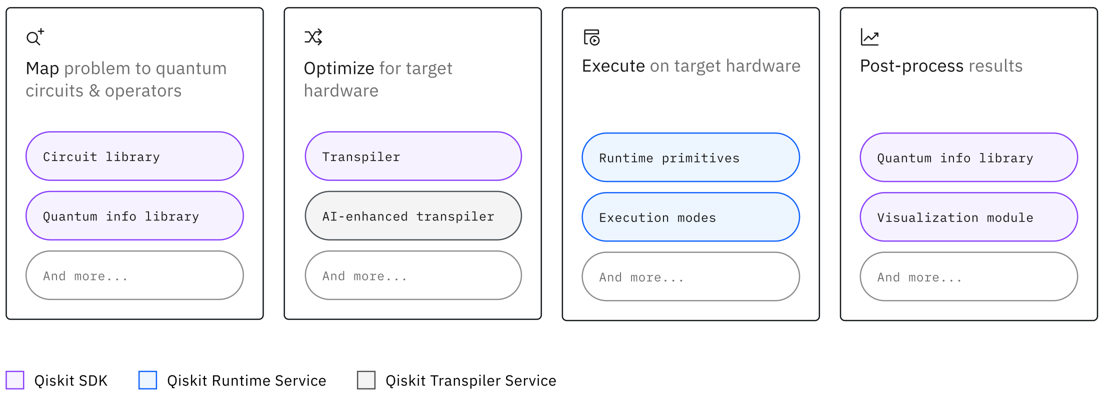
Qiskit

- a collection of software for executing programs on quantum computers.
- most notable softwares:
    - Qiskit SDK (open-source)
    - runtime environment of Qiskit Runtime (to execute workloads on IBM Quantum Computers)



Qiskit SDK

package name : `qiskit` (library)

- open-source SDK for working with quantum computers at the level of **extended quantum circuits, operator and primitives**.
    - Extended Quantum Circuits:
        - static
        - dynamic
        - scheduled
- core component of and the largest package under Qiskit (the collection)
- has the broadest suite of tools for quantum computing
- useful features [*name (module)*]:
    - Circuit-building tools (`qiskit.circuit`)
        - for initialising and manipulating:
            - registers
            - circuits
            - instructions
            - gates
            - parameters
            - control flow objects
    - Circuit library (`qiskit.circuit.library`)
        - a vast range of:
            - circuits
            - instructions

                - gates
            - key building blocks for circuit-based quantum computations
    - Quantum info library (`qiskit.quantum_info`)
        - toolkit for working with:
            - quantum states
            - operators
            - channels
        - using exact calculations (no sampling noise)
        - module used to:
            - specify input observables -- ??
            - analyse fidelity (here, accuracy), of outputs from primitive queries
    - Transpiler (`qiskit.transpiler`)
        - for transforming and adapting quantum circuits to suit specific device topology
        - for optimisation of real Quantum Processing Units (QPUs) execution
    - Primitives (`qiskit.primitives`)
        - module that contains the base definitions and reference implementations of :
            - Sampler and Estimator primitives
                - used by different quantum hardware providers to derive their own implementations.
        - **More** : https://docs.quantum.ibm.com/guides/primitives

Qiskit Runtime

- open-source elements:
    - The Qiskit Runtime client (the interface for users to access the Qiskit Runtime service)
    - The Qiskit SQK running on the server side
    - some of the software used for error mitigation
    - More to get involved with the Qiskit open-source efforts: Qiskit | Qiskit Extensions
- not open-source element:
    - The Qiskit Runtime service software that handles the technicalities of running our quantum program on IBM Quantum device (including any error mitigation and suppression)
- used for executing quantum computations on IBM Quantum hardware
- cloud-based service
- package name : `qiskit-ibm-runtime` (library)
    - a client for that service
    - successor to Qiskit IBM Provider - ***DEPRECATED***
- streamlines quantum computations
- provides optimal implementations of the Qiskit primitives for IBM hardware
- More on selecting an IBM Quantum Channel : https://docs.quantum.ibm.com/guides/setup-channel

- designed to use additional classical and quantum compute resources
  - including techniques such as:
    - error suppression
    - error mitigation
      - Example : zero-noise extrapolation (ZNE)
      - More on configuration : https://docs.quantum.ibm.com/guides/configure-error-mitigation
    - to return a higher-quality result from executing quantum circuits on quantum processors
- includes 3 types of execution modes for running the quantum program on IBM hardware:
  - Job
    - Single query to a primitive that can be run over a specifies number of shots
  - Sessions
    - Allow us to efficiently run multiple jobs in iterative workloads on quantum computers
  - Batch
    - Allows us to submit all our jobs at once for parallel processing

Qiskit Serverless

- package name: `qiskit-serverless`
- creating utility-scale quantum applications generally requires a variety of computer resource requirements
- premium users cam use Qiskit Serverless to easily submit quantum workflows for remote, managed execution
- More on how to use this collection of tools: https://docs.quantum.ibm.com/guides/qiskit-serverless

Qiskit Transpiler as a Service

- package name : `qiskit-transpiler-service`
- new experimental service
- provides remote transpilation capabilities on the cloud to IBM Quantum Premium Plan users
- additionally to local Qiskit SDK transpiler capabilities, our transpilation tasks can benefit from both IBM Quantum cloud resources and AI-powered transpiler passes using this service
- More on how to integrate cloud-based transpilation into our Qiskit workflow : qiskit-transpiler-service

The Qiskit ecosystem

- other open-source projects using the 'Qiskit' name (but not part of Qiskit itself)
    - can provide valuable additional functionality to supplement the core Qiskit workflow
    - some maintained by IBM Quantum teams
    - other supported by broaded open-source community
- The Qiskit SDK is designed in a modular, extensible way to make it easy for developers to create projects liek these that extend its capabilities.
- Popular projects:
    - Qiskit Aer (`qiskit-aer`) - Maintained by IBM Quantum
        - package for quantum computing simulators with realistic noise models
        - provides interfaces to run quantum circuits with or without noise using multiple different simulation methods
    - qBraid SDK (`qbraid`) - Maintained by qBraid
        - a platform-agnostic quantum runtime framework for both quantum software and hardware providers
        - designed to streamline the full lifecycle management of quantum jobs:
            - from defining program specifications to job submission
            - through the post-processing and visualisation of results
    - mthree (`mthree`) - Maintained by IBM Quantum
        - package for implementing M3 (Matrix-free Measurement Mitigation)
        - a measurement mitigation technique
        - solves for corrected measurement probabilities using a dimensionality reduction step followed by either:
            - direct factorization
            - or, a preconditioned iterative method that:
                - nominally converges in O(1) steps
                - can be computed in parallel
    - Catalog of projects and information about how to nominate our own project: [Qiskit Ecosystem]()