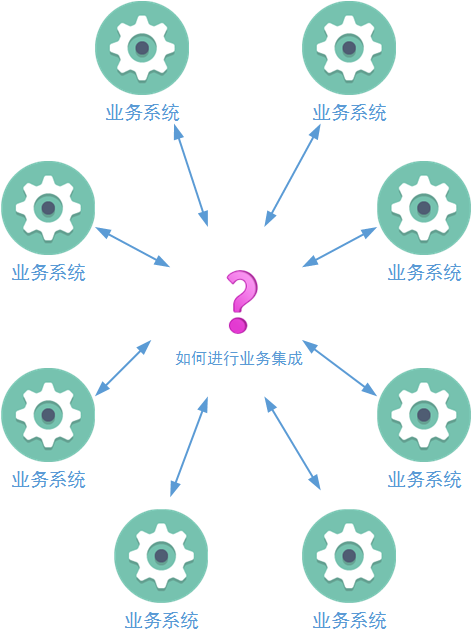
ESB与SOA

SOA(Service-Oriented Architecture)面向服务的架构

SOA是一种架构模式思想，主要围绕多个“服务”，如何进行集成以达到某种目的进行讨论。在业务系统中被发布出来供用户使用，能够完成一个完整业务过程的功能，就是服务



**服务着眼于完整的业务**

从以上对”服务”的定义可以看出，服务的定义对象中的完整业务功能。例如电商系统中“确认订单”这个功能就是一个服务、计费系统中”当月费用结算”功能就是一个服务;但是CRM系统中，系统完成”工单生成”功能所进行的”用户登录”动作就不是服务的定义，因为使用者进行”用户登录”是为了完成”工单生成”功能的权限验证步骤，并不是为了”登录”而登录

**服务的粒度虽然相对粗放，但却可控，目标也重用**

“用户登录”这个功能在某些情况下也可以满足”服务”的定义;当用户中心系统为其他所有业务系统统一提供的“登录”服务被公布出来时。**服务粒度的拆分完全根据业务系统中业务过程进行定义和分析**，所以服务的粒度都相对粗放。就像上一段文字中所举例”费用结算”服务那样;可能完成“费用结算”功能，在计费系统内部需要完成 用户身份确认-->上月费用查询-->套餐清单查询-->费用明细生成 这几个过程。**但是这些每一个过程都不会有任何其他业务系统进行单独使用**，也就是说是否单独公布这些出来过程对于其他业务系统来说没有任何意义

如果在后续的业务变更/业务重编时，业务设计人员发现某一个业务系统**需要单独使用计费系统的“上月费用查询”功能**，这时就需要计费系统将这个功能作为一个独立的服务向第三方业务系统公布。这时，这个功能就满足”服务”的定义

**集成的目的是形成一个新的服务**

对企业内部(或者企业间)的业务服务进行集成，被集成的业务服务称之为**原子服务**，集成的目的是重用这些原子服务形成的一个新的服务。**这样保证了技术团队/业务团队能以最小的代价，最高的效率使用既有服务**，但同时也对实现SOA架构思想的软件提出了更高的也要求

**SOA需要保证屏蔽细节**

使用SOA架构思想构建多个业务系统的集成关系，需要保证每个业务系统屏蔽细节。这些细节包括技术细节和业务细节。从技术层面看，无论业务系统使用哪种开发语言、哪种对外传输协议、哪种消息格式都可以使用SOA进行集成，并且能够在SOA架构的实现软件上完成不同传输协议的转换和不同消息格式的转换;从业务细节层面看，SOA系统屏蔽业务系统的功能步骤细节。也就是说第三方系统只需要知道调用某一个服务就可以达到业务目的，至于提供服务的业务系统如何实现业务过程则无需关心

**SOA让各业务系统保持松散**

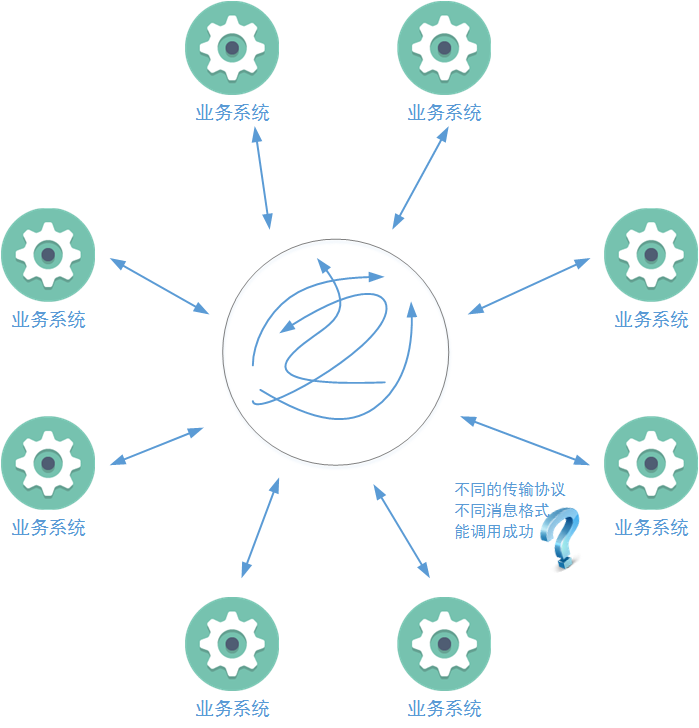
SOA架构模式为多个业务系统进行松散集成提供了一个良好的思路;通过屏蔽各业务系统技术细节和业务细节，兼容各业务系统的不同传输协议和不同消息格式，可以让通过SOA进行业务集成的各个业务系统保持低耦合度状态。这是因为所有协议和消息格式都处于开发状态，业务集成时各业务系统不需要单独进行额外的转换工作，甚至不需要为基于SOA的业务集成进行任何额外工作，也无须知道对方系统的存在

**注意：如果您所在企业或者客户的业务系统还没有达到一个较高的复杂等级，则不建议立即使用SOA架构模型进行业务集成。因为目前SOA架构模型的各种实现本身就具有一定的复杂性。**

ESB

ESB(Enterprise Service Bus)企业服务总线，是SOA架构思想的一种实现思路。

企业的信息化建设一般经历很长的时间的发展，少则5、6年多则10几年。所以我们看到某大型企业的信息系统最可能的情况是：存在着多个业务系统，甚至各业务系统负责的功能职责还存在重叠。这些系统采用不同时代的编程语言、编程框架、通讯协议、消息格式和存储方案



为了满足SOA架构思想的设计要点，达到既定的工作目标，ESB总线技术至少需要帮助这些业务系统完成以下工作：

**多协调协议支持和转换**

无论业务系统向外公布的服务使用哪种调用协议，都可以通过ESB中间层实现相互转换。例如A业务系统的服务只接受Web Service SOAP形式的调用，B业务系统的服务却可以使用Thirft RPC进行调用(不必为了调用A业务系统而专门去适应A业务系统的协议)。在基于ESB服务的中间层帮助实现两种协议的转换，

**多消息格式支持和转换**

无论调用协议携带哪一种消息描述格式，通过ESB中间层也可以实现相互转换。ESB中间层应该支持将JSON格式的信息描述转换成目标业务系统能够识别的XML格式，或者将XML描述格式转换成纯文本格式，又或者实现两种不同结构的XML格式的相互转换等等

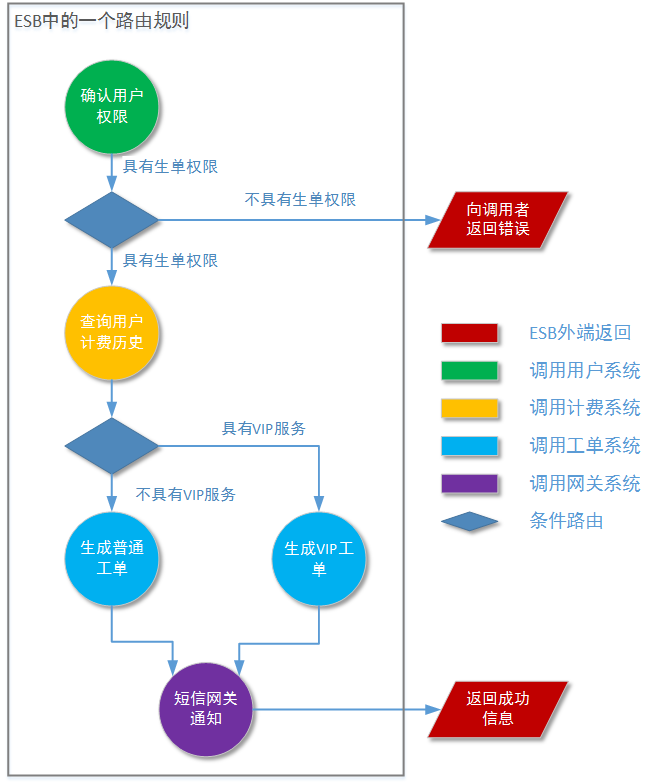
**服务监控管理(注册、安全、版本、优先级)**

既然ESB要对原子服务进行集成，考虑的问题就比较多了。首先，业务系统提供的服务可能会以一定周期发生变化，例如周期性升级;时刻的业务系统甚至可能呈现完全无预兆无规律的变化，例如突发性数据切割导致服务接口变动。那么ESB的实现软件中应该有一套功能，能够保证在这样的情况下集成服务依然能够工作。其次，并不是业务系统所提供的所有服务都可以在ESB中进行集成，也并不是所有的服务都能被任何路由规则所编排。ESB应该有一套完整的功能来保证服务集成的安全性和权限

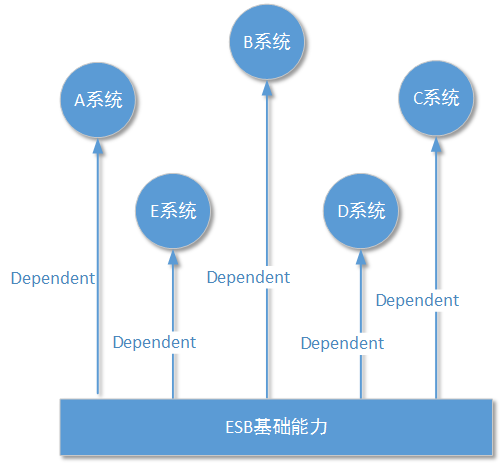
作为被集成的业务系统，ESB中如何集成它提供的原子服务，前者是不需要关心的

**服务集成和编排**

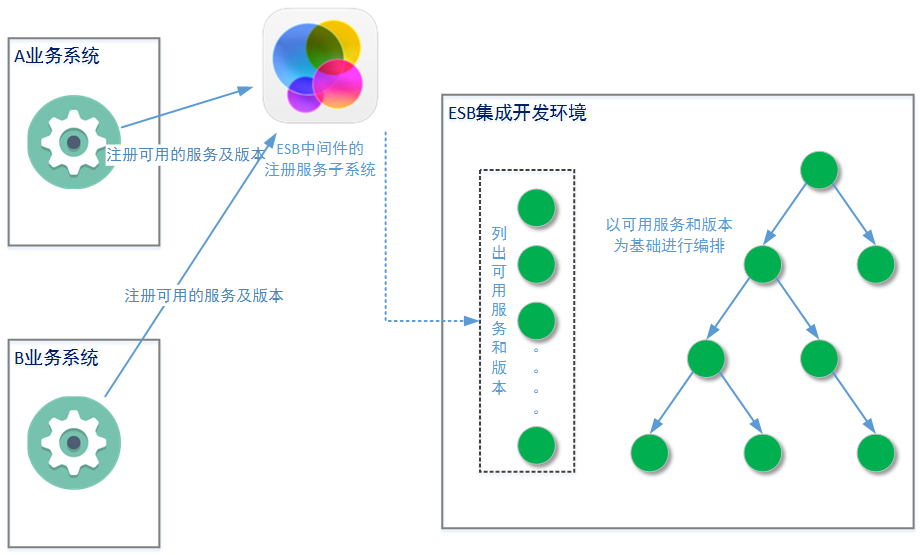
为了将多个服务通过ESB技术进行集成形成一个新的服务，ESB技术必须能够进行服务编排。服务编排的作用就是明确原子服务执行的先后顺序。判断原子服务执行的条件。确保集成后的新服务能够按照业务设计者的要求正常工作。下图示例了新服务”工单派发”通过多个业务系统提供原子服务，按照设置的执行条件在ESB总线上进行工作的过程



实际上ESB技术和本专题之前讲过的服务治理技术在架构层面属于同一层;都是对SOA思想的实现思路。但是两者的应用场景和侧重点完全不一样



**注意：**虽然ESB承担了原本在系统内部完成的业务集成工作。但是根据依赖倒置设计原则，ESB也只是依赖于各业务系统注册在ESB总线上的调用接口，业务系统中功能的具体实现对于ESB来说是透明的。



上图中，进行版本控制的关键功能就是ESB中的**注册中心模块**。在其中管理了各个业务系统已经向ESB总线注册的能够提供业务服务的接口定义和接口版本信息。ESB总线在进行服务编排时，**只能使用已经向注册中心注册的业务接口和相应版本**

**注册中心在下线某个版本的服务时，会检测是否还有业务功能对这个服务和版本存在调用关系**，如果检测发现服务依赖关系依然存在，则不允许下线操作。注册中心还可以在服务的新版本注册后，对现有编排逻辑中的老版本进行批量升级

需要注意：ESB中的注册中心模块并不像协议转换功能哪有，是ESB技术的必备功能。但是又注册管理中心和版本管理机制的ESB实现更能保证总线业务上功能的稳定运行

ESB与监控

对原子服务单元的监控性进行监控

对原子服务的调用情况进行监控

对集成后的新业务调用情况进行监控

对ESB中间件自身监控性进行监控

ESB的顶层设计

