

协同办公应用开发与服务平台技术规范

Technical specification for collaborative office application
development and service platform

(征求意见稿)

(本草案完成时间: 2022.11.10)

在提交反馈意见时, 请将您知道的相关专利连同支持性文件一并附上。

XXXX - XX - XX 发布

XXXX - XX - XX 实施

目 次

前言	II
引言	III
1 范围	4
2 规范性引用文件	4
3 术语和定义	4
4 缩略语	5
5 基本要求	5
6 功能规范	6
6.1 基础开发框架	6
6.2 基础开发组件	7
6.3 基础应用组件	9
7 兼容性规范	11
7.1 CPU 兼容性	11
7.2 操作系统兼容性	12
7.3 数据库兼容性	12
7.4 中间件兼容性	12
7.5 浏览器兼容性	12
8 技术支持规范	12
8.1 开发手册	12
8.2 使用手册	13
8.3 培训手册	13

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由辽宁省工业和信息化厅提出并归口。

本文件起草单位：大连倚天软件股份有限公司、大连理工大学、辽宁嘉讯软件有限公司。

本文件主要起草人：聂尚宇、梁晓伟、张磊、叶鑫、侯程、陈雪龙、王宁、宿晓燕、李守明、辛星、白洁。

本文件发布实施后，任何单位和个人如有问题和意见建议，均可以通过来电和来函等方式进行反馈，我们将及时答复并认真处理，根据实际情况依法进行评估及复审。

本文件归口单位通讯地址：沈阳市北陵大街45-2号，联系电话：024-86913384

本文件起草单位通讯地址：大连市高新园区软件园路80号科技园B座三层，联系电话：0411-84706767

引 言

随着信息技术的飞速发展和数字化转型的深入推进，协同办公已成为现代企事业单位提升工作效率、优化资源配置、促进信息流通的重要手段。特别是在信创（信息技术应用创新）体系日益完善的背景下，构建自主可控、安全可靠的协同办公应用开发与服务平台显得尤为重要。为了规范并提升信创体系下协同办公应用开发与服务平台的技术水平，确保各厂商能够开发出符合国家标准、兼容性强、技术支持完善的协同办公产品，同时满足用户单位在项目实施、验收及后期运维中的实际需求，特制定本《协同办公应用开发与服务平台技术规范》地方标准（以下简称“本文件”）。

本文件旨在通过明确协同办公应用开发与服务平台的基本要求、功能规范、兼容性规范、技术支持规范等关键要素，为行业内相关厂商提供统一的技术开发指南和测试基准，促进技术交流与合作，推动产业生态的健康发展。同时，本文件也为项目使用部门在采购、部署及验收协同办公类的平台与系统时提供了科学依据，确保系统能够稳定高效地运行，满足组织在数字化转型过程中的多元化需求。

通过本文件的实施，不仅能够提升我国协同办公应用开发与服务平台的技术创新能力，增强产品的市场竞争力，还能够有效保障信息安全，促进数据资源的共享与利用，为构建智慧化、数字化的工作环境奠定坚实的技术基础。因此，本文件的制定与实施，对于推动信创体系下协同办公应用开发与服务平台的技术创新、产业发展及广泛应用具有重要意义。

协同办公应用开发与服务平台技术规范

1 范围

本文件规定了信创体系下协同办公应用开发与服务平台功能、兼容性、技术支持等方面的技术要求。

本文件适用于各协同办公系统应用开发与服务平台厂商进行产品研发、指导测试机构进行应用开发服务平台产品测试。

本文件适用于各协同办公系统使用部门进行协同办公类的平台与系统项目验收。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

- GB / T 33476.1-2016 党政机关电子公文格式规范
- GB / T 33477-2016 党政机关电子公文标识规范
- GB / T 33478-2016 党政机关电子公文应用接口规范
- GB / T 33479-2016 党政机关电子公文交换接口规范
- GB / T 33480-2016 党政机关电子公文元数据规范
- GB / T 33481-2016 党政机关电子印章应用规范
- GB / T 33482-2016 党政机关电子公文系统建设规范
- GB / T 33483-2016 党政机关电子公文系统运行维护规范

3 术语和定义

GB/T 37696-2019、DB21/T 1793界定的以及下列术语和定义适用于本文件。

3.1

平台化开发 Platform-based Development

是一套综合的工具和共享的最佳平台，旨在按需应变的世界中开发软件和基于软件的服务。它形成完整、久经考验、开放和模块化的解决方案，使开发小组能够跨合作伙伴、供应商和客户自动化和集成软件开发的核心业务流程。平台化开发通过提供灵活性和速度，帮助企业获得竞争优势，迅速响应市场变化。其核心是面向大小型项目的灵活流程架构，支持多种开发语言和系统，提升工作效率，缩短投资回报时间，并增加业务灵活性。

3.2

服务网关 Service Gateway

服务网关介入客户端和服务端之间，是系统唯一的访问入口，主要功能包括：路由转发、权限控制、限流、熔断、负载均衡、健康检查、状态监控等，确保服务的安全性和稳定性。服务网关不仅作为单一访问点，还充当多项服务的代理，实现跨服务的变换、路由和公共处理，是微服务架构中不可或缺的组成部分。

3.3

微服务 Microservices

通过把原有SOA的业务逻辑实现根据微服务设计规范按照一定的微服务标准进行更细粒度的拆分，使得服务能够独立运行，服务之间的耦合性，分布式部署后实现服务的高可用及可扩展等特性，通过服务间的组合实现完整业务逻辑。

3.4

业务敏捷 Business Agility

在快速的业务变化过程中响应变化以及应对不确定环境实现业务目标所具备的能力。业务敏捷的目标是尽可能的缩短业务需求上线时间，所带来的直接影响是缩短交付周期、保证软件交付质量、提高组织协同效率等。

3.5

界面编排 User Interface Orchestration

在构建系统界面时，对图、文等视觉元素进行有序组织和布局的过程。通过精心的编排设计，使系统界面直观易懂、操作便捷，并具备良好的用户体验。这一过程旨在提升系统的可用性和用户满意度，确保信息的有效传递和高效交互。

3.6

逻辑编排 Logic Orchestration

将复杂的编程逻辑通过图形化界面进行直观展示和编辑的过程。它允许开发者通过拖拽、配置等方式快速构建业务逻辑，降低编程门槛，提高开发效率。可视化逻辑编排通常包含节点物料、逻辑编排编辑器、编排描述数据等组成部分，支持前后端通用的业务逻辑实现。

4 缩略语

下列缩略语适用于本文件。

SDP: 软件开发平台 (Software Development Platform)

DevOps: 开发与运营维护 (Development and Operations)

XML: 可扩展标记语言 (Extensible Markup Language)

JSON: JavaScript对象表示法 (JavaScript Object Notation)

5 基本要求

5.1 概述

协同办公应用开发与服务平台旨在促进不同团队、部门或组织之间的高效协作，提供强大的应用开发和部署能力，同时保障系统的稳定性、安全性与易用性。平台的系统架构、技术架构及其关键组件应遵循相应的设计原则和要求。

5.2 系统结构

系统结构的设计原则和要求应包括：

- a) 多租户支持：平台应支持多租户架构，确保不同租户之间的数据隔离和安全性；
- b) 模块化设计：系统应采用模块化设计，将不同功能划分为独立的模块，以提高系统的可维护性和可扩展性；
- c) 服务化架构：平台应采用微服务架构，将各个功能组件以服务的形式提供，降低模块之间的耦合度，提高系统的灵活性和可伸缩性；
- d) 负载均衡与容错：系统应支持负载均衡技术，确保在高并发场景下服务的稳定运行。同时，应具备容错能力，能够在部分服务出现故障时自动切换或恢复。

5.3 技术架构

技术架构的设计原则和要求应包括：

- a) 前后端分离：平台应采用前后端分离的技术架构，确保前端与后端的独立性，便于开发和维护；
- b) 容器化部署：采用容器化技术（如 Docker）进行应用部署，提高应用的可移植性和部署效率；
- c) 云计算基础设施：平台应基于云计算基础设施进行构建，充分利用云计算的弹性伸缩、高可用性和安全性等优势；
- d) 安全架构：构建完善的安全架构，包括数据加密、访问控制、身份认证等安全机制，确保用户数据的安全性和隐私性；
- e) 监控与日志：平台应提供完善的监控和日志系统，对系统的运行状态、性能指标等进行实时监控和记录，便于问题的及时发现和解决。

6 功能规范

6.1 基础开发框架

6.1.1 微服务架构

平台采用微服务架构，将应用拆分为一系列独立的服务，每个服务负责一个特定的业务功能。这种架构模式有助于降低系统复杂性，提高开发效率，并允许服务独立扩展和部署。

6.1.2 服务治理与注册发现

提供服务注册与发现机制，确保服务之间的动态发现和调用。服务治理功能包括负载均衡、容错处理、服务降级等，以提高系统的可用性和稳定性。

6.1.3 服务网关

引入服务网关作为服务的统一入口，实现请求的路由、鉴权、限流、监控等功能。API网关有助于保护后端服务，提高系统的安全性和可管理性。

6.1.4 容器化支持

支持Docker等容器化技术，实现服务的快速部署和迁移。容器化有助于减少环境差异带来的问题，提高系统的可移植性和可扩展性。

6.1.5 持续集成与持续部署

引入持续集成于持续流程，实现代码的自动化构建、测试和部署。这有助于缩短开发周期，提高开发效率，并保障代码的质量和可维护性。

6.1.6 错误处理与异常捕获

提供全局异常捕获机制，确保程序在遇到异常时能够优雅地退出并记录错误信息。
提供友好的错误提示信息，帮助用户快速定位问题。

6.1.7 版本控制

使用Git等版本控制系统，实现代码的集中存储、版本管理、协作开发等功能。版本控制有助于追踪代码变更历史，协同多人开发，并确保代码的可追溯性和可维护性。

6.1.8 可扩展性

设计可扩展的架构和接口，支持新功能的快速开发和部署。通过水平扩展和垂直扩展等方式，满足系统不断增长的性能需求。

6.1.9 可维护性

遵循良好的编程规范和设计模式，确保代码的可读性和可维护性。提供清晰的文档和注释，帮助开发人员快速理解系统架构和代码实现。

6.2 基础开发组件

6.2.1 前端引擎组件

前端引擎组件应包括但不限于以下功能和特性：

- a) 前端业务组件管理能力：能够将前端按照标准规范开发的前端业务组件进行可视化管理；
 - 1) 布局组件：包括用于系统整体布局的上所有、上下等布局性组件；
 - 2) 表单组件：包括文本框、选择框、文件选择框等元素及其组合构成的简单和复杂表单；
 - 3) 大屏组件：包括区块、树表、图表、曲线、折线、散点、雷达等统计图形组件；
 - 4) 列表组件：普通列表、复杂表头列表、其他用于列表的前端组件；
 - 5) 其他组件：树、弹窗、按钮、消息提示、等其他类型的组件；
- b) 业务组件开发能力：能够根据业务对前端业务组件提供开发能力，包括组件开发、参数设定、事件绑定等；
- c) 前端业务组件的解系能力：能够使用对用 XML 和 JSON 设定的前端组件进行解析，并生成相应的页面并绑定相对应的事件；
- d) 提供前端路由管理能力：能够对前端的路由及其跳转流向进行管理；
- e) 多端自适应能力：能够通过一次开发编排的前端组件能够在 PC、移动、Pad 等多端自适应能力；
- f) 可视化开发能力：能够提供可视化的开发界面，通过拖拽的方式实现组件的拼装和配置。

6.2.2 逻辑编排引擎组件

逻辑编排引擎组件应包括但不限于以下功能和特性：

- a) 业务逻辑管理能力：能够对按照标准开发的的业务逻辑进行管理；
- b) 算子管理能力：能够对组成业务逻辑的各类算子程序及函数进行管理；
- c) 业务逻辑解析运行能力：能够使用对用 XML 和 JSON 设定的业务逻辑进行解析后再内存中运行并反馈运算结果；
- d) 可视化开发能力：能够提供可视化的开发界面，通过拖拽的方式实现算子的拼装和配置并最终组合成业务逻辑。

6.2.3 流程引擎组件

流程引擎组件应包括但不限于以下功能和特性：

- a) 流程管理能力：能够对按照标准开发的的业务流程进行管理；
- b) 流程运行能力：能够按照流程的配置驱动业务进行流转；
- c) 可视化配置能力：能够提供可视化的开发界面，通过拖拽的方式实现流程的可视化配置；
- d) 流程节点类型：包含但不限于以下类型：起始、结束、自动、人工、汇聚网关、分发网关；

- e) 流程节点的属性：包含但不限于以下：名称、进入条件、推出条件、串行、并行、会签、权限设定。

6.2.4 数据模型引擎组件

数据模型引擎组件应包括但不限于以下功能和特性：

- a) 数据模型管理能力：能够对业务系统所需要的建立的数据模型进行管理，包含不限以下功能：
 - 1) 模型结构：对数据表的属性、其字段的属性和主键进行管理；
 - 2) 模型关系：对数据模型之间的关系进行管理，例如外键；
- b) 数据模型实例化能力：能够按照数据模型的配置在关系型数据库中对数据模型进行创建；
- c) 可视化配置能力：能够提供可视化的开发界面，对数据模型进行可视化配置。

6.2.5 输入输出引擎组件

输入输出引擎组件应包括但不限于以下功能和特性：

- a) 对于实现业务过程中的数据的输入和输出进行管理，包括但不限于：
 - 1) 数据导入：对需要导入数据模版进行建模并和数据模型件建立映射关系的全过程进行管理；
 - 2) 数据导出：对需要导出的数据实现数据模型到数据模板的映射管理；
- b) 可视化配置能力：能够提供可视化的开发界面，对数据模板的可视化设定，并与数据模型进行数据绑定。

6.2.6 数据解析引擎组件

数据解析引擎组件应包括但不限于以下功能和特性：

- a) 应支持将数据模型解析为 SQL 语句，涵盖对业务数据进行增、删、改、查操作；
- b) 应支持根据过滤条件获取业务实体数据；
- c) 应支持根据权限过滤获取业务实体数据；
- d) 应提供查询数据组织的自定义扩展机制；
- e) 应支持数据权限过滤查询结果。

6.2.7 报表与数据分析引擎组件

报表与数据分析引擎组件应包括但不限于以下功能和特性：

- a) 应能够利用数据解析引擎，对需要生成的数据进行预处理；
- b) 能够与前端引擎中的大屏组件进行配合，实现数据报表的定制；
- c) 应支持根据需要设定数据分析逻辑，并能够定制执行的能力。

6.2.8 定时任务与调度组件

定时任务与调度组件应包括但不限于以下功能和特性：

- a) 定时任务管理能力：能够对系统中的定时任务进行管理，并对定时任务的执行属性进行配置；
- b) 定时任务执行能力：能够按照配置对定时任务进行执行，当出现问题能够尝试重试；
- c) 定时任务的监控能力：能够对执行的定时任务的状态、异常进行监控，并对异常进行报警；
- d) 可视化配置能力：能够提供可视化的开发界面，对定时任务及其执行逻辑进行可视化配置。

6.2.9 消息引擎组件

消息引擎组件应包括但不限于以下功能和特性：

- a) 消息类型的管理能力：能根据业务需求对消息类型、消息结构进行定义；
- b) 消息发布能力：提供统一的消息发布客户端，能够按照业务需求发布消息；
- c) 消息转发能力：提供统一的消息转发能力，确保消息按照需要及时和延时发送到消费端；
- d) 消息消费能力：提供统一的消息消费客户端，能够消息能够被安全消费；
- e) 可视化配置能力：能够提供可视化的开发界面，对消息及其分发逻辑进行配置。

6.3 基础应用组件

6.3.1 布局管理组件

布局管理组件应对系统的整体布局进行可视化设定：

- a) 整体布局管理：包括用于系统整体布局的上所有、上下等布局性组件；
- b) 门户首页组件：根据用户权限，对系统内各个组织部门的门户进行可视化设定。

6.3.2 业务管理组件

业务管理组件应包括但不限于以下功能和特性：

- a) 系统内业务的新增能力：能够为系统增加新的功能模块；
- b) 系统内业务的更新能力：能够对系统已经存在的功能模块进行更新；
- c) 系统内业务的删除能力：能够对系统已经存在的功能模块进行删除；
- d) 系统内业务的暂停能力：能够对系统已经存在的功能模块进行暂停；
- e) 系统内业务的授权能力：能够对系统已经存在的功能模块进行授权，限定使用范围。

6.3.3 流程管理组件

流程管理组件应包括但不限于以下功能和特性：

- a) 流程管理能力：为用户提供可视化的流程定义、修改和运行能力；
- b) 流程消息处理：业务门户通过消息集成服务，聚合来自多个业务的待办任务等业务消息，并对其进行统一管理和分类展示。开发平台应提供消息中心，支持消息的推送配置，提供统一的消息推送服务；
- c) 审批管理：无须重新修改流程模型，可动态修改业务规则，如流程启动条件、参与人条件、流转条件等；审批时可自由选择退回的业务处理节点；审批时可指定下一步处理人；审批时可转交其他人处理；支持审批时的加签、跳转； workflow 审批须支持加签、协办、退回、取回等流程操作；工作流选人应支持按部门、按人员、按自定义分组等多种方式；应支持委托办理，支持按流程模板委托给不同人员，支持按条件委托；
- d) 流程监控管理：应提供可视化的流程监控图，实时监控流程的运行状况；支持管理员对流程进行回滚、任务改派、跳转、终结、删除、恢复等维护操作；支持对故障流程实例的智能识别与维护。

6.3.4 分布式文件存储组件

分布式文件存储组件应包括但不限于以下功能和特性：

- a) 提供分布式的文件存储能力：能够对业务系统中流转的各类文件进行统一的存储管理；
- b) 数据恢复能力：当分布式文件存储某个节点故障可以通过其他节点进行恢复。

6.3.5 组织机构组件

组织机构组件应包括但不限于以下功能和特性：

- a) 应提供组织机构管理，支持新增、修改、删除、迁移、合并等，机构岗位员工信息维护，机构、岗位、员工数据以及关系维护；
- b) 应提供人员管理，对组织机构下的人员进行管理，用户信息维护，用户数据管理，用户状态管理、密码周期管理；
- c) 应提供多组织管理，且支持组织结构的树形结构管理；
- d) 应支持组织的启用、禁用；
- e) 应支持组织中设定分级管理员的功能；
- f) 提供组织分层管理和配置功能。要求提供统一的组织机构、人员管理功能，能够按照业务需求提供基于单位和部门的精细化组织人员管理模式；
- g) 系统支持单租户、多租户、租户群等集团多组织架构模型，满足大型客户实现集权管控、分权管控和平衡管控的管理需求。提供分别面向用户组、角色和用户的权限管理，实现对业务功能的授权。

6.3.6 统一授权组件

统一授权组件应包括但不限于以下功能和特性：

- a) 应提供岗位管理；
- b) 应提供人员与岗位的设定功能；
- c) 应提供岗位与业务权限（菜单）的设定功能；
- d) 系统支持多种权限控制模型，提供功能授权、字段授权、数据授权、特殊授权和流程授权，还需要支持三权分立体系，实现权限制衡。提供菜单级、表单级、页面字段级的权限管理；
- e) 应支持多级授权机制，包括组织、用户类型、角色、用户等多维度授权，支持权限的继承和复制；
- f) 支持审计查询，认证与操作过程审计。

6.3.7 统一认证组件

统一认证组件应包括但不限于以下功能和特性：

- a) 提供跨业务的统一认证管理；
- b) 提供组织机构推送和同步能力；
- c) 支持关键资源总数统计，如：用户、系统、客户端数统计；
- d) 支持系统在线用户数 TOP 排行，例如：统计 12 小时内系统在线用户数排行；
- e) 支持客户端凭证生成，支持对客户端的凭证进行管理控制；
- f) 支持系统登录权限设置，应用系统的登录权限控制，可为机构、工作组、用户设置登录权限；
- g) 支持 OAuth 2.0 认证流程，客户端认证，支持客户端认证模式，实现系统间服务调用安全控制；
- h) 支持用户认证，支持用户认证模式，实现用户登录、状态保持与安全控制；
- i) 支持访问令牌状态管理，对安全访问令牌声明周期进行统一管理；
- j) 支持访问令牌安全管理，对访问令牌进行证书加密控制；

6.3.8 安全管理组件

安全管理组件应包括但不限于以下功能和特性：

- a) 应保证应用安全，提供三员管理、身份认证、权责分离、资源访问控制、在线用户监控等功能；
- b) 应支持多种认证方式，包括用户名+密码认证、CA 认证、单点登录认证等认证方式；
- c) 应提供统一的日志服务，支持开发人员写入业务日志；

- d) 应提供跟踪日志记录，自动记录用户操作信息，包括但不限于访问时间、IP、访问路径、提交内容等；
- e) 应提供密码策略，包括但不限于密码强度、用户锁定策略、输入次数、密码有效期、初始修改密码等；
- f) 应提供锁定用户的解锁机制，提供管理员使用；
- g) 应保证数据传输安全，支持软硬件加密机制，确保传输数据的完整性和安全性；
- h) 应保证应用服务安全，提供访问代理机制，客户端不能直接访问应用服务器；
- i) 应保证数据存储安全，提供数据库的备份和还原功能。应具备防 SQL 注入、防 XSS 攻击、防 CSRF 攻击等防护能力；支持对敏感数据配置的加密能力；
- j) 应保证数据访问安全，提供对数据库的访问控制机制；
- k) 异常处理：在业务应用中，异常产生后，由框架统一进行处理，而不需要在上层应用的代码中去处理抛出的异常。框架会在服务层通过 AOP 实现异常的统一捕获，产生唯一的异常跟踪号，并将异常的详细信息记入日志。在界面展示仅含业务含义的说明，并附异常跟踪号。异常业务含义可以通过可配置和查找机制获得。

6.3.9 应用配置组件

应用配置组件应包括但不限于以下功能和特性：

- a) 支持配置应用管理，在配置中心生成应用模型；
- b) 支持配置集群管理，生成与应用组对应的集群模型；
- c) 支持配置分组管理，通过分组管理应用下的配置；
- d) 支持配置项管理，涉及增、删、改、查配置项；
- e) 支持配置发布时，发布配置并生成新版本；
- f) 支持回退配置版本；
- g) 配置发布后，能够在应用实例生成配置缓存；
- h) 配置发布后，配置中心中能够主动的将配置推送到微服务应用；
- i) 支持在微服务应用中通过接口热更新配置值；
- j) 支持自动与配置中心进行配置版本比对，低于配置中心版本，自动获取最新配置版本。

6.3.10 业务日志监控组件

业务日志监控组件应包括但不限于以下功能和特性：

- a) 业务调用全链路监控；
- b) 业务调用全链路图形化展示；
- c) 业务日志集中收集；
- d) 业务日志格式转换；
- e) 业务日志检索；
- f) 业务 API 调用统计分析，比如：调用次数、失败次数、响应时间等；
- g) 监控业务资源，比如 SQL、逻辑层，的调用次数、响应时间；
- h) 支持对业务应用的 CPU、内存的资源使用情况监控；
- i) 支持断路器监控。

7 兼容性规范

7.1 CPU 兼容性

平台应支持多种主流CPU架构，包括但不限于Intel x86、AMD64、ARM架构、龙芯、申威、飞腾、兆芯。在开发过程中，应确保应用程序能够在这些架构上稳定运行，并充分利用各自架构的特性进行优化。对于特定指令集或功能的利用，应确保在不支持这些特性的CPU上能够优雅降级或提供备选方案。

7.2 操作系统兼容性

平台应支持多操作系统环境，包括但不限于欧拉、UOS、中标麒麟、中科方德、Windows、Linux、macOS等。应用程序在开发时，应遵循跨平台开发原则，避免使用特定于某一操作系统的API或功能。对于必须依赖特定操作系统的功能，应提供清晰的文档说明和替代方案。

7.3 数据库兼容性

平台应兼容多种主流关系型数据库和非关系型数据库，如达梦、神通、金仓、瀚高、Gbase、MySQL、PostgreSQL、Oracle、MongoDB、Redis等。在数据库访问层，应采用统一的数据访问接口，确保应用程序能够无缝切换不同的数据库系统。同时，对于特定数据库的特性或优化，应提供清晰的文档说明和配置选项。

7.4 中间件兼容性

中间件作为平台的重要组成部分，本平台支持多种主流中间件产品，如消息队列(RabbitMQ、Kafka)、缓存系统(Redis、Memcached)、负载均衡(Nginx、HAProxy)、Web中间件(东方通、中创、金蝶、保兰德)等。在开发过程中，应遵循中间件的使用规范，避免直接操作中间件底层资源。对于中间件之间的交互或协作，应提供清晰的接口定义和文档说明。

7.5 浏览器兼容性

对于前端应用程序，本平台支持主流浏览器，包括360、红莲花、Chrome、Firefox、Safari、Edge等。在开发前端应用时，应遵循跨浏览器开发原则，避免使用特定于某一浏览器的API或功能。对于不同浏览器之间的样式差异或功能差异，应提供兼容性解决方案或备选方案。同时，对于前端应用的性能优化和用户体验提升，应充分考虑不同浏览器的特性和限制。

8 技术支持规范

8.1 开发手册

为了确保协同办公应用开发与服务平台的开发过程顺利进行，需要以离线或在线方式，对开发者提供指导手册，技术组件层面，提供平台前后端属性、事件、方法的说明；业务组件层面提供模块使用方法、服务接口等。

- a) 提供平台的整体架构说明以及模块划分方法，以便于开发人员能够快速的了解平台的整体结构；
- b) 提供平台升级机制说明，无缝实现开发平台的升级；
- c) 提供平台的开发环境搭建的说明，知道开发人员快速的搭建开发环境；
- d) 提供平台的开发手册，确保开发出的应用符合平台的技术规范和标准；
- e) 提供必要的API文档和SDK，方便开发人员快速集成和使用平台的功能；
- f) 提供诸如在线交流等技术支持形式，对开发者提供问题帮助；
- g) 开发规范，应提供业务开发的最佳实践，进一步提高开发效率和应用扩展性；
- h) 用户社区，应提供在线交流等技术支持形式，对开发者提供问题帮助。

8.2 使用手册

为了让用户更好地理解和使用平台提供的功能，需要以离线或在线方式，对开发者提供指导手册使用手册。

- a) 使用手册以图文并茂的方式，介绍了平台的界面布局、功能操作、数据管理等基本使用方法；
- b) 提供了相应的解决方案和常见问题解答。用户应仔细阅读使用手册，并遵循其中的指引进行操作，以确保能够充分利用平台的功能，提高工作效率。

8.3 培训手册

为了帮助用户更好地掌握平台的使用技巧和方法，需要以离线或在线方式提供专门的培训手册。

- a) 培训手册针对不同类型的用户（如管理员、普通用户、开发人员等），分别提供了相应的培训内容和指导；
 - b) 培训手册还提供了培训课程的安排和进度，以及培训后的评估方式，确保用户能够真正掌握平台的使用技巧和方法；
 - c) 提供平台的在线或现场培训，以及配套培训材料。
-