SOA 为什么不“香”了？丨建设数据中台系列（三）

* 耿立超

发布于：2020 年 8 月 6 日 10:44

SOA 所有的理念都是基于现有应用系统展开的，不管是对服务的梳理还是服务之间的交互，都是以现有应用系统为载体的，中台不同于 SOA 的地方在于：中台是一种平台化思维，它并不是从系统集成的角度去思考问题，而是从架构层面上重构了整个 IT 生态。相比之下，中台无疑是一种更深刻、更底层的变革，因为它完全破除了应用之间的壁垒，把企业的核心业务能力“中心化”，把它们提炼并沉淀到中台的各个业务中心上，而不是面向单一业务方向或渠道的应用系统上。这在 SOA 架构下是很难实现的，因为中台的业务中心与 SOA 的服务载体（即应用系统）之间有着本质区别，它们的定位和服务对象都不同，这些区别决定了 SOA 依然是一种相对松散的分治式的架构，很难与中台这种更加中心化、更为强力的架构体系相抗衡。

烟囱式的生态系统并不是今天才突显出来的，很多企业已经被这个问题困扰多年了，并且尝试过各种措施试图进行改善。回顾企业的 IT 生态变迁史，一段不得不提的历程就是 SOA（面向服务的架构）。本文核心观点援引自作者所著的[《大数据平台架构与原型实现：数据中台建设实战》](https://item.jd.com/12677623.html)一书，全书对数据中台的理念、架构和具体实现做了详细论述。

大概在 2005 年前后的七八年间，随着 SOA 理念和相关技术（如 ESB）的不断发展和完善，SOA 在当时被认为是改善僵化的 IT 生态、解决烟囱架构等弊病的终极方案而被业界寄予厚望，很多企业在那个时期纷纷上马 SOA 项目，希望凭借 SOA 将企业的 IT 生态拉回到一种理想的状态。十多年后回首当初那场 SOA 热潮，我们发现最终在 SOA 改造上取得成功的企业少之又少，即使曾经取得了一定的成效，伴随后来新业务系统的冲击，当年辛苦建立的 SOA 生态也大都名存实亡，是什么原因导致了这样的结果呢？

人们很早就意识到点对点式的系统间交互是非常糟糕的，在 SOA 起源之前，已经出现了基于消息队列的“消息总线”架构，各个应用系统与消息总线连通，由消息总线负责将消息路由到接收方，从而让应用系统通过中心化的消息总线完成交互，这样就可以消除点对点式的系统交互。但是消息总线用于系统集成时在某些方面依然有所欠缺，例如消息都是静态的、预定义的、无法自描述的，消息接口无法被注册和发现。同时，另外一种以“服务”为视角看待和思考系统间交互的架构思想一直在不断地发展，后来，随着 Web 服务（Web Service）技术的兴起，IT 系统的对外接口逐渐向平台中立的第一代 Web 服务标准（WSDL+SOAP）靠拢，这为实施这一架构打开了大门，这就是 SOA。

从系统集成的角度看，SOA 是一种非常理想的方案，SOA 体系的两大核心如下：

* 对系统提供的对外交互进行提炼、组织和梳理，通过封装、组合与编排，将接口以“服务”的形式发布出去；
* 系统间的交互统一通过中心化的企业服务总线（ESB）完成。

典型的 SOA 架构如图 1 所示。

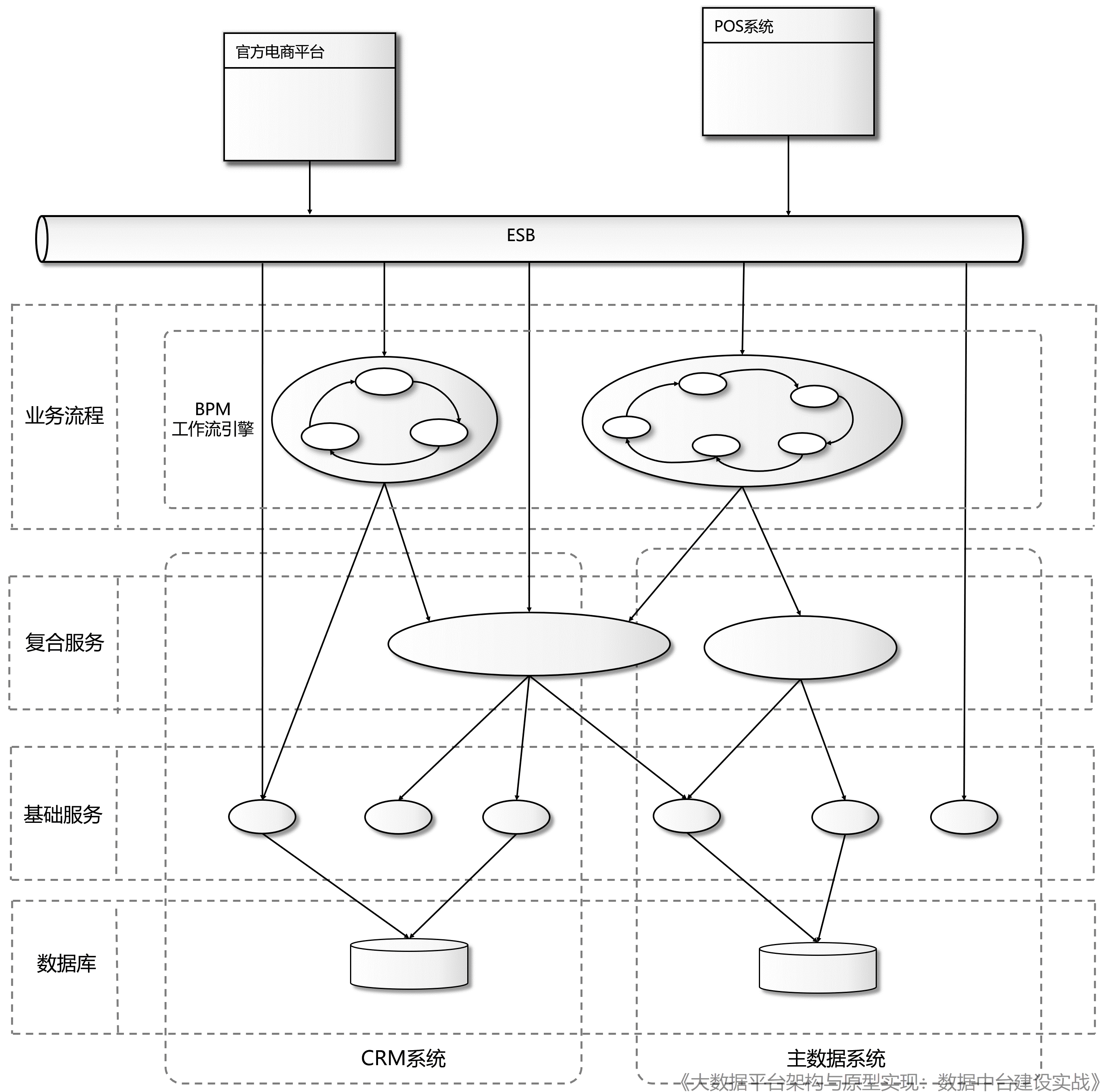


图 1 典型的 SOA 架构

SOA 成功的基础是对“服务”的提炼、组织和梳理，只有服务足够灵活才能支撑各种外部系统的复杂需求，而这一工作需要建立在对业务的深入了解之上，同时要融合良好的设计思想才能达到要求。

SOA 中的“服务”在技术上以 Web Service 为载体，但是在粒度（或者说抽象程度）上会有所不同，主要有如下三种粒度的服务：

* 基础服务：最细粒度的服务，最基本、最原子的服务都会在这一层，从服务数量上看，这一层也是最多的；
* 复合服务：是基于多个基础服务组合叠加而成的粗粒度服务，多用于封装并简化由多个基础服务组合实现的共性服务；
* 业务流程：是通过工作流引擎将多个服务编排起来，形成一个完整的业务流程，这是一种粒度更粗的服务，常用于实现一个标准的、可复用的大尺度业务流程，如审批等。

在应用系统之间，SOA 依靠 ESB 实现系统集成，ESB 是治理点对点式的系统集成的核心手段，它肩负着如下重担：

* 实现系统间的连通；
* 数据转换；
* 智能路由；
* 安全控制；
* 可靠性控制；
* 服务管理；
* 监控与日志。

以上是对 SOA 的一个基本介绍，SOA 针对烟囱架构的治理主要依赖于两个方面：

* 一方面立足于每个应用系统，要求系统对提供的“服务”进行提炼和抽象，确保其灵活、可重用，这是让服务满足外部复杂需求的根本保障；
* 另一方面是通过中心化的交互媒介——ESB 来约束系统间的交互，消除点对点式集成的负面影响。

但令人感慨的是：走到今天，SOA 已经很少被人提及了，回看企业曾经在 SOA 上做出的尝试和努力，最后的效果多数都不够理想。在完成 SOA 改造之后的若干年间，受到后续各种新系统的冲击，很多企业都没能坚守住自己的 SOA 体系，最终又回到了烟囱架构下的野蛮生长状态，这其中的原因主要是：

* 沟通与协作成本高：新系统迫于业务需求和市场压力，急需上线，对负责的团队而言，与周边系统的对接和调试属于外部不可控因素，团队总是倾向于在内部可控的范围内解决问题，因此会刻意避开对外部服务的依赖，选择自建相关功能，这样一来，系统间的交互会向着衰减的方向发展，重复建设也因此随之蔓延；
* 组织架构制约：团队往往缺乏为响应其他系统的诉求而改造和升级自身服务的意愿，因为新系统与他们没有直接的利益关系，企业也缺乏适当的奖惩机制促使各团队之间的积极协作，本质上，这是组织架构决定的；
* 缺乏长效机制：SOA 改造常常是作为一个项目实施的，项目结束之后就不再有专门的组织和团队对 SOA 架构进行持续把控了，后续新的系统在融入 SOA 生态时受到的支持就减弱了，而新系统本身提供的服务也缺乏必要的梳理和管控，有的新系统甚至不对外提供服务。

这些问题并不是 SOA 自身的问题，而是一些普遍的现实问题，也是治理烟囱架构过程中遇到的深层次问题，这些问题阻碍了 SOA 在企业的落地和持续发展。所以说 SOA 是曾经的“救赎”，企业 IT 生态现在面临的问题依然没有得到很好的解决。