Share your integration challenges with us!

Get in touch.

Our hybrid integration experts are happy to help:

+44 203 983 9860 | info@mach.health | mach.health

BUILD YOUR HYBRID

HEALTHCARE INTEGRATION

xtention
IT with care.

Technical Brief

MEDICAL APPLICATION CONNECTION HUB – MACH

Fast, frictionless and secure transfer of big data through data streaming in MACH Cloud Connector (CC)

MACH aims to address the integration and data challenges faced by the healthcare industry by providing hybrid cloud integration for health data. MACH is a secure, open standards-based platform for quick and simple deployment of on-premises and cloud-native integration services. MACH allows you to manage instance creation, updates and replacement centrally. It is highly scalable, API-driven, and privacy-first by default. MACH seeks to address every use case and serve as a one-stop shop for

partners facing integration challenges. It can be used in a variety of ways: standalone or fully embedded into a partner solution; as a classical integration engine connecting on-premise applications and systems, or helping to bridge local edge systems with cloud-hosted applications. Due to the modularity of the various options, customers benefit from additional flexibility and can tailor the architecture to their individual needs.

MACH provides a secure, reliable and compliant platform for sharing health data between medical applications.

The MACH | Orchestra integration engine on which MACH is built provides industry standards-based message transformation tools for all flavors of HL7 (v2 and v3), FHIR[1], and supporting standards such as ASTM, XDT, EDIFACT, CDA, CCD, CCDA and DICOM. MACH also supports device integrations[2] and all widely used communication technologies, such as MLLP, TCP/IP, JDBC, SOAP, REST, DICOM Worklist, and TLS certificate-based exchange. The built-in DICOM routing features offer DICOMweb and DIMSE, MPPS, plus change and (semi-automated) anonymization functions, which can be implemented as pre- and post-processing steps facilitated by MACH. MACH also offers a hybrid configuration: if required, you can customize the centrally managed cloud platform by connecting to an on-premise instance.

How you manage and monitor data flows determines how effective a data integration platform is. On the MACH | Portal, monitoring is possible from a cluster perspective all the way down to individual instances, allowing for high-level status investigation as well as in-depth scenario exploration and log search capabilities using tools such as Grafana.

The use case illustrated next uses MACH's Cloud Connector (CC) to transfer DICOM studies from an on-premises image source to a cloud-hosted AI solution. The fully fledged Health Service Bus that is included in MACH | Orchestra ensures that MACH CC functions flawlessly and can securely and reliably transfer protected health information (PHI[3]) right out of the box, with optional pre- and post-processing steps.
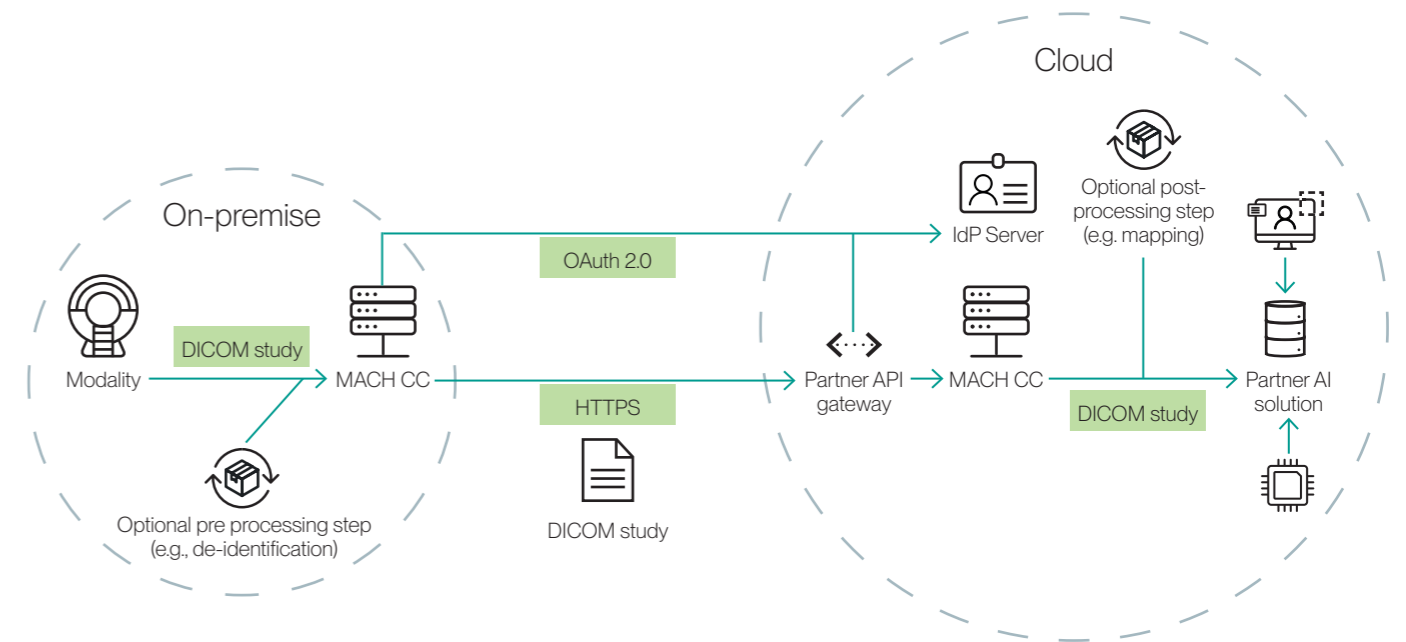
Standard MACH deployment options include:[4]

- **Lightweight**
  Docker or VM based image delivery, local database therefore limited data resilience suitable for on-premise and cloud use. Will be managed from a central point and receive pushed configuration changes.

- **Default**
  Micro-Kubernetes Cluster with shared databases, including cloud backup suitable for on-premise and cloud use. The default option can be provided in multiple variations depending on the focus or number of scenarios. It can be understood as a mid- to high-range interoperability layer capable of handling heavy loads.

- **High availability and cluster**
  Zero downtime and high availability is the main goal of implementations in a k8s cluster providing a robust operations-centric architecture that is highly scalable and resilient. Offering container self-healing capabilities and designed to efficiently manage large amounts of containers that are distributed across complex cloud environments.

Basic deployment options with close-to-zero downtime are available for all MACH instances. Rolling deployments can be provided in a container environment such as Docker or k8s.

Additional deployment scenarios can be implemented based on the client's needs.