

CEN/CLC/JTC 22/WG 3 "Quantum Computing and Simulation"

Convenor: Paul Alexandra Mme



HAL_requirements_Draft01

Document type	Related content	Document date	Expected action
Project / Draft		2026-01-12	COMMENT/REPLY by 2026-02-11

Description

Dear members,

Please find attached the first draft of the HAL WI.

This will be discussed during the WG 3 meeting on February 11th.

Kind regards,

CEN/TC XXX

Date: 20YY-XX

prEN XXXXX:20YY

Secretariat: XXX

**JTC 22 WG3 Quantum Computing
Hardware Abstraction Layer
Functional requirements
Draft 01, 2026-01-12**

CCMC will prepare and attach the official title page.

Contents		Page
European foreword.....		3
Introduction.....		4
1	Scope.....	5
2	Normative references.....	5
3	Terms and definitions.....	6
3.1	6	
3.2	6	
4	Abbreviations.....	6
5	Overview.....	7
5.1	Organisation of qubits.....	7
5.2	Concept of native and primitive gates.....	7
5.3	Concept of measurement.....	7
5.4	Interfacing considerations.....	7
6	Instruction Set.....	8
7	Libraries.....	8
7.1	Library for Fault Tolerant Quantum Computing (FTQC).....	8
7.2	Library with Optimised Circuits.....	8
7.3	Library for Mapping and Routing.....	8
8	Virtualisation and resource management.....	8
8.1	Multi-User Task Scheduling and Cooperative Execution.....	8
8.2	Partitioned Sequences and Instruction-Flow Boundaries.....	8
Bibliography.....		9

European foreword

This document (prEN XXXX:20YY) has been prepared by Technical Committee CEN/TC JTC22/WG3 “Quantum Computing and simulation”, the secretariat of which is held by XXX.

This document is currently submitted to the CEN Enquiry/Formal Vote/Vote on TS/Vote on TR.

This document has been prepared under a Standardization Request given to CEN by the European Commission and the European Free Trade Association, and supports essential requirements of EU Directive(s) / Regulation(s).

Introduction

A hardware abstraction layer (HAL) is a software layer within a quantum computing software stack to enable higher-level software and compilers to operate independently of hardware-specific details. This TS defines functional descriptions and functional requirements for the HAL, while specific implementation details and values remain out of scope. The HAL resides between the assembly layer and the control software layer, as described in TR 18202:202 on “Layer Models for Quantum Computing”. Its purpose is to provide a standardised interface that abstracts hardware-specific details away and to expose essential capabilities to higher layers. This capability enables portability and interoperability across diverse quantum architectures.

The HAL is designed to support a wide range of quantum hardware platforms, each with unique characteristics such as qubit topology, native gate sets, and error correction schemes. Its functionality is plug-in extendable via libraries to accommodate future developments. Rather than prescribing strict implementation details, this TS focuses on defining the functional roles of the HAL, including instruction translation, resource management, fault tolerant quantum computing, mapping and routing, virtualisation for multi-user environments, and reporting of hardware capabilities. These functionalities allow compilers and programming frameworks to optimise execution without requiring direct interaction with proprietary hardware interfaces.

By establishing these functional requirements, the HAL becomes a cornerstone for modular quantum computing systems, ensuring flexibility, scalability, and vendor-neutral integration.

1 Scope

This TS describes the functionalities of the hardware abstraction layer (HAL) for use in the software stack of quantum computers, as described in TR 18202:202 on “Layer Models for Quantum Computing”. The present TS is focussed on a high-level functional description of the HAL while additional details on implementation and interfacing are reserved for other future CEN/Ts.

The word “functional” means within this context that a precise definition of implementation, interfaces, information flows (via instructions, commands, signals, etc.) and values is out of scope. This TS concentrates therefore on general descriptions of what the HAL supports and which properties or quantities are to be considered for future specification. It offers mainly an enumeration of characteristics that are considered as relevant as well as a motivation why they are relevant.

The aim of the Hardware Abstraction Layer is to inform higher layers with capabilities and limitations supported by the underlying hardware. It hides many implementation-specific details, and provides a more unified interface to higher layers. This includes instruction translation, resource management, fault tolerant quantum computing, mapping and routing, virtualisation for multi-user environments, and reporting of hardware capabilities.

Not all quantum computers make use of the same paradigm. Annealing quantum computers behave differently from gate-based quantum computers, and therefore their HALs might behave different as well. The HAL can therefore provide information about the underlying architecture, such as for instance being “gate-based”, “annealing” or “simulation”. When applicable, the HAL can also select between multiple quantum architectures and even partition a single task into multiple queues to perform calculations in parallel on multiple architectures.

In order to speed-up progress, the first version of this TS may put a focus on gate-based quantum computer, but that does not exclude functional requirements for other architectures. If needed this first version may leave details of those architectures for future revisions of this TS.

2 Normative references

- CEN/CLC/TR 18202:2025, "*Layer model of Quantum Computing*", sept 2025

3 Terms and definitions

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

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp/>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1

ISA (Instruction Set Architecture)

a lower-level method of defining operations on a quantum computer.

Note 1: Instead of defining specific gates, this method defines gates (or other instructions) as operations, using pulses pulsed for a certain time, on specific qubits.

3.2

Universal gate-based quantum computing

a quantum computer being capable of processing an arbitrary quantum circuit.

Note 1: A universal gate-based quantum computer ought to have a gate set which is universal. A gate set is said to be universal if any unitary operation may be approximated to arbitrary accuracy by a quantum circuit involving only those gates [2]. The definition also comprises non-fault-tolerant universal quantum computers, which can process an arbitrary quantum circuit reliably only up to a certain length, size or gate count.

4 Abbreviations

ISA – Instruction Set Architecture

HAL – Hardware Abstraction Layer

FTQC - Fault Tolerant Quantum Computing

QEC – Quantum Error Correction

5 Overview

The hardware abstraction layer ...bla bla

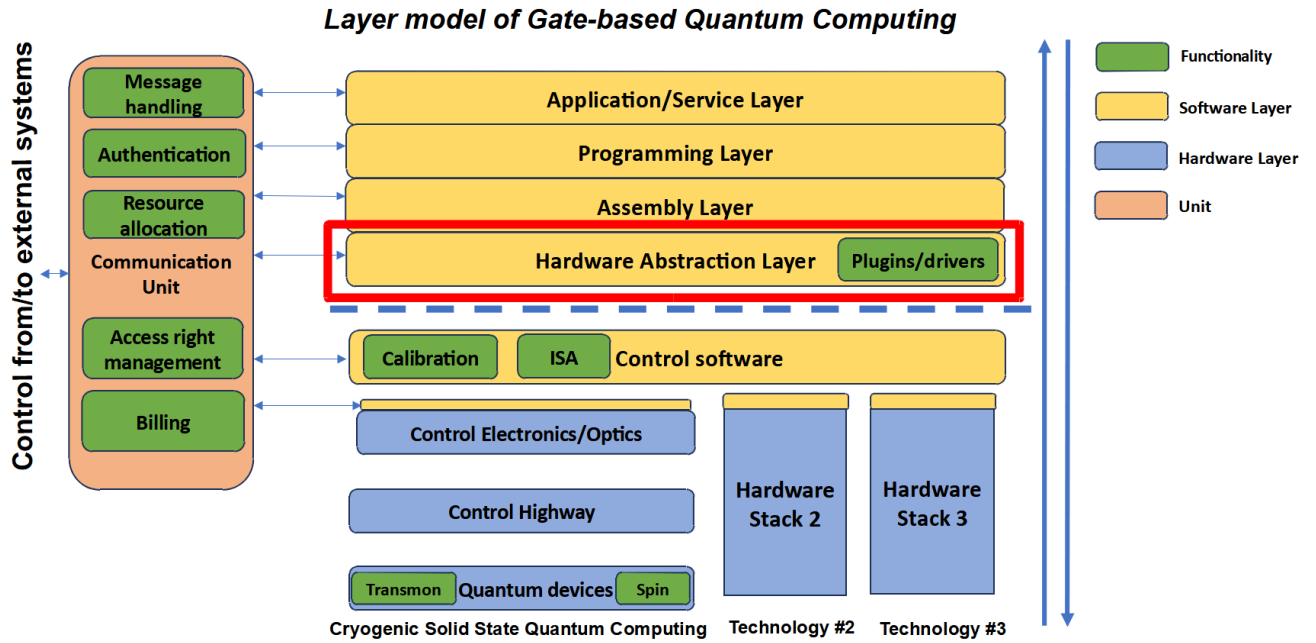


Figure 1 - Overview of the layer model of quantum computing. Only the layer within the red box is within the scope of this document.

Figure 1 shows bla bla

Editor's Note:

This chapter can reuse a lot of the text created for the (accepted) paper about the "Layer Model", which in turn is an improvement of the originating text about the HAL in TR 18202:2025

5.1 Organisation of qubits

5.2 Concept of native and primitive gates

5.3 Concept of measurement

5.4 Interfacing considerations

6 Instruction Set

Editor's Note:

This chapter can have a lot of similarities with the ISA description in the project on "Cryogenic Solid State Quantum Computing"

7 Libraries

Editor's Note:

The libraries use the Dynamic Linking mechanisms offered by the underlying operating system. The use of such "plug-in" libraries enable future extension of functionalities or insertion of solutions from other vendors.

7.1 Library for Fault Tolerant Quantum Computing (FTQC)

7.2 Library with Optimised Circuits

7.3 Library for Mapping and Routing

8 Virtualisation and resource management

8.1 Multi-User Task Scheduling and Cooperative Execution

8.2 Partitioned Sequences and Instruction-Flow Boundaries

Bibliography

- [1] *A first reference*