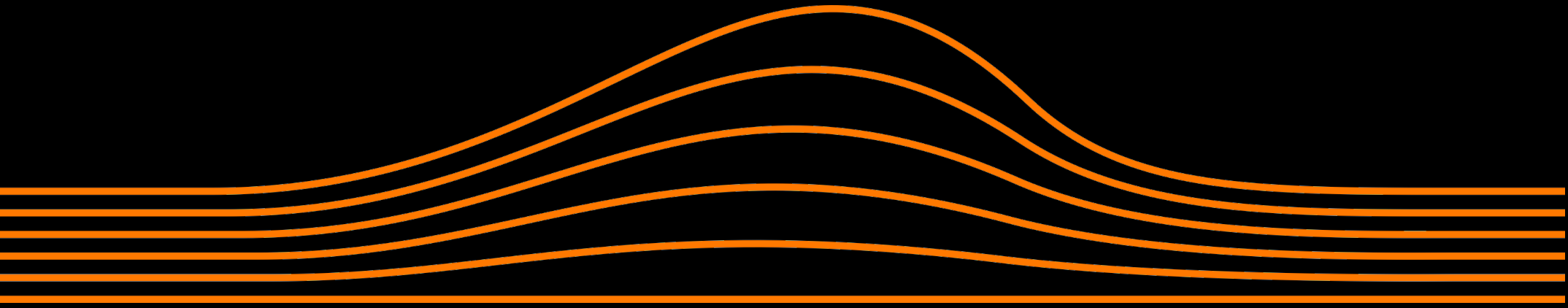
Orange Research

Shape the future and value innovation

October 15th, 2026

Classic-quantum hybrid cloud computing

Joël Penhoat – Orange Innovation





Classic-quantum hybrid cloud computing

We bring out a global view of the software infrastructure needed to support the deployment and orchestration of classic-quantum hybrid applications. This view comes from the Institute of Architecture of Application Systems (IAAS), University of Stuttgart.

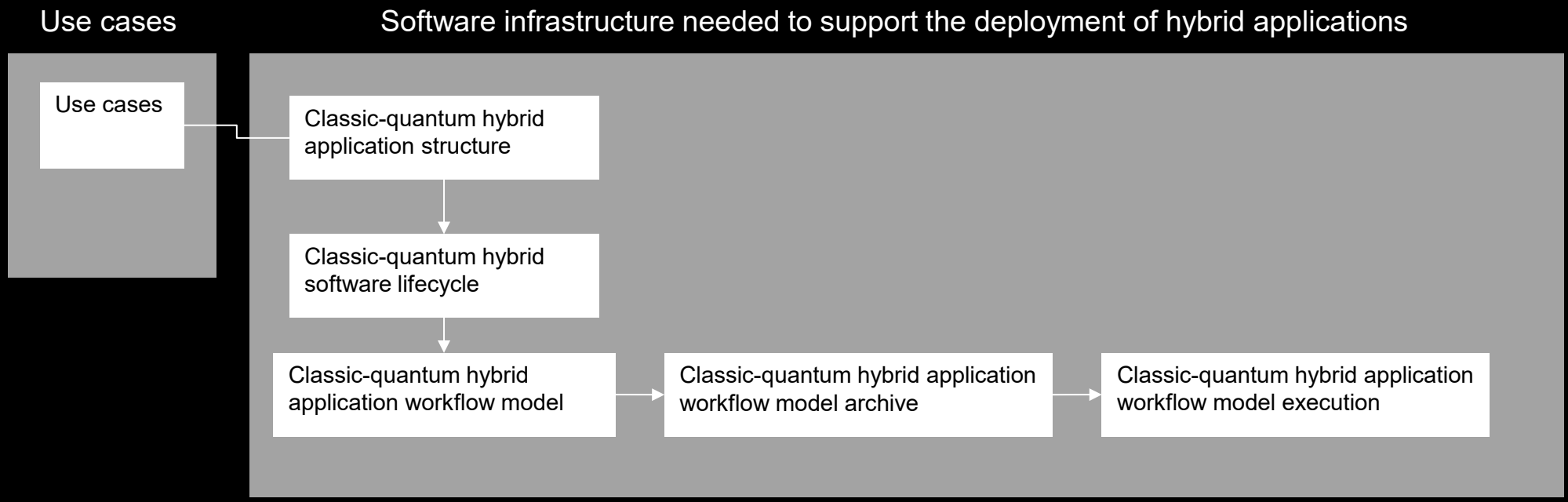
(i) the classic-quantum hybrid software lifecycle;

(ii) the classic-quantum hybrid application workflow model;

(iii) the classic-quantum hybrid application workflow model archive;

(iv) the classic-quantum hybrid application workflow model execution.

Classic-quantum hybrid cloud computing



Use cases @ Orange

Use cases in the NISQ era

At Orange Innovation the use cases currently studied – in the NISQ era - are optimization problems in network routing, network resilience, and network security.

Software infrastructure needed to support the deployment of hybrid applications

Classic-quantum hybrid application structure

Classic-quantum hybrid applications consist of interwoven classical and quantum tasks:

1) **Pre-processing tasks :** they are implemented by classical programs, executed on a classical computer; they include generating state preparation circuits based on input data to initialize the register of the quantum computer when executing the quantum programs;

2) **Quantum tasks :** they are executed on a quantum computer; they include preparing the required state in the register depending on the generated state preparation circuit, performing the unitary transformation specified by the quantum algorithm, measuring the output;

3) **Post-processing tasks :** they are executed on a classical computer; they include mitigating readout-errors in the result distribution, interpreting the results.

Classic-quantum hybrid software lifecycle

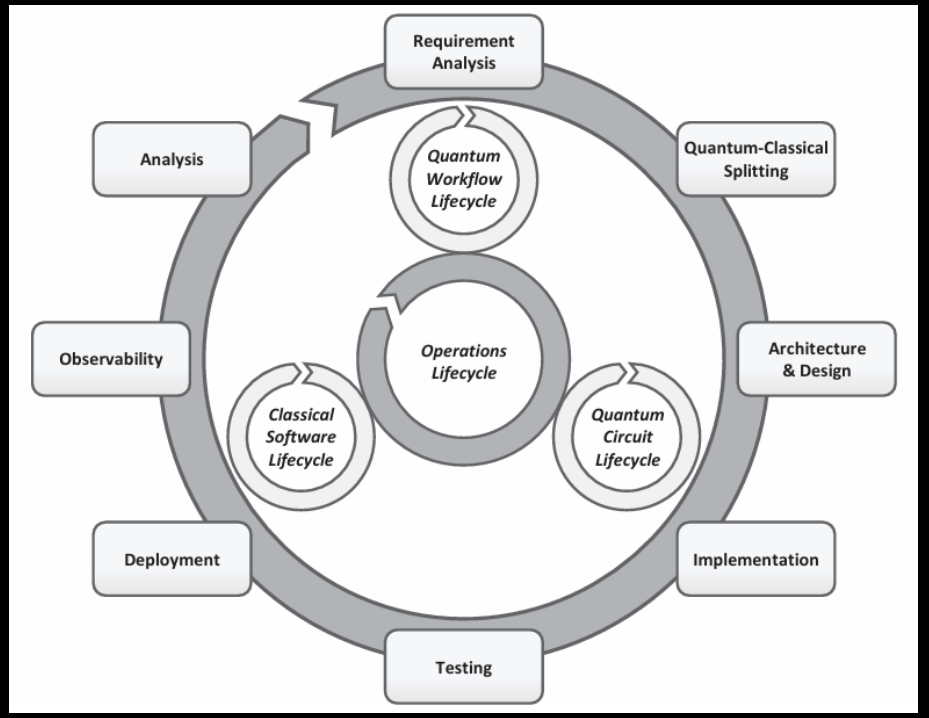
Classic-quantum hybrid software development lifecycle documents the process of designing, implementing, deploying, monitoring, and updating an application and its workflow model.

Requirement Analysis – it contains the functional and non-functional requirements of the hybrid classic-quantum software.

Analysis – it finds possible improvements of the application. After the analysis phase, the next iteration of the lifecycle can be entered.

Observability – the current state of the workflow and its execution environment are monitored by collecting data.

Deployment – the execution environment for the hybrid classic-quantum workflow is set-up.



Testing – the workflow and software artifacts are tested to verify their behavior according to the specified requirements.

Quantum-Classical Splitting – it determines which part of the application will run on a quantum computer or on a classical computer, and the programing models used by each part.

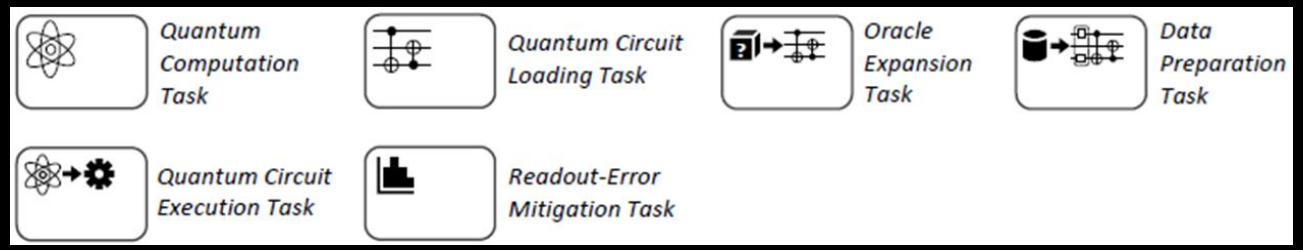
Architecture & Design – it specifies the application workflow, the software components, their functionalities, their interfaces, the data structure. The workflow specifications can be defined with BPMN (Business Process Model and Notation) extended for the quantum domain; the software specifications can be defined with UML (Unified Modeling Language) extended for the quantum domain.

Implementation – it includes the development of the workflow and the software artifacts.

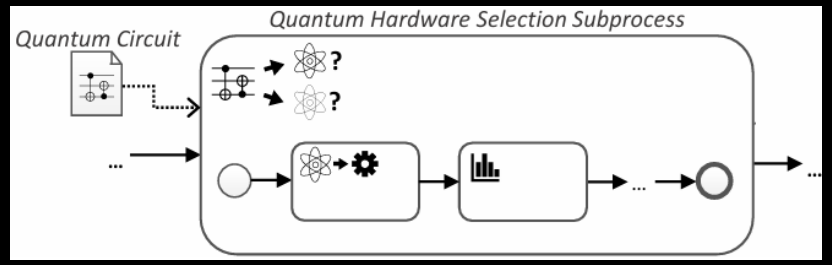
Classic-quantum hybrid application workflow model

The sequence of tasks in a hybrid workflow can be designed with quantum extensions of “Business Process Model and Notation” (BPMN) (ISO/IEC 19510 standard).

These quantum modeling extensions (QuantME) model classic-quantum tasks and their configuration attributes, quantum hardware selection subprocess, input-output data objects.

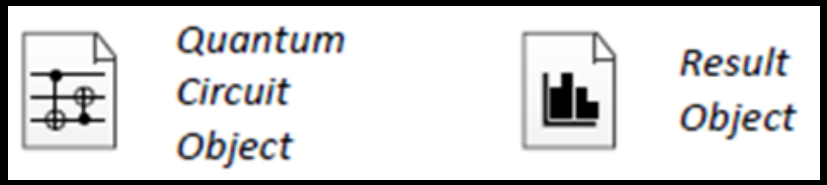


Quantum Workflow Task
it represents a unit of work performed by an automated system, or a human.



Quantum Hardware Selection Subprocess
it performs the selection of quantum hardware for a given quantum circuit, and the execution of all contained tasks according to this selection.

Quantum Circuit Object
It is an object for storing and transferring the data about a quantum circuit.



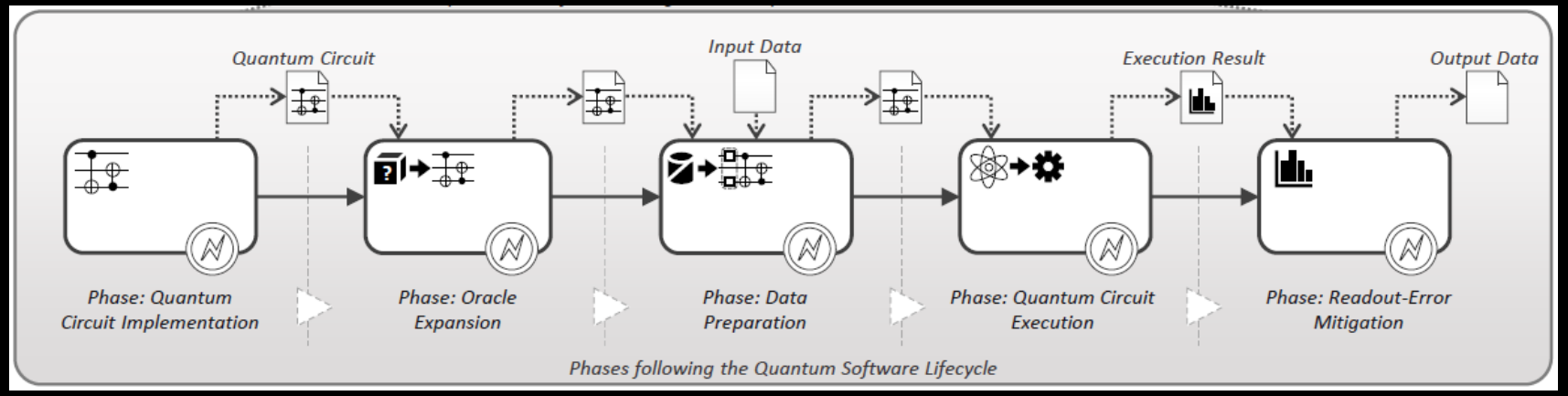
Result Object
it is an object for storing and transferring the data about results of quantum computations.

Classic-quantum hybrid application workflow model

BPMN Quantum Modeling Extension (QuantME) models the workflow as a sequence of tasks : (1) “quantum circuit loading task”, (2) “oracle expansion task”, (3) “data preparation task”, (4) “quantum circuit execution task”, (5) “readout error mitigation task”:

(2) Oracle Expansion task
it replaces the oracles contained in the quantum code by circuits implementing their functionalities. The task has three attributes: (i) OracleId to identify the oracle to replace; (ii) OracleURL to load from a repository the circuit to replace the oracle; (iii) OracleCircuit to directly specify the circuit to replace the oracle.

(4) Quantum Circuit Execution task
it executes the quantum circuit and returns a Result Object containing the probability distribution of the execution results. The task has three attributes: (i) Provider to define quantum computers; (ii) QPU to define quantum computers; (iii) Shots to define the executions number.



(1) Quantum Circuit Loading task
it loads the hybrid code into the workflow and creates a Quantum Circuit Object containing the code and its data. The task has two attributes : (i) URL to define a location to load the circuit code from; (ii) Quantum Circuit to insert the code into the workflow.

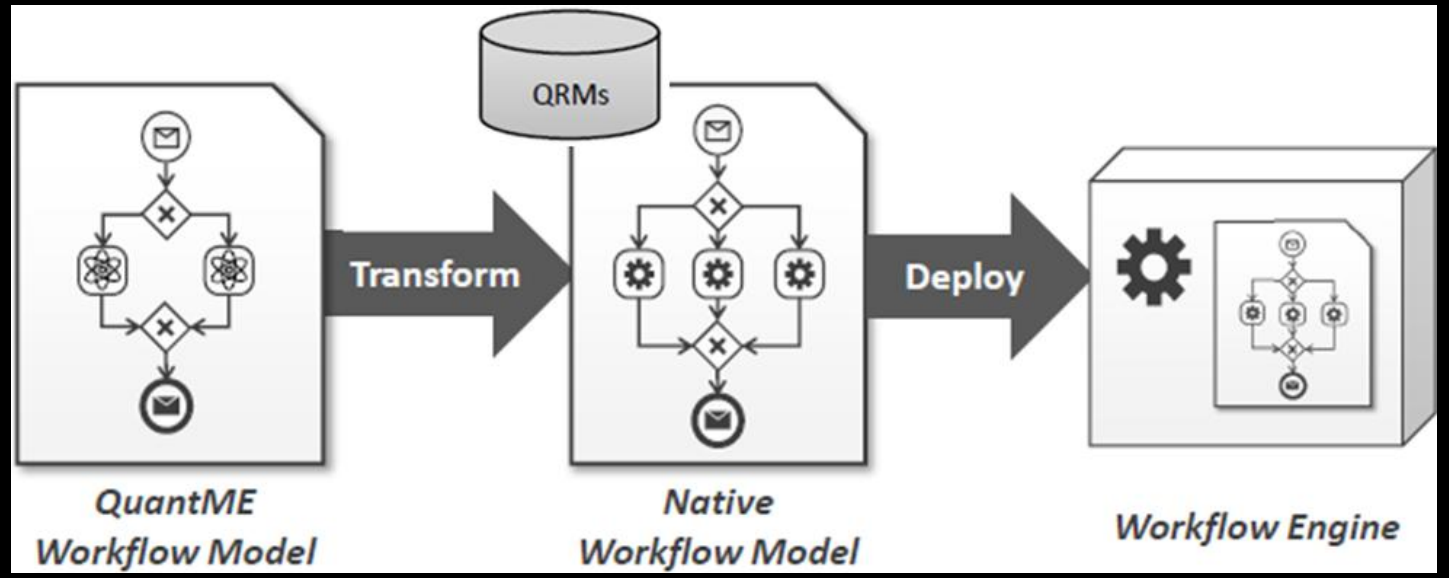
(3) Data Preparation task
it encodes the quantum circuit input data and prepares the state of the quantum register. The task has two attributes: (i) the encoding scheme, to define how the input data is encoded; (ii) the programming language, to identify the programming language of the quantum circuit to be initialized by the task.

(5) Readout Error Mitigation task
it reduces the readout errors in the results probability distribution. The task has three attributes: (i) Unfolding Technique to define the technique to mitigate the error; (ii) QPU to identify the quantum computer; (iii) MaxAge for the maximum age of the data to use for the mitigation.

Classic-quantum hybrid application workflow model

The workflow engine understands only native BPMN syntax; it is therefore necessary to replace quantum tasks by tasks that implement the same logic but written in native BPMN syntax.

Quantum Replacement Models (QRMs)
The repository contains two task types : “classic-quantum hybrid tasks” and “native BPMN tasks”. Rules are defined to select a native BPMN task for a given classic-quantum hybrid task.



QuantME Workflow Model
it creates the workflow with Quantum Modeling Extension (QuantME).

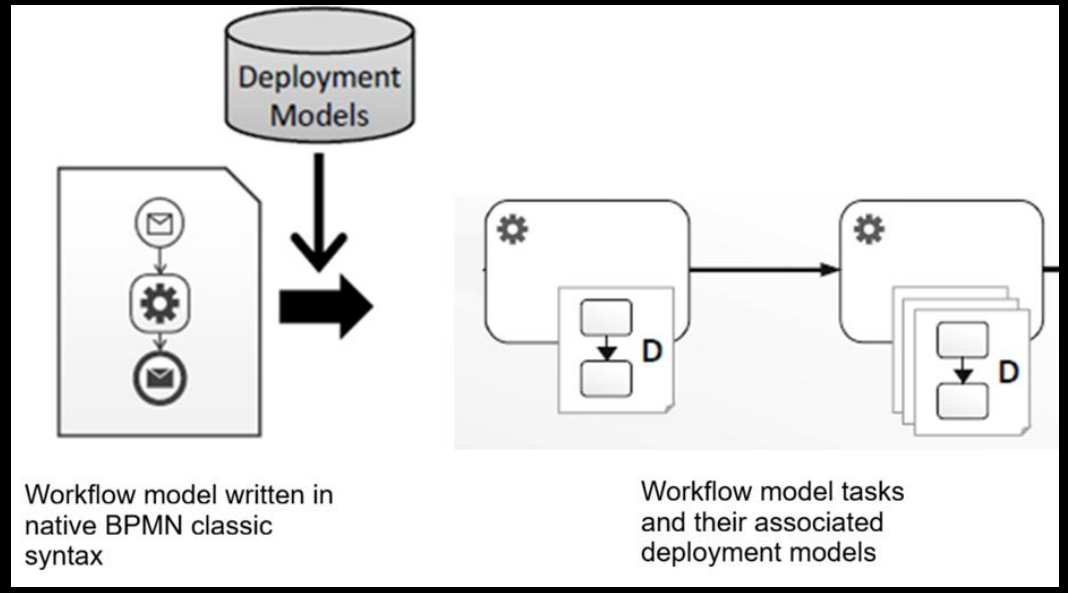
Native Workflow Model
it replaces the QuantME modeling tasks by tasks that implement their logic but written in native BPMN syntax.

Workflow Engine
it executes the native BPMN workflow model.

Classic-quantum hybrid application workflow model archive

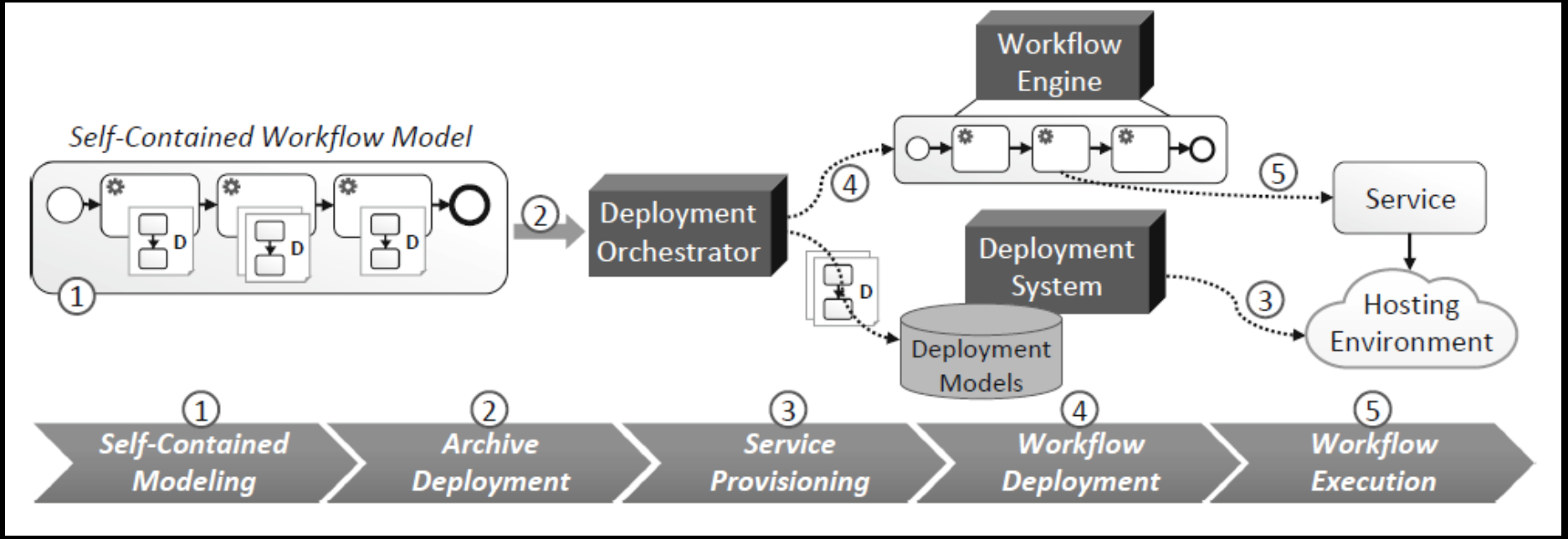
An archive – called Quantum Application Archive (QAA) – contains the workflow written in native BPMN syntax, the data required by the workflow, and the deployment models used to deploy the services needed by the workflow tasks. The deployment model is written in TOSCA (Topology and Orchestration Specification for Cloud Applications) service templates.

Deployment Models
The repository contains a set of TOSCA service templates to deploy an instance of a given service.



Classic-quantum hybrid application workflow model execution

The execution model covers the steps from the definition of the workflow model to its execution.



5) Workflow Execution
the workflow engine executes the workflow, and the tasks access the deployed services.

1) Self-Contained Modeling
it contains the workflow modeled with a native BPMN syntax.

2) Archive Deployment
the quantum application archive (QAA) is sent to the deployment orchestrator, which selects a deployment system and a workflow engine.

3) Service Provisioning
the deployment orchestrator triggers the deployment of the required services.

4) Workflow Deployment
the deployment orchestrator binds the services to the workflow tasks.

Classic-quantum hybrid application workflow model in the FTQC era

In the Fault-Tolerant Quantum Computing era where does an orchestrator deployment fit in ?

Thank you

