文章编号:1000-5641(2015)06-0134-09

SaaS 云服务环境下的管理信息系统方案研究

王倩宜, 欧阳荣彬, 龙新征
(北京大学 计算中心,北京 100871)

摘要: 为了降低北京大学各院系等基层单位的信息化门槛,深化学校信息化建设,根据学校院系管理的需求特点,提出了一种 SaaS 云服务环境下的管理信息系统方案. 探讨了方案的多层次可扩展的应用架构,并且重点研究了其中的 3 个关键技术:多层次数据存储模型,基于 IAAA 的多租户访问控制策略 MT-IAAA,以及面向多租户的可配置方法和运行机制.

关键词: SaaS; 多租户; 数据存储; 访问控制策略; 个性化配置; 管理信息系统
中图分类号: TP315 **文献标识码**: A **DOI**:10.3969/j.issn.1000-5641.2015.06.017

Research of SaaS-based MIS scheme design

WANG Qian-yi, OUYANG Rong-bin, LONG Xin-zheng
(*Computer Center, Peking University, Beijing* 100871, *China*)

Abstract: In order to improve the information construction of Peking University and facilitate the steps of each department, this paper presents a multi-tenant-oriented SaaS solution by analyzing the requirements. It discusses the extendable multi-layer application framework and focuses on three key technologies, which are multi-layer data storage model, multi-tenant-oriented IAAA access control strategy and configuration/deployment methodology.

Key words: SaaS; multi-tenant; data storage; access control policy; personalized configuration; management information system (MIS)

0 引 言

SaaS(Software as a Service,软件即服务)[1]作为云计算的一种软件应用模式,越来越受到重视. SaaS 应用借助云计算带来的低成本、可扩展和可配置的共享服务资源,为多个企业或组织提供具有共性的软件服务,解决企业或组织信息化建设中的新技术应用、软件构建、运维成本、管理成本等问题.

北京大学管理服务信息化已全面深入到学校教学、科研、管理与服务的各个层面,建成了人事、学生、设备、财务等多个主干信息系统. 这些主干信息系统侧重学校级管理要求,满足学校各行政主管部门的管理需求,对院系级需求考虑不足. 然而,随着学校信息化建设的深化,院系等基层单位自身的管理信息化要求越来越迫切. 按照传统的软件建设模式进行院系信息化

收稿日期:2014-10
基金项目:北京大学"北大研究"课题(2014YB09)
第一作者:王倩宜,女,硕士,研究方向为高校教育信息化. E-mail: qywang@pku.edu.cn.

建设，需要根据每个院系自身的业务需求，定制开发完全属于院系自己的信息系统.

为了缩短建设时间，降低建设成本，更有效提高信息化建设效能，本文根据院系级管理信息系统的需求特点，结合按需服务、无需用户维护、便于扩展的 SaaS 软件应用模式，提出了基于多租户的 SaaS 云服务模式的应用方案. 具体工作包括提出多层次可扩展的应用架构，并且重点研究其中的 3 个关键技术：多层次数据存储模型，基于 IAAA 的多租户访问控制策略 MT-IAAA，以及面向多租户的可配置方法和运行机制.

1　系统需求

院系级管理信息系统面对的主要是本院系的职工和学生，较大的院系也就数千人. 用户数量不多，并且业务需求不复杂，数据规模也不大，属于小型的管理信息系统.

各院系的管理模式相近，业务需求具有共性，完全可以抽取、规范和归类院系级功能需求. 院系级管理信息系统是整个学校电子校务系统的一个缩影，覆盖的业务功能广泛，涉及了全部的学校管理业务，如人事、学生、科研、财务、办公等，是一个综合性的管理信息系统.

当前，院系的管理业务和功能多分散在学校各校级主干信息系统中，院系级管理信息系统需要将这些功能整合在一起，方便管理服务工作. 同时院系级管理信息系统需要在这些校级业务功能上进行自己业务功能和工作流程的扩展.

院系级的管理信息系统需要能够及时共享和使用学校级的业务数据，如职工数据、学生数据、科研数据等，并且需要对这些业务数据进行整合和重构，使数据不再被纵向分割，数据可以从平行的角度进行查询和利用.

此外，根据学校的整体规划，所有管理信息系统都需要在北京大学电子校务的整体框架内进行建设，需要符合大框架的安全规范、数据规范等标准和要求，特别是访问控制需要接入学校 IAAA 统一安全系统.

如果采用传统模式进行软件开发和部署，每个院系都需要开发和部署一套院系管理信息系统，每个院系都需要配备一系列硬件设备，以及维护一个安全稳定的运行环境，每一个院系管理信息系统都需要分别制定与学校主干系统在业务、数据、访问控制等方面的对接方案. 而采用 SaaS 软件服务模式，只需架构一套院系级管理信息系统，院系按需求定制使用，无需自己进行维护，从而降低成本，提高软件利用率，节约资源.

2　研究内容

SaaS 模式是云计算的三大模式之一，其在应用架构、开发部署等方面与传统软件有着很大不同，国内外学者对其软件架构、访问控制、数据存储模型、个性化定制、服务质量等多方面开展了研究工作.

SaaS 的架构按照成熟度分为四级：定制开发、可配置、多租户单实例架构、可伸缩的多租户架构. KANG 等[2]对 SaaS 应用案例进行分析，总结了 SaaS 服务的技术和业务要素，以一个 SaaS 服务成熟度模型为基础，提出可配置的多租户 SaaS 平台概念架构. 周学权等[3]从较为宏观的角度提出面向多租户的多层次可伸缩 SaaS 软件架构，这个软件架构在分析 SaaS 软件可伸缩需求的基础上，将 SaaS 软件的伸缩层次划分为业务层和数据层两个部分. SaaS 的架构大都采用分层模型，并且通过将服务、应用、流程组件化，以及使用内存数据库等策略，满足系统灵活、可伸缩、高性能等要求，同时也带来组件优化、应用调度、数据一致性

等新的挑战.

在 SaaS 多租户应用的数据存储研究方面,李晓娜等[4]提出一种支持 SaaS 应用的多租户数据划分模型和算法,有效地实现云环境中系统规模动态扩展,降低分布式事务代价. WEISSMAN 等[5]提出元数据驱动的多租户数据共享存储结构,应用功能、租户数据和配置等使用元数据表示,通过基于元数据的运行引擎产生虚拟应用组件. Force. com 支持多租户、按需定制,是一个高效的可伸缩的应用平台. 但是其对元数据和运行时引擎的依赖较多,也易使元数据访问成为运行瓶颈,需要通过元数据缓存加速来提高读取效率,从而增加系统实现的复杂度.

SaaS 多租户的特性给传统应用访问控制模型带来挑战. 马旭[6]提出一种基于 SaaS 的云计算安全模型,通过封闭的执行环境,保证客户虚拟机安全运行. LI 等[7]扩展 RBAC 模型和 ARBAC97 模型,提出 S-RBAC 模型,使用分层结构来实现系统级和租户级访问控制,租户可以自治管理自己用户的访问权限. 北京大学管理信息系统都采用统一访问控制模块,因此需要找出最适合的访问控制策略,既能满足 SaaS 应用的自身特性,又能融合学校的访问控制模型.

针对用户的个性化定制请求,SaaS 应用需要提供定制机制. 史玉良等[8]提出一个基于 TLA 支持租户业务流程定制行为建模及验证的框架. 张一川等[9]提出基于 ex-WSCL 的 SaaS 业务-租约 SBTM 模型和个性化业务定制框架,将业务端与租约端相分离,提高多租户个性化业务定制的方便性和灵活性. SaaS 模式下的个性化体现应当适度,如果过多强调和关注个性化,会增加 SaaS 定制的多样性,丧失定制应用或服务的共性、兼容性,使应用和服务很难标准化和模块化,同时也增加验证和确保租户定制结果正确性的难度.

总体来看,围绕 SaaS 模式的系统架构、系统伸缩性技术、数据隔离技术、安全访问控制、个性化配置、数据存储利用率、数据查询优化、资源调度、负载均衡等方面已经开展许多研究工作,也有许多成功和成熟的实际案例可以借鉴. 但是基于 SaaS 模式的院系管理信息系统有一些自己的需求特点,在设计和实施中需要重点考虑. 院系管理信息系统是一个小型应用系统,其面向的租户以及租户的用户数量规模不大并且相对稳定,因此系统的应用架构和数据存储模型的方案更侧重如何使系统易于实施和维护,而在系统的性能、数据存储量、可伸缩性等方面则不需过多顾及. 由于院系管理信息系统不是孤立的系统,其处于学校电子校务整体环境中,因此系统方案中需要研究如何利用和整合现有的功能、服务、数据等信息资源,同时也需要探讨如何既符合电子校务框架下要求的安全、标准等规范,又能满足 SaaS 应用的自身特性. 当然由于各院系会有个性化业务需求存在,如何满足个性化配置也是必须研究的内容.

3 系统方案

本文根据院系级管理信息系统的需求特点,结合按需服务、无需用户维护、便于扩展的 SaaS 软件应用模式,提出了 SaaS 云服务环境下的多租户模式的应用方案.

3.1 多层次可扩展的应用架构

图 1 所示为一个面向多租户的多层次可扩展的 SaaS 应用架构,该架构主要由 7 个部分组成. 界面控制层用以识别访问用户所属的租户及其身份信息如角色、权限等,并通过访问控制和个性化配置动态构建用户界面. 虚拟应用层是更高层次上的模型,在虚拟应用层每个院系租户都对应一个虚拟应用,组成该虚拟应用的功能和服务是由下层的组合服务层定义

并提供. 组合服务层具有业务整合功能，将相关联的基本业务功能单位组合成独立的、完整的功能服务单元. 业务逻辑层实现系统的基本业务逻辑处理，对应系统的基本功能操作，主要是基于访问控制和业务规则对系统的数据进行操作. 数据访问层封装对数据逻辑存储层的数据操作以及访问控制. 数据逻辑存储层实现数据库物理数据的逻辑存储. 可扩展组件的主要任务是对功能定制、应用配置、数据层扩展、数据安全保证等方面提供支持.

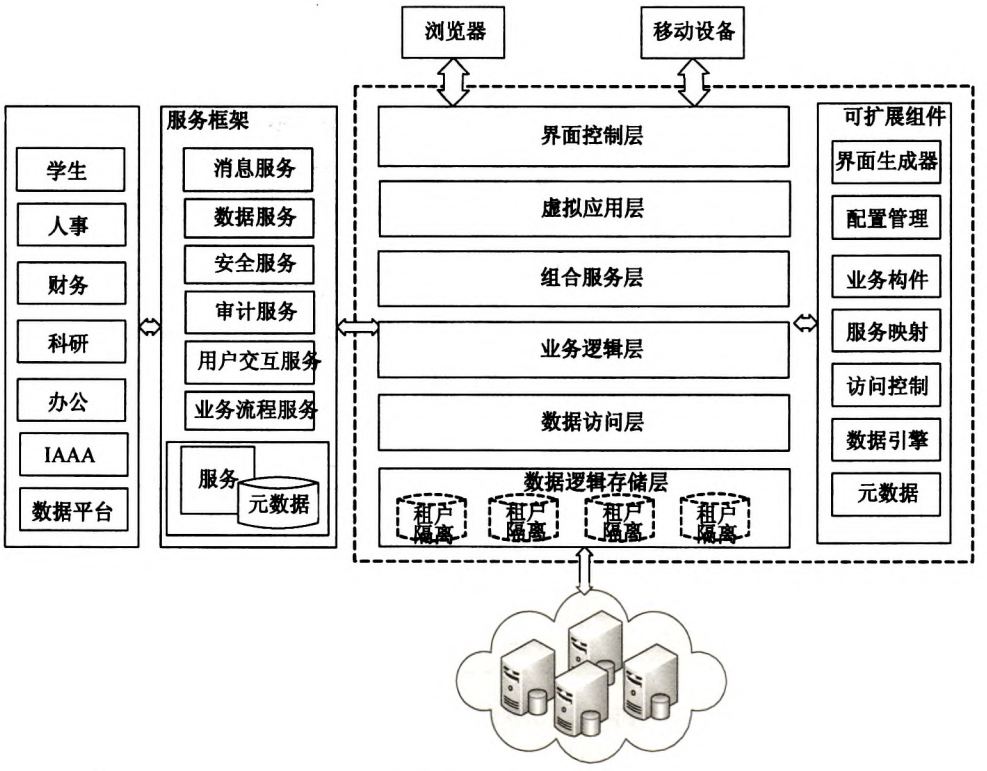

图 1　应用架构图

Fig. 1　Application architecture

北京大学电子校务框架采用 SOA 架构实现应用和流程整合，从主干信息系统抽取业务逻辑组件，封装成一个个对外提供的服务. 为了充分利用现有的功能服务和数据服务，本架构的系统采用服务映射组件与电子校务的服务集成架构进行接口. 系统通过服务映射组件将电子校务的服务引入业务逻辑层，作为一个基本的业务逻辑处理提供给上层组合服务层使用.

3.2　多层次数据存储模型

有 3 种数据库方案实现多租户的数据存储[10]：独立数据库；共享数据库，独立数据模式；共享数据库，共享数据模式. 本文方案采用共享数据库，共享数据模式.

租户由应用层管理，在应用层租户之间是彼此隔离和独立的. 但对于数据层来说，本文方案中系统所有的数据共享使用一个存储空间，租户之间没有区别.

如图 2 所示，本文方案采用多层次存储模型，除提供对各院系租户的数据支持外，还实现对数据可扩展、可伸缩性的支持. 该模型包括院系数据视图层、逻辑存储视图层、云存储视图层 3 个层次，通过元数据驱动引擎和云映射完成 3 个层次之间的协作和映射. 在该模型中，多租户的数据以透明的方式存储在云端，系统应用层只需要考虑业务逻辑，无需关心数据的物理存储.

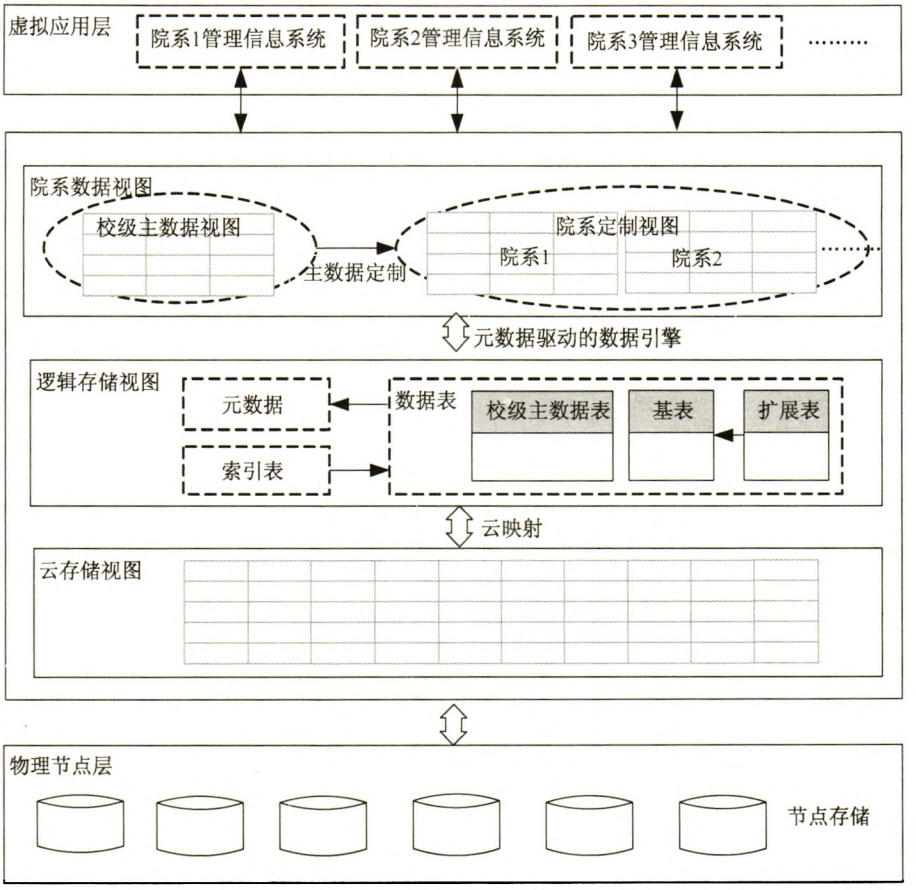

图 2 多层次数据存储模型

Fig. 2 Multi-level data storage model

院系数据视图是系统应用开发过程中所面对的视图,与数据存储模式、云资源调度等无关. 在系统应用中可以直接使用传统的 SQL 语句定义应用的数据模式和访问接口. 院系数据视图由校级主数据视图和院系定制视图两类视图组成.

(1) 校级主数据视图是面向所有院系的统一的数据视图,描述了所有院系管理需要的学校管理业务数据,包括职工数据、学生数据、科研数据、课程数据等. 校级主数据通过北京大学数据综合服务管理平台从校级数据中心实时获取. 系统中校级主数据只能查询不能修改.

(2) 院系定制视图是面向院系租户的虚拟独立视图,系统为每个租户的业务逻辑提供独立数据定制视图,支持租户数据隔离及数据定制的特征.

(3) 主数据定制定义和建立了校级主数据视图和院系定制视图之间的映射对应关系,院系租户可以对校级主数据进行按需定制,满足自己的个性化需求.

逻辑存储视图描述多租户数据逻辑存储模型,与云中数据具体存储位置无关. 逻辑存储视图统一了存储视图,支持多租户数据存储,易于扩展;并且屏蔽了云数据物理存储的技术细节. 系统只有一个逻辑存储视图,采用元数据驱动的基本表加扩展表方式存储多租户数据. 院系数据视图与逻辑存储视图之间的对应和转换通过元数据驱动引擎实现.

云存储视图是多租户数据存储的底层抽象，描述了云计算中数据伸缩、多数据存储节点等特征. 云存储视图提供了底层云数据的物理存储模式及多租户数据的放置策略. 逻辑存储视图与云存储视图之间的对应关系由云映射定义，租户数据通过该映射关系放置到云中不同的数据节点中. 当底层云存储视图发生变化时，只需要调整云映射，即可保证逻辑存储视图的相对稳定.

3.3　基于 IAAA 的多租户访问控制策略 MT-IAAA

北京大学 IAAA 统一安全系统实现对整个电子校务用户身份的集中管理、统一认证，支持多应用系统的功能权限和数据权限管理以及提供审计服务. 北京大学 IAAA 统一安全系统采用基于属性规则的用户功能授权实现模型[11]和数据权限模型[12]，主要思想是将角色作为用户与权限的中间层，通过为角色配置基于属性的功能和数据权限约束规则，自动建立用户、角色和数据对象之间的关联关系. IAAA 现有的授权模式不适应面向多租户的 SaaS 应用系统.

SaaS 模式下的访问控制具有多租户的特点，访问控制分为租户数据访问隔离和租户内部访问控制两部分. 租户数据访问隔离指每个租户只能看到自己的访问控制定制的信息，这是 SaaS 应用基本的数据安全保证. 租户内部访问控制是租户内部用户的角色权限管理. 每个租户相当于一个独立的安全域，各租户内部的角色和用户只能访问该租户拥有的资源，不能超出租户的资源范围进行操作.

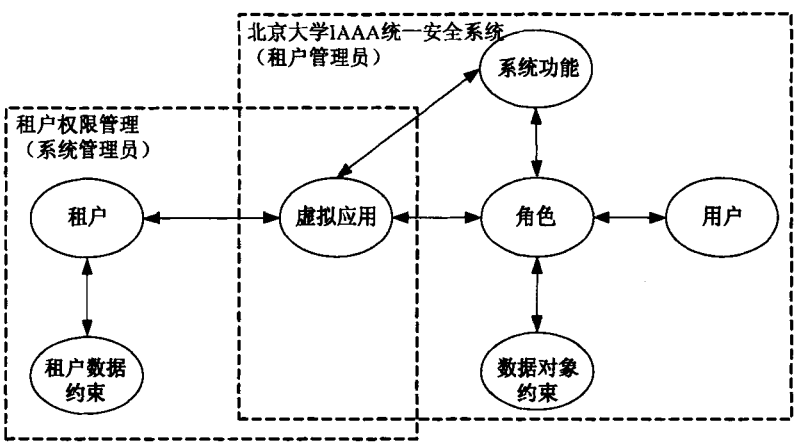

图 3　MT-IAAA 多租户权限管理模型

Fig. 3　multi-tenant authorization model

MT-IAAA 模型充分利用现有 IAAA 成熟的认证和授权管理资源，在 IAAA 访问控制的基础上，扩展出租户层的访问控制，租户层和 IAAA 的访问控制通过虚拟应用进行关联. 在租户层，每个租户对应一个虚拟应用，该虚拟应用作为一个独立应用自动在 IAAA 中注册. 在租户层会为每个租户分配一个租户管理员，该租户管理员同时作为 IAAA 中该对应应用的应用管理员存在.

MT-AAA 访问控制采用集中式和分布式相结合的授权管理方式. 租户身份审核和租户授权由系统管理员进行集中管理. 而租户内部的访问控制是由各租户管理员通过 IAAA 系统分布式进行管理.

用户登录系统时，首先通过 IAAA 认证进行身份认证，然后通过访问控制组件判定其

租户域,最后根据租户域在 IAAA 中获取该用户的相应访问规则. 用户在访问数据时,由应用层发出逻辑 SQL 语句,元数据驱动的数据引擎根据租户数据约束在逻辑 SQL 语句上增加租户的数据权限控制信息传递到数据存储层,返回的数据结果集在系统应用层应用租户内该用户的权限规则进行过滤返回给用户,租户内用户的访问控制是通过 IAAA 提供的数据权限服务完成.

3.4 面向多租户的可配置方法

在本文方案中,所有院系租户共享该系统的应用服务程序. 由于各院系租户的管理需求会有一定差异,因此院系租户对系统的数据、界面、功能等会有个性化要求. 这就要求系统能够支持院系租户不同级别服务的定制需求,具有一定的可配置性. SaaS 应用的可配置性主要体现在四大方面:数据可配置、功能可配置、界面可配置和流程可配置.

本文方案的个性化配置主要体现在以下 3 个方面.

(1) 租户数据的个性化配置. 采用基于元数据的基表加扩展表的方式为院系租户提供可扩展的数据结构,实现数据的可配置性. 基表由系统根据通用业务功能预先设置数据字段. 针对每个基表增设一个扩展关联表来存放各院系租户需要扩展的数据. 扩展表采用"键值对"存储方式,以行的形式存储记录行中的每一列. 使用元数据描述各院系租户的应用数据,如数据类型、长度等描述性信息. 元数据表、基表、扩展表共同决定租户定制的数据值. 用户访问定制数据时,首先获取元数据,依据元数据信息,访问相应基表和扩展表获取数据.

(2) 系统功能的个性化配置. 首先,将系统的功能、服务等可定制的对象,按照最小化、不能交叉重叠、不可循环依赖等原则,细分为一个个相对独立的单元,并且定义功能之间的依赖关系. 如果院系租户提出个性化的功能需求,则将其的定制需求按照同样原则划分为多个最小粒度的基本功能单元,然后在系统所有可定制对象中进行查找匹配和组合,在通过功能使用验证后以 XML 方式保存定制数据. 租户用户进入系统时,由功能配置管理组件解析 XML 文本并保存在用户的会话中.

(3) 操作界面的个性化配置. 操作界面的个性化可配置主要体现在操作页面的内容和菜单的个性化两个方面. 将系统功能菜单和页面显示元素参数化,将配置点以参数形式进行统一管理. 操作界面的个性化配置信息以 XML 方式组织和保存.

图 4 描述了个性化界面的运行机制. 菜单加载器用来加载租户个性化配置菜单,它使用功能配置管理组件来加载菜单和功能单元的配置信息. 页面数据生成器根据数据的元数据信息加载显示租户页面中个性化的数据元素. 个性化页面生成器根据租户页面配置参数生成个性化操作界面.

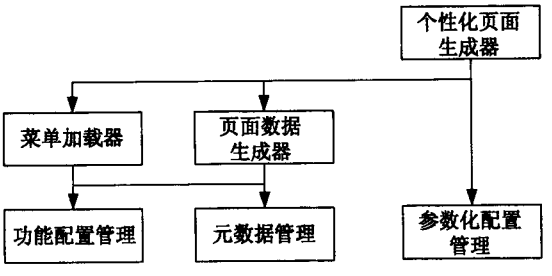

图 4 个性化配置运行机制

Fig. 4 Personalized configurable function design

4　应用实践

总结的说，在北京大学电子校务环境下，院系级管理信息系统的需求主要有以下 3 个特点：功能整合——整合学生、人事、科研、办公等多种业务功能；数据整合——基础数据来源于学校的多个主干信息系统；共性和个性并存——院系的管理需求大多具有共性，但是各院系也有自己的个性化需求. 并且，北京大学电子校务的整体框架有安全规范、数据共享、应用整合等方面的标准和要求，也提供许多功能服务、数据、安全等方面的共享资源.

应用本文提出的方案，院系级管理信息系统既符合电子校务整体框架要求，又能安全可靠的最大程度共享框架内的资源. 最终可以便捷有效的实现如下目标：

(1) 共享学校业务数据. 例如，院系管理使用的职工数据包括职工基本信息、联系信息、职工发表论文、获奖、专利、承担项目、课堂教学、指导学生、担任职务等多方面信息. 其中，职工部分基本信息和担任职务信息存储于学校人事业务数据库，课堂教学和指导学生信息存储于学生业务数据库，而承担项目信息则源自科研业务数据库. 在本文方案中，系统可以便捷安全的共享这些学校业务数据.

(2) 接入学校 IAAA 统一安全系统. 北京大学 IAAA 统一安全系统是电子校务的基础安全平台，实现对整个电子校务用户身份的集中管理、统一认证，支持多应用系统的访问控制管理. 在本文方案中系统可以最大程度的利用现有 IAAA 统一安全服务资源.

(3) 整合和扩展学校业务功能及流程. 院系管理中的一些业务功能是在学校业务管理上的扩展. 例如学校招聘管理业务流程中，院系只是其中一个环节，院系只需要提交自己的初选和终选结果，而院系内部的招聘流程并没有实现. 在本文方案中，系统可以在学校招聘管理业务流程上扩展实现院系内部的招聘管理.

(4) 界面可配置和功能可定制. 在本文方案中，系统可以满足各个院系租户的个性化需求，提供灵活的定制机制，当院系的管理需求发生变化时，可以方便增加或者修改定制的数据和服务等.

实践表明，每个院系或基层单位相当于一个租户，整个系统的租户在 100 个左右，规模不大. 院系级管理信息系统定位为院系内部的信息管理，每个租户的用户数量不多，院系管理的业务规模、数据规模也都不大. 应用本文方案建设，在性能、可靠性和投入成本等方面都能比较容易满足院系级管理信息系统的服务质量要求.

5　结　　论

综上所述，本文提出了 SaaS 云服务环境下的管理信息系统方案，设计了多层次可扩展的应用架构，并且对体系架构中的数据存储模型、访问控制策略和可配置方法进行重点论述. 该方案能够缩短学校院系等基层单位的信息化建设时间，降低建设和运维成本，从而降低学校基层单位的信息化门槛，有效深化学校信息化建设.

[参　考　文　献]

[1] CHONG F, CARRARO G. Architecture strategies for catching the long tail[R]. MSDN Library, Microsoft Corporation, 2006: 9-10.

[2] KANG S, MYUNG J, YEON J, et al. A general maturity model and reference architecture for saas service[C]//

Proceedings of the 15th International Conference on Database Systems for Advanced Applications,2010:337-346.

[3] 周学权,战德臣,聂兰顺,等. 面向多租户的多层次可伸缩 SaaS 软件架构研究[J]. 华中科技大学学报 :自然科学版, 2013,41(增刊Ⅱ):131-136.

[4] 李晓娜,李庆忠,孔兰菊,等. 基于共享模式的 SaaS 多租户数据划分机制研究[J]. 通信学报,2012(9):110-119.

[5] WEISSMAN C D, AND BOBROWSKI S. The design of the force. com multitenant internet application development platform[C]//Proceedings of the ACM SIGMOD International Conference on Management of Data,2009: 889-896.

[6] 马旭. 一种基于 SaaS 的云计算安全模型(英)[J]. 宁夏师范学院学报:自然科学版,2011,32(6):39-45.

[7] LI DC, LIU C, WEI Q, et al. RBAC-Based Access Control for SaaS Systems[C]//Proceedings of 2nd International Conference on InformationEngineering and Computer Science,ICIECS,2010:1-4.

[8] 史玉良,栾帅,李庆忠,等. 基于 TLA 的 SaaS 业务流程定制及验证机制研究[J]. 计算机学报,2010,33(11): 2056-2067.

[9] 张一川,张斌,刘莹. 支持多租约个性化业务租约模型的 SaaS 业务-租约模型[J]. 东北大学学报,2012,33(5): 636-640.

[10] CHONG F, CARRARO G, WOLTER R. Multi-tenant data architecture[EB/OL]. MSDN Library, Microsoft Corporation, 2006[2015-04-17]. http://msdn. microsoft. com/en-us/library/aa479086. aspx.

[11] 欧阳荣彬,王倩宜,李丽,等. 基于属性规则的用户授权模型的研究与实现[J]. 中山大学学报, 2009,48(增刊): 277-279.

[12] 欧阳荣彬,王倩宜,李丽,等. 基于属性规则的数据权限模型的研究与实现[J]. 大连海事大学学报, 2010, 36(2): 81-83.

(责任编辑 李万会)