009, IDT)

## Road vehicles – Modular vehicle communication interface (MVCI) –
## Part 2: Diagnostic protocol data unit application programming interface (D-PDU API)
## (ISO 22900-2:2009, IDT)



SWEDISH STANDARDS INSTITUTE

# Hitta rätt produkt och ett leveranssätt som passar dig

Standarder
Genom att följa gällande standard både effektiviserar och säkrar du ditt arbete. Många standarder ingår dessutom ofta i paket.

Tjänster
Abonnemang är tjänsten där vi uppdaterar dig med aktuella standarder när förändringar sker på dem du valt att abonnera på. På så sätt är du säker på att du alltid arbetar efter rätt utgåva.
e-nav är vår online-tjänst som ger dig och dina kollegor tillgång till standarder ni valt att abonnera på dygnet runt. Med e-nav kan samma standard användas av flera personer samtidigt.

Leveranssätt
Du väljer hur du vill ha dina standarder levererade. Vi kan erbjuda dig dem på papper och som pdf.

Andra produkter
Vi har böcker som underlättar arbetet att följa en standard. Med våra böcker får du ökad förståelse för hur standarder ska följas och vilka fördelar den ger dig i ditt arbete. Vi tar fram många egna publikationer och fungerar även som återförsäljare. Det gör att du hos oss kan hitta över 500 unika titlar. Vi har även tekniska rapporter, specifikationer och "workshop agreement". Matriser är en översikt på standarder och handböcker som bör läsas tillsammans. De finns på sis.se och ger dig en bra bild över hur olika produkter hör ihop.

Standardiseringsprojekt
Du kan påverka innehållet i framtida standarder genom att delta i någon av SIS ca 400 Tekniska Kommittéer.

# Find the right product and the type of delivery that suits you

Standards
By complying with current standards, you can make your work more efficient and ensure reliability. Also, several of the standards are often supplied in packages.

Services
Subscription is the service that keeps you up to date with current standards when changes occur in the ones you have chosen to subscribe to. This ensures that you are always working with the right edition.
e-nav is our online service that gives you and your colleagues access to the standards you subscribe to 24 hours a day. With e-nav, the same standards can be used by several people at once.

Type of delivery
You choose how you want your standards delivered. We can supply them both on paper and as PDF files.

Other products
We have books that facilitate standards compliance. They make it easier to understand how compliance works and how this benefits you in your operation. We produce many publications of our own, and also act as retailers. This means that we have more than 500 unique titles for you to choose from. We also have technical reports, specifications and workshop agreements. Matrices, listed at sis.se, provide an overview of which publications belong together.

Standardisation project
You can influence the content of future standards by taking part in one or other of SIS's 400 or so Technical Committees.

Den internationella standarden ISO 22900-2:2009 gäller som svensk standard. Detta dokument innehåller den officiella engelska versionen av ISO 22900-2:2009.

The International Standard ISO 22900-2:2009 has the status of a Swedish Standard. This document contains the official English version of ISO 22900-2:2009.



Upplysningar om sakinnehållet i standarden lämnas av SIS, Swedish Standards Institute, telefon 08-555 520 00.
Standarder kan beställas hos SIS Förlag AB som även lämnar allmänna upplysningar om svensk och utländsk standard.

Information about the content of the standard is available from the Swedish Standards Institute (SIS), tel +46 8 555 520 00.
Standards may be ordered from SIS Förlag AB, who can also provide general information about Swedish and foreign standards.

SIS Förlag AB, SE 118 80 Stockholm, Sweden. Tel: +46 8 555 523 10. Fax: +46 8 555 523 11.
E-mail: sis.sales@sis.se Internet: www.sis.se

# Contents

Page

# Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 22900-2 was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 3, *Electrical and electronic equipment*.

ISO 22900 consists of the following parts, under the general title *Road vehicles — Modular vehicle communication interface (MVCI)*:

— *Part 1: Hardware design requirements*

— *Part 2: Diagnostic protocol data unit application programming interface (D-PDU API)*

— *Part 3: Diagnostic server application programming interface (D-Server API)*

# Introduction

ISO 22900 is applicable to vehicle electronic control module diagnostics and programming.

This part of ISO 22900 was established in order to more easily exchange software and hardware of vehicle communication interfaces (VCIs) among diagnostic applications. It defines a generic and protocol independent software interface towards the modular vehicle communication interface (MVCI) protocol module, such that a diagnostic application based on this software interface can exchange the MVCI protocol module or add a new MVCI protocol module with minimal effort. Today, the automotive after market requires flexible usage of different protocol modules for vehicles of different brands. Many of today's protocol modules are incompatible with regard to their hardware and software interface, such that, depending on the brand, a different protocol module is required.

The objective of this part of ISO 22900 is to specify the diagnostic protocol data unit application programming interface (D-PDU API) as a generic software interface, and to provide a “plug and play” concept for access onto different MVCI protocol modules from different tool manufacturers. The D-PDU API will address the generic software interface, the protocol abstraction, its exchangeability as well as the compatibility towards existing standards such as SAE J2534-1 and RP1210a.

The implementation of the modular VCI concept facilitates co-existence and re-use of MVCI protocol modules, especially in the after market. As a result, diagnostic or programming applications can be adapted for different vehicle communication interfaces and different vehicles with minimal effort, thus helping to reduce overall costs for the tool manufacturer and end user.

Vehicle communication interfaces compliant with ISO 22900 support a protocol-independent D-PDU API as specified in this part of ISO 22900.

# Road vehicles — Modular vehicle communication interface (MVCI) —

## Part 2: Diagnostic protocol data unit application programming interface (D-PDU API)

## 1 Scope

This part of ISO 22900 specifies the diagnostic protocol data unit application programming interface (D-PDU API) as a modular vehicle communication interface (MVCI) protocol module software interface and common basis for diagnostic and reprogramming software applications.

This part of ISO 22900 covers the descriptions of the application programming interface (API) functions and the abstraction of diagnostic protocols, as well as the handling and description of MVCI protocol module features. Sample MVCI module description files accompany this part of ISO 22900.

Migration from and to the existing standards SAE J2534-1 and RP1210a is addressed. This part of ISO 22900 contains a description of how to convert between the APIs. Corresponding wrapper APIs accompany this part of ISO 22900.

The purpose of this part of ISO 22900 is to ensure that diagnostic and reprogramming applications from any vehicle or tool manufacturer can operate on a common software interface, and can easily exchange MVCI protocol module implementations.

## 2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 9141-2, *Road vehicles — Diagnostic systems — Part 2: CARB requirements for interchange of digital information*

ISO 14229-1, *Road vehicles — Unified diagnostic services (UDS) — Part 1: Specification and requirements*

ISO 14230 (all parts), *Road vehicles — Diagnostic systems — Keyword Protocol 2000*

ISO 15031-5, *Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 5: Emissions-related diagnostic services*

ISO 15765 (all parts), *Road vehicles — Diagnostics on Controller Area Networks (CAN)*

ISO 22901-1, *Road vehicles — Open diagnostic data exchange (ODX) — Part 1: Data model specification*

ISO/IEC 8859-1, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*

## 3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

**3.1**
**application**
way of accessing the diagnostic protocol data unit application programming interface (D-PDU API)

NOTE From the perspective of the D-PDU API, it does not make any difference whether an application accesses the software interface directly, or through an MVCI D-Server. Consequently, in this part of ISO 22900, the term “application” represents both ways of accessing the D-PDU API.

**3.2**
**ComLogicalLink**
logical communication channel towards a single electronic control unit (ECU) or towards multiple electronic control units

**3.3**
**COMPARAM-SPEC**
protocol-specific set of predefined communication parameters (ComParams), the value of which can be changed in the context of a layer or specific diagnostic service

NOTE This part of the model can also contain OEM-specific ComParams.

**3.4**
**ComPrimitive**
smallest aggregation of a communication service or function

EXAMPLE A request message to be sent to an ECU.

**3.5**
**Ethernet**
physical network media type

## 4 Symbols and abbreviated terms

| | |
|---|---|
| API | Application Programming Interface |
| ASCII | American Standard for Character Information Interchange |
| CAN | Controller Area Network |
| CDF | Cable Description File |
| CLL | ComLogicalLink |
| ComParam | Communication Parameter |
| COP | Communication Primitive |
| CRC | Cyclic Redundancy Check |
| DLC | Data Link Connector |
| DLL | Dynamic Link Library |
| D-PDU | Diagnostic Protocol Data Unit |
| D-Server | Diagnostic Server |
| ECU | Electronic Control Unit |

| | |
|---|---|
| HDD | Hard Disk Drive |
| HI | Differential Line — High |
| HW | Hardware |
| IEEE 1394 | Firewire serial bus |
| IFR | In-Frame Response |
| IGN | Ignition |
| IOCTL | Input/Output Control |
| K | UART K-Line |
| KWP | Keyword Protocol |
| L | UART L-Line |
| LOW | Differential Line — Low |
| MDF | Module Description File |
| MVCI | Modular Vehicle Communication Interface |
| ODX | Open Diagnostic Data Exchange |
| OEM | Original Equipment Manufacturer |
| OSI | Open Systems Interconnection |
| PC | Personal Computer |
| PCI | Protocol Control Information |
| PGN | Parameter Group Number |
| PROGV | Programmable Voltage |
| PWM | Pulse Width Modulation |
| RDF | Root Description File |
| RX | UART uni-directional receive |
| SCI | Serial Communications Interface |
| SCP | Standard Corporate Protocol |
| TX | UART uni-directional transmit |
| USB | Universal Serial Bus |
| USDT | Unacknowledged segmented data transfer[1)] |
| UUDT | Unacknowledged un-segmented data transfer[2)] |

1) ISO 15765-2 network layer includes protocol control information for segmented data transmission, which results in a maximum of 7 data bytes for normal addressing and 6 data bytes for extended addressing.

2) Single CAN frames do not include protocol control information, which results in a maximum of 8 data bytes for normal addressing and 7 data bytes for extended addressing.

| | |
|---|---|
| VCI | Vehicle Communication Interface |
| VPW | Variable Pulse Width |
| XML | Extensible Markup Language |

# 5 Specification release version information

## 5.1 Specification release version location

Specification release version information is contained in each modular VCI release document specification under the same clause title “Specification release version information”. It is important to check for feature support between modular VCI release specifications if the most recent API features shall be implemented. The D-PDU API supports the reading of version information by the API function call PDUGetVersion.

Release version information is also contained in the following files:

— root description file (RDF);

— module description file (MDF);

— cable description file (CDF);

— D-PDU API library file.

## 5.2 Specification release version

The specification release version of this part of ISO 22900 is: 2.2.0.

# 6 Modular VCI use cases

## 6.1 OEM merger

In the past, several OEMs in the automotive industry have merged into one company.

All companies try to leverage existing (legacy) components and jointly develop new products, which are common across different vehicle types and badges.

If OEMs already had modular VCI compliant test equipment, it would be simple to connect MVCI protocol modules from merged OEMs into one chassis or device. All protocols would be supported by a single MVCI protocol module configuration without any replacement of MVCI protocol module hardware at the dealer site. The same applies for engineering and some of this concept might also work for production plants (end of line).

## 6.2 OEM cross vehicle platform ECU(s)

OEMs specify requirements and design electronic systems to be implemented in multiple vehicle platforms in order to avoid re-inventing a system for different vehicles. The majority of design, normal operation, and diagnostic data of an electronic system are re-used if installed in various vehicles. The engineering development centres are located worldwide. A great amount of re-authoring of diagnostic data is performed to support different engineering test tools.

Providing diagnostic data in an industry standard format like ODX and XML will avoid re-authoring into various test tool specific formats at different system engineering locations. The D-PDU API supports this re-use concept by fully abstracting vehicle protocols into the industry supported ComParam descriptions.

### 6.3 Central source diagnostic data and exchange during ECU development

Single source origin of diagnostic data (as depicted in Figure 1 — Example of central source engineering diagnostic data process), combined with a verification and feedback mechanism and distribution to the end users, is highly desirable in order to lower engineering expenses. Engineering, manufacturing, and service specify which protocol and data shall be implemented in the ECU. This information will be documented in a structured format like XML. Furthermore, the same structured data files can be used to setup the diagnostic engineering tools to verify proper communication with the ECU and to perform functional verification and compliance testing of the ECU. Once all quality goals are met, these structured data files shall be released to the OEM database. An Open Diagnostic data eXchange (ODX) schema has been developed for the purpose of supporting these structured formatted files used for ECU diagnosis and validation.

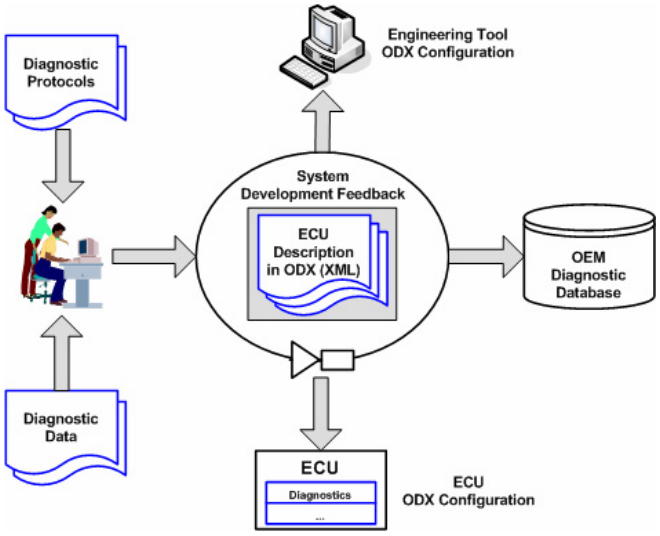



**Figure 1 — Example of central source engineering diagnostic data process**

### 6.4 OEM franchised dealer and aftermarket service outlet diagnostic tool support

The service shop uses the modular VCI hardware and software for vehicle diagnosis and enhanced procedural testing. By using the same engineering, manufacturing, and service functions as those used for individual ECU testing, the reliability of the data is maintained. A modular VCI protocol module can be used with any PC (handheld or stationary) and can be utilised as an embedded device.

## 7 Modular VCI software architecture

### 7.1 Overview

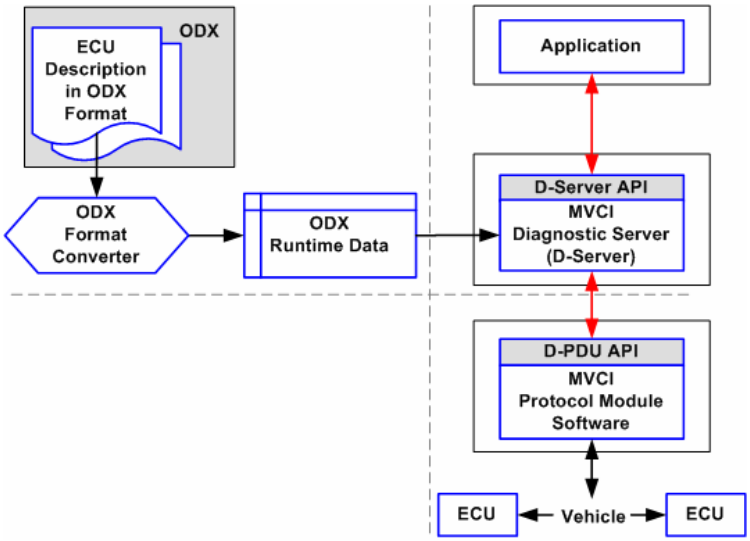
The modular VCI concept is mainly based on three software components (see Figure 2 — MVCI software architecture):

— MVCI D-Server software;

— runtime data based on ODX;

— MVCI protocol module software.

The application accesses the MVCI D-Server through the MVCI D-Server API. The D-Server obtains all required information about the ECU(s) out of the ODX runtime data. Using the ODX runtime data information, the D-Server converts the application's request into a byte stream, which is called a diagnostic protocol data unit (D-PDU). The D-PDU is handed over to the MVCI protocol module through the D-PDU API. The MVCI protocol module transmits the D-PDU to the vehicle's ECU(s). The other way around, the MVCI protocol module receives the vehicle's response(s) and reports the response data to the D-Server. Again using the ODX runtime data, the D-Server interprets the D-PDU and provides the interpreted symbolic information to the application.




NOTE The grey shading of symbols indicates reference to the following International Standards:

— ODX: ISO 22901-1;

— D-Server API: ISO 22900-3;

— D-PDU API: ISO 22900-2;

— MVCI protocol module: ISO 22900-1.

**Figure 2 — MVCI software architecture**

## 7.2 Modular VCI D-Server software

The MVCI D-Server is accessible through the MVCI D-Server API. By accessing this API, the application may browse the available features for each ECU and initiate a request towards an ECU using simple symbolic expressions. If the request requires input parameters, they can be specified using symbolic expressions as well. The MVCI D-server takes the symbolic request, including input parameters, and converts them into a diagnostic request message as defined at the protocol level. The diagnostic request message represents the diagnostic protocol data unit (D-PDU) as passed to the MVCI protocol module through the D-PDU API. Vice versa, the D-Server converts diagnostic response messages as retrieved from the MVCI protocol module back to symbolic information and provides it to the application.

For a detailed description and the complete MVCI D-Server API definition, see ISO 22900-3.

## 7.3 Runtime format based on ODX

For every conversion from symbolic requests to diagnostic request messages, and vice versa for responses, the MVCI D-Server obtains the required information from the runtime database. The database defines the structure of every diagnostic request and response as supported by an ECU. The database defines byte and bit positions, width, and type of every input and output parameter.

Even though the MVCI D-Server obtains its information from a runtime database, the runtime database and format are not specified by the MVCI standard. Instead, the MVCI standard defines an exchange format to import and export the ECU description across OEMs and tool suppliers. The runtime format is left up to the system designers.

The exchange format is called open diagnostic data exchange format (ODX format). For a detailed description, see ISO 22901-1.

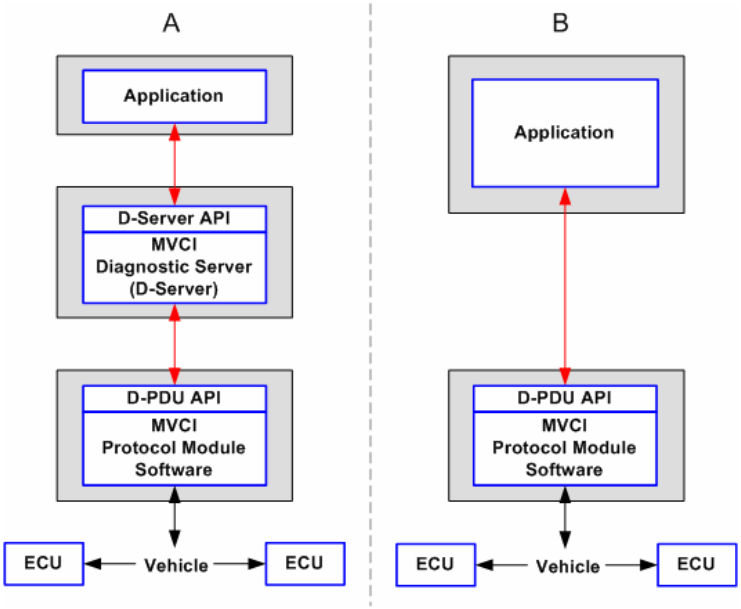
## 7.4 MVCI protocol module software

The MVCI protocol module is accessible through the D-PDU API. The application issues diagnostic requests through the D-PDU API. The MVCI protocol module takes the request D-PDU and transmits it to the vehicle's ECU(s) according to the vehicle communication protocol. Header type, checksum information, and D-PDU segmentation depend on the vehicle communication protocol, and shall be handled transparently by the MVCI protocol module. Also, the MVCI protocol module observes the timing between message frames and requests and responses on the physical interfaces. After completion, the MVCI protocol module simply has to deliver the response back to the application or report an error condition.

## 7.5 MVCI protocol module configurations

The D-PDU API and MVCI protocol modules work in many configurations. A MVCI D-Server is not required as the application interface to the D-PDU API.

Figure 3 — MVCI configurations shows two such configurations to point out the differences.




**Key**

A application with MVCI D-Server

B application without MVCI D-Server

NOTE From the perspective of the D-PDU API, it does not make a difference whether an application accesses the software interface directly, or through an MVCI D-Server. Consequently, in this part of ISO 22900, the term “application” represents both ways of accessing the D-PDU API.

**Figure 3 — MVCI configurations**

# 8 D-PDU API use cases

## 8.1 Overview

The MVCI protocol module is the key component to exchange implementations of diagnostic protocols among OEMs and tool suppliers without re-engineering already implemented software. By relying on the D-PDU API, the application may easily access other or additional MVCI protocol module implementations. In a similar way to existing standards like SAE J2534-1 and RP1210a, applications compliant to the MVCI standard are basically independent of the underlying device as long as the required physical interface is supported and no tool supplier specific feature is required.

Even though the D-PDU API extends the capabilities beyond the definitions of SAE J2534-1 and RP1210a, the existing standards and their related devices and applications do not become obsolete by introducing the D-PDU API. Instead, the transition and co-existence of all standards are facilitated to save development and investment costs. The definition of the D-PDU API is closely related to SAE J2534-1 and RP1210a to allow mapping of functionality in both directions. However, it extends their definitions to cover the full width of enhanced diagnostics.

The fulfilment of the following use cases is crucial for the inter-exchange of protocol module implementations according to MVCI, SAE J2534-1 and RP1210a.

NOTE In the use case figures below, the grey boxes suggest a specific software component architecture. This representation is not intended to be construed as the only possible architectural solution. Depending on the situation, there can be more software components, or fewer software components.

## 8.2 Use case 1: Single MVCI protocol module

The single MVCI protocol module configuration (see Figure 4 — MVCI configuration with single MVCI protocol module) is the simplest configuration where the D-PDU API implementation and the MVCI protocol module hardware are obtained from the same vendor. The application will access the single MVCI protocol module through a single D-PDU API. Parallel access onto multiple D-PDU APIs is not required. Resource handling is completely covered inside the D-PDU API implementation.

This use case applies to basically all stand-alone MVCI protocol module device configurations.

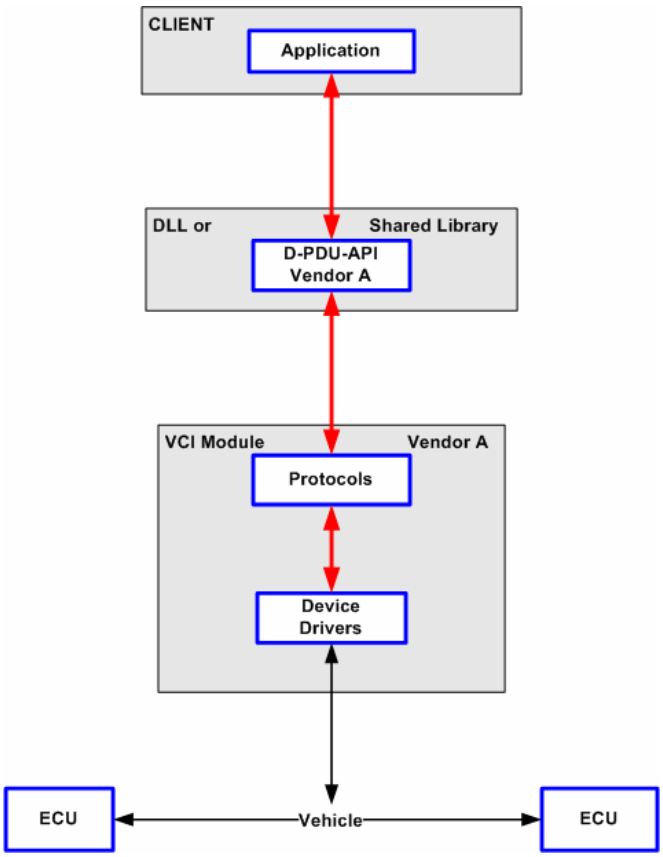



**Figure 4 — MVCI configuration with single MVCI protocol module**

## 8.3 Use case 2: Multiple MVCI protocol modules supported by same D-PDU API implementation

There are different configurations with multiple MVCI protocol modules. In this use case, a D-PDU API implementation may support more than one MVCI protocol module at a time, where both modules and D-PDU API implementations are from a single vendor (see Figure 5 — Multiple MVCI protocol modules supported by same D-PDU API implementation). The application will access the MVCI protocol modules through a single D-PDU API. Parallel access onto multiple D-PDU APIs is not required. However, the application may access and operate the MVCI protocol modules at the same time in parallel if the D-PDU API implementation provides the capabilities. Resource handling is completely covered inside the D-PDU API implementation.

This use case applies to MVCI protocol module device configurations where the vendor integrates support for multiple MVCI protocol modules into one software package.