SOA（Service Oriented Architecture）面向服务架构，面向服务的架构（SOA）是一个组件模型，它将应用程序的不同功能单元（称为服务）通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种各样的系统中的服务可以以一种统一和通用的方式进行交互。

微服务架构，微服务架构是一项在云中部署应用和服务的新技术。大部分围绕微服务的争论都集中在容器或其他技术是否能很好的实施微服务，而红帽说API应该是重点。微服务不需要像普通服务那样成为一种独立的功能或者独立的资源。定义中称，微服务是需要与业务能力相匹配，这种说法完全正确。不幸的是，仍然意味着，如果能力模型粒度的设计是错误的，那么，我们就必须付出很多代价。如果你阅读了Fowler的整篇文章，你会发现，其中的指导建议是非常实用的。在决定将所有组件组合到一起时，开发人员需要非常确信这些组件都会有所改变，并且规模也会发生变化。服务粒度越粗，就越难以符合规定原则。服务粒度越细，就越能够灵活地降低变化和负载所带来的影响。然而，利弊之间的权衡过程是非常复杂的，我们要在配置和资金模型的基础上考虑到基础设施的成本问题。

SOA和微服务的区别：

SOA（Service Oriented Architecture）面向服务架构，面向服务的架构（SOA）是一个组件模型，它将应用程序的不同功能单元（称为服务）通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种各样的系统中的服务可以以一种统一和通用的方式进行交互。

微服务架构，微服务架构是一项在云中部署应用和服务的新技术。大部分围绕微服务的争论都集中在容器或其他技术是否能很好的实施微服务，而红帽说API应该是重点。微服务不需要像普通服务那样成为一种独立的功能或者独立的资源。定义中称，微服务是需要与业务能力相匹配，这种说法完全正确。不幸的是，仍然意味着，如果能力模型粒度的设计是错误的，那么，我们就必须付出很多代价。如果你阅读了Fowler的整篇文章，你会发现，其中的指导建议是非常实用的。在决定将所有组件组合到一起时，开发人员需要非常确信这些组件都会有所改变，并且规模也会发生变化。服务粒度越粗，就越难以符合规定原则。服务粒度越细，就越能够灵活地降低变化和负载所带来的影响。然而，利弊之间的权衡过程是非常复杂的，我们要在配置和资金模型的基础上考虑到基础设施的成本问题。

SOA和微服务的区别：

微服务架构是一项在云中部署应用和服务的新技术。大部分围绕微服务的争论都集中在容器或其他技术是否能很好的实施微服务，而红帽说API应该是重点。