芝士架构系统架构设计师论文押题

适用 2025 年 5 月 – 芝士架构凯恩编辑整理 v1.0.0

目录

1.历年论文题目

2024 年 11 月	论软件维护及其应用
	论面向服务的架构设计
	论多源异构数据集成方法
	论分布式事务及其解决方案
2024 年 5 月	论大数据 Lambda 架构
	论单元测试及其应用
	论模型驱动架构设计方法及其应用
	论云上自动化运维及其应用
2023 年 11 月	论面向对象分析的应用与实现
	论多数据源集成的应用与实现
	论软件可靠性评价的设计与实现
	论边云协同的设计与实现
2022 年 11 月	论基于构件的软件开发方法及其应用
	论软件维护方法及其应用
	论区块链技术及应用
	论湖仓一体架构及其应用
2021 年 11 月	论面向方面的编程技术及其应用
	论系统安全架构设计及其应用
	论企业集成平台的管理与应用
	论微服务架构及其应用
2020 年 11 月	论企业集成架构设计
	论软件测试中缺陷管理及其应用
	论云原生架构及应用
	论数据分片技术及其应用
2019 年 11 月	论软件设计方法及其应用
	论软件系统架构评估及其应用

	论数据湖技术及其应用
	论负载均衡技术在 Web 系统中的应用
2018 年 5 月	论软件开发过程 RUP 及其应用
	论软件体系结构的演化
	论面向服务架构设计及其应用
	论 NoSQL 数据库技术及其应用
2017 年 11 月	论软件系统建模方法及其应用
	论软件架构风格
	论无服务器架构及其应用
	论软件质量保证及其应用
2016 年 11 月	论体系系统架构评估其应用
	论软件设计模式及其应用
	论数据访问层设计技术及其应用
	论微服务架构及其应用
2015 年 11 月	论应用服务器基础软件
	论软件系统架构风格
	论面向服务的架构及其应用
	企业集成平台的技术与应用
2014 年 11 月	论软件需求管理
	论非功能性需求对企业应用架构设计的影响
	论软件的可靠性设计
	论网络安全体系设计
2013 年 11 月	论软件架构建模技术与应用
	论企业应用系统的分层架构风格
	论软件可靠性设计技术的应用
	论分布式存储系统架构设计
2012 年 11 月	论基于架构的软件设计方法及应用
	论企业应用系统的数据持久层架构设计
	论决策支持系统的开发与应用

	论企业信息化规划的实施与应用
2011 年 11 月	论模型驱动架构在系统开发中的应用
	论企业集成平台的架构设计
	论企业架构管理与应用
	论软件需求获取技术及应用
2010 年 11 月	论软件的静态演化和动态演化及其应用
	论数据挖掘技术的应用
	论大规模分布式系统缓存设计策略
	论软件可靠性评价
2009 年 11 月	论基于 DSSA 的软件架构设计与应用
	论信息系统建模方法
	论基于 REST 服务的 Web 应用系统设计
	论软件可靠性设计与应用

2.本年论文押题（4 篇）

凯恩根据系统架构设计师论文命题趋势，给大家提供了 4 篇有训练价值的题目，并且都给出了我亲自写的范文，这里的内容搜集了大量网上的真实案例进行创作，在可信度、严谨性和反模板（个性化上）上个人认为都做得相对较好。这里的题目你至少要根据自己的项目背景进行修改，至于原因我在红宝书论文宝典里都有具体说明。

2.1.软件测试——《论测试驱动的软件开发》

测试驱动开发是一种软件开发方法，它强调在编写实际的功能代码之前，先编写测试用例。开发过程是由测试驱动的，即先定义好预期的功能和行为，通过测试用例来描述，然后编写代码使测试通过。请围绕“论测试驱动的软件开发”论题，依次从以下三个方面进行论述。

1. 简要叙述你参与的软件开发项目以及你所承担的主要工作。
2. 请简要概述测试驱动开发有哪些优点。
3. 具体阐述在你所参与的项目中是如何进行测试驱动开发的。

适用主题	测试，测试驱动的软件开发，TDD
题目解析	近些年考试中不管是架构还是系分对测试的考察频率有点过于夸张，单元测试静态测试性能测试其实书本上讲得都不太多，但是论文题是每次都在出现。那么这次把测试开发引入作为未来的考点其实也说得过去。TDD 之前还是比较热门的概念，相关书籍和实践也非常多。这里关键是要积累测试用例的编写的一些依据或者说最佳实践，另外 TDD 的一些基本概念也要掌握。
摘要 300 字	为摆脱对于第三方电商平台的依赖，进一步满足消费者日益增长的多元化需求，为公司开拓新的营收增长渠道，2021 年 11 月我司开始自主研发电商平台。笔者作为技术负责人，主要负责整体方案设计和项目落地。在项目推进过程中，笔者发现，随着软件基础设施的不断演进，亟需一种能够提高代码质量的开发方法。而测试驱动开发，强调测试先行，完全可测试的代码和不断演化的简洁设计，能避免开发陷入代码越写越糟的恶性循环。笔者在该项目中通过设计测试用例，运行测试用例以及重构代码等测试开发实践，实现了项目的可靠交付。该项目于 2022 年 4 月初版正式上线之后，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 500 字	我司成立于 2015 年，在跨境电商领域已积累长达 9 年的行业经验。品牌始终致力于为全球消费者呈献极具魅力的快时尚服饰产品。一直以来，我司专注于构建服饰专业品牌线，并凭借自主研发深入耕耘该领域，从而确立了精准清晰的品牌定位。2021 年 11 月，公司管理层深入且全面地对未来电商战略格局展开研判，认为随着电商行业蓬勃发展，市场竞争愈发白热化，消费者的需求必定会呈现出高度多元化态势，这必将会给企业带来新的机会。为开拓新的增长渠道，摆脱对于第三方电商平台的依赖，公司决定通过自建电商平台，打造专属的销售与服务生态，直接触达消费者，实现业务拓展与品牌影响力提升。 该项目从平台规模和复杂度上来看，属于中等复杂度平台。主要包含如商品展示、购物车、支付系统和简单的订单管理等功能，另外根据业务需求还包含了会员系统（包括会员等级、积分、优惠券等）、支付集成、复杂的商品分类和搜索、营销工具（如限时折扣、满减活动等）等在内的复杂功能。项目初期阶段公司投入的团队规模为 15 人，后续随着业务扩大，团队成员增至 23 人。
技术方法说明 500 字	测试驱动开发（Test-Driven Development，简称 TDD）是一种软件开发方法论，它强调在编写功能代码之前先编写测试代码。在笔者的理解中，TDD 最大的好处是通过使用 TDD 可以显著提高代码质量。这是因为 TDD 确保每一行代码都有明确的测试覆盖，对比传统先

	开发后测试的做法，可以减少缺陷的引入，提高代码的可维护性和可靠性。第二是优化代码的设计。因为 TDD 迫使开发者在编写代码之前思考接口和对象之间的交互，例如笔者在设计一个电商订单系统时，通过测试用例明确订单、商品、支付等对象的交互，使系统各模块职责清晰，结构简洁，降低耦合度，进而更易于扩展和维护，提升整体设计质量。最后一个是笔者现在看来使用测试驱动开发最意外的好处。使用测试驱动开发，通过遵循短的红－绿－重构循环，你的大脑会得到大量的积极强化，这使得编程成为一项非常愉快的活动，并且这种感觉会在开发过程中得到持续的积极强化。尽管 TDD 在开发初期由于团队不适应可能会导致开发速度减慢，但随着时间的推移，它最终能够达到减少缺陷、提高可维护性，提高开发效率的目的。
主要内容 1000	在实践中，笔者发现测试驱动开发需要遵循一定的流程，即需求分析与测试用例编写，代码实现和代码重构。在此，笔者也从这三个方面展开，谈谈一些实践体会。 在需求分析与测试用例编写环节，最关键的步骤是需求理解与澄清，明确做什么，怎么做。要能够将复杂的业务需求分解为可管理的、具体的功能单元。引用 IEEE 标准（IEEE 软件需求规格说明书指南）来说，需求应该是具体的、可验证的。尽管这里涉及大量沟通和确认，但是明确的需求可以更好地指导后续的编码实现。在用例编写环节，考虑到 IDE 支持程度以及生态的丰富度，笔者所在团队主要使用 JUNIT 框架来实现单元测试。这个环节的关键是要确保测试用例的独立性和原子性。以笔者在该项目中遇到的某个订单计算场景为例，在设计订单计算用例时，应该只关注商品价格、数量、运费、折扣等因素，而不考虑比如折扣叠加规则在内的其他无关的因素。这块功能会由单独的折扣计算用例进行测试。通过将测试重点精准地聚焦在订单计算的核心要素上，避免测试用例的过度复杂和混乱，使得每个测试用例都能够独立，方便功能的实现和错误的定位。 在代码实现环节，这个阶段笔者的经验是不要过度设计代码，要以最快的速度实现功能，满足测试用例的要求即可。一般来说，过度设计的方案在前期需要花费大量时间进行复杂的架构搭建、数据库设计和代码编写，导致功能的开发周期变长，开发人员可能会在很长一段时间内都无法看到一个完整可用的功能，影响开发效率。所以笔者要求团队，在代码实现环节尽量不要做过度设计，如过度缓存数据、提前进行数据库索引优化等。特别是过度缓存数据会导致缓存的数据与数据源不一致。尤其在数据频繁更新的场景下，为了维护缓存和数据库一致性，会需要生产许多代码来做到这一点，不知不觉中就会让代码结构变得复杂，难以维护。

	代码重构是在 TDD 中容易被忽略的部分，实际上笔者认为这是最关键的部分，它让整个代码有了自我迭代更新的能力。马丁.福勒（MartinFowler）在他的著作《重构改善既有代码设计（第 2 版）》（下称《重构》）中提到，重构是对软件内部结构的一种调整，目的是在不改变软件可观察行为的前提下，提高其可理解性，降低其修改成本。因此重构的目的绝不是为了代码好看，最终一定是为了让系统的可维可测可扩展性更强，当然这必将带来生产力的提高。重构规模笔者有大型重构，常用的重构手段有分层、模块化、解耦、抽象、复用等。这类重构一般是架构重构。它的特点是代码改动多，影响范围大，修改难度大，耗时较长，当然风险也大。与之相对的是小型重构，如规范命名注释，拆分函数，提取重复代码，类调整等。它的特点是修改集中，操作性强，耗时短，风险小，它本质上不改变系统的架构。出于风险考虑，笔者一直强调，应当在日常开发中持续进行小规模重构（实际情况是大家常常不愿这么做），不然就会慢慢破坏系统架构从而累积为大型重构。而大型重构往往需要花费更多的人力，时间。
结尾 400 字	得益于测试驱动开发的实践，使得该项目虽经历了多次的需求变更，大大小小数十次迭代，但每次都能按时交付上线，取得了比较满意的结果。在实践中，笔者也深切感受到，TDD 不是银弹，它也存在一些不可忽视的问题。首先 TDD 由于测试先行，开发步骤相对来说比较繁琐，主要适用于功能明确、需求稳定的项目，对于需求变化大或探索性开发的场景不太适用。另外，假如开发团队习惯于“先写代码，后做测试”的传统流程，突然转向 TDD，就像是要求一名习惯右手写字的人改用左手。这种转变不仅需要时间，还需要整个团队的共识和决心。以笔者所在的项目组为例，起初团队对此抵触，但经过一系列内部培训和试点项目，团队成员逐渐意识到 TDD 带来的长期收益。所以对于想要落地 TDD 的组织，笔者建议定期的 TDD 研讨会和工作坊，邀请行业专家分享成功案例。小步快跑，从一个小模块或新特性开始实践 TDD，逐步扩大范围。

2.2.软件工程——《论敏捷开发方法及应用》

敏捷软件开发遵循一套软件开发的价值和原则，在开发中，需求和解决方案通过自组织跨功能的团队达成。敏捷软件开发主张适度计划，迭代开发，提前交付与持续改进，并且快速灵活地应对变更，作为敏捷开发方法之一，Scrum 方法被广泛应用。请围绕敏捷开发方法论题从以下 3 点进行论述。

1.概要叙述你参与管理和开发的软件项目，以及你在其中担任的主要工作。

2.请简要概述 Scrum 开发方法中的角色、工件和活动。

3.具体阐述你参与管理和开发的项目是如何基于 Scrum 敏捷开放方法进行系统开发的。

适用主题	敏捷，Scrum
题目解析	敏捷是目前常用的软件开发模型，系分考过，架构那么多年没有再考，但是案例出现过，所以仍然作为重点的，潜在的论文题目放这里给大家练习。敏捷方法众多，但是流行的只有 Scrum。**只要出题人还正常，就只能出这个题目**。所以这里按照 Scrum 来进行练习。你要知道 Scrum 的核心要素有哪些，有哪些角色配置，有哪些关键活动，每个活动做什么的。
摘要 300 字	为摆脱对于第三方电商平台的依赖，进一步满足消费者日益增长的多元化需求，为公司开辟新的营收增长渠道，2021 年 11 月我司开始自主研发电商平台。笔者作为技术负责人，主要负责整体方案设计和项目落地。在项目推进过程中，笔者发现，随着软件的复杂度不断提升，相应的管理手段也要发生改变才能突破研发效率瓶颈。而敏捷模型，以客户反馈、沟通协作为最主要的手段，比较适合我们彼时面临的项目快速迭代快速上线的建设要求。笔者在该项目中通过组织协同，项目过程控制，业务需求管理等敏捷实践实现了项目全生命周期的敏捷管理。该项目于 2022 年 4 月初版正式上线之后，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 500 字	我司成立于 2015 年，在跨境电商领域已积累长达 9 年的行业经验。品牌始终致力于为全球消费者呈献极具魅力的快时尚服饰产品。一直以来，我司专注于构建服饰专业品牌线，并凭借自主研发深入耕耘该领域，从而确立了精准清晰的品牌定位。2021 年 11 月，公司管理层深入且全面地对未来电商战略格局展开研判，认为随着电商行业蓬勃发展，市场竞争愈发白热化，消费者的需求必定会呈现出高度多元化态势，这必将会给企业带来新的机会。为开拓新的增长渠道，摆脱对于第三方电商平台的依赖，公司决定通过自建电商平台，打造专属的销售与服务生态，直接触达消费者，实现业务拓展与品牌影响力提升。 该项目从平台规模和复杂度上来看，属于中等复杂度平台。主要包含如商品展示、购物车、支付系统和简单的订单管理等功能，另外根据业务需求还包含了会员系统（包括会员等级、积分、优惠券等）、支付集成、复杂的商品分类和搜索、营销工具（如限时折扣、满减活动等）等在内的复杂功能。项目初期阶段公司投入的团队规模为 15 人，后续随着业务扩大，团队成员增至 23 人。

技术方法 说明 500 字	由于本系统复杂程度较高,且项目过程中面临较多的变更需求,所以在项目实施过程中,我们采用了 Scrum 敏捷开发方法。Scrum 作为敏捷方法论的重要组成部分,着重强调团队间的协作、灵活应对各种变化以及持续交付高品质的产品。这里的重点是,Scrum 包括了 3 个核心角色,涵盖产品负责人、敏捷教练以及开发团队。其中,产品负责人肩负着定义产品愿景以及管理待办事项列表的任务,敏捷教练他作为流程的专家,确保 Scrum 流程得到正确执行,保障团队遵循 Scrum 的原则和规则开展工作。开发团队则去完成具体的编码测试上线工作。在工作物件方面,Scrum 包含产品待办事项列表、冲刺待办事项列表以及增量。前者用于记载客户的需求以及重要特性,后者乃是在每个冲刺周期(通常为 2 至 4 周)内团队需要完成的具体任务,而增量则是每个冲刺结束时所产生的可交付成果。最后,Scrum 的主要活动包含冲刺计划会议、每日站会、冲刺评审以及冲刺回顾。这些流程主要就是为了加强团队之间的沟通协作,并且及时对问题进行复盘,做到早发现早解决。 Scrum 敏捷开发让团队以一个整体走完全程,产品开发过程从一个精心挑选的多学科团队的不断互动中产生。团队成员从开始到结束都在一起工作,该过程不是在定义好的、高度结构化的阶段中进行的,而是在团队成员的相互作用下产生的。

主要内容 1000	在实践中，其实有很多项目管理工具已经集成了常见开发模型。笔者所在团队使用禅道工具进行敏捷管理，它可以涵盖 Scrum 定义的所有开发过程。包括收集需求、规划冲刺、开发任务分解、执行冲刺、交付与验收 Sprint 以及回顾 Sprint 这六个环节。受限于作答篇幅，在此，笔者选取其中三个环节进行展开。 在收集需求环节，主要由产品经理负责根据业务需求的价值理顺需求要点，并按照顺序排列优先级。由于电商业务的多样化特性，笔者深切感受到，想要确保项目顺利交付，核心在于关键需求的挖掘。在这个过程中，笔者带领产品团队主要从用户调研和竞品分析两个角度展开。以竞品分享拆解，对新市场的主要竞争对手进行研究。竞品 A 以其超快速的物流配送服务赢得了用户口碑，例如在当地主要城市实现了当日达或次日达服务。竞品 B 推出了会员专属的虚拟社区，会员可以在社区内交流购物心得、分享商品使用体验并获得积分奖励，这一功能增强了会员的黏性和活跃度，该电商平台以此模式，打造了自己的会员社区。我们通过拆解竞品的核心亮点功能背后的设计深意，来判断此功能是否要在我们的项目中融入这些功能。 在开发任务分解环节，开发团队组织 Sprint 任务分解会议，由产品经理、敏捷教练参加。这个阶段的主要工作是对 Sprint Backlog 的每条进行开发任务分解，制定详细的开发计划。在笔者的实践中，我们将每项任务的工作量大小拆分标准为 2-4 工时，为每项任务指派负责人、预估任务的开始和结束日期（日期标准定为 2 天内）。这里最大的困难详细开发计划的粒度和实际工时的确定。如果开发计划粒度太粗，没有进一步细分，那么在实际开发过程中很难准确判断任务的进度。开发人员可能会对某项功能产生理解偏差，无法明确到底应该完成哪些部分。这使得项目经理和团队成员难以评估项目是否按计划进行。所以在实践中，笔者尽可能在设计阶段多考虑一些问题，把一些不明确的问题提前通过会议形式沟通掉，避免后续返工带来的进度延误。 到了执行冲刺环节，我们的主要任务就是完成本次冲刺的所有待办事项。在这个阶段，需要考虑的问题很多，包括团队通过每日站会进行任务进度更新、问题反馈与今日计划阐述，代码开发、单元测试、代码集成与联调工作，同时敏捷教练或产品经理要每日对比进度，推动团队内部及与外部相关方保持沟通协作以推进冲刺任务完成。以站会为例，在项目后期阶段，每晚八点，我们召开进度沟通会议，参会成员包括项目总负责人、各专项负责人及质量控制负责人，从协作的角度出发，审查当天的完成情况。针对遇到的问题，我们会探讨可行的解决方案，并根据项目进展调整第二天各团队的工作任务。为了防止会议流于形式，笔者给会议设立时间限制，对于合理把控每个问题的讨论时间。例如，对于一般性问题，讨论时间控制在 10 - 15 分钟；对于较为复杂的关键问题，最长不超过 30 分钟。这样可以促使参会成员在有限的时间内集中精力思考，提高讨论效率，避免因无休止的争论而浪费时间。

结尾 400 字	得益于 SCRUM 敏捷开发，使得该项目虽经历了多次的需求变更，大大小小数十次迭代，但每次都能按时交付上线，取得了比较满意的结果。在实践中，笔者也深切感受到，Scrum 能否成功实施，关键要先获得高层的认同和理解，让高层们理解 Scrum 的要义、利弊，如果 Scrum 能带来高效、优质的开发成果，那就在制定绩效结果并在实施过程中放权，让每个成员真正意识到项目成果是自己的事，而不是领导的事。如果是职能型的研发团队，同时也要获得各需求方的认可和支持，分享在这种方法下对整体的收益最大化，否则可能会面临各种不理解，最终可能导致实施失败。因此，我们要落地 Scrum 敏捷开发，就要做好“猪”与“鸡”两种角色之间心理上的平衡与和谐，“鸡爷爷”切不可把“小猪”们看成一群猪八戒，空有一身本领，但好吃懒做。“小猪”们也不可把“鸡爷爷”想象成周扒皮，只会半夜鸡叫，影响正常的开发进度。猪和鸡双方相互理解，达到项目开展过程中的平衡点，才能让整个项目顺利地完成。

2.3.安全设计——《论系统安全架构设计及其应用》

信息安全的特征是为了保证信息的机密性、完整性、可用性、可控性和不可抵赖性。信息系统的安全保障是以风险和策略为基础，在信息系统的整个生命周期中提供包括技术、管理、人员和工程过程的整体安全，在信息系统中保障信息的这些安全特征，并实现组织机构的使命。许多信息系统的用户需要提供一种方法和内容对信息系统的技术框架、工程过程能力和管理能力提出安全性要求，并进行可比性的评估、设计和实施。

请围绕“论系统安全架构设计及其应用”论题，依次从以下三个方面进行论述。

1.概要叙述你所参与管理或开发的软件项目，以及你在其中所承担的主要工作。

2.详细论述系统安全架构设计涉及的主要内容，面临的主要威胁，并说明其危害。

3.阐述你在系统安全架构设计的落地过程中通过什么手段解决了哪些问题。

适用主题	系统安全
题目解析	系统安全以前没有单独成一章来考，现在新教材单独拿出来考证明了它的重要性和优先权已经不容忽视了。从考察频率来看，最近一次考察在 2021 年 11 月，那再次出现的可能性非常之大。之前安全考察的是访问控制相关的内容，这个部分其实很难写出亮点来，因为点很小，想要丰富只能往下钻细节。假如再次考试，凯恩认为很可能从应用安全，系统安全等更高的纬度进行考察。大家在练习的时候一定要对这两块有针对性地积累素材。

摘要 300 字	为应对日益激烈的市场竞争和消费者的多元化需求，开辟新的增长渠道，我司于 2021 年 11 月启动了自建电商平台研发工作。笔者作为技术负责人，主要负责整体方案设计和落地。该项目工期紧难度大，在项目推进过程中，笔者发现，团队不仅仅要确保平台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。这一方面出于用户隐私等信息安全的要求，另一方面也是出于平台交易安全的要求。在该项目中，笔者着重从应用安全，系统安全，安全管理等三方面进行设计和规划，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。该平台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 500 字	我司成立于 2015 年，在跨境电商领域已积累长达 9 年的行业经验。品牌始终致力于为全球消费者呈献极具魅力的快时尚服饰产品。一直以来，我司专注于构建服饰专业品牌线，并凭借自主研发深入耕耘该领域，从而确立了精准清晰的品牌定位。2021 年 11 月，公司管理层深入且全面地对未来电商战略格局展开研判，认为随着电商行业蓬勃发展，市场竞争愈发白热化，消费者的需求必定会呈现出高度多元化态势，这将会给企业带来新的机会。为开拓新的增长渠道，摆脱对于第三方电商平台的依赖，公司决定通过自建电商平台，打造专属的销售与服务生态，直接触达消费者，实现业务拓展与品牌影响力提升。 该项目从平台规模和复杂度上来看，属于中等复杂度平台。主要包含如商品展示、购物车、支付系统和简单的订单管理等功能，另外根据业务需求还包含了会员系统（包括会员等级、积分、优惠券等）、支付集成、复杂的商品分类和搜索、营销工具（如限时折扣、满减活动等）等在内的复杂功能。项目初期阶段公司投入的团队规模为 15 人，后续随着业务扩大，团队成员增至 23 人。
技术方法 说明 500 字	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 1000	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。在笔者的理解里，SQL 注入本质上是由于程序在执行 SQL 语句时，用户输入的数据未被妥善处理，结果这些输入意外成为执行的代码。而实际上根据笔者多年经验，这个问题通过成熟的持久层插件就可以解决。而实现的原理归结到本质就是参数化查询。通过参数化查询，我们不会把用户的输入直接拼接进 SQL 语句里，而是把这些输入当作参数传递给 SQL 引擎，让 SQL 引擎预先编译 SQL 语句，再去执行，从而避免危险代码的注入。在电商项目中涉及大量需要对外展示评论的场景，这些场景往往会被别有用心之人使用来做跨站脚本攻击。这个场景中我们一般在后端采用 Java 的第三方库来对用户提交到数据库中的数据库进行编码，防止传到前端之后被浏览器解释为可执行的脚本。这样做的好处是一次性后端处理完毕即可，而不用前端每次都做 html“杀毒”工作。 在系统安全方面，我们主要关注的是操作系统层面的安全。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、特权操作环境和基础设施管理、操作系统补丁管理等。笔者在实践中，根据不同的应用场景分配不同的账户权限，设定最小特权原则，防止管理活动和特权账户被恶意行为者破坏。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，测试无误方部署上线。 在安全管理方面，安全管理是该系统必不可缺少的一个环节。安全管理是以人为主导从事安全生产工作，管得好坏在于人的意识和责任划分。依据笔者的经验，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，为确保信息安全管理体系的有

	效运行，定期会对项目的信息安全状况进行评估和审计。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 400 字	得益于系统安全架构的落地实施，本项目的安全风险得到了较好地控制。虽经历了多次的需求变更，大大小小数十次迭代，但该项目仍然安全稳定运行至今，取得了设计之初的预期结果。作为技术负责人，笔者也深切感受到，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，随着业务竞争越来越激烈，安全风险越来越高，所以安全是一个动态的过程，我们需要持续跟进同行或外部环境并进行调整。最后，业务团队人员的变动，可能会导致我们的安全制度和措施变样。那么笔者认为这要从两个方面考虑，一个是安全团队在业务团队有专职或兼职的接口人，加强部门间的工作沟通，另一个是在业务管理过程中，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。

2.4.云原生架构——《论服务网格在云原生架构中的作用及其实践》

服务网格 （ServiceMesh）是分布式应用在微服务软件架构之上发展起来的新技术，旨在将那些微服务间的连接、安全、流量控制和可观测等通用功能下沉为平台基础设施，实现应用与平台基础设施的解耦。这个解耦意味着开发者无需关注微服务相关治理问题而聚焦于业务逻辑本身，提升应用开发效率并加速业务探索和创新。换句话说，因为大量非功能性从业务进程剥离到另外进程中，服务网格以无侵入的方式实现了应用轻量化。

请围绕“论服务网格在云原生架构中的作用及其实践”论题，依次从以下三个方面进行论述。

1. 简要叙述你参与的软件开发项目以及你所承担的主要工作。

2. 简单阐述说明服务网格提供哪些常用功能。

3. 详细说明你所参与的软件开发项目中，如何使用服务网格实现应用轻量化。

适用主题	容器化，微服务，服务网格
题目解析	服务网格绝对是新技术新题目，书本上单独成章节进行介绍。这里作为重点给出来。这个题目曝光很少（目前来看）会的人不多，建议参照范文进行改写。这里你要注意的是，这个题目适用于多个主题（**容器化，微服务**），不要把它只看作服务网格。比如容器化其实也是个重点出题方向，那么除了实现容器的编排，部署之外，容器的监控是否可以用服务网格来做？微服务也是，传统的微服务做法有哪些，通过引入服务网格效果如何。但是你一定要记住，服务网格可以作为这些主题的一个亮点补充，但是切记不要跑题，张冠李戴。
摘要 300 字	为应对日益激烈的市场竞争和消费者的多元化需求，开辟新的增长渠道，我司于 2021 年 11 月启动了自建电商平台研发工作。笔者作为技术负责人，主要负责整体方案设计和落地。在项目推进过程中，笔者发现，在云原生应用中服务的管理和消息传递十分复杂。而服务网格可以通过提供服务熔断、重试、负载均衡、熔断降级等功能，来管理那些必须运行在复杂环境中的服务。在该项目中，笔者通过深入研究服务网格技术的原理与特性，结合项目实际需求，充分利用服务网格服务发现、流量管理等功能，使得项目在预期时间内高质量交付，各项性能指标均达到或优于预定标准。该平台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 500 字	我司成立于 2015 年，在跨境电商领域已积累长达 9 年的行业经验。品牌始终致力于为全球消费者呈献极具魅力的快时尚服饰产品。一直以来，我司专注于构建服饰专业品牌线，并凭借自主研发深入耕耘该领域，从而确立了精准清晰的品牌定位。2021 年 11 月，公司管理层深入且全面地对未来电商战略格局展开研判，认为随着电商行业蓬勃发展，市场竞争愈发白热化，消费者的需求必定会呈现出高度多元化态势，这将会给企业带来新的机会。为开拓新的增长渠道，摆脱对于第三方电商平台的依赖，公司决定通过自建电商平台，打造专属的销售与服务生态，直接触达消费者，实现业务拓展与品牌影响力提升。 该项目从平台规模和复杂度上来看，属于中等复杂度平台。主要包含如商品展示、购物车、支付系统和简单的订单管理等功能，另外根据业务需求还包含了会员系统（包括会员等级、积分、优惠券等）、支付集成、复杂的商品分类和搜索、营销工具（如限时折扣、满减活动等）等在内的复杂功能。项目初期阶段公司投入的团队规模为 15 人，后续随着业务扩大，团队成员增至 23 人。
技术方法 说明	云原生应用中的服务管理和消息传递始终是一个比较棘手的问题。以某个新服务升级为例，传统架构下，如果要做新产品上线的灰度发布，需要由业务代码中来判断哪些用户请求

500 字	可以导向新的服务版本，哪些导向旧版本。这种硬编码方式有很多缺点，诸如导致代码臃肿，策略变更不灵活等等。经过深入调研，笔者发现可以使用服务网格来解决这些问题。从本质上来看，服务网格是一个专用的基础设施层，用于管理服务之间的通信。相比传统的分布式解决方案，服务网格不管从配置上还是效果上都有明显优势。在笔者所在的项目中，主要使用服务网格的三大功能，分别是服务发现和注册，流量管理和可观测性。从服务发现与注册来看，服务网格能够自动感知服务实例的状态变化。它通过数据平面代理（如 Istio 中的 Envoy）来实时监控服务实例的上线和下线情况。在流量管理方面，服务网格可以通过设置特定的策略，实现自定义的流量分配，为我们实现诸如灰度发布，多版本路由提供了可能。在可观测性方面，服务网格能够收集三大类数据——Metric、Trace 和 Access Log，可以让我们轻松实现服务的调用依赖与性能分析、服务访问审计和错误排查工作，保障了云原生应用在复杂环境下的稳定、高效运行。
主要内容 1000	通过笔者在项目前期的技术调研，笔者发现 Istio 与 Kubernetes 紧密耦合，它是为云原生应用构建在 Kubernetes 之上的服务网格解决方案。由于之前在云原生的生态环境中，Kubernetes 已经成为笔者团队的容器编排标准，所以我们最终选择 istio 作为项目的服务网格底座。受限于作答篇幅，笔者将选取服务注册发现、流量控制和监控告警三个方面展开。 在服务注册发现环节，传统的做法要么是基于 DNS 和负载均衡器来做，要么是基于特定框架的服务注册中心来做。比如笔者团队之前常用 Eureka 来做服务注册中心。所有服务将自己的信息(如服务名称、IP 地址、端口等)注册到 Eureka 服务器，服务消费者从 Eureka 获取服务提供者的列表，并根据一定的规则选择一个服务实例进行调用。相比于传统做法，服务网格的优势比较明显，Istio 对服务注册与发现功能做了简化。当服务启动时，Envoy 代理会向控制平面注册自身，并上报其所在的服务实例信息。控制平面接收到这些信息后，会维护一个全局的服务注册表，记录所有服务的地址和端口等信息。当其他服务需要调用某个服务时，Envoy 代理会向控制平面查询目标服务的地址信息，并根据查询结果建立连接。在实际应用中，我们部署 Istio 后，无需手动配置服务的注册与发现，Istio 会自动识别并管理所有加入网格的服务。底层的原理是在 Kubernetes 环境中，Istio 可以通过与 Kubernetes API 的集成，自动获取 Pod 和服务的信息，并将其注册到服务注册表中。这样，无论是新服务的加入还是旧服务的下线，Istio 都能实时更新服务注册表，实现服务之间的通信。 在流量控制环节，笔者团队最常应对的场景就是多版本路由。以某个会员积分场景为例，一开始网站计划推出会员积分抵扣运费的活动以发展会员，策划了会员积分抵扣订单金额的

	新功能。当前部署的 order 服务由 v1 deployment 提供，没有运费抵扣的功能，而新开发了 order 服务的 v2 版本，有积分抵扣运费的功能。通过服务网格，我们可以通过判断请求的 header 中是否会员的 cookie 信息进行路由，会员路由至 order v2（有运费抵扣功能），非会员路由至 order v1（无运费抵扣功能）。Istio 通过 VirtualService 和 DestinationRule 这两个核心资源对象，为微服务架构提供了路由分发能力。其中 VirtualService 定义了服务的路由规则，通过 VirtualService 定义按流量特征进行路由，请求的 header-cookie 中 vip=false 时路由至 order 服务的 v1 subset，vip=true 时路由至 order 服务的 v2 subset。即会员的请求路由至 order v2，非会员的请求路由至 order v1，从而实现了多版本路由。 在监控告警环节，传统微服务监控方法主要关注服务自身的运行状态，对于服务之间的交互和整体系统的运行情况掌握程度有限。虽然笔者也尝试过通过一些分布式追踪工具（如 Zipkin）来跟踪请求在多个微服务之间的请求，但这种方法仍然无法完全掌握服务间的网络连接质量、服务间通信的详细协议等信息。笔者团队通过 Prometheus 收集 Envoy 代理上报的 Metric 数据，提供网格拓扑、服务拓扑、服务监控（请求数量、请求状态码分布、请求耗时、请求大小），然后把 Grafana 作为可视化工具，通过定制监控面板，将 Prometheus 收集的数据以图表的形式展现出来，实时监控服务网格的关键指标，比如笔者比较关注服务请求量，服务响应时间，服务间流量分布，通过掌握这些指标，可以反映出服务的整体健康状态。
结尾 400 字	得益于服务网格的实施，本项目在运维管理的成本上得到了极大改善。虽经历了多次的需求变更，大大小小数十次迭代，但该项目至今仍然稳定运行，取得了设计之初的预期结果。作为技术负责人，笔者也深切感受到，服务网格能否成功应用，关键在于分析项目的实际需求和预期收益，不能为了使用新技术和使用新技术。服务网格并非万能，它在为项目带来诸如流量治理，流量路由、熔断和限流能力的同时，也对团队的技术能力和运维经验提出了要求。在技术引入的时候，技术负责人不仅要充分考虑团队的技术能力和运维经验，同时也要考虑技术的成熟性和稳定性。只有在项目建设初期考虑周全，并且在项目实施的过程中对方法细节进行不断改进，才能让项目获得成功。