**75% 新项目都可以“无脑”选择单体架构**

<https://mp.weixin.qq.com/s/yG3JuDUNDhlrixHg-Qyj9A>

## 易于理解

很多朋友都觉得微服务架构的理解难度更低……但事实真是这样吗？

当然了，我们只需要具体管控各个肩负明确职责的微服务项目，所以每种元素在干嘛、需要干嘛、系统整体状况如何不就更清晰了吗？毕竟我们面对的是一个个服务，而非彼此交织的架构整体。

但我要给大家泼点冷水：这完全就是骗人的。没错，确实有一些微服务架构做出了优秀的边界定义，任何半路介入的参与者都能快速理解某项微服务的实际作用。但也有很多项目做得不好，导致某项服务要么做得太多、要么做得太少；甚至部分单一功能也被过度拆分成了多项服务，因此系统的混乱度大幅提升，任何单一服务的故障都可能将整体应用拖入崩溃的深渊。

大家可能会说，“对，这种情况是有，但那是你的问题、不是微服务的问题”，或者“你笨啊，笨还能怪架构？”有道理，但问题是项目绝对不可能百分之百受控，真的不可能。团队、组织、项目，各个层面都有出错的可能，所以边界定义不清的几率会远远高于边界定义良好的几率。

但我也承认，这种情况在单体式架构中也可能带来麻烦。我也见过那些搞不清在干什么的单体式应用，所有功能就像飘香拌面一样混杂成大坨，代码库硕大无朋、缺少必要的测试、说明文档含糊不清、不同功能的编程风格格格不入等等。

不过单体式架构修改起来还是更轻松一点，它相较于微服务的优越性也正在于此——模块化。我们同样像为微服务定义边界那样进行代码构建，只是不再将这些“服务”或者说模块视为其他应用的元素，而是同一整体中的各个组成部分。

所以对我来说，这种杂乱的毛病主要还是出在微服务架构身上。只是我也承认，如果要想搞砸，那在单体式架构中也一样可以搞砸。