论软件系统架构风格

系统架构风格（System Architecture Style）是描述某一特定应用领域中系统组织方式的惯用模式。架构风格定义了一个词汇表和一组约束，词汇表中包含一些构件和连接件类型，而这组约束指出系统是如何将这些构件和连接件组合起来的。软件系统架构风格反映了领域中众多软件系统所共有的结构和语义特性，并指导如何将各个模块和子系统有效地组织成一个完整的系统。软件系统架构风格的共有部分可以使得不同系统共享同一个实现代码，系统能够按照常用的、规范化的方式来组织，便于不同设计者很容易地理解系统架构。

请围绕“软件系统架构风格”论题，依次从以下三个方面进行论述：

1.概要叙述你参与分析和开发的软件系统开发项目以及你所担任的主要工作。

2.分析软件系统开发中常用的软件系统架构风格有哪些？详细阐述每种风格的具体含义。

3.详细说明在你所参与的软件系统开发项目中，采用了哪种软件系统架构风格，具体实施效果如何。

范例

摘要部分：

本人于**年**月参与**项目，该系统为***，在该项目组中我担任系统架构师岗位，主要负责系统整体架构设计。本文以该车联网项目为例，主要讨论了软件架构风格在该项目中的具体应用。底层架构风格我们采用了虚拟机风格中的解释器，因该公交共有几十种不同的数据协议，使用解释器风格可以满足整车数据协议兼容性需求；中间层关于应用层的数据流转采用了独立构件风格中的隐式调用，以减低系统间耦合度、简化软件架构，提高可修改性方面的架构属性；应用系统层采用了 B/S 的架构风格，统一解决公交行业性难题“实施推广难、维护难”问题。最终项目成功上线，获得用户一致好评。

【注意：实际写作中相关项目情况应介绍清楚，摘要字数（包括标点符号）一般写 280 到 300 字】

正文部分：

随着**，**集团自**年**月起****，规划***。【项目背景内容可分 2 段写，第 1 段简要说明下项目来龙去脉】

年月我司**委托建设***项目。本项目****，我在项目中担任系统架构师职务，主要负责系统整体架构设计，该系统主要***，主要功能模块包括**。【第 2 段对系统整体情况进行细致介绍，项目背景第 1、2 段内容可以写 400 到 450 字】

在架构工作开始阶段，我们便意识到，架构风格是一组设计原则，是能够提供抽象框架模式，可以为我们的项目提供通用解决方案的，这种能够极大提高软件设计的重用的方法加快我们的建设进程，因此在我司总工程师的建议下，我们使用了虚拟机风格、独立构件风格以及 B/S 架构风格这三种较常用风格。虚拟机风格中的解释器架构风格能够提供灵活的解析引擎，这类风格非常适用于复杂流程的处理。独立构件风格包括进程通讯风格与隐式调用风格，我们为了简化架构复杂度采用了隐式调用风格，通过消息订阅和发布控制系统间信息交互，不仅能减低系统耦合度，而且还能提高架构的可修改性。B/S 架构风格是基于浏览器和服务器的软件架构，它主要使用 http 协议进行通信和交互，简化客户端的工作，最终减低了系统推广和维护的难度，以下正文将重点描述架构风格的实施过程和效果。

底层架构我们使用解释器风格来满足整车数据协议兼容性需求。解释器风格是虚拟机风格中的一种，具备良好的灵活性，在本项目中我们的架构设计需要兼容好 86 种不同 can 数据协议，一般来说这种软件编写难度非常高，代码维护难度压力也很大，因此这个解释器的设计任务便很明确了，软件设计需要高度抽象、协议的适配由配置文件来承担。具体的做法如下：我们对各个车厂的 can 数据结构进行了高度抽象，由于 can 数据由很多数据帧组成，每个数据帧容量固定并且标识和数据有明确规定，因此我们将 can 协议中的 ID 和数据进行关系建模，将整体协议标识作为一个根节点，以 canid 作为根节点下的叶子节点，使用 XML 的数据结构映射成了有整车协议链-数据帧-数据字节-数据位这 4 层的数据结构，核心的代码采用 jdom.jar 与 java 的反射机制动态生成 java 对象，搭建一套可以基于可变模板的解释器，协议模板的产生可以由公交公司提供的 excel 协议文档进行转换得到，解释器支持协议模板热部署，这种可以将透传二进制数据直接映射成 java 的可序列化对象，将数据协议的复杂度简化，后期数据协议更改不会对软件产生影响，仅仅更改协议模板文件即可，最终我们使用了 86 个协议描述文件便兼容了这些复杂的 can 数据协议，规避了 can 数据巨大差异带来的技术风险。

中间层我们使用独立构件风格中的隐式调用来简化构件间的交互复杂度，降低系统耦合度。主要的实现手段是我们采用了一个开源的消息中间件作为连接构件，这个构件是 apache 基金会下的核心开源项目 ActiveMQ，它是一款消息服务器，其性能和稳定性久经考验。由上文提到的解释器解析出对象化数据经过 ActiveMQ 分发到各个订阅此消息的应用系统，这些应用系统包括运营指挥调度、自动化机护、新能源电池安全监控等，这种多 web 应用的情况非常适合采用消息发布与消息订阅的机制，能够有效解决耦合问题，我们在编码的过程中发现只要采用这种风格的 web 应用，整个迭代过程效率极高，错误率降低，而且我们使用的 spring 框架，消息队列的管理完全基于配置，清晰简单，维护性良好，例如整车安全主题、运营调度主题、机护维修主题等消息队列分类清晰，可以随时修改其结构也能够随时增添其他主题的消息队列，不同的 web 系统监听的队列也可以随时变换组合，基于消息中间件的架构设计能够让系统的构件化思路得到良好实施，总体来说这种架构风格带来了非常清晰的数据流转架构，简化了编码难度，减低本项目的二次开发的难度。

应用系统层我们主要采用 B/S 的架构风格，主要用于解决公交推广难、维护难的问题。公交行业有一个明显的特点，公交子公司分布在全市各个地区，路途很远，且都是内网通讯，车联网络也是走的 APN 专网，一般是无法远程支持的，这给我们的系统推广以及后期维护带来了很大难题，我们可以想象如果使用 C/S 架构，更新客户端一旦遇到问题很可能需要全市各个站点跑一遍。这让我们在系统推广和维护方面面临较大压力。我们采用的 B/S 架构风格能够解决这个难题，并充分考量了现在相关技术的成熟度，例如现在的 html5 完全能够实现以前客户端的功能，项目中我们使用了大量的前端缓存技术与 websocket 技术，能够满足公交用户实时性交互等需求。这种风格中页面和逻辑处理存储在 web 服务器上，维护和软件升级只要更新服务器端即可，及时生效，用户体验较好，例如界面上需要优化，改一下 Javascript 脚本或者 CSS 文件就可以马上看到效果了。

项目于**年**月完成验收，这 1 年内共经历了 2 次大批量新购公交车辆接入，这几次接入过程平稳顺利，其中协议解释器软件性能没有出现过问题，消息中间件的性能经过多次调优吞吐量也接近了硬盘 IO 极限，满足当前的消息交互总量，另外由于我们的项目多次在紧急状态下能够快速适应 can 协议变动，得到过业主的邮件表扬。除了业主机房几次突发性的网络故障外，项目至今还未有重大的生产事故，项目组现在留 1 个开发人员和 1 个售后在维护，系统的维护量是可控的，系统运行也比较稳定。

不足之处有两个方面，第一在架构设计的过程中我们忽略 PC 配置，个别 PC 因为需要兼容老的应用软件不允许系统升级，这些电脑系统老旧，其浏览器不支持 html5，导致了系统推广障碍。第二在系统容灾方面还有待改善。针对第一种问题，我们通过技术研讨会说服业主新购 PC，采用两台机器同时使用方式解决。针对第二种问题我方采用了服务器冗余和心跳监测等策略，在一台服务暂停的情况下，另外一台服务接管，以增加可用性。

论云原生架构及其应用

近年来，随着数字化转型不断深入，科技创新与业务发展不断融合，各行各业正在从大工业时代的固化范式进化成面向创新型组织与灵活型业务的崭新模式。在这一背景下，以容器和微服务架构为代表的云原生技术作为云计算服务的新模式，已经逐渐成为企业持续发展的主流选择。云原生架构是基于云原生技术的一组架构原则和设计模式的集合，旨在将云应用中的非业务代码部分进行最大化剥离，从而让云设施接管应用中原有的大量非功能特性（如弹性、韧性、安全、可观测性、灰度等），使业务不再有非功能性业务中断困扰的同时，具备轻量、敏捷、高度自动化的特点。云原生架构有利于各组织在公有云、私有云和混合云等新型动态环境中，构建和运行可弹性扩展的应用，其代表技术包括容器、服务网格、微服务、不可变基础设施和声明式 API 等。

请围绕“论云原生架构及其应用”论题，依次从以下三个方面进行论述：

1.概要叙述你参与管理和开发的软件项目以及承担的主要工作。

2.服务化、弹性、可观测性和自动化是云原生架构的四类设计原则，请简要对这四类设计原则的内涵进行阐述。

3.具体阐述你参与管理和开发的项目是如何采用云原生架构的，并且围绕上述四类设计原则，详细论述在项目设计与实现过程中遇到了哪些实际问题，是如何解决的。

范例

摘要部分：

年月，我作为技术负责人参与了某市**平台建设项目，该项目主要包括**，打造**，构建**。本文将以**建设为例，阐述云原生架构在本项目中具体实践。基于云原生弹性原则，解决以往项目中系统资源无法根据需求进行弹性缩扩容的问题。基于云原生自动化原则，解决项目中应用上线周期长，复杂度高的问题。基于云原生服务化原则，解决以往系统建设中各系统共性功能重复建设的问题。通过云原生架构的成功使用，平台实现资源层和应用层的弹性伸缩能力，提升了平台对外提供服务的能力，及项目整体自动化水平。现在平台已经顺利上线运营，取得非常良好社会效益和经济效益。

【注意：实际写作中相关项目情况应介绍清楚，摘要字数（包括标点符号）一般写 280 到 300 字】

正文部分：

年月公司中标我市**项目，项目总投资**万元人民币，项目建设周期**年，我作为技术负责人全程参与了本项目。【项目背景内容可分 2 段写，第 1 段简要说明下项目来龙去脉】

本项目**，项目总体建设内容包括***，为***提供***；采用***，我们基于****技术路线，将平台****。【第 2 段对系统整体情况进行细致介绍，项目背景第 1、2 段内容可以写 400 到 450 字】

我们在项目中采用云原生架构，充分体现云原生架构的自动化、服务化、弹性，及可观测性四大原则。自动化原则，体现应用的自动发布，自动化代码检查，及自动化测试等自动流程；服务化原则，体现在云原生架构实现了“微服务，轻应用”，按需对外提供服务；弹性原则主要来源于云计算特有的对云资源的弹性伸缩能力；可观测性，通过相关采集系统和监测工具的使用，系统的可观测性进一步加强。接下来，我将基于本项目的实践着重论述自动化原则，服务化原则，及弹性原则。

一、自动化原则。以往的项目中上线一个应用，需要准备环境，搭建服务器，有时因为开发时没考虑上线时的需要，如浏览器及客户端等适配问题，经常导致上线工作周期长，问题多，经常返工。本项目中我们一方面采用了 K8S 来进行容器的编排和镜像的管理，基于容器技术实现了应用的自动化部署，及按需灵活缩扩容。为了实现自动化上线和自动运维，我在项目倡导了 DevOps 实践，开发人员从写下第一行代码开始，就要考虑相关后期上线及运维的工作，运维人员也在项目一开始就介入其中，使得从编码到上线更加顺畅高效。同时我也引入了相关自动代码检查，代码安全漏洞扫描工具，代替以往的人工走查和桌面检查，我们的测试团队也开发了相关自动测试工具，引入相关云服务商提供的自动压测工具，极大提升了团队的工作效率和项目质量。

二、服务化原则。以往系统建设中，多是竖井式建设，烟囱式系统，小而全，大而全，但工业互联网平台将要开发大量的工业应用，如果还是按照以前方式，一是工作量巨大，二是一定会有大量的重复建设，造成工期和成本的巨大浪费。项目中我们基于 SOA 的架构思想，采用现在流行的“微服务，轻应用”理念，采用 spring cloud+docker 的技术路线，我们将平台沉淀的相关设备管理、生产管理、供应链管理，及工业大数据分析能力拆分解耦，封装为一个个微服务，实现平台整体的“高内聚，低耦合”，这些服务通过微服务网关支撑各类上层工业应用服务建设，及工业 app 开发。通过这种方式，我们一方面实现了共性能力的复用，达到了集约化建设目标，另一方面也将平台能力通过微服务的方式开放给第三方开发者，将平台沉淀的相关数据分析，智能制造，设备管理能力对外赋能，为构建开放共享的区域工业互联网应用生态打下了坚实基础。

三、弹性原则。以往的系统平台建设中，如何按需对项目所需要的存储、计算、网络资源进行动态缩扩容，以及在面临海量高并发的情况下，如何能够实现应用服务的高可用性也是一直没有很好解决的难题。我通过打造两方面的能力，实现在云原生架构的区域工业互联网平台的弹性伸缩能力。为了解决这个问题，我们从三方面着手解决：一是在 IaaS 层通过 VMware 对存储，计算资源进行了虚拟化，构建了统一的虚拟化云资源池，并基于 penStack 打造了云资源服务统一管理平台，也就是区域工业云平台，基于 OpenStack 实现了云资源的弹性供给，动态管理，及按需缩扩容。二是在 PaaS 层通过 K8S，实现根据服务请求数量，实时调整相关微服务数量，使用系统就算在高并发条件下也具有非常高的可用性，比如在实际使用中我们设备平台的协议转换服务请求较多时，通过 k8s 及时上线相关备用服务，满足忙时业务需求，当闲时也可以非常方便的下线相关应用服务，释放系统资源，提升了平台服务层的弹性。

年月项目正式上线运营，目前平台已经为区域内 50 多家工业企业提供上云服务，接入工业设备超 200 台，沉淀工业大数据近 50PB，提供工业微服务近 200 个，打造了工业 app 近 30 个，初步构建了开放共享的区域工业互联网生态，取得了比较良好的社会效益和经济效益。

通过项目的实践，我深刻体会到云原生架构不仅是技术创新，更是组织与管理的变革，如果不能建立起开发、运维、及质量保证融合推进的组织架构，没有敏捷开发及快速迭代的文化，云原生架构带来自动化，服务化，弹性，及可观性优势都会大打折扣，流于形式。虽然项目中取得一定成绩，但也暴露了比较多的问题，项目初期由于大量采用开源的新技术和新架构，技术复杂开发难度大，导致了项目推进举步维艰，一方面我引进外部专家，开展多轮全员培训，并招聘相关专业技术人员进入团队；另一方面根据我们项目的实际情况，提出直接按需购买相关云厂商产品服务的方案。通过综合施策，项目得以顺利开展。项目的成功不是我一个人的功劳，在此向参与项目各位成员表示最真诚地感谢。凡是过往，皆为序章，我将继续保持空杯心态，不断学习提升自己的综合素养，继续为祖国的信息化事业贡献自己的绵薄之力。

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.详细论述安全架构设计中鉴别框架和访问控制框架设计的内容，并论述鉴别和访问控制所面临的主要威胁，并说明其危害。

3.阐述你在软件开发的过程中都遇到了哪些实际问题及解决方法。

范例

摘要部分：

本人于**年**月参与**项目，该系统是**。在该项目组中我担任架构师，主要负责整体架构设计和中间件选型。本文以该演进项目为例，主要讨论系统安全架构设计及其应用在本项目中的具体应用。本项目通过采用面部识别和证件照片验证等生物特征技术解决客户因为证件丢失,被他人盗用证件办理业务后用于从事非法活动的问题;通过采用基于角色的访问控制来限制营业员的业务受理范围和用户的业务受理范围；通过采用访问控制列表技术实现该运营商的网上商城业务，拓展了新业务领域，增加了该运营商的收入。项目最终顺利上线，获得一线员工和用户的一致好评。

【注意：实际写作中相关项目情况应介绍清楚，摘要字数（包括标点符号）一般写 280 到 300 字】

正文部分：

随着**的迅猛发展，需求的快速变化，**，经过认真调研后决定采用平滑演进的方式来实现**系统，因此**。【项目背景内容可分 2 段写，第 1 段简要说明下项目来龙去脉】

系统是业务的核心系统，包括了**等几十个菜单。该系统实现了**，整合了**技术。以**实现了该系统的**的建设，与甲方共建**多个专题建设。本项目组共**人，我在项目中担任系统架构师职位，主要负责整体架构设计及中间件选型，项目于**年**月成功启动。整个项目共耗时**个月，第一版于**年**月顺利通过验收。同时也实现了**等重点业务。【第 2 段对系统整体情况进行细致介绍，项目背景第 1、2 段内容可以写 400 到 450 字】

系统的安全架构设计主要有鉴别服务、访问控制、数据完整性校验、数据保密性和抗抵赖等 5 个方面，鉴于该运营商的业务特点，我们着重考虑了鉴别服务的能力和访问控制能力。鉴别技术主要有用户名+口令模式、数字证书和生物特征识别等技术。鉴别技术面临的主要目的是验明用户或者信息的正身，面临的威胁主要有用户名、口令被盗；生物特征被窃取、数字证书重放冒用等。访问控制主要有自主访问控制、访问列表控制、强制访问控制、基于角色的访问控制和基于任务的访问控制等技术。访问控制监控哪些资源可以被访问和使用，以及在某种情况下最大限度的降低未经授权行为的风险。无论是终端用户因疏忽而导致的敏感信息泄漏，还是因公共 web 服务器软件漏洞而导致的敏感信息泄漏，都可能造成灾难性后果。

在系统中，我们着重采用了生物特征识别方式来验证客户身份的准确性，采用访问控制列表和基于角色的访问控制来实现系统的访问控制技术。下文着重讨论这些技术在系统中的具体应用和实现。

一、生物特征识别

首先我们在受理业务的首页设置了客户认证菜单页，在客户认证通过的情况下才可以办理该客户下的业务，若认证未通过，则不允许受理任何业务。在用户受理业务时，我们首先要确保是实名办理业务，客户需要出示证件，系统调用第三方系统核实证件真伪，其次需要现场拍照，系统进行照片比对核对证件和证件持有人是否一致并将照片信息存储在后台 hbase 中，防止用户的业务被非法篡改，也防止别有用心的人利用账号从事非法活动，给国家和人民造成经济损失。只有当客户认证通过之后才可以正常办理业务。在受理业务之前通过生物特征识别校验用户的真伪性，在业务受理最后一个阶段需要采集用户的指纹或者在线签名，确保用户知道自己受理的具体业务内容，以及日后的防抵赖性。对于特殊类代办业务，系统需要验证代办人的证件资料，采集代办人的人脸信息，以及代办业务用户所收到的实时验证码，当验证码校验通过之后才运行受理该代办业务。

二、访问控制列表

访问控制列表主要是针对该运营商的网上商城业务而设计的，该运营商为了增加集团的收入，在手机 app 客户端和网上商城均开放了购物业务，用户可以使用话费购买商城的产品，比如京东卡、腾讯会员、生活用品等。由于账号和口令可能被暴力破解或者泄漏，所以如何在消费话费的时候确认该购买行为是该客户有意识的主动发起是我们遇到的最大的难题。因为客户可能是误操作，也可能账号是被盗取。还有可能是购买意愿不强烈，订单取消风险较大，但是某些商品如京东卡一旦售卖成功是不允许取消订单的，导致消费者的购买体验较差。经过慎重研究决定采用访问控制列表的方式来实现该购买行为。当购买行为进入支付阶段的时候系统会给用户发一条动态短信，若用户按照短信内容回复短信，则可以判断是用户的真实操作，才会进入真正的支付环节以及产品交付环节。该技术的应用成功拓展了业务领域，增加了集团的收入，同时也得到了用户的好评。

三、基于角色的访问控制

基于角色的访问控制，主要是针对营业厅营业员以及运维系统的运维人员，基于角色的访问控制可以限制营业员受理业务的范围，比如是否可以受理跨地市的业务，是否可以受理新上线的业务，是否可以给用户减免违约金，是否可以给予某种优惠等操作，一旦操作不当就有可能导致用户的投诉，给集团带来直接经济损失也会带来口碑上的下滑，不利于客户关系的维护。运维人员如果因误操作对用户数据造成不可恢复的损害，也将造成严重后果，所以需要实现“最小权限原则”和“权限分离原则”，将试图访问信息的用户只能访问执行自身角色职能所必须的那部分数据。实现基于角色的访问控制，还有一大好处就是便于对营业员和运维人员账号的维护，当员工需要某个权限的时候，只需要在该工号上配置上一条相应的角色就可以实现营业员账号权限的扩展，提供给工号管理系统的就是一个选项的按钮而已，可以以最快的时间实现营业员权限的扩展和收缩。

通过上述鉴别服务和访问控制方案的实施，该系统于**年**月完成第一个版本的顺利验收。上线后虽然出现了某些特殊用户到某个营业厅办理不了某些业务需要到指定的营业厅才能办理的场景，但是并没有发生角色越权行为，没有给集团带来经济损失。项目至今仍然保持着每月一次或者两次的版本迭代，顺利实现了 1.0 到 2.0 架构演进的平滑式过渡，得到了甲方的肯定，实践证明鉴别服务和访问控制方案适用于该项目安全架构的实现。

当然，在系统的后期运行维护过程中，也陆续发现了一些问题和不足，在服务调用链上没有做访问量控制，没有做调用链剥离，当推广的某一业务量突然上来后，会导致调用链崩溃而不可用，也会因为运维修复大量数据重跑某些业务，导致关键业务崩溃，波及实时业务。同时也导致了很多异常数据，给运维人员增加了工作量。在后续的改进方案中对重点调用链进行了剥离，并做了访问量控制。运维大批量的数据修复工作只能在晚上执行。运维人员对数据库的一切增删改查操作均需要有日志记录，并定期将该运维人员的操作日志发给该运维人员确认是否是本人操作，确保生产数据的安全性。

高可靠性系统中软件容错技术的应用

容错技术是当前计算机领域研究的热点之一，是提高整个系统可靠性的有效途径，许多重要行业（如航空、航天、电力、银行等）对计算机系统提出了高可靠、高可用、高安全的要求，用于保障系统的连续工作，当硬件或软件发生故障后，计算机系统能快速完成故障的定位与处理，确保系统正常工作。

对于可靠性要求高的系统，在系统设计中应充分考虑系统的容错能力，通常，在硬件配置上，采用了冗余备份的方法，以便在资源上保证系统的可靠性。在软件设计上，主要考虑对错误（故障）的过滤、定位和处理，软件的容错算法是软件系统需要解决的关键技术，也是充分发挥硬件资源效率，提高系统可靠性的关键。

请围绕“高可靠性系统中软件容错技术的应用”论题，依次从以下三个方面进行论述。

1．简述你参与设计和开发的、与容错相关的软件项目以及你所承担的主要工作。

2．具体论述你在设计软件时，如何考虑容错问题，采用了哪几种容错技术和方法。

3．分析你所采用的容错方法是否达到系统的可靠性和实时性要求。

范例

摘要部分：

年月，本人在公司参加了**系统的开发。该项目主要功能是做**业务。本人在该项目中担任了系统架构师职位，负责该系统的架构设计工作。本文以该项目为例，主要论述了高可靠性系统中软件容错技术的应用。为了防止硬件和软件的单节点故障问题，我们采用了集群部署的方式来提高软硬件服务的可靠性；数据库是所有业务服务都依赖的基础服务，我们采用了主备部署的方式提高数据库的可靠性；为了提高单个业务服务软件健壮性与可靠性，我们采用了防卫式程序设计、降低复杂度设计和错误日志记录的方案来提高服务的可靠性。通过上述方案的使用，保障了服务的高可靠性，最终项目顺利上限，持续稳定运行。

【注意：实际写作中相关项目情况应介绍清楚，摘要字数（包括标点符号）一般写到 280 到 300 字】

正文部分：

某公司**为了应对***，因此**研发**系统。【项目背景内容可分 2 段写，第 1 段简要说明下项目来龙去脉】

年月，本人在公司参加了**系统的开发。该系统的核心业务场景是**。该系统的主要功能模块有**等。**系统为提供了**；**系统提供了**；**系统提供了**等。本人在该系统中担任系统架构师职位，负责该系统的架构设计工作。【第 2 段对系统整体情况进行细致介绍，项目背景第 1、2 段内容可以写到 400 到 450 字】

系统可靠性包括硬件可靠性和软件可靠性，两个方面都需要我们考虑到，软件的复杂度比硬件高，所以软件系统发生故障的概率也会高于硬件系统。硬件在使用时间过长的情况下，物理退化的现象加重，故障发生概率就会变高，通常我们会采用主动冗余、监控检错配合报警的方案来解决。软件系统包含各种的业务逻辑，复杂度较高，故障概率高，通常我们采用防卫式程序设计、N 版本程序设计、降低复杂度设计等方案来提高单个服务的自身可靠性；单个服务的可靠性不可能达到百分百，总归避免不了特殊场景下导致服务崩溃无法提供服务支持的情况，针对这一问题通常采用集群部署配合监控报警的方案来解决，即使一台服务崩溃，还有其他服务提供者进行业务支持。合理的采用容错技术方案，可以大大提高服务的可靠性。

在系统中，我们采用了多种容错技术来提高系统的可靠性，下文着重讨论应用系统集群部署、数据库主备部署和程序设计方面在该项目中容错方面具体的应用和效果。

一、系统集群部署容错

单个硬件设备或者说单个软件服务进程，都不可避免的会发生故障导致服务中断。如果单个硬件设备上部署一个或多个软件服务进程，当硬件发生故障，所有软件服务都将不可正常服务；如果一个软件服务就部署了一个进程在提供服务，一旦这个软件进程发生故障，服务也将不可正常运行。针对上述问题，集群部署方案可以很好的解决。同一个软件服务，在不同的多台物理设备上做集群部署，即使一台物理设备发生故障或者一个软件服务进程发生故障，都有其他的多台物理设备和软件进程在提供服务，达到服务永不中断的目的。集群部署就要考虑到请求分发和状态的问题，因为我们的业务是 pos 机交易，走的是 TCP/IP 协议，采用了硬件 F5 做负载分发业务；每个业务进程都尽量做成无状态服务，实在需要处理状态的，我们采用了 redis 集群来保存状态数据的方案。发生问题的物理设备和软件服务要尽快被发现然后修复，我们采用了 zabbix 分布式服务监控系统来实现，一旦监控发现问题，立马发送邮件给相关维护人员进行处理。通过集群和监控报警的配合，大大提高了整个系统的可靠性和实时性。

二、数据库主备部署容错

数据库服务是整个收单交易系统的基础服务，每个业务模块都要使用到，所以一旦数据库出现问题，整个服务将中断，影响重大。单个数据库服务不可能说永远不可能出现故障，我们要做到的首先是提升单个数据库服务的可靠性，其次是如果单个数据库服务发生故障，立刻快速使用备用数据库继续支持服务。针对上述问题我们采用了数据库主备部署方案，主备都同时提供服务，每台服务的压力就会减少，提高单台数据库服务的可靠性。主库服务于全部的写操作和少量的读操作，比如说交易业务；备库服务于实时性要求不高的读取操作，比如说内部管理平台查看流水、构建各种报表业务。一旦主库发生故障无法提供服务，备库可以快速的升级为主库，继续提供服务，保证了整个数据库系统的整体可靠性。在数据库的物理文件保存方面，通过磁盘阵列技术做到磁盘容错，防止数据丢失。通过数据库主备部署和磁盘阵列方案，大大提升了数据库服务的可靠性。

三、程序设计方面容错

程序设计方面我们采用了防卫式程序设计以及降低复杂度的策略。防卫式程序设计方面首先是报文字段的检测，报文中各个参数按照预定的规则先进行检测，检测通过后再进行后续的业务处理，比如说交易金额，必须是一个正整数，比如说商户编号必须为固定 15 位长的字符串，参数检测通过后会大大降低后续业务出现不可控制的错误的概率；其次是异常处理，程序正常逻辑要处理，发生异常的情况下我们也要考虑如何处理，比如说查看商户法人证件照片，读取文件的过程中很可能会出现文件未找到的错误，一旦发生这个错误我们就要按照异常的流程处理，提示说照片文件未找到，不能进行处理；再次是进行错误信息日志记录，程序发生错误时，把相关的详细信息都记录到文件中，方便后续查找定位问题。整体的系统设计我们都采用了合理的架构模式，遵循高内聚、低耦合、单一职责、依赖倒置、接口隔离等基本的设计原则，这些都会降低整个系统的复杂性，增加系统的可靠性。

该项目于**年**月完成测试并上线，通过上述容错方案的实施，整个系统持续稳定的运行，未出现过服务中断的故障，即使系统变更上线，也可以通过集群优势，逐台设备更新上线部署，达到服务不中断的效果。后续我们又完善了 zabbix 监控系统，丰富了监控的内容，包括硬件设备监控、网络流量监控、软件进程监控、软件业务数据监控等，实现了更加全面的监控策略。错误日志记录追踪发挥了良好的作用，定期我们会排查一遍错误日志，从中分析出系统存在的问题再进行修复，进一步完善系统，进一步提高系统的可靠性。

当然，在我们设计系统和后期运行维护过程中，也陆续发现了一些问题和不足，比如说集群部署导致运维成本增加，后期我们增加了 jekins 自动部署的功能来解决一些问题。另一个问题就是单个服务不可用时，还是需要人工来处理，不能及时的自动新增一个服务节点上去，目前我们正在研究使用容器技术来解决该问题，等成熟稳定后也会陆续投产使用。

论基于架构的软件设计方法（ABSD）及应用

基于架构的软件设计（Architecture-Based Software Design, ABSD）方法以构成软件架构的商业、质量和功能需求等要素来驱动整个软件开发过程。ABSD 是一个自顶向下，递归细化的软件开发方法，它以软件系统功能的分解为基础，通过选择架构风格实现质量和商业需求，并强调在架构设计过程中使用软件架构模板。采用 ABSD 方法，设计活动可以从项目总体功能框架明确后就开始，因此该方法特别适用于开发一些不能预先决定所有需求的软件系统，如软件产品线系统或长生命周期系统等，也可为需求不能在短时间内明确的软件项目提供指导。

请围绕“基于架构的软件开发方法及应用”论题，依次从以下三个方面进行论述。

1.概要叙述你参与开发的、采用 ABSD 方法的软件项目以及你在其中所承担的主要工作。

2.结合项目实际，详细说明采用 ABSD 方法进行软件开发时，需要经历哪些开发阶段？每个阶段包括哪些主要活动?

3.阐述你在软件开发的过程中都遇到了哪些实际问题及解决方法。

范例

摘要部分：

年月，某集团公司开始了**系统的开发，该系统主要实现**等功能。主要包括**等主要功能模块。我在该系统中担任系统架构师，主要负责该系统的架构设计工作。本文以该系统为例，主要论述了基于架构的软件设计方法在该系统中的具体应用。在架构需求阶段，以构件图、包图、类图和对象图描述系统结构，为系统整体的上层结构进行建模；在架构复审阶段，采用架构权衡分析法来对架构设计方案进行复审，以应对评估质量属性的满足程度问题；在架构演化阶段，采用需求变动管理工具对收集的需求进行归类，以应对需求变动管理问题。经过多轮迭代演化，本系统最终顺利上线，并运行稳定，获得用户一致好评。

【注意：实际写作中相关项目情况应介绍清楚，摘要字数（包括标点符号）一般写 280 到 300 字】

正文部分：

某集团公司**为了应对***，因此**研发**系统。【项目背景内容可分 2 段写，第 1 段简要说明下项目的来龙去脉】

该项目在**年**月正式启动，旨在通过**，优化**。我在该项目中担任架构师，并负责整体系统的架构设计工作。本系统主要由**等主要模块构成。此外，该系统能**，通过如**得到**。**从而最终实现加强集团公司在银行业的竞争力的目的。【第 2 段对系统整体情况进行细致介绍，项目背景第 1、2 段内容可以写 400 到 450 字】

基于架构的软件设计方法，简称 ABSD 方法，主要包含了架构需求、架构设计、架构文档化、架构复审、架构实现和架构演化六个阶段。其中，架构需求主要包含了需求获取、标识构件、需求评审等活动。架构设计，主要包括提出架构模型、映射构件、分析构件相互作用、产生架构、设计评审等活动。架构文档化，用于把架构设计的成果进行分析与整理并进行文档化。架构复审，用于对架构设计进行复审并标识其中潜在的风险、缺陷和错误。架构实现，包括了架构分析与设计、构件实现、构件组装、系统测试等活动。架构演化，主要活动包括需求变化归类、架构演化计划、构件变动、更新构件的相互作用、构件组装与测试、技术评审等活动。

考虑项目建立初期的需求不稳定，且后续的需要应对大量的新需要，我们决定使用 ABSD 方法来对系统进行构建与演进。下面将着重描述 ABSD 方法在本项目应用过程中所遇到的问题和采取的应对方案。

一、架构需求阶段，以构件图、包图、类图和对象图描述系统结构

为了应对架构需求阶段描述系统架构结构的问题，我们在标识构件活动中，使用构件图、类图等来对系统的整体结构进行建模，实现了通过标识构件活动描述架构结构的目的。首先，我们根据获取的需求以类图和包图的形式为系统的下层结构进行建模。若某个构件的业务构成较为复杂，则以对象图进行辅助说明。然后，我们对这些类图和包图进行分组，从而拟定构件的边界。接着，我们安排项目负责人基于上一步得出的类图和包图的分组，并以需求获取活动得到的会议记录和用例图作为辅助，使用构件图设计出需要的构件，再使用这些构件为系统整体的上层结构进行建模。最后，我们使用《需求构件关系表》标识并记录构件与需求之间的关系，用以指导后续的工作。通过这个方案，我们在标识构件活动中，使用构件图、包图、类图等结构更为清晰的视图为系统的整体结构建立了一个更为稳定的基线，从而达到了提升需求到系统设计的转化效率的效果。

二、架构复审阶段，使用 ATAM 对架构的质量属性进行复审

为了应对对架构设计方案的质量属性满足程度进行评估的问题，我们在架构复审阶段，使用架构权衡分析法（简称 ATAM）来对文档化后的设计方案进行复审，从而实现了确定非功能性需求在本系统的架构设计中是否得到满足的效果。首先，考虑到参与 ATAM 会议的与会人员来自于不同领域的部门，为了便于会议的进行，我们在 ATAM 的描述和介绍阶段中，为相关人员提供 ATAM 评估方法、系统业务动机和架构整体设计作为知识基础。然后，我们在 ATAM 的调查和分析阶段中，使用基于质量属性的评估方法（如建立质量效用树等），对评估系统架构满足非功能需求的情况进行评估。接着，我们在 ATAM 的测试阶段中，基于场景的验证方案来对上一步评估结果进行验证，从而通过质量场景和场景的优先级对架构方案进行调整。通过这个方案，我们在架构复审阶段中，使用 ATAM 来为架构设计方案的质量属性进行评估，从而确定非功能需求在架构设计中的满足程度并通过质量场景调整架构方案的目的。

三、架构演化阶段，对需求变化进行归类

为了应对架构演化阶段的需求变动管理问题，我们在需求变化归类活动中，使用数个工具对收集的需求变动进行整理与分类，实现了对需求变动与构件之间的关系进行管理的目的。首先，我们把新的需求变动登记到《需求管理列表》中，并通过该表获得新的需求变动与旧需求的对应关系。而当某个新的需求变动和旧需求建立起关系时，我们可以通过上次迭代生成的《需求构件关系表》和《开发工作登记表》，推导出新需求变动与系统已有构件及这些构件的开发人员的对应关系。然后，把新需求与系统中已有构件和候选处理人员的对应关系回填到《需求功能关系表》中。而当某个新需求的需求点和旧需求无法建立起关系时，我们直接把新需求写到《需求构件关系表》中并标记为新增。然后我们再通过召开相关干系人的会议，议定每个变动的实现优先级和跟进人员。通过这个方案，我们不但达到了对需求变动与构件之间的关系进行管理的目的，还能划定较适合处理这些需求变动的候选人名单以指导后续工作之用。

由于使用了以上 ABSD 方法来对项目进行开发与演进，该系统改进工作十分顺利，经过多轮迭代演化，本系统最终于**年**月顺利上线，并运行稳定，用户普遍反馈功能可以满足工作需要且使用体验良好。实践证明 ABSD 方法适用于该项目的变动管理。

然而，我们在应用 ABSD 方法的过程中还存在着一些不足，如我们发现因人手更替，导致一些构件的跟进人员产生变动，这将会产生额外的学习成本。因此，我们决定在后续的演化中，每个构件都需要维护一份更详尽易懂的文档，并在技术评审阶段对本轮迭代中的这些文档进行审核来缓解这个问题。通过本次项目，我认识到紧贴业务需求、数据和开发小组的构成情况来选择具体的软件设计方法将会大大地影响一个项目的演进能力。而在选定软件设计方法后并不意味着结束，通过对方法的细节进行不断改进，将是项目成功的关键。