

应用系统

国产适配迁移方案

拟制人 _____

审核人 _____

批准人 _____

目录

1 系统建设理解	6
1.1 背景.....	6
1.2 建设内容.....	6
1.3 应用开发重点难点理解.....	6
1.3.1 应用开发重点难点理解.....	6
1.3.1.1 应用系统开发周期短，任务重.....	6
1.3.1.2 应用系统数据迁移风险分析及应对措施.....	7
1.3.1.3 信创产品更新换代技术风险.....	8
1.3.2 项目实施管理重点难点理解.....	9
1.3.2.1 组织风险及应对措施.....	9
1.3.2.2 管理风险及解决方案.....	9
2 需求分析	10
2.1 硬件及基础软件需求.....	10
2.1.1 基础硬件.....	10
2.1.2 基础软件.....	11
2.2 应用系统功能需求.....	12
2.2.3 应用系统.....	12
3 系统建设方案	12
3.1 总体目标.....	12
3.2 整体安排.....	12
3.3 总体架构设计.....	12
3.3.1 设计原则.....	12
3.4 技术架构设计.....	14
3.4.1 技术路线.....	14
3.4.2 技术架构.....	15
3.5 系统部署方案.....	17
3.5.1 部署说明.....	17
3.5.2 系统部署工作流程.....	17

3.6 适配验证方案	18
3.6.1 适配改造主要问题	19
3.6.2 系统适配重难点	19
3.6.3 适配方案设计	20
3.6.3.1 环境搭建和应用迁移	20
3.6.3.2 全面适配测试	20
3.6.3.3 适配改造	21
3.6.3.3.1 需求分析和系统设计	21
3.6.3.3.2 系统开发（适配改造）	22
3.6.3.3.3 系统调优	22
3.6.3.4 应用部署和数据迁移	22
3.6.3.5 适配验证	23
3.6.3.6 试运行和正式运行	23
3.6.4 终端适配改造	23
3.6.4.1 适配国产化浏览器	23
3.6.4.1.1 系统表单控件从前端渲染改为后端渲染输出	23
3.6.4.1.2 表单控件加载数据源模式修改	25
3.6.4.1.3 首页及打开列表的方式改为 TAB 页模式	25
3.6.4.1.4 列表从 easyUI 改为 Jgrid	26
3.6.4.1.5 Js 文件合并及延时加载	26
3.6.4.1.6 浏览器兼容性修改	29
3.6.4.2 适配国产化操作系统	29
3.6.4.3 适配国产化文字处理软件	29
3.6.5 服务器端适配改造	30
3.6.5.1 适配国产化数据库改造	30
3.6.5.2 适配国产化中间件改造	30
3.7 系统迁移方案	30
3.7.1 数据迁移方案	30
3.7.1.1 数据迁移目标	30
3.7.1.2 数据迁移原则	31
3.7.1.3 数据迁移范围	32
3.7.1.4 项目迁移方法	32
3.7.1.5 数据迁移过程中需注意的问题	33

3.7.2 应用系统迁移	33
3.7.2.1 研发测试阶段	33
3.7.2.2 上线试运行阶段	33
3.7.3 业务切换阶段	34
3.7.3.1 硬件环境	34
3.7.3.2 软件环境	34
3.8 关键技术	35
3.8.1 向国产数据库管理系统的迁移	35
3.8.2 向国产办公软件的迁移	36
3.8.3 应用系统性能优化	37
3.8.4 迁移后系统与现有系统的兼容	38
3.9 测试方案	39
3.9.1 准入要求	39
3.9.2 准出要求	39
3.9.3 进度计划	41
3.9.4 人员安排	42
3.9.5 功能测试	43
3.9.6 性能测试	43
3.9.6.1 单交易基准测试	44
3.9.6.2 单交易负载测试	44
3.9.6.3 混合容量测试	44
3.10 工作进度计划	45
4 项目管理	46
4.1 项目实施组织机构	46
4.1.1 项目组织原则	46
4.1.2 项目组织体系	47
4.1.3 项目组织结构	47
4.1.3.1 项目领导小组	48
4.1.3.2 专家顾问组	49

4.1.3.3 项目经理	49
--------------	----

4.1.3.4 建设组	50
4.1.3.4.1 需求分析小组	50
4.1.3.4.2 系统设计小组	51
4.1.3.4.3 系统开发小组	51
4.1.3.4.4 系统测试小组	52
4.1.3.5 培训组	53
4.1.3.6 文档管理组	53
4.1.3.7 技术服务组	53
4.1.3.8 质量保障组	54
4.1.3.9 商务组	55
4.1.4 公司内部资源支撑	55
4.2 项目进度管理方案	55
4.2.1 实施进度计划	55
4.2.2 进度承诺及保证措施	57
4.2.2.1 合理的资源配置	57
4.2.2.2 严格进行进度控制	58
4.2.2.3 做好进度偏差管理	61
4.3 项目沟通管理方案	62
4.3.1 沟通的组织保障	62
4.3.2 渠道和方式	62
4.4 项目变更管理方案	63
4.5 风险管理方案	64
4.5.1 项目风险的产生	64
4.5.2 风险管理定义	65
4.5.3 风险管理原则	65
4.5.4 风险管理目的	66
4.5.5 风险管理机制	66
4.5.5.1 监督机构	67
4.5.5.2 长期的合作关系	67
4.5.5.3 项目全过程的监督	67
4.5.6 项目风险识别的技术与工具	67

4.5.6.1	风险矩阵	67
4.5.6.2	检查表	68
4.5.6.3	流程图	69
4.5.7	项目风险应对	69
4.5.7.1	项目风险应对概述	69
4.5.7.2	项目风险应对过程	70
4.5.7.3	项目风险应对策略	70
4.5.7.4	项目风险应对技巧	71
4.5.8	项目风险监控	71
4.5.8.1	项目风险监测与控制	71
4.5.8.2	风险监测与控制的工具与技术	72
4.5.8.3	技术风险的识别	73
4.5.8.4	技术风险的量化	73
4.5.9	风险管理和规避	75
4.5.9.1	事前控制—风险管理规划	75
4.5.9.1.1	风险形势评估	75
4.5.9.1.2	风险识别	76
4.5.9.1.3	风险分析和评价	76
4.5.9.2	事中控制—风险管理方法	77
4.5.9.2.1	风险监视	77
4.5.9.2.2	风险规避	77
4.5.9.3	事后控制—风险管理报告	78

1 系统建设理解

1.1 背景

信息技术创新领域的建设是落实“世界科技强国”的重要行动，更是推动经济发展的重要抓手。近些年来来自国际的威胁国家安全的事件案例不断发生，如何有效应对不断变换的国际局势对我国的冲击，逐步推进“科技强国”的国家方针，是信息行业持续探索的主要目标。

我国已推进信息技术创新工作多年，在各个领域均有不菲成果，已经在信息技术创新领域获得初步成果，实现软硬件设备的创新及使用。在此过程中，金融行业作为国家经济产业的核心之一，必然需要逐步完成信息技术创新工作，打破国际厂商对我国金融科技的垄断。

1.2 建设内容

在现有资源基础上，围绕“替、迁、改、建”开展全面建设，主要建设任务包括：

1. 信息化设备替代：对现有服务器、中间件、数据库等相关软硬件进行安全可靠替换，替换后的产品全部采用国产自主可控技术。

2. 应用系统的适配、改造及迁移：对现有业务应用系统进行开发或国产化适配改建后，迁移到国产化服务器中。

本项目将应用系统迁移到国产软硬件环境，达到国产自主可控的总体目标。

1.3 应用开发重点难点理解

1.3.1 应用开发重点难点理解

1.3.1.1 应用系统开发周期短，任务重

本项目应用系统开发周期短，任务重。因此制定合理的实施方案和路径是本次安可替代项目的难点之一。

解决方案：根据用户要求提供的实施总体方案制定详细实施方案。提供清晰、科学并满足用户时间进度要求的实施路径，充分分析系统需求和适配改造对原有相关系统可能造成影响和实施过程中可能遇到的问题，并提供解决方案，保障系统的平滑过渡，不降低原有功能和用户体验，提升运维保障的信息化水平。充分考虑组织实施重难点，调用技术骨干人员组建成熟的项目实施团队，分组分批并行进行项目开发实施，集中解决应用系统开发、外部系统对接等建设重难点和安全保障重难点，并提供相应解决方案，严格按照项目实施计划对项目进行监督管理，保障项目按预计时间运作。

1.3.1.2 应用系统数据迁移风险分析及应对措施

针对可能出现的数据源风险、数据质量风险、数据迁移时间风险、数据量风险等风险制定应对策略。

1. 数据源风险

将这些不同数据源的数据汇集到国产数据库中，是一个非常复杂的工作。建议解决该问题的办法首先是将不同异构数据源先通过一定的技术手段装载到同种类型的数据库中，然后再进行相应的数据转换工作。

2. 数据质量风险

数据质量风险主要包括有原始数据质量差、在数据转换中由于数据元拆分、数据元合并或者两个不同数据源中数据合并产生的问题等。需要对基层具体数据核对业务人员统一思想，加强认识，加强责任心，避免由于人为疏忽带来的数据质量问题，另一方面加大数据核对工作，将现有数据与企业进行核对，以保证基础数据的正确性。在数据转换过程中，加大数据测试力度，保证原始数据正确无误的转换过来。

3. 数据量风险

数据量庞大时对数据迁移工作形成了巨大的挑战。需要根据业务应用的需要和时间对原有系统的数据进行分割，区分业务类数据和历史查询类数据，分类进行迁移。

4. 数据迁移时间风险

数据迁移的时间风险主要表现在旧系统宕机到新系统正式启用的时间范围内，由于数据量的原因而无法完成原有系统数据的迁移工作。

应对策略：

(1) 切实安排好上线时间，尽量避免在业务运行高峰期上线，必要时可安排在国定休息时间加班上线，为数据迁移预留足够时间。

(2) 提前规划好系统上线流程，合理安排数据迁移各项工作的执行顺序，确保数据迁移各项工作并行开展。

(3) 借鉴各大厂商数据迁移 ETL 工具的优点，提高数据迁移脚本的执行效率。

根据数据迁移策略，分批次、分步骤迁移数据，首先确保上线启用功能的数据迁移，在试点功能上线成功后再迁移其余功能的业务数据。

1.3.1.3 信创产品更新换代技术风险

安全可靠计算机系统是安全可靠 CPU、安全可靠操作系统构建安全可靠服务器及计算机，同时结合安全可靠中间件、数据库共同构成安全可靠基础软硬件环境。其技术体系目前正处于不断改进，逐渐完善的阶段，并且，不同安全可靠 CPU 厂商在产品设计上选择的技术路线也各不相同，因此本次项目建设需要综合考虑实际业务需求和安全可靠软硬件的技术现状，设计相应的建设内容和产品选型。如何选择合适的技术路线，对本次安全可靠替代工作推进至关重要。

应对措施：本项目采用统一的接口标准，满足应用系统适配建设的需要，同时，采用最新《产品名录》的产品适配清单，解决建设阶段的产品适配问题；对本项目软硬件系统运行和运维，建设运维管理系统及运维管理制度，进行科学运维管理；对本项目各应用系统，强制要求按照安全设计方案的要求，采用身份认证，对重要数据进行加密，加强安全记录和审计。全面落实信息系统安全体系，尽快全面组织落实本项目安全体系方案。

1.3.2 项目实施管理重点难点理解

1.3.2.1 组织风险及应对措施

本次信息系统安全可靠替代项目需要有效地沟通、协作以及配合。如果组织内部成员对目标未达成一致，项目参与人员知识与技能不完备、技术储备不足、人员安排不当等因素导致本项目建设队伍不稳定。此外，本次安全可靠替代项目尚处在经验积累阶段，大部分业务部门未开展过相关建设，缺乏实践经验，存在一定组织风险。

应对措施：建设安全可靠替代项目的管理领导小组，由建设单位在建设时成立，组长由相关领导担任，该小组由相关单位的主要业务人员、主要信息化技术人员，以产品及供应商、系统集成商的技术支持人员共同组成。管理领导小组能够支撑本项目的项目组织管理，并保障本项目高效地实施，统筹协调各单位实施工作有关事项。

1.3.2.2 管理风险及解决方案

本项目涉及业务系统的应用开发和数据迁移，在实际操作中存在一定的难度。如果在进行过程中，出现计划不周到、资金使用不当、办事效率差、进度安排不合理等管理上的偏差，就难以确保本项目按时按质地完成。

应对措施：为确保项目管理的高效率，将对项目进行有效策划，制定并落实严格的项目实施具体计划，应用先进管理工具和方法提高进度计划管理、跟踪水平。同时将借鉴类似已经实施的安全可靠替代项目管理实践的经验，合理估算项目工作量，明确项目间依赖关系和先后顺序，突出关键项目，进一步分解项目工作任务，使每个里程碑阶段均应有工作量估算、时间进度、以及可操作、可管理和可检查的交付物。

2 需求分析

2.1 硬件及基础软件需求

具体阐述系统架构，网络、安全、部署情况及基础软硬件等情况。

2.1.1 基础硬件

序号	类型	品牌型号	配置	数量	备注
1	数据库服务器	曙光	配置 1: CPU 海光 C86 5280 2.5GHz/ RAM 512GB /8TB HDD+960GB SSD		
		宝德	配置 2: CPU 鲲鹏 920 2.6GHz/ RAM 512GB /8TB HDD+960 SSD		
		浪潮	配置 3: CPU 飞腾 FT-2000+ 2.2GHz/ RAM 512GB /8TB HDD+960GB SSD		
2	应用系统服务器	曙光	配置 1: CPU 海光 C86 5280 2.5GHz/ RAM 512GB /8TB HDD+960GB SSD		
		宝德	配置 2: CPU 鲲鹏 920 2.6GHz/ RAM 512GB /8TB HDD+960 SSD		
		浪潮	配置 3: CPU 飞腾 FT-2000+ 2.2GHz/ RAM 512GB /8TB HDD+960GB SSD		
3	客户端	长城	配置 1: CPU 飞腾 FT-2000 2.6GHz/ RAM 8GB /256GB SSD		
		联想	配置 2: CPU 兆芯 ZX-E KX-U6780A/ 2.7GHz/RAM 8GB /256GB SSD		
4	存储设备	浪潮	CPU 鲲鹏 920 2.6GHz/ RAM 256GB /5*2TB HDD+3*480G SSD		可选
5	安全设备	天融信			可选
6	密码设备	三未信安			可选
7	网络设备	交换机（华为）			可选

		路由器（华为）			
		负载均衡（深信服）			
8	其他				

2.1.2 基础软件

序号	类型	名称	版本	数量	备注
1	应用服务器中间件	XX 应用服务器软件[简称：PAS]	V6.5		
2	数据库系统	达梦数据库管理系统	V8		
		海量数据库 Vastbase	G100		
3	服务器操作系统	麒麟	V10		
		统信	V20		
4	客户端操作系统	麒麟	V10		
		统信	V20		

2.1.3 应用软件

序号	类型	名称	版本	数量	备注
1	浏览器	火狐浏览器	V52.9.0 (64 位)		
		360 浏览器	V10		
2	其他	永中 Web Office	V3.0		可选
		金山 WPS 办公软件	V11		可选
		永中 OFD 在线预览转换软件	V1.0		可选
		数科 OFD 阅读	V5.0		可选
		福昕 OFD 版式办公套件软件	V8.0		可选
		金格电子签章系统	V8.2		可选

2.2 应用系统功能需求

2.2.4 应用系统

① 系统架构：基于技术平台进行开发，采用部署方式及服务器整体情况。开发语言、应用架构、关键技术，架构图等。

② 网络、安全、部署情况

③ 应用系统功能模块情况

3 系统建设方案

3.1 总体目标

阐述最终建设完成该系统的目标，如：承载机构约 10 万用户，日均交易约 40000 笔，高峰交易并发数约 3 笔/秒，改造后需满足系统运行指标要求，指标具体参数以实际项目要求为准。

3.2 整体安排

阐述建设完成该系统需要经历那几个阶段，每个阶段主要任务是什么，阶段达成的目标是什么。

3.3 总体架构设计

3.3.1 设计原则

本项目是一项规模大、范围广、结构复杂的系统性工程。为确保项目建成能够达到预期目标，在系统设计、开发和建设过程中应遵循以下设计原则：

1、安全保密防护

建设在安全可靠软硬件环境基础上，基于等级保护要求进行应用系统开发，结合自主化的安全防护手段实现信息安全，不改变原有系统的安全保密等级。

2、标准规范

业务应用系统建设遵循业务、技术、运维等相关标准规范，保证业务应用系统建设的标准化、规范化，实现系统间互联互通。满足用户的实际要求与使用习惯，体现以人为本的设计思想。页面简洁直观，各项功能清晰。

3、可扩展性

为适应将来的发展，要求应用系统和安全保障系统应具有良好的可扩展性，可实现服务和应用的灵活扩展。

4、可维护性

要求应用系统和安全保障系统应具有良好的可维护性，部署便利、使用简便、维护集中。

5、规避风险

要充分考虑政策、应用环境、人员等风险，尤其是存在业务系统建设或数据迁移的单位，并制定相应的风险对策和管理措施。

本项目依托建设单位已有业务网进行建设，系统的安全防护遵循国家关于信息系统等级保护的相关要求，结合业务应用的特性进行设计和建设。

本项目业务应用系统设计，采用多层架构，由门户层、应用层、应用支撑层、数据资源层、基础设施层、标准规范体系、安全保密体系和运维管理体系等几个部分组成，其总体架构设计如下图所示：

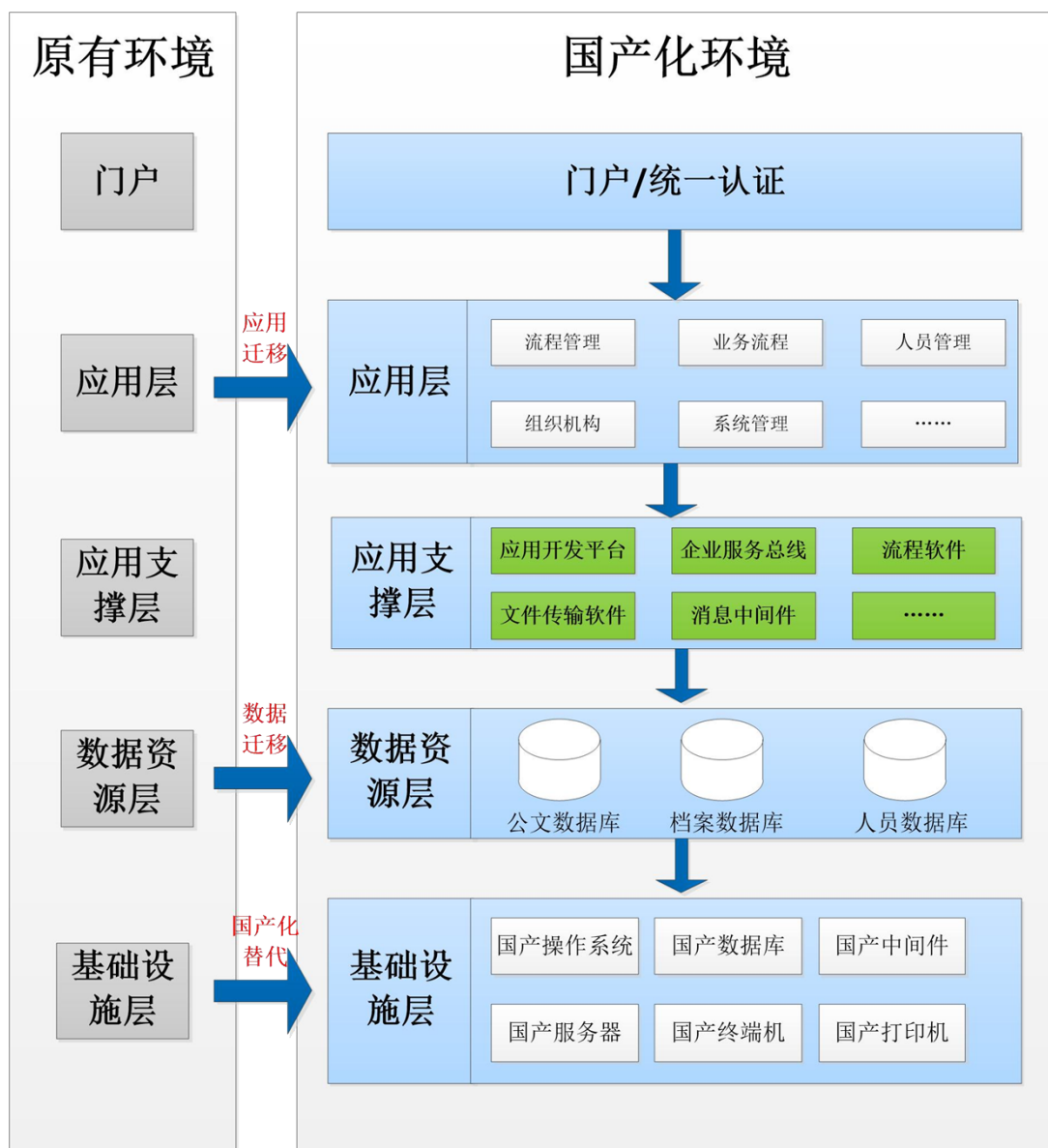


图 3-1 总体架构图

3.4 技术架构设计

3.4.1 技术路线

综合考虑安全可靠软硬件产品成熟度、产业链支撑能力、芯片产业发展现状及国家战略等因素，选择合适的技术路线进行安全可靠替代工作实施。

本次安全可靠替代项目采用具有良好的稳定性、高可靠性、跨平台、可移植性和扩展性的编译技术进行设计开发。

客户端采用国产操作系统，支持 Firefox 浏览器和基于 Chrome 内核的浏览器，支持国产安全浏览器；能支持高可用性的服务器集群和网络负载均衡；系统采用面向服务体系架构设计和平台化、组件化的模式开发，实现全浏览器界面下的操作和管理功能。

采用 OFD 公文版式，实现常见格式文档向 OFD 格式的单向转换和标引、封装的功能，阅读模块提供 OFD 文件的阅读、打印和盖章验章功能。

采用流式编辑技术，实现内容编辑、格式调整、修订批注和转版输出等全部功能。提供在线应用系统集成时按照浏览器规范封装为对应的浏览器插件，支持不同的网页浏览器和操作系统平台。

电子签章组件通过调用电子公章载体中的算法接口和公章使用标准接口，为客户端使用电子公章提供取单、印章及验章服务，实现与客户端用章系统（如 OFD 公文阅读软件）的结合。

采用国产安全数据库和国产应用中间件，最大程度的保证本次项目建设的应用系统的信息安全。

采用国产专用计算机、国产专用服务器等搭建本次安全可靠替代项目的基础环境，本次迁移改造的应用系统需要与安全可靠环境进行适配验证。

3.4.2 技术架构

本项目的总体技术框架可以分为五个层次和四个体系。5 个层次自下而上分别为基础设施层、数据资源层、应用支撑层、应用层和用户层,2 个体系包括运维管理体系和安全保障体系，如下图所示。

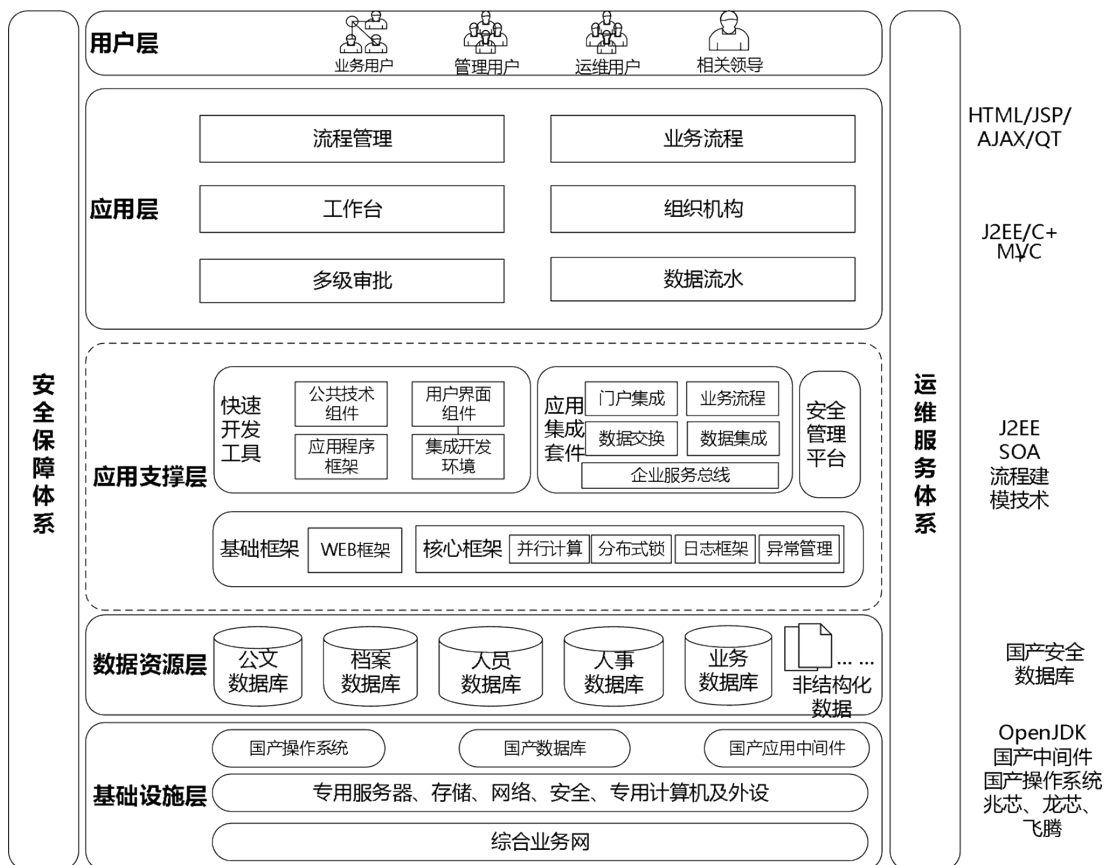


图 3-2 技术架构图

用户层主要包括系统业务用户、系统管理用户（包括系统管理员、安全保密管理员、安全审计员等）、系统运维用户以及相关领导等。

应用层是业务应用的主要实现部分，包括流程管理、业务流程、跟着他、组织机构、多级审批、数据流水。本方案中的业务应用都基于安全可靠基础平台的服务器和计算机、国产操作系统、国产中间件、国产数据库等。

应用支撑平台为用户现有软件资产带来了更好的重用性，基于已建应用支撑平台创建新的应用系统，也能够通过更新相关服务而无需重构整个应用来完成系统升级任务。

数据资源层为各系统提供数据存储和访问，包括持久化数据以及运行期缓存数据。主要可分为基础支撑数据区、综合办公数据区、业务应用系统数据区等几个部分。数据资源层需要国产数据库系统、共享存储系统支持。

基础设施层为系统提供硬件网络设施以及基础运行环境。包括专用服务器、专用计算机、网络设备等硬件环境，以及国产操作系统、国产应用服务器中间件、数据库服务器中间件等基础软件环境。

3.5 系统部署方案

3.5.1 部署说明

安全域的科学、合理划分是对信息系统进行边界整合的前提，是选择和部署安全防护措施，制定信息系统安全保护策略的基础。安全域划分目标是通过
对系统进行分区域划分和防护，构建起有效的纵深防护体系，有效抵御潜在威胁，降低风险，保证系统的顺畅运行，保证业务服务的持续、有效提供。

充分考虑综合业务网现状，结合国产化设备架构与传统设备架构不同的客观实际，方便日常管理和运行维护，在现有网络安全域划分的基础上部署本次项目新增的软硬件设备，在保障服务现状的基础上，提供国产化应用服务。

本项目建设的应用系统，部署在业务应用区。

3.5.2 系统部署工作流程

首先是产品选型，产品组合确定之后，依托公司产品资源，搭建仿真的国产化应用平台测试环境，进行组合适配。在设备到货之前既可以提供应用系统开发迁移资源，便于提早发现、解决适配过程中遇到的各类问题，仿真环境适配通过后进行生产环境应用部署工作，然后进行系统割接和数据迁移工作。

在本次工程的割接迁移中，需要对原有网络及承载的业务系统访问需求进行充分了解及详尽的技术准备，在整个割接中尽量不间断原有的业务运行或者将影响控制在用户可接受的范围。在割接前要准备详细的割接文档，并在仿真环境对方案的可行性进行测试验证；在割接过程中采用以往割接时使用的成功的测试手段来验证割接是否成功，一旦出现问题，按照割接文档中的应急预案解决问题，使得割接可控。

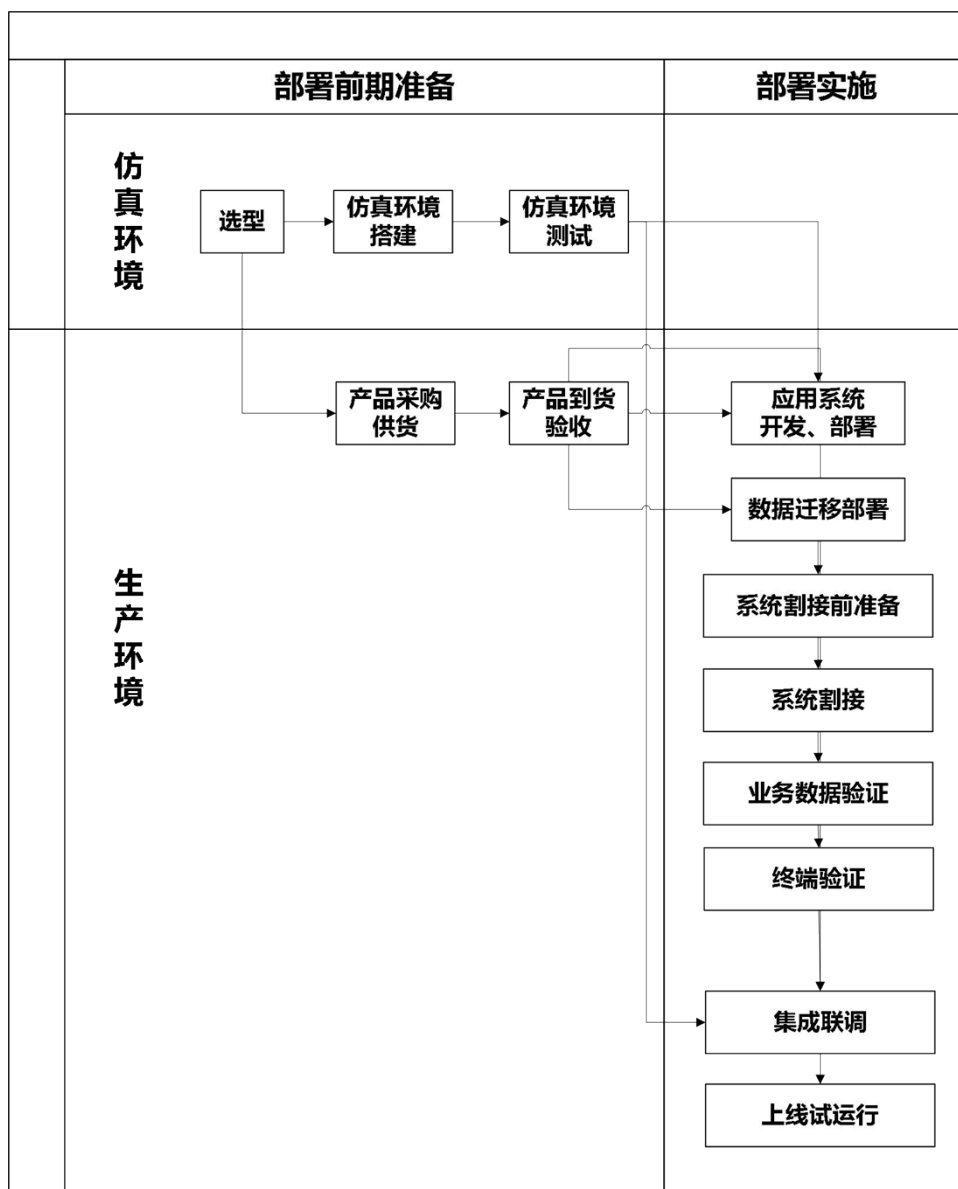


图 3-3 系统部署工作流程

3.6 适配验证方案

本项目需要进行应用系统的改造适配，以适应国产专用服务器和国产专用计算机的环境。需要针对国产专用服务器和国产专用计算机及国产化的软硬件环境下进行开发和测试。

3.6.1 适配改造主要问题

基于各应用系统的开发架构、开发技术和功能需求，分析本次 AQKK 替代适配改造面临的涉密专用计算机和涉密专用服务器端的产品特点，本次国产化改造适配的主要问题如下：

1. 安装问题：文档无法打开，因为浏览器不支持 word 控件，会始终阻止安装控件
2. 数据问题：对文件进行了相关操作后，出现数据不一致问题，或者部分字段数据丢失
3. 流程问题：由于浏览器兼容性的差别，应用系统的流程图可能会展示异常，出现错误
4. 乱码问题：部分情况系统会出现乱码
5. 显示问题：系统会出现 banner 条、字体、样式等诸多显示问题
6. 功能响应问题：某些按钮按下没有响应，或者报错
7. 性能问题：在某些检索场景下，响应时间可能会过长

3.6.2 系统适配重难点

根据系统现状和测试问题的分析，基于国产化软硬件环境，应用系统进行国产安全环境适配的重难点如下：

应用迁移与兼容性：应用系统是重点业务系统，非常重要。需要基于全新的国产化安全环境，将应用系统迁移过来，经过适配后需能正常使用，不能出现数据丢失和各类功能问题。是系统适配的重点和难点。

性能问题：在新的国产化安全环境下，系统要保证正常的性能，不能因为迁移到国产化安全环境后性能出现大幅下降，使用的体验差很多，让用户丧失使用本系统的耐心和兴趣。但是，国产化安全环境下使用的专用计算机、专用服务器普遍存在性能不足的问题。这是系统适配的难点，需要在本项目实施过程中解决好。对于性能问题，主要从两个方面来解决：一个是建议提升硬件配置；一个是进行应用系统调优。

业务流畅性：在新的安全可靠环境下，需要保证现有系统的业务流畅性：既要保证原有业务流程顺利流转和开展，也要保证各业务节点的丰富功能都能够正常使用。

接口问题：本系统和其他系统有相关接口。由于本系统需要进行国产化安全环境下升级改造，其他系统也可能需要在安全可靠环境下的升级和改造，本系统与其他系统的各种接口就需要重新适配，保证接口的有效性。

3.6.3 适配方案设计

基于前期的需求了解和适配重难点分析，本次应用系统国产化适配包括前期准备、基础环境搭建、应用迁移、全面适配测试、升级改造、应用部署、数据迁移、适配验证、试运行、正式运行等阶段。围绕着应用系统国产化改造升级的核心目标，完成好每一个阶段的任务，确保应用系统在国产化安全环境下开发和适配的顺利完成。

3.6.3.1 环境搭建和应用迁移

1. 前期准备

前期准备阶段需要完成项目小组建立、制定项目计划和项目组织管理方案。

2. 基础环境搭建

由于本次系统适配和开发项目涉及到服务器和客户端的适配，所以国产专用计算机和国产专用服务器等基础环境是本次项目实施的基础。基础环境搭建主要包括客户端硬件、客户端软件、服务器端硬件、服务器端软件的准备。

3. 应用迁移

将应用系统迁移到国产专用服务器的基础环境中。

3.6.3.2 全面适配测试

为了对系统适配工作进行摸底，前期进行了初步的测试。本环节需要进行科学全面的适配测试，以便基于测试情况明确升级改造的内容，进行有针对性的安可适配开发改造。全面适配测试主要包括以下几个方面：

1. 测试环境

服务器端采用国产专用服务器环境，客户端采用国产专用计算机环境。2.

测试内容

测试内容包括应用系统的各个模块，不留死角。

3. 测试方式

测试方式主要包括功能测试和性能测试。其中，功能测试对各个模块的功能在新的客户端和服务端环境下的使用状况进行全面测试；性能测试对系统的性能在新的国产专用服务器环境下进行测试。测试发现的问题完整详细的记录下来，生成测试报告。

3.6.3.3 适配改造

适配改造需要基于适配测试产生的问题进行有针对性的开发改造，对系统从架构和代码等方面进行调整，从而让系统在新的涉密专用计算机环境下能正常使用。通过升级改造，使系统兼容涉密专用计算机和涉密专用服务器，并在混合环境下正常访问和使用，有效开展相关业务。升级改造有以下几个方面的内容：

3.6.3.3.1 需求分析和系统设计

1. 需求分析

根据全面适配测试报告，整理出项目升级改造的需求并进行相应的分析，生成需求分析报告。

2. 系统设计

基于需求分析报告，对系统进行有针对性的设计，并完成相关设计文档。

本应用系统的设计侧重以下几个方面：

浏览器兼容性设计

数据库兼容性设计

中间件兼容性设计

数据防丢失设计

业务流程设计

字处理软件控件设计

性能调优设计

接口设计

3.6.3.3.2 系统开发（适配改造）

系统开发是本次应用系统适配和开发项目的重点和难点。基于前期项目情况，依据需求分析和系统设计，本项目适配改造主要包括国产专用计算机终端适配改造和国产专用服务器端适配改造。适配改造的具体内容请参见下两个章节：《终端适配改造》和《服务器端适配改造》。

3.6.3.3.3 系统调优

由于国产专用计算机和国产专用服务器端普遍存在一定的性能不足问题，需要进行应用系统调优，以满足本应用系统的性能要求。系统调优主要包括以下内容：

- 数据层优化
- 模拟数据量
- 分库
- 分区
- 数据库参数
- SQL 优化应用
- 应用层优化
- 布式应用
- Web 应用配置
- 中间件配置
- 业务逻辑优化

3.6.3.4 应用部署和数据迁移

系统升级改造完成后，需要部署新的应用并进行数据迁移。

1. 应用部署

对升级改造完成的应用系统进行部署。服务器部署环境是国产专用服务器，客户端采用国产专用计算机。

2. 数据迁移

基于重构后的应用系统，将原系统的相关数据迁移到新的应用系统中。通过合理的数据迁移方案，保证新旧系统数据及业务的连续性。基于现系统数据库结构和详细设计说明书中的数据库设计，对现有系统中的历史数据分别按横向的时间和纵向的模块两个角度进行内容划分，划定需要迁移的数据范围，进而就数据清洗、转换、装载新系统形成一个可行的迁移方案。

3.6.3.5 适配验证

系统升级改造完成后，需要部署新的应用并进行适配验证。

适配验证用于验证经升级改造的应用系统是否完全适配了新的国产专用计算机和国产专用服务器端环境。适配验证完全通过后才能进行系统试运行。

3.6.3.6 试运行和正式运行

应用系统经过适配验证并通过后，进行试运行阶段。试运行阶段的日期和期限由公司是与客户商量并由客户决定。

1、试运行

部署适配验证后的应用系统

2、培训

3、试运行

4、正式运行

系统在试运行通过后，进行正式运行。

3.6.4 终端适配改造

3.6.4.1 适配国产化浏览器

3.6.4.1.1 系统表单控件从前端渲染改为后端渲染输出

自 Web2.0 以来，随着计算机硬件和前端技术的不断发展，Web 项目越来越多的使用前端渲染技术。前端渲染的核心是由客户端浏览器在前端对页面元素进行渲染。其运作流程如下：

从后端直接输出 Html 框架，该框架只包含前端所需的模板结构及相应的 Js 代码。

前端通过 Ajax 异步向后台请求数据，后台返回纯格式数据，通常使用 Json 或 XML 进行数据封装。

前端获取到数据包后由 Javascript 分析数据，并通过对 Html 的 dom 元素进行操作达到数据渲染的效果。

当前前端技术日新月异，各种新框架层出不穷，这些新框架大多支持或推荐使用前端渲染模式。其优势如下：

端口分离，专注于各自的角色：前端专注视图，后端专注数据。

可维护性高

模板成型，不需要进行后端的字符串拼接，繁琐，更容易出现错误；

模板进行同一化的管理，视图层的改变，容易更改，不需要再去后端代码中分离；

需求变更时，后端在需要时提供数据接口即可，前端只需更改视图层。跨

平台：现有技术已经完全能够满足 js 这门语言，调用系统设备。后端渲染的项目，无法进行跨平台（特指移动端平台）的处理。前端渲染的项目，跨平台时，后端代码完全无需更改，前端代码复用率可达到百分之七十五以上，效率极高。

但由于目前使用国产 CPU 的计算机与当前时代的 X86 计算机存在较大的性能差异，同时国产化的浏览器版本较低，在处理 Html 的 Dom 以及执行 Javascript 的引擎速度都偏弱，如采用前端渲染的模式，则在使用时会有明显的延迟和卡顿的情况出现，用户体验较差。同时由于客户端是单机，除了软硬件升级外无法通过其他常规手段进行速度提升。

因此，在本次应用系统适配及开发中，我们改为使用传统的后端渲染技术来打破这一僵局。后端渲染指的是在后台服务器上就已经将数据与 Html 混合完毕，直接将整个页面吐出到浏览器，浏览器直接解析 Html 并展示的模式。我们通常会使用后台模板引擎如 Velocity 等实现后端渲染的功能。

后端渲染的运作流程如下：

编写 Velocity 模板，使用 Html 和 Java 变量混排技术，可以支持条件、循环等语句的输出方式。

后端直接获取到数据，并调用 Velocity 引擎，将 Velocity 模板与数据混合后形成标准 Html 并输出到前端。

前端浏览器直接解析并显示 Html 的页面内容。

后端渲染仅将 Html 和数据作为字符串进行处理，非对象化的处理模式使得其在执行效率上要较 Javascript 和 Dom 执行引擎高出 1-2 个数量级。同时 Web 服务器由于可以采用负载均衡等手段进行横向扩充，当系统压力上升时可以通过添加 Web 服务器解决压力问题。

在本次应用升级改造中，我们抛弃以前的表单、控件及其他页面的前端渲染模式，改为使用 Velocity 模板引擎和 Jsp 引擎的方式进行后端渲染输出，以适配国产专用计算机客户端环境。

3.6.4.1.2 表单控件加载数据源模式修改

表单都是由一个个的控件组成的，有简单控件也有复杂控件。简单的控件如文本框、下拉框等，复杂的控件如文件编号、上传下载、意见显示控件等。这些控件复制业务数据的加载、展示以及获取数据进行保存的功能。

表单加载时，我们通过表单数据源来进行页面数据的初始化，以前的表单数据源中只包含了简单控件的数据，简单控件可直接从数据源中获取数据进行加载和展示，但是复杂控件的数据获取都是通过控件本身与后台独立进行交互后获取数据再进行加载和展示。每一个复杂控件都包含了一次后台的独立请求。在这种模式下由于产生了多次请求，当页面上复杂控件较多且国产化硬件性能较弱的情况下就产生了表单加载过慢，用户体验不好的情况。

本次修改中，我们对表单的数据源进行扩充和优化，将复杂控件所需的数据一并查询并加入到表单上下文中。我们对所有复杂控件进行修改，从独立加载数据模式修改为同简单控件一样的上下文数据加载模式，这就很好的减少了与数据层的交互，优化了性能体验。

3.6.4.1.3 首页及打开列表的方式改为 TAB 页模式

原系统中表单打开采用新窗口方式实现。经在国产化平台实际测试中发现，使用浏览器新窗口时会消耗较多的计算能力，单独新窗口打开一个空

链接界面也要大约一秒的时间，这样在实际使用中加上表单加载，速度会更慢，严重影响用户的使用体验。新平台采用 Tab 页方式来打开新表单；经测试，速度上能提高 50%以上。

3.6.4.1.4 列表从 easyUI 改为Jgrid

原系统列表使用 easyUI 前端框架的表格组件。easyUI 是一个很重的框架，其功能非常强大，但是性能上却不是最好的选择。

新平台中，我们将使用 Jgrid 来替换原来的 easyUI 表格，Jgrid 是一个很轻的表格框架，基于 JQuery 开发而成，能实现我们列表的绝大部分功能。在平台改造中我们需要根据 Jgrid 的数据标准和 Api 标准重新开发列表组件。

3.6.4.1.5Js 文件合并及延时加载

每一个 Js 或 css 连接都会消耗 Web 服务器的连接资源，在安全可靠环境下我们要充分利用资源，尽可能的减少连接资源的浪费情况。

首先我们将 Js 文件和 Css 文件尽量合并，减少服务器连接。同时，我们针对第一次页面加载速度较慢的情况，将部分无需页面初始化时使用的 Js 进行延迟加载。

JS 延迟加载，也就是等页面加载完成之后再加载 JavaScript 文件。JS 延迟加载有助于提高页面加载速度。要实现延迟加载大致有以下几种方法：

defer 属性

HTML 4.01 为 <script>标签定义了 defer 属性。

用途：表明脚本在执行时不会影响页面的构造。也就是说，脚本会被延迟到整个页面都解析完毕之后再执行。

在<script> 元素中设置 defer 属性，等于告诉浏览器立即下载，但延迟执行。

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="test1.js" defer="defer"></script>
<script src="test2.js" defer="defer"></script>
</head>
<body>
<!-- 这里放内容 -->
</body>
</html>
```

说明：虽然<script>元素放在了<head>元素中，但包含的脚本将延迟浏览器遇到</html>标签后再执行。

HTML5 规范要求脚本按照它们出现的先后顺序执行。在现实当中，延迟脚本并不一定会按照顺序执行。

defer 属性只适用于外部脚本文件。支持 HTML5 的实现会忽略嵌入脚本设置的 defer 属性。

async 属性

HTML5 为 <script>标签定义了 async 属性。与 defer 属性类似，都用于改变处理脚本的行为。同样，只适用于外部脚本文件。

目的：不让页面等待脚本下载和执行，从而异步加载页面其他内容。

异步脚本一定会在页面 load 事件前执行。

不能保证脚本会按顺序执行。

```
<!DOCTYPE html>
<html>
<head>
  <script src="test1.js" async></script>
  <script src="test2.js" async></script>
</head>
<body>
<!-- 这里放内容 -->
</body>
</html>
```

async 和 defer 一样，都不会阻塞其他资源下载，所以不会影响页面的加载。

缺点：不能控制加载的顺序

动态创建 DOM 方式

//这些代码应被放置在</body>标签前(接近 HTML 文件底部)

```
<script type="text/javascript">
    function downloadJSAtOnload() {
        varelement = document.createElement("script");
        element.src = "defer.js";
        document.body.appendChild(element);
    }
    if (window.addEventListener)
        window.addEventListener("load",downloadJSAtOnload, false);
    else if (window.attachEvent)
        window.attachEvent("onload",downloadJSAtOnload);
    else
        window.onload =downloadJSAtOnload;
</script>
```

使用 jQuery 的 getScript() 方法

\$.getScript("outer.js",function() { //回调函数，成功获取文件后执行的函数

```
    console.log("脚本加载完成")
});
```

使用 setTimeout 延迟方法

让 JS 最后加载

把 js 外部引入的文件放到页面底部，来让 js 最后引入，从而加快页面加载速度。

3.6.4.1.6 浏览器兼容性修改

原系统基于 Windows 平台进行开发，主要使用 IE 浏览器进行访问使用。由于部分客户端机器老旧以及需要同时支持其他业务系统的浏览器兼容使用，所以我们向下兼容 IE6.0。但是在安全可靠环境下，浏览器为国产化 Firefox，其对 Javascript 和 Css 的兼容性与 IE6.0 不同。整个系统页面都需要基于国产化 Firefox 进行兼容性改造，主要改造点为 Javascript 和 Css 的兼容性。

3.6.4.2 适配国产化操作系统

由于采用国产化操作系统，导致出现变化字符编码兼容性问题。针对这类问题，对输出页面的字符集编码需要统一为 UTF-8，需要对所有页面进行统一修改字符集编码信息。基于 UTF-8，能很好的解决字符编码兼容性问题。

3.6.4.3 适配国产化文字处理软件

原系统使用 Windows 客户端，同时使用 IE 浏览器和微软的 Office 作为文字处理软件对系统内上传的一些正文附件进行在线编辑和相关处理。由于使用的是 IE，所以在线编辑控件使用基于 IE 浏览器的 ActiveX 控件来实现。

客户端改为国产化操作系统后，客户端已不可能使用 IE 和微软 Office，改为国产化的 Firefox 浏览器和 Wps 字处理软件。我们通过以下技术修改实现正文附件进行在线编辑功能：

- 1) 使用公司提供的基于国产化Linux 操作系统和 Firefox 浏览器下的Wps 插件。
- 2) 根据插件的需要，重新编写上传、下载、在线编辑、套红等功能的修改。
- 3) 因为 Wps 和 Office 之间的差异，还需要对在线编辑中所需要使用的宏进行调整和修改。

3.6.5 服务器端适配改造

3.6.5.1 适配国产化数据库改造

国产化数据库跟现有数据库从结构到功能均存在一定的差别，需要从以下三个方面进行数据库的迁移和适配改造：

1. 使用国产数据库厂商提供的 jdbc 连接 jar 包并修改数据库连接串，使程序能正常访问到数据库。

2. 由于国产数据库与我们目前使用的数据库的语法、数据结构、内部函数等方面可能存在差异，所以应根据数据库厂商的语法差异文档对整个系统的数据访问 SQL 语句进行改造，改造的工作量与系统的页面数和国产化安全数据库与原数据库语法的差异性成正比。

3. 使用国产数据库厂商提供的数据库迁移工具将已有的数据库的历史数据迁移到国产安全数据库中。若国产数据库厂商无法提供相应工具或提供的工具有缺陷，则通过开发数据库迁移工具来进行数据迁移。

3.6.5.2 适配国产化中间件改造

国产应用中间件跟现有中间件从结构到功能均存在一定的差别，需要从以下两个方面进行国产应用中间件的迁移和改造：

1. 由于国产应用中间件与我们目前使用的中间件的语法、数据结构、内部函数等方面可能存在差异，所以应根据国产应用中间件厂商的相关差异文档对整个系统的相关内容进行修改。

2. 使用国产应用中间件，测试验证，使系统能正常访问。

3.7 系统迁移方案

3.7.1 数据迁移方案

3.7.1.1 数据迁移目标

数据迁移工作的目标是：以保证新系统中的初始数据的数据质量、从而保证新系统顺畅运行为首要目的，同时对历史数据中有价值的信息要充分利用，

保证数据零录入、零丢失，数据质量 100%转入新的业务应用系统，确保新系统的成功上线。

3.7.1.2 数据迁移原则

数据迁移应注意一致性、完整性、自治性、规范化、标准化原则，同时在实施过程中应注意数据迁移的重点与策略，由业务部门主导，体现数据迁移为业务服务。

1. 一致性

数据迁移应保证支持业务应用系统的业务数据一致性，数据迁移需要保障：

(1) 系统之间数据一致性保障。部分替换模式下，新原有系统同步运行 需要保障系统数据交互通畅，数据集成完整。

(2) 子系统之间、子系统内部数据一致性保障。对重构或新建的系统保障系统之间业务功能数据衔接完整一致。

(3) 业务数据与决策统计数据一致性保障。在迁移的核心业务数据基础上，迁移必要的历史数据，以满足统计决策数据要求。

2. 完整性

数据迁移的数据应包含结构化的数据库系统数据和非结构化的文件系统数据，应包含电子档案库的图形文件信息、电子文件系统数据和存储于数据库中的各种办公业务数据及其它相关的数据。

3. 标准化

数据迁移应设定对应标准，保障迁移过程高效有序进行，规范整个迁移过程对提高数据质量尤其重要。对应的数据迁移标准应包括：数据对照标准、数据的转换标准和中间文本的标准，提前制定并在迁移过程中严格依据标准实施。除指定对应的迁移标准外，还需要制定数据问题解决机制及流程，规范数据清理、数据问题反馈机制，将有效提高数据迁移质量。

4. 性能

在数据迁移过程中应注意提高数据迁移的性能，确保数据迁移方案在时间要求方面的可行性。由于数据迁移分布执行，对执行过程的时间要求十分严格，为保障数据迁移的过程效率，需要对数据库设计、迁移开发、数据环境部署等方面进行优化，提高整个迁移的性能。

3.7.1.3 数据迁移范围

需要迁移的数据不限于生产数据、系统配置数据、日志数据、汇总数据等，按照存储形式分为结构化数据和非结构化数据：

1、数据库数据迁移

在国产化数据库中使用数据迁移工具连接原有 X86 平台数据库进行数据的迁移。

2、非结构化数据迁移

将原有 X86 平台存储的音频、视频、文档数据等非结构化数据文件通过网络拷贝至国产化平台的指定目录。

3.7.1.4 项目迁移方法

数据迁移可以采取不同的方法进行，归纳起来主要有三种方法，即系统切换前通过工具迁移、系统切换前采用手工录入、系统切换后通过新系统生成。

1. 系统切换前通过工具迁移

在系统切换前，利用数据迁移工具把旧系统中的历史数据抽取、转换，并装载到新系统中去。其中数据迁移工具可以购买成熟的产品，也可以是自主开发的程序。这种方法是数据迁移最主要，也是最快捷的方法。其实施的前提是，历史数据可用并且能够映射到新系统中。

2. 系统切换前采用手工录入

在系统切换前，组织相关人员把需要的数据手工录入到新系统中。这种方法消耗的人力、物力比较大，同时出错率也比较高。主要是一些无法转换到新系统中的数据，和新系统启用时必须需要而旧系统无法提供的数据采用这种方法，可作为第一种方法的有益补充。

3. 系统切换后通过新系统生成

在系统切换后，通过新系统的相关功能，或为此专门开发的配套程序生成所需要的数据。通常根据已经迁移到新系统中的数据来生成所需的信息。其实施的前提是，这些数据能够通过其它数据产生。

在适配改造具体的实施过程中，并不能固定模式的选择其中的任何一种模式，系统的数据迁移过程可能是这三种手段的综合体。

3.7.1.5 数据迁移过程中需注意的问题

1. 最大限度的保证原系统数据迁移到新系统中，即使是对错误数据进行一些处理，然后在新系统中调整。
2. 新旧系统的对应关系一定要完整。
3. 原系统的数据在新系统中一定要有备份，不能数据迁移完成以后就将原系统数据删除掉。原系统备份数据至少保留一年的时间。
4. 新系统开发过程中，数据迁移负责人一定要与软件项目负责人保持经常沟通，以保证转换数据的正确性。同时软件项目负责人熟悉原系统的业务流程和用户的习惯操作方式也是有必要的。
5. 数据质量测试是非常重要的一个环节。

3.7.2 应用系统迁移

为了保证业务应用系统迁移的平滑过渡，首先需要将业务应用系统软件与国产操作系统、数据库、中间件、服务器和终端计算机环境进行充分适配和测试。适配测试通过后，在国产化平台的软硬件环境中重新部署相应的业务应用系统。

3.7.2.1 研发测试阶段

主要任务：搭建测试环境，包括桌面终端、服务器、存储、交换机、负载均衡器、操作系统、中间件、数据库等。按照总体方案执行研发工作，根据测试情况做好缺陷修复。按照测试用例执行测试工作，同时做好测试结果记录。

达成目标：完成系统功能测试案例，且所有测试案例 100%通过；完成系统性能测试案例，且各性能指标满足要求。

3.7.2.2 上线试运行阶段

主要任务：进行系统上线及试运行，同时做好试运行环境业务验证。

达成目标：顺利完成系统上线，业务验证无误，试运行期间系统稳定运行。

3.7.3 业务切换阶段

主要任务：评估上线试运行情况，满足要求则进行业务切换。

达成目标：顺利完成业务切换，系统在新环境下稳定运行。

3.7.3.1 硬件环境

序号	类型	主机名称	ip 地址	用途
1	数据库服务器			
2	应用系统服务器			
3	客户端			

3.7.3.2 软件环境

序号	类型	名称	版本
1	服务器操作系统	麒麟	V10
		统信	V20
2	数据库系统	达梦数据库管理系统	V8
		海量数据库 Vastbase	G100
3	应用服务器中间件	XX 应用服务器软件[简称：PAS]	V6.5
4	客户端操作系统	麒麟	V10
		统信	V20
5	浏览器	火狐浏览器	V52.9.0(6 4 位)
6	其他	Primeton EOS	V7.5
7		永中 Web Office	V3.0
8		金山 WPS 办公软件	V11
9		永中 OFD 在线预览转换软件	V1.0
10		数科 OFD 阅读	V5,0

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/227035114012010004>