Issue Date:	**2021-08-18**	**CEN-CLC/FGQT**	**N113a**
Deadline:	**n/a**	**Supersedes:**	**N113**
Status:	FOR APPROVAL		

Focus Group on Quantum Technologies

TITLE	**Use case: Cloud-based quantum computing**
PROJECT	FGQT Roadmap
REFERRING TO	N020f
SOURCE	Niels Neumann (TNO)
CONTACT	niels.neumann@tno.nl

Supported by:

- Michele Amoretti (CINI Consortium)
- Pim Venderbosch (QuiX Quantum)
- Rob van den Brink (Delft Circuits)

ABSTRACT

Explanation of changes

This document outlines the examples of functionalities users of cloud-based quantum services might request from different viewpoints. Additionally, the impact for future cloud-based quantum services are identified based on these user requests. We furthermore propose to remove Annex D *Cloud-based quantum computing interfaces* of the FGQT roadmap N020f, as the proposed changes below replace the intended content of Annex D.

Why are the changes needed?

It is expected that quantum computers will be made available via cloud mostly, at least for the near-term. There are already some quantum cloud-services available, such as those offered by IBM, Rigetti, D-Wave, Xanadu and others. However, these are still limited in available functionalities. Furthermore, there is already much knowhow about classical cloud-services and many of them are already available. As a result, quantum cloud-service can be set-up with the knowledge of classical cloud-services. Furthermore, there should be a close link between classical cloud-services and quantum cloud-services, as most algorithms will be hybrid, where only part of the operations are performed on a quantum device, with classical intermediate processing.

At this moment, the FGQT roadmap N020f is lacking text on cloud-based quantum computing interfaces, other than a largely empty Annex D.

This contribution provides texts that explains some of the requirements and functionalities required from such hosts. It presents first the classical setting and subsequently it introduces extensions to the classical cloud-services with quantum capabilities. Finally, some specific aspects of quantum cloud-services are discussed.

This contribution is related to section 11.1.2.1 *Use Case: Using a Quantum Computer as Secondary Processor in the Cloud* of the roadmap document. That section contains a specific use case for quantum computing, whereas this proposed contribution considers cloud-based quantum computing from a more general perspective from a user.

Instructions for editor

This document should be added to N020f as subsection 11.1.2.6 in the section on use-cases. Furthermore, Annex D *Cloud-based quantum computing interfaces* can be removed from of the FGQT roadmap N020f.

Instructions for the Editor: Remove *Annex D: Cloud-based quantum computing interfaces* from N020f.

Instructions for the Editor: The text below has to be inserted in N020f as new Section 11.1.2.6.

11.1.2.6 Use case: Cloud-based quantum computing

This use case serves to identify possible desirable functionalities of future quantum computers from a user perspective, with quantum computers hosted in a cloud-environment. This use case hence also helps in identifying possible requirements for cloud-based quantum computing. It relies on common developments for digital computing, and may also fulfil important needs for end-users of future quantum computers. As such, this use case has the potential to become the most dominant business-case of quantum computing in the future.

11.1.2.6.1 Current setting of classical cloud-services

Cloud computing is an all-purpose term for offering a flexible solution to storage, computing power and software. Instead of buying a fixed amount of resources, users can opt for resources for a limited time only. Various types of contracts are possible, for instance, a single hour of computations, or the possibility to flexibly use a software package for a period of three years. Therefore, the structure and way of working of cloud-functionalities are in essence similar to those of local hardware and software. Many companies are already migrating or have already migrated some of their processes to public clouds, as overall, cloud-services require lower upfront investments and can reduce the ICT total-cost of ownership.

In general, three types of cloud-services are proposed: Infrastructure as a Service (Iaas), Platform as a Service (PaaS) and Software as a Service (SaaS), with increasing functionalities offered by the cloud-provider. In the first, IaaS, the hardware is provided, typically this includes computers and data storage. In PaaS, the cloud-provides also takes care of the infrastructure, often including the operating system. Finally, in SaaS, a software package or license is provided by the cloud-provider. A key difference between PaaS and SaaS is that PaaS provides a platform to develop software, whereas SaaS offers the software itself.

There are other types of cloud-services located between the three basic types mentioned above. An example is Container as a Service (CaaS), located between IaaS and PaaS, where services are offered as containers. This is automatically taken care of in PaaS.

Note that each of these services also comes with limitations and potential risks: security and vendor lock-in are two of them. Using cloud-services is therefore often a trade-off between cost, performance, flexibility and security aspects.

11.1.2.6.2 Extending cloud-services with quantum capabilities

Classical cloud-services have already been around for a long time and have gained significant attention the last few years. Quantum cloud-services on the other hand have been around only shortly and are still emerging.

When considering the three types of cloud-services of the previous section, we mainly see PaaS being available now. The first platforms that offer IaaS or SaaS are currently upcoming. Below we will sketch the situation for each of the three services in the quantum context:

- IaaS: Users are given access to a quantum computer and have complete control over which hardware fundamental operations are applied to which quantum states at what time. This in general requires sufficient knowledge of the quantum device and possibly requires calibration of it.

  - Note that though possible, quantum computer providers will in general be reluctant to opening up their devices to IaaS users. This service is likely most relevant for specific academic contexts or for use by employees of quantum computer providers.
- PaaS: Users are given access to a quantum computer and can program it using a set of operations, which might be parametrized. For gate-based quantum devices this can for instance imply that users have access to the qubits on a quantum device and they may manipulate them using simple rotations around certain axes or a controlled-phase gate between two qubits. The operations can also be more complex, such as a quantum Fourier transform. The user should not be bothered by mapping these instructions to the quantum hardware, i.e., quantum compiling is performed automatically by the system.
- SaaS: Users are given algorithm implementations on quantum hardware. Users are only bothered with specific problem instances they want to solve. Users might be given the option to tweak the algorithm implementation, which in some cases is even a necessity (consider for instance the oracle implementation in Grover's algorithm). The implementations given to the users can already be optimized for specific quantum devices or based on specific metrics.

Note that the descriptions above work equally well for a network of quantum computers, instead of an individual quantum computer. In IaaS in a distributed setting, users are given the flexibility how to correspond between which computers in what way. For PaaS, users are flexible in assigning which operations are performed on which device and which non-local operations (between quantum states on different devices) are performed. The way how these are implemented is taken care of for the user. This is not yet applicable for current hardware.

Finally, in hybrid quantum-classical settings, where only part of the computations are performed on a quantum computer, there has to be a seamless integration between the quantum devices and classical (cloud) computers that perform the classical operations and instantiate new quantum operations. Most current quantum-cloud services require the classical computations to be performed on a local device, instead of in the cloud, which might result in communication overhead. This hybrid quantum-classical setting can even be integrated in SaaS, for instance with the classical computations to update the quantum parameters being done automatically by the system in variational quantum eigensolvers.

11.1.2.6.3 Desirable functionalities for quantum cloud-services

From a user point-of-view, there are various functionalities that allow for a sufficient adoption of quantum cloud-computing and an integration with classical functionalities. Below we list a few, including some suggestions on how to implement this. Before we present these functionalities, we first give some assumptions on quantum computing:

- There will only be a few quantum computer providers. Their quantum devices can be made available via a self-hosted platform or via a third-party cloud-platform;
- The number of quantum resources will remain limited. Similarly, will the quality and the fidelity of these quantum resources be limited. Error correction will help suppress errors, but will not yet result in fault-tolerant quantum computing. This point will likely hold especially for the near-future;
- Quantum devices can be linked together via a quantum network, possibly via fibre optic cables;
- Mid-circuit measurements and subsequent classically controlled operations are possible;
- There will not be a single agreed upon hardware technology. This holds for both the quantum paradigm (gate-based, photonics, quantum annealing, Hamiltonian simulation and others) and the specific implementation of it (for gate-based e.g. transmon or spin qubits);
- There will not be a single approach to quantum computing (e.g. gate-based quantum computing, quantum annealing or Hamiltonian simulation);

- Users of SaaS are in general only interested in the end-result of a quantum routine. This can be either the measurement results themselves or the interpretation thereof.

Some of the following functionalities relate to the early nature of quantum computing, where especially for the near-future, device characteristics including different noise rates are a key-part in determining the performance of a quantum device. Examples of functionalities which users might request are set out below and are formulated generally, and should be specified for different hardware backend technologies. Therefore: quantum cloud services could provide

- **Characterization of quantum devices:** Relevant properties of quantum devices should be characterized correctly such that users can make a conscious choice on which quantum backend to use. For gate-based devices, such properties include for instance the number of qubits, the qubit type, single qubit T1 and T2 times (average and worst-qubit values), topology of the qubits and fidelity of single and two-qubit gates.
    - Preferably these metrics should be presented in a uniform manner across different vendors.
- **User choice in computational device:** Users should have the option to choose the quantum backend on which the computations are run. This choice can be made based on the characteristics of the available quantum devices. If users do not have a preference, any available and sufficient device may be used for the computations.
    - Once quantum computing devices are sufficiently large, a single device may be portioned to provide multiple smaller devices to the users. This requires sufficient shielding between different parts.
- **Programming opportunities at different complexity levels:** Users should be able to program quantum algorithms on different levels of complexity, i.e., using different granularities. This means that the complexity of the quantum operations given by the user can differ from for instance elementary quantum operations to a higher-order routine, such as *Use Shor's algorithm to decompose this biprime* or *Solve this linear system of equations*.
- **A hybrid programming environment:** Users should be able to program their algorithms in a hybrid programming environment, which can be made available by the vendor. Users should be able to program their algorithm in the classical language of choice, with the option to call quantum subroutines. Results of these quantum subroutines can directly be used in the classical computations.
    - At least the most common programming languages should be supported in this hybrid programming environment, including Python and C++/C#
- **Appropriate resource allocation:** With only a limited number of quantum devices, allocating computation time across different users, but also maintenance and calibration time of the device self is an optimisation problem. There should be *service level agreements* between the quantum cloud-host and the users on which quantum devices can be used when.
    - Especially for IaaS users this is important, as such users have the most control over quantum devices and their fundamental operations.
- **Security of computations:** The vendor should provide sufficient shielding and isolation between the computations of different users, such that computations by one user are secure and inaccessible to other users.
    - Ideally computations of a user should also be secure against the cloud-provider. Blind quantum computation will offer this security.
- **Possibility to communicate with other devices:** The vendor should make it possible for the quantum device to communicate with other quantum devices, possibly hosted by other parties and/or used by other users.
    - This allows for some form of blind computation, where users cannot learn data of other parties, while still being able to do computations with them. Furthermore, quantum teleportation allows for transferring quantum states from one device to another.
- **Automatic quantum operation compiling routines:** Users should be able to have their quantum instructions automatically compiled to hardware-agnostic low level instructions or

even further to hardware-specific fundamental instructions. Users should also be able to provide such a compilation themselves.

- **An integration with classical cloud-processes:** The quantum device should be callable directly from a remote user, however, the quantum devices should also be able to communicate and interact with different classical computers hosted in the cloud.

To achieve the points listed above, it is important that various aspects work together seamlessly. Interfaces are a vital concept to achieve these requirements. Therefore, there should be interfaces between quantum computer providers and cloud-providers for a seamless interaction between different relevant components in cloud-based quantum computing. These interfaces furthermore allow modularity, where different parts are provided by different parties.

Below, in Figure X.1, an overview of different interfaces is presented. Note that where the distinction between local and cloud lies, depends in a large part on the services the user rented. Typically, in SaaS, this distinction lies at the User-Software interface, whereas for PaaS and IaaS this distinction is at the Software-Hardware interface. Note that different interfaces, with different functionalities, can both be Software-Hardware interfaces. See Figure X.2 as an example, the interface can be at the *Intermediate quantum instructions* separation, indicating that quantum instructions are presented at relatively high level and decompositions are performed on the host side. On the other hand, the separation can also be at the *Low-level quantum instructions* separation, where the quantum algorithm is sent to the host in terms of low-level hardware-agnostic quantum instructions, or even in terms of hardware-specific quantum instructions.

In Figure X.1, also an interface is required between quantum computers and cloud-hosted classical devices. Such an interface is fundamental for a sufficient adoption of quantum computing in complex business processes.

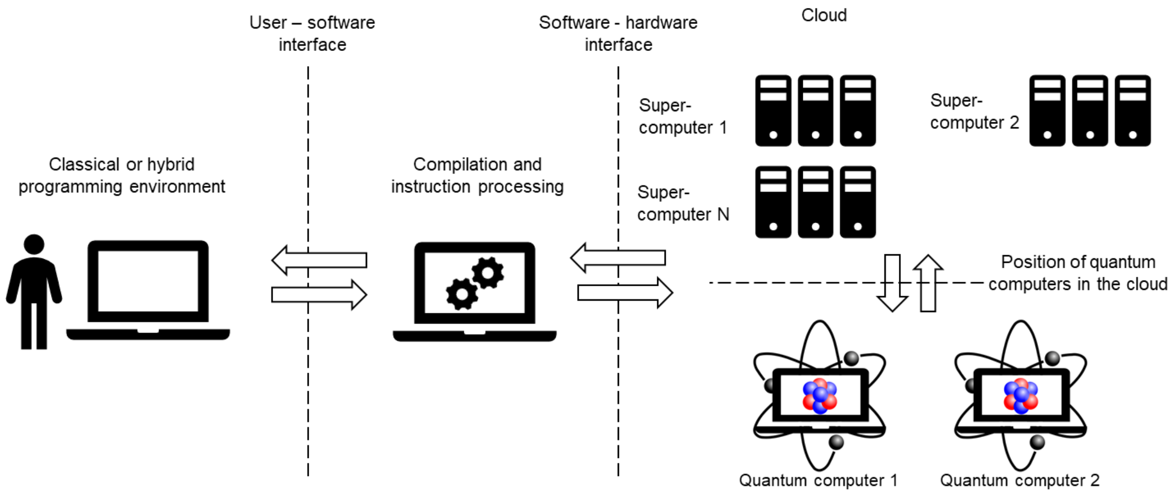

Figure X.1: An overview of different interfaces in a hybrid cloud setting. The separation between a local user an the cloud depends on the services agreed upon.

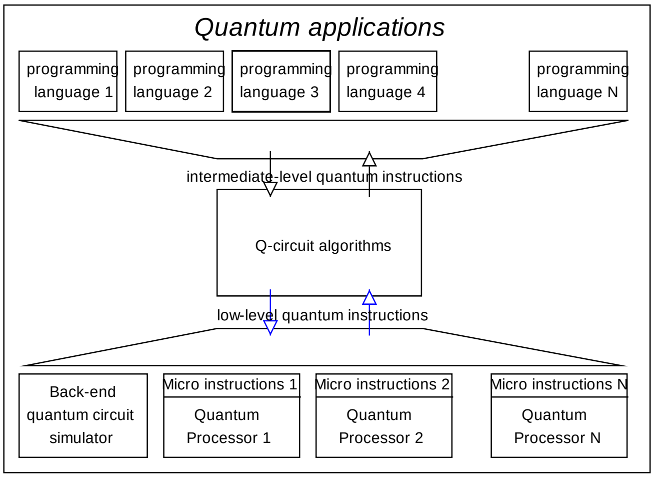

Figure X.2: A software stack for quantum computing (from: Van den Brink, Neumann, Phillipson, "Vision on Next Level Quantum Software Tooling", 2019).