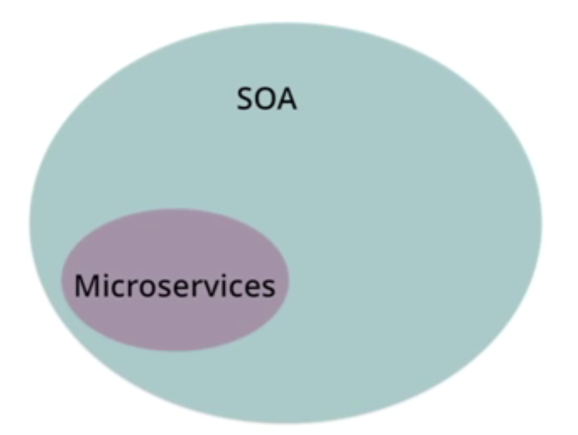
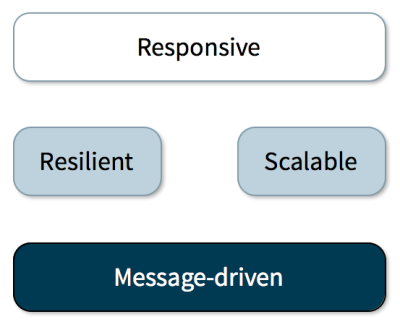
* The core difference between SOA and microservices lies in the size and scope.
* As the word "micro" suggests, it has to be significantly smaller than what SOA tends to be.
* Microservice is a small(er) independently deployable unit.
* A SOA can be either a monolith or it can be comprised of multiple microservices.
* Martin Fowler says he likes to think SOA to be a superset of microservices.

[](http://i.stack.imgur.com/thqNG.png)

* The typical SOA model, usually has more dependent [ESBs](http://en.wikipedia.org/wiki/Enterprise_service_bus), with microservices using faster messaging mechanisms.
* SOA also focuses on imperative programming, whereas microservices architecture focuses on a responsive-actor OR Reactive programming style.
* Moreover, SOA models tend to have an outsized relational database, while microservices frequently use NoSQL or micro-SQL databases (which can be connected to conventional databases).
* But the real difference has to do with the architecture methods used to arrive at an integrated set of services in the first place.

**The four Reactive principles**

Reactive applications are built on four guiding principles.



The core building blocks of Reactive applications.

* A ***responsive*** application is the goal.
* A *responsive* application is both***scalable*** and ***resilient***. Responsiveness is impossible to achieve without both scalability and resilience.
* A ***message-driven*** architecture is the foundation of scalable, resilient, and ultimately responsive systems.