**基于 Scrum 敏捷开发的 EDI管理链式微服务实践**

**摘要严格300字**

为应对全球制造业供应链日益激烈的竞争态势和企业间多元化的数据交互需求，提升供应链协同效率与数据交互安全性，我司于 2020 年 5 月启动了 EDI 管理系统研发项目。笔者作为技术负责人，全面负责项目整体方案设计与落地实施工作。该项目规模大、周期长，在推进过程中，笔者发现，团队不仅需要确保系统达到预期的业务性能要求，还要充分考虑到系统的整体安全架构设计。这不仅是满足企业数据隐私保护等信息安全合规要求，更是保障供应链数据传输安全的必然选择。

在该项目中，笔者着重从应用安全、系统安全、安全管理三大维度展开设计与规划，以配置安全策略和提高加密手段相结合的方式，将系统的安全风险降到最低，实现了平台全面的安全保障。自 2021 年 12 月初版系统上线后，该系统在经历一系列的业务优化升级变更，版本的并行切换后，仍然高效稳定运行至今。

**项目背景**

我司作为全球领先的企业资源管理解决方案服务商，自成立以来，始终致力于为制造业企业提供高效、稳定的 ERP 解决方案，凭借深厚的技术沉淀与对市场趋势的敏锐洞察，已成功为众多企业实现了生产、采购、财务等核心业务的数字化管理升级。随着全球制造业供应链体系加速变革，企业间数据交互效率成为制约产业协同发展的关键因素。尽管 EDI 技术广泛应用于供应链数据自动化交互，但 ANSI X12、EDIFACT 等国际主流 EDI 标准与企业个性化业务需求之间存在适配难题。据行业数据显示，传统模式下企业 EDI 数据处理错误率高达 15%，平均订单处理周期长达 3 - 5 天，严重影响供应链响应速度与企业运营成本控制。

2020 年 5 月，公司基于制造业数字化转型趋势，启动 EDI 管理系统研发项目。该项目旨在打造自主可控、适配企业业务的系统，实现数据协议全流程自动化管理，打破企业间数据交互壁垒。项目采用标准化与定制化结合模式，满足多样化业务场景需求，提升客户供应链协同效率。

该项目从规模和复杂度来看，属于中大型企业级应用项目，聚焦 EDI 数据交互核心需求，打造三大核心功能组件：订单管理组件实现全流程数字化管控，覆盖订单全生命周期；数据格式映射组件支持多标准数据格式双向转换，保障数据传输精准高效；EDI 结果监控及展示组件实时追踪交互状态，智能预警异常并可视化呈现处理结果。项目初期组建了12人的专业团队，成员覆盖业务分析、软件开发、测试等多个领域。随着项目推进及业务需求深化，后期团队规模扩展至18人，进一步充实了系统优化、功能拓展及运维保障等方面的力量，确保系统稳定运行与持续迭代升级。

**技术方法说明**

对比ToB项目，不仅需要满足企业数据隐私保护等信息安全合规要求，更需要保障供应链数据传输安全，任何环节的疏漏都可能引发数据泄露、系统瘫痪等灾难性后果，直接威胁项目的存续与企业的生命线。通常意义上来讲，信息系统安全架构一般需要包括五方面要素即：物理安全、系统安全、网络安全、应用安全和安全管理。

其中在物理安全方面，主要是保障信息系统硬件设备、基础设施以及其所处环境的安全，确保服务器和存储设备处于物理上受保护的状态。在此项目中，大部分应用都跑在大厂的云服务器上，因此对于云服务的物理安全和网络安全参与不多，这块服务保障主要由云服务提供商负责。

在系统安全方面，一般重点在于保护服务器、数据库等基础软件服务，防止未经授权的访问或操作，确保系统的稳定性和数据完整性。一般来说，操作系统、数据库等未被及时修复的漏洞可能会被黑客利用。

在应用安全方面，主要是防止第三方非法进行用户账户窃取以及通过应用漏洞或越权访问数据，这个也是笔者所带团队重点关注的方向。因为攻击者往往会利用一些常见的应用程序的漏洞，如SQL 注入、跨站脚本攻击（XSS）、跨站请求伪造（CSRF）等，在用户不知情的情况下执行一些如修改用户的账户设置、发起转账等高危操作。

最后，也是最容易出问题的，就是安全管理。笔者的个人体会是，就算你前面的策略做得再好再完善，一旦在安全管理上失守，那就相当于前功尽弃。受限于篇幅，在本文中，笔者将从自身实践出发，重点从应用安全，系统安全，安全管理等三方面展开，谈谈系统安全架构落地过程中的心得体会。

**主要内容**

在应用安全方面，笔者主要关注 SQL 注入、跨站脚本攻击（XSS）等常规的安全漏洞防范。在 EDI 管理项目中，由于系统需要频繁处理企业间大量的订单、交易等数据交互，SQL 注入风险尤为突出。在笔者的理解里，SQL注入本质上是程序在执行 SQL 语句时，对用户输入数据处理不当，致使输入数据意外成为可执行代码。根据笔者多年经验，此问题可借助成熟的持久层插件解决，核心原理在于参数化查询。通过参数化查询，用户输入不再直接拼接进 SQL 语句，而是作为参数传递给 SQL 引擎，SQL 引擎预先编译语句后再执行，有效杜绝危险代码注入，保障数据交互的安全性。在 EDI 系统的业务交互场景中，数据传输频繁，跨站脚本攻击（XSS）威胁同样不容小觑。对此，我们在后端采用 Java 第三方库对用户提交至数据库的数据进行编码处理，防止数据传输到前端时被浏览器误解释为可执行脚本。如此一来，仅需在后端一次性完成处理，而不用前端每次都做 html“杀毒”工作，既提升了开发效率，又增强了系统安全性。

在系统安全方面，我们主要关注操作系统层面的安全防护。一般来说，用户正确的操作系统管理可以降低操作系统被攻击的风险，进而保证操作系统的安全性。笔者团队主要的工作包括操作系统流程管理、特权操作环境与基础设施管理、操作系统补丁管理等。笔者在实践中，严格遵循最小特权原则，依据不同应用场景分配账户权限，防止特权账户被恶意利用。例如针对 Web 服务器管理员，主要赋予管理和维护Web 服务器（如Apache 或 Nginx）及其相关的应用程序和配置文件的权限。针对安全管理员，主要赋予安装和配置安全工具、查看安全日志、进行安全审计等权限。除此之外，应用安全补丁对于确保应用程序、驱动程序、操作系统和固件的持续安全性至关重要。在这过程中，出于谨慎考虑，笔者在部署补丁之前，往往要求团队成员创建一个与生产环境相似的测试环境，在测试环境中先应用安全补丁，观察系统性能、应用程序功能和兼容性，测试无误方能部署上线。

在安全管理方面，安全管理是该系统不可或缺的重要环节。安全管理以人员为核心，其成效取决于人员的安全意识与责任划分。依据笔者经验，安全管理体系也是一项体系工程，目的推动参与的人系统地开展安全生产工作。体系的完善和落实到位有利于避免安全生产事故的发生和应对突发情况。在笔者公司内部，为保障信息安全管理体系高效运行，定期会对 项目的信息安全状况进行评估与审计。通过各种技术手段对系统进行实时监控，如使用入侵检测系统、日志分析工具等及时发现和响应安全事件。对系统和网络的安全等进行定期检查，如扫描、漏洞评估等，确保安全控制措施的有效性。通过这种内部的安全优化机制，不断提升我司的信息安全防护能力和管理水平。

**结尾**

得益于系统安全架构的落地实施，项目的安全风险得到了有效控制。尽管在项目推进过程中经历了多次需求变更与数十次迭代升级，系统至今仍保持安全稳定运行，成功达成了设计预期目标。作为项目技术负责人，笔者深刻体会到，安全是保障和促进业务发展的，而业务发展过程中，必然有场地的变化、人员的变化、系统的变化、流程的变化、权限的变化，所以安全的力度、粒度都需要紧紧跟随业务的变化而变化，红线（底线）以上、安全需求（上线）以下才是我们安全的发展空间。

同时，随着制造业供应链竞争的日益白热化，数据泄露、恶意攻击等安全风险持续攀升，这决定了安全建设必然是一个动态演进的过程。我们需要持续跟进同行或外部环境并进行调整。此外，业务团队人员的流动可能会对既定的安全制度与措施执行产生影响。对此，笔者认为可从两个关键方向着手解决：其一，在业务团队中设置专职或兼职的安全接口人，通过常态化的沟通协作，确保安全要求能够精准传达与落实；其二，在业务管理过程中，融入的安全措施能够提供良好的用户体验，使得业务人员能够主动积极地拥抱安全，为安全站位。