

Microservices: Designing for innovation

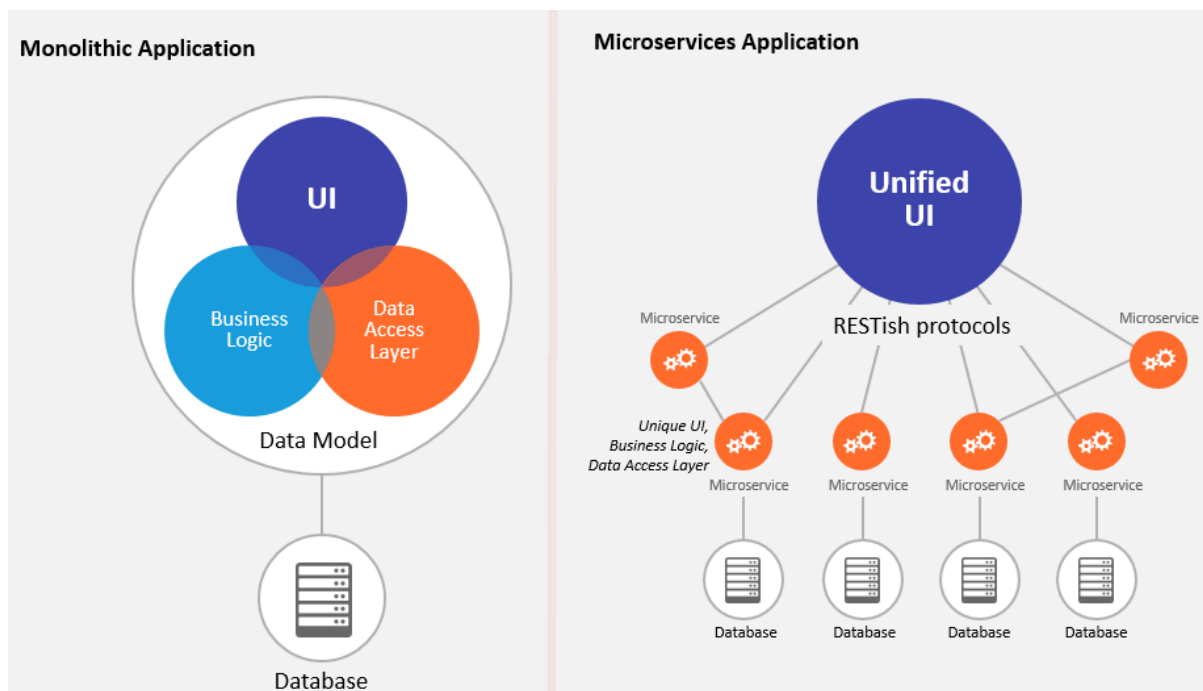
DIGITAL HEALTH LEARNING NETWORK,

CAMBRIDGE DIGITAL INNOVATION

Hughes Hall, University of Cambridge

Microservices architecture has been gaining popularity as a design approach over the past few years. What are microservices and can they enable innovation? In order to explain microservices, we start by considering the composition of software applications and distinguishing microservices from its predecessor, monolithic architecture. At their most basic level, applications consist of three architectural layers: 1) the user interface (what is visible to the end user), 2) the data access layer, and 3) the database (hidden behind a black box).

Monolithic design consists of a single user interface behind which sits a data access layer connected to a single database. Monolithic architecture is built as a single unit or code base where all the business logic including the algorithms and security for the entire application sits within a unified backend. Within an innovation ecosystem, re-use is possible but not with the application "as is."



In comparison, microservices design breaks down the application to its core business capabilities and each business capability has its own unique user interface, tied to its own data access layer and mini-database. Each microservice is a mini-application, which can be configured, re-configured, and replaced without compromising the integrity of the whole application. A multitude of microservices are brought together through REST-ish protocols

(e.g. HTTP request-response with resource APIs) to form a unified user-interface. Therefore, microservices are decoupled, yet cohesive. Decoupled because they are federated (self-contained responsible for their own data model and data) and cohesive because they communicate with each other or are choreographed through a well-defined interface.

From the perspective of the end-user, there is no difference between a monolithic architecture and microservices as what they see, manipulate and play with are no different. Microservices is the "as-is" reconfiguration of components to create a whole new application. The practice of designing microservices is indicative of the deliberateness in which the boundaries of a microservice is constructed. A microservice is the sociomaterial process of design decision-making, which is a combination of the designers' interests, and the material need of a very particular situation. The resultant application is a sociomaterial embodiment of the process undertaken.

So can microservices enable innovation? It depends on 1) how microservices architecture are implemented in practice and 2) on how innovation is "performed" within an organization. The emphasis here is that microservices are both the practice of designing software applications as well as the outcome suite of independently deployable application components or mini-applications available as-is for innovation. Re-use without (or with limited) re-work depends on how the boundaries are drawn around a microservice.

Microservices, depending on boundary management in practice can facilitate on open innovation ecosystem across multiple stakeholder groups. Microservices could enable an organizational innovation strategy with strong external collaborations, partnerships, and multiple pathways to market.

