目录

[目录 1](#_Toc448427884)

[科探Infoplus平台技术白皮书 4](#_Toc448427885)

[1. 总体设计 5](#_Toc448427886)

[1.1. 总体设计原则 5](#_Toc448427887)

[1.1.1发挥计算机的自动化处理能力 6](#_Toc448427909)

[1.1.2突出业务过程的智能表单和流程开发平台 7](#_Toc448427910)

[1.1.3以人为本，管理与服务并重 8](#_Toc448427911)

[1.1.4表单的定制开发原则 8](#_Toc448427912)

[2. 系统需求分析 9](#_Toc448427913)

[2.1. 功能性需求 9](#_Toc448427914)

[2.2. 非功能性需求 10](#_Toc448427915)

[3. 校园服务框架总体设计 12](#_Toc448427916)

[3.1. 项目理解 12](#_Toc448427917)

[3.2. 通用服务的梳理与规范 12](#_Toc448427918)

[3.3. 服务框架的组成和架构 12](#_Toc448427919)

[3.4. 服务调度架构的选择 13](#_Toc448427920)

[3.4.1服务调度架构的设计目标 13](#_Toc448427921)

[3.4.2服务调度解决的问题 13](#_Toc448427922)

[3.4.3基于微服务架构的服务调度架构 13](#_Toc448427923)

[3.5. 服务调度的模型和实现 19](#_Toc448427924)

[3.5.1业务层面的服务调度 19](#_Toc448427925)

[3.5.2业务无关的服务调度 20](#_Toc448427926)

[3.6. 服务调度的支撑机制 20](#_Toc448427927)

[3.6.1服务的底层封装 20](#_Toc448427928)

[3.6.2性能可靠性管理 21](#_Toc448427929)

[3.6.3安全管理 21](#_Toc448427930)

[4. 系统设计方案 24](#_Toc448427931)

[4.1. 建设内容 24](#_Toc448427932)

[4.2. 标准规范建设内容 25](#_Toc448427933)

[4.3. 基本数据描述结构与逻辑关系 26](#_Toc448427934)

[4.4. WFaaS总体架构 27](#_Toc448427935)

[4.5. 可视化的流程设计 29](#_Toc448427936)

[4.6. 可视化的表单设计 30](#_Toc448427937)

[4.7. 标准的事件驱动模型 32](#_Toc448427938)

[4.8. 开放性与云端化 32](#_Toc448427939)

[5. 系统核心特点与优势 34](#_Toc448427940)

[5.1. 与数字校园完善对接“即插即用” 34](#_Toc448427941)

[5.2. 深度优化的校园服务可视化开发 35](#_Toc448427942)

[5.3. 无缝融合的移动应用环境支持 38](#_Toc448427943)

[5.4. 程序语言无关的完全开放性 39](#_Toc448427944)

[6. 系统主要功能特性 41](#_Toc448427945)

[6.1. 全功能流程引擎 41](#_Toc448427946)

[6.1.1可视化工作流编辑/Visual Workflow Editor 41](#_Toc448427947)

[6.1.2岗位-步骤级基本权限控制/Posts-Step Authorization 42](#_Toc448427948)

[6.1.3执行人限定-部门限定的高级权限控制 43](#_Toc448427949)

[6.1.4Shell脚本支持/Shell Scripting 43](#_Toc448427950)

[6.1.5实例和步骤级别超时 44](#_Toc448427951)

[6.1.6灵活的选人方式 44](#_Toc448427952)

[6.1.7工作流版本控制 44](#_Toc448427953)

[6.2. 全功能表单引擎 45](#_Toc448427954)

[6.2.1可视化编辑 45](#_Toc448427955)

[6.2.2字段级功能设置 46](#_Toc448427956)

[6.2.3字段-步骤级权限控制 46](#_Toc448427957)

[6.2.4可嵌套的重复字段、重复节 47](#_Toc448427958)

[6.2.5表单打印功能 47](#_Toc448427959)

[6.2.6丰富的控件渲染类型 47](#_Toc448427960)

[6.2.7基于正则表达式表单验证 48](#_Toc448427961)

[6.2.8条件公式和条件样式 48](#_Toc448427962)

[6.2.9表单变量 48](#_Toc448427963)

[6.2.10表单版本控制 48](#_Toc448427964)

[6.3. 代码表（Data Exchange） 49](#_Toc448427965)

[6.3.1内部代码表 49](#_Toc448427966)

[6.3.2外部代码表 49](#_Toc448427967)

[6.3.3代码表的首字母拼音感知检索 49](#_Toc448427968)

[6.4. 事件驱动的标准化接口体系 50](#_Toc448427969)

[6.4.1应用-工作流多对多的消息接口 50](#_Toc448427970)

[6.4.2贯穿表单实例的整个生命期 50](#_Toc448427971)

[6.4.3应用端数据干预 51](#_Toc448427972)

[6.4.4应用端表单验证 51](#_Toc448427973)

[6.4.5应用端流程干预 51](#_Toc448427974)

[6.4.6应用端数据持久 51](#_Toc448427975)

[6.5. 流程与表单驱动的大学服务中心门户 51](#_Toc448427976)

[6.5.1服务大厅 52](#_Toc448427977)

[6.5.2待办任务 53](#_Toc448427978)

[6.5.2进行中任务 54](#_Toc448427979)

[6.5.3已完成任务 54](#_Toc448427980)

[6.5.4待评价任务 54](#_Toc448427981)

[6.6. 统一的流程监控管理与分析 55](#_Toc448427982)

[6.7. 数据管理中心 56](#_Toc448427983)

[6.7.1数据模型的自动持久化 56](#_Toc448427984)

[6.7.2数据管理功能界面的快速开发 57](#_Toc448427985)

[6.8. 系统接口设计 58](#_Toc448427986)

[6.9. 流程开发规范 63](#_Toc448427987)

[6.8.1表单开发 64](#_Toc448427988)

[6.8.2数据代码表开发 68](#_Toc448427989)

[6.8.3事件驱动开发 69](#_Toc448427990)

[7. 安全体系设计 71](#_Toc448427991)

[7.1. 安全体系设计原则 71](#_Toc448427992)

[7.2. 安全体系解决方案 71](#_Toc448427993)

[7.2.1物理层安全技术方案 71](#_Toc448427994)

[7.2.2网络层安全技术方案 72](#_Toc448427995)

[7.2.3系统层安全技术方案 73](#_Toc448427996)

[7.2.4应用层安全技术方案 74](#_Toc448427997)

[7.3. 数据备份方案 76](#_Toc448427998)

[7.4. 安全管理方案建议 76](#_Toc448427999)

[8. 方案可行性及案例验证 78](#_Toc448428000)

[上海交通大学 78](#_Toc448428001)

[上海大学的应用案例 79](#_Toc448428002)

[哈尔滨工程大学 82](#_Toc448428003)

科探Infoplus平台技术白皮书

# 总体设计

## 总体设计原则

1、标准化和开放性

系统设计遵从中华人民共和国国家标准和教育部标准、教育部数据交换标准等，能够与遵循上述标准的其他系统交换信息，实现信息资源一致与共享。

2、可操作性和易维护性

整个系统是由多个部分组成的较为复杂的系统，为了便于系统的日常运行维护和管理，要求解决方案和产品必须具有良好的可管理性和可维护性，便于日常运行维护和管理。

3、可扩展性及易升级性

充分利用目前系统的成果，保护前期投资，与现有系统协同工作。 随着管理与服务的变化，系统必然会有横向、纵向扩展、系统负载的重大变化。系统采用分层次、模块化设计，便于系统扩展。应用平台模块间应完全独立，接口清晰，内部的业务流程升级和改造与其它模块无关，所有模块基于组件开发，并为二次开发提供开发API 接口，该平台需要支持集群部署方式。为适应应用不断升级的需要，应用系统的软硬件环境必须有良好的可升级性。

4、可持续发展和整体性

一张表工程系统的建设是一项科学的系统工程，必须本着“承接历史、照顾现状、考虑未来”的原则进行规划和建设，使得最终的系统做到开放性、标准化、易扩展、易升级。在应用系统整体化、功能化和规模化的同时，保证系统在技术上、经济上的可持续发展，能支持用户数和业务量不断扩展的需求。

系统的建设需要综合考虑对原有投资的保护，容原有的系统，保护已有投资，照顾用户已有的使用习惯，保护并利用这些系统已积累的宝贵数据；同时应用平台的软硬件环境必须有良好的平滑可扩充性，保证对未来应用深化和扩展的支持。

5、先进性和成熟性

系统设计既要采用超前思维，先进技术和系统工程方法，又要注意思维的合理性，技术的可行性，方法的正确性。系统的设计不但能反映当今的先进技术和理念，而且具有发展潜力，能保证未来若干年内占主导地位。先进性与成熟性并重，并考虑到近年来的应用发展特点，把先进性放在重要位置。

6、可靠性和稳定性

在考虑技术先进性和开放性的同时，还应从系统结构、技术措施、系统管理等方面着手，确保系统运行的可靠性和稳定性，达到最大的平均无故障时间。

7、实用性和可行性

系统的主要技术和产品必须具有成熟、稳定、实用的特点，实用性放在首位，既要便于用户使用，又要便于系统管理。

8、安全性和保密性

按照系统性、立体性设计系统的安全方案，包括系统的运行环境温度、湿度、供电等；系统在硬件、网络、数据库、应用操作权限、数据权限、身份认证方面，提供全面的、多级别的安全防护；数据的异地备份。即要考虑信息资源的共享，更要注意信息资源的保护和隔离，应针对不同的应用和不同的网络通信环境，采取不同的措施，包括系统安全机制、安全策略、数据存取的权限控制等。

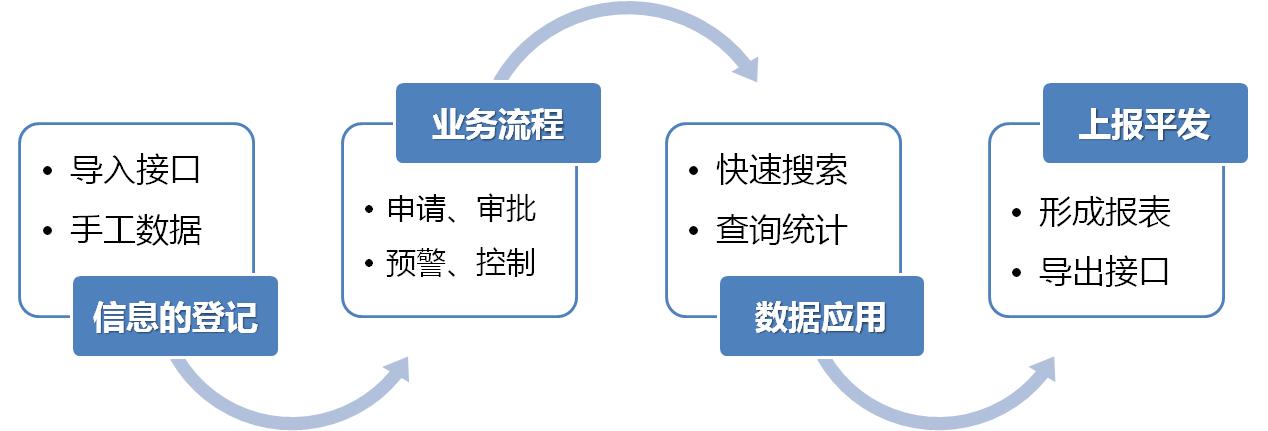
9、系统整体建设设计兼顾中山大学现有的平台及标准，各个阶段的文档齐全，能够为第三方软件提供相关接口并且支持集群部署方式，以确保系统可满足后续业务发展的需要。

# 



### 1.1.1突出业务过程的智能表单和流程开发平台

一站式流程服务中心具有业务表单通用信息自动填报、业务流程电子审批等功能，同时提供表单、工作流自定义开发平台，师生可根据需求自定义表单，开发人员可基于此平台轻松完成开发任务。利用数据仓库技术，针对学校的业务特点，构建学校关键绩效指标所依赖的分析数据模型，并逐步完成数据获取（ETL），数据管理以及数据应用的过程，为各级管理人员提供完整、统一、准确的数据视图。提供计划、预警、跟踪和分析，便于加强计划性和事中控制，通过事后数据的分析进行流程的优化提升。

****

**图：业务管理过程示意**

与此同时，学校业务流程变化频繁，且自身管理模式、培养目标各有差异，造成校内存在大量的形态各异的业务执行过程，给系统的设计和业务管理带来了困难。当整体业务发生变化，不再适应学校业务流程时，很多学校采用抛弃重建的方法，这样使学校损失了宝贵的数据，同时也无法谈及基于数据积累的一站式流程服务中心。面对这样的情况，仅仅通过单一固化的产品，难以满足长期发展需求，需要有整体化的解决方案。

### 1.1.2以人为本，管理与服务并重

传统的管理信息系统主要面向行政管理人员提供基本业务资源管理、查询和统计，实现业务流程的管理和控制，为师生提供的服务，以及为校领导提供的决策分析支撑有限。然而，随着互联网的发展、通信手段的不断丰富，使得网络的服务从学校的管理者慢慢扩展到广大的师生用户，不同角色的用户被服务的需求越来越强烈。

本期项目的建设应改变以往的被动式服务，应遵循“以人为本”的设计理念，在实现基础信息管理、业务流程的基础下，突出业务系统的服务能力，扩展面向教师、学生的综合服务；并从使用者角度出发，根据不同角色的不同业务需求，智能化的为用户提供各类信息和服务。

与此同时，本次建设的系统需具备良好的开放性和可扩展性，可以无缝集成于数字化校园应用支撑平台之上，采用统一身份认证方式，最终用户可通过信息门户平台单点登录后统一获取系统提供的各类服务。

### 1.1.3表单的定制开发原则

“数据记录、减轻手工劳动工作量、降低管理成本、加快教育信息化进程……”这些都是各个业务系统带来的好处，但系统存在的意义绝不仅仅只有这些。信息化建设最大的意义在于上层的商务智能，即如何通过有效、深入的数据挖掘，帮助学校发现在教学、管理中的问题，从而指导学校在教学、管理方面的改革，提升学校的整体竞争实力。

业务表单及对应流程可以实现定制开发。

# 系统需求分析

## 功能性需求

* 对于终端用户（师生、管理人员、领导）而言：

所有的信息服务可以在同一个空间完成：查询、申请、审核、互动等等。

平台完全屏蔽后台管理应用和管理部门，用户通过同一种方式获取各种服务，大幅度降低宣传、培训、学习成本，符合目前主流的应用设计理念

* 对于部门业务管理人员而言：

所有的面向师生的服务可以用统一的方式开发，并挂接到流程中心。而部门业务系统只需要管理好数据，并实现部门内部的管理功能即可。部门信息服务将更容易的推广到师生用户，系统建设的成功率提高了

* 对于部门业务系统的开发人员而言：

事实上，根据分析与经验，出了教务的排课、选课等特殊模块，校园信息化各种功能中50%以上可以通过流程模型解决。

因此，统一的流程开发模式可以从根本上改善定制型应用的开发，降低成本，提高开发成功率，提高稳定性。业务系统开发人员可以重新回到提升业务理解、增强咨询能力、提高业务系统专业性的职责中去。

本平台一些重要的功能特性如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 类别 | 描述 | 备注 |
| 1. | 工作流 | 全功能流程支持：人工节点、自动节点、条件节点、串行、并行、各种会签、分支、归并、回退、子流程等等，及其各种组合 |  |
| 2. | 工作流 | 全功能流程控制与调度，脚本支持 |  |
| 3. | 工作流 | WebUI可视化工作流编辑 |  |
| 4. | 工作流 | 可视化流程运行监控 |  |
| 5. | 工作流 | 版本管理机制 |  |
| 6 | 表单 | 可视化编辑 |  |
| 7 | 表单 | 丰富的控件支持 |  |
| 8 | 表单 | 与工作流进行无缝整合 |  |
| 9 | 表单 | 强大的表达能力，嵌套、重复、表格、节等等，及其自由组合，无需任何代码 |  |
| 10 | 表单 | 与数字校园基础数据无缝整合 |  |
| 11 | 服务中心 | 所有流程入口，待办／已办／在办中心，统一体验 |  |
| 12 | 服务中心 | 用户互动，包括处理过程中的互动 |  |
| 13 | 服务中心 | 多租户支持 |  |
| 14 | 应用整合 | 完整的RBAC权限模型 |  |
| 15 | 应用整合 | 虚拟信使模型的接口，完整的事件模型 |  |
| 16 | 应用整合 | 第三方应用界面整合 |  |

## 非功能性需求

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 类别 | 描述 | 备注 |
| 1. | 性能 | 单页面展示性能0.1-3秒 |  |
| 2. | 性能 | 流程数无限水平扩展，负载均衡 | 第一阶段集中式数据库，第二阶段分布式数据库 |
| 3. | 性能 | 支持3万以上终端用户同时在线 |  |
| 4 | 性能 | 50个以上应用系统接入，实时事件响应 |  |
| 5 | 性能 | 支持云存储，容量无限扩容 |  |
| 6 | 安全 | Internet主流授权协议 |  |
| 7 | 安全 | Internet主流安全协议 |  |
| 8 | 安全 | 应用间调用的安全保护 |  |
| 9 | 安全 | 通过主流安全漏洞扫瞄 |  |
| 10 | 安全 | 访问日志，安全审计，数据备份与恢复 |  |
| 11 | 可靠性 | 水平扩展多机热备＋负载均衡＋自动监控 |  |
| 12 | 可靠性 | 标准化的单节点，快速更换、升级、维护，客户管理工作最小化 |  |

# 校园服务框架总体设计

## 项目理解

大学服务中心建设是一项围绕所有智慧校园信息化的重大工程，其核心是通过建立一个完整的信息服务建模、开发、编排、运行、维护的校园信息服务管理框架，从而使得校园信息化的建设从传统的信息系统建设逐步转向基于智慧校园服务框架的信息服务建设。

其技术核心是建立基础性的信息服务底层支持，能过通过基础服务的有效定制和组合编排实现传统意义上需要通过管理信息系统进行定制开发的信息服务。

本方案基于互联网服务目前主流的微服务架构，建成一个从数据管理、服务封装、服务编排，到最终面向用户的应用／微应用服务的校园信息服务框架，通过服务框架能够覆盖绝大部分校园信息服务场景，并实现长期、可持续演进的校园信息服务建设。

## 通用服务的梳理与规范

校园信息化历经多年发展，已从原先的应用建设逐步发展到服务整合阶段。在智慧校园的建设阶段，首先要以顶层设计的方法，重新梳理高校信息化形成的各类通用的、可复用的组件化服务，在整体层面提出智慧校园的通用服务规范和基本服务内容。

## 服务框架的组成和架构

智慧校园服务框架首先面向校园已有的应用系统，将现有的应用系统通过服务改造后集成到服务框架中，提供通用基础的功能服务；其次要面未来新建的应用系统，通过通用的框架结构，实现新建服务功能的不断融入；此外，还要面向应用系统的开发者，提供开放的应用系统开发框架，帮助开发者快速地构建应用系统。因此，通过参考阿里等服务框架的服务组成，以及百度等服务框架的开发架构，结合高校信息化建设特点，研究智慧校园服务框架的组成和架构，为后续基于服务框架的智慧校园应用系统开发奠定基础。

## 服务调度架构的选择

### 3.4.1服务调度架构的设计目标

服务调度架构需实现对信息服务的灵活编排、统一管控、统一优化，实现最终交付的应用系统功能在个性化与标准化方面的平衡，既能提供基于服务组装的应用系统的足够功能和定制需求，又能通过标准化的服务编排技术大幅度提升应用开发与维护的效率，提高可扩展性，并有效降低开发维护成本。

### 3.4.2服务调度解决的问题

服务调度架构设计需解决的问题包括：

* 既支持业务层面的服务调度，也支持业务无关的服务调度
* 有效控制服务的颗粒度，保障合理的服务划分，同时控制服务开发与持续维护的门槛。
* 能够快速有效的将业务服务／通用服务组合编排，形成面向用户的微应用
* 服务调度机制与服务内部逻辑的边界清晰，降低耦合关系，同时具备足够的应用场景表达能力

### 3.4.3基于微服务架构的服务调度架构

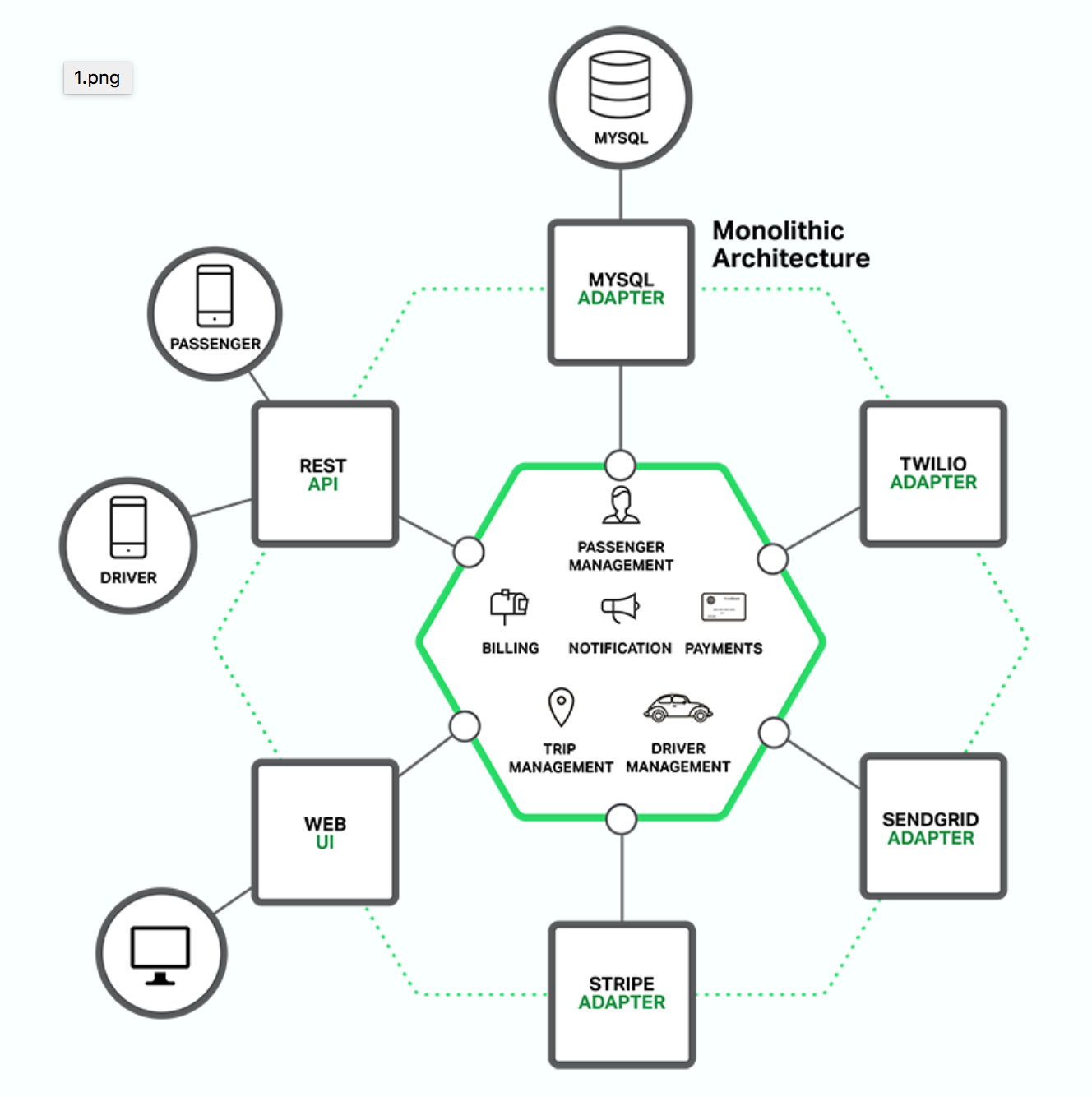
通过大量分析对比与实践经验总结，选择微服务架构理念设计服务调度的架构实现。微服务架构可以理解为：

采用一组服务的方式来构建一个应用，服务独立部署在不同的进程中，不同服务通过一些轻量级交互机制来通信，例如 RPC、HTTP 等，服务可独立扩展伸缩，每个服务定义了明确的边界，不同的服务甚至可以采用不同的编程语言来实现，由独立的团队来维护。

微服务架构被Gartner列为2016年十大战略性技术趋势，同时被Google等主流互联网企业及业务分布式计算专家评价为将在未来十年给软件开发的方式带来重大的变化。

#### 3.4.3.1与单体架构的对比

基于MVC框架、集中数据库等技术的ERP／MIS系统是典型的单体架构（Monolithic structure）应用。如下图示意：

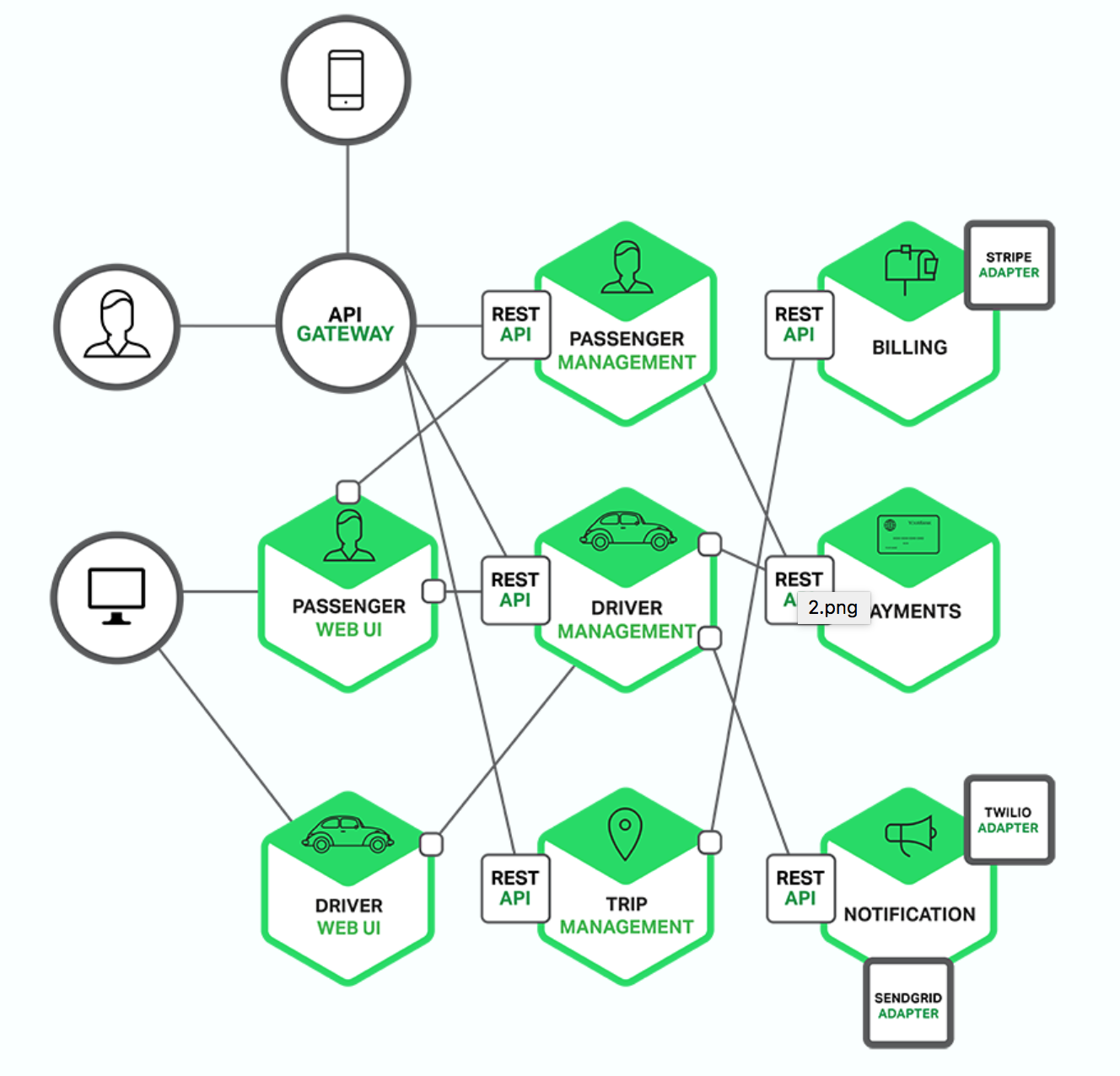


* 应用程序被部署为一个单一的文件或者同一个目录下的文件合集
* 所有应用程序代码都运行在相同的进程中
* 在缩放过程中，开发者需要将应用程序代码的多个副本部署到多个所需的服务器上

单体架构被普遍认为是松耦合系统的罪魁祸首，而在微服务架构中，

* 每个微服务都对应了一个独立的业务功能，也只定义了该功必须的一些操作
* 取代运行包含大量CPU和内存的大服务器，微服务可以被部署在更小的主机上，这些主机只需要满足其部署服务的需求
* 不需要遵循单一的MVC开发框架和开发语言，开发者可以根据业务的需求选择开发语言

由下图示意：



#### 3.4.3.2对SOA架构的发展与改进

SOA架构与微服务架构有很多共通性，如都是以服务封装的形式出现并进行组装编排。

SOA架构：

* + 集成重度依赖于中间件，尤其是ESB
  + ESB包含了大量逻辑——用于消息路由、模式验证、消息翻译和业务规则
  + 基于SOAP，较重量级的接口
  + 连接大型的Monolithic系统

微服务架构：

* + 消息驱动
  + 接口及数据的定义逻辑简单，易于理解，无业务依赖
  + 轻量级API
  + 连接简单的、高度解耦的服务代码

SOA架构解决不少问题，使数字校园具备了整体架构，解决了一些跨系统的整合问题。但在快速迭代的敏捷开放环境／互联网式的应用开发需求下，依然很糟糕：

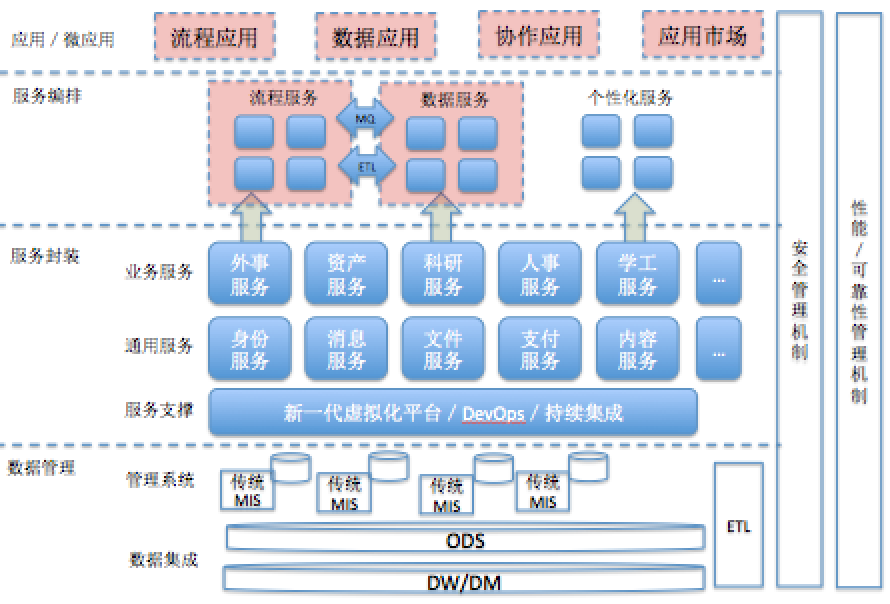
* + SOA中间件过于复杂，学习、开发、维护门槛依然很高，只有大团队、重投入的大型用户能够应对
  + 开发周期长成本高，在互联网＋的快速迭代的时代已经很难适应。很多技术已经在逐步被弃用
  + 经过多年的应用，SOA的价值和问题都已逐步清晰，互联网发展的影响下，无法依赖SOA解决问题

#### 3.4.3.3智慧校园服务框架下的微服务架构

服务框架的核心要素是：流程、数据。

最终面向用户的服务被分为三类：标准化的流程服务、标准化的数据服务、完全个性化的信息服务。

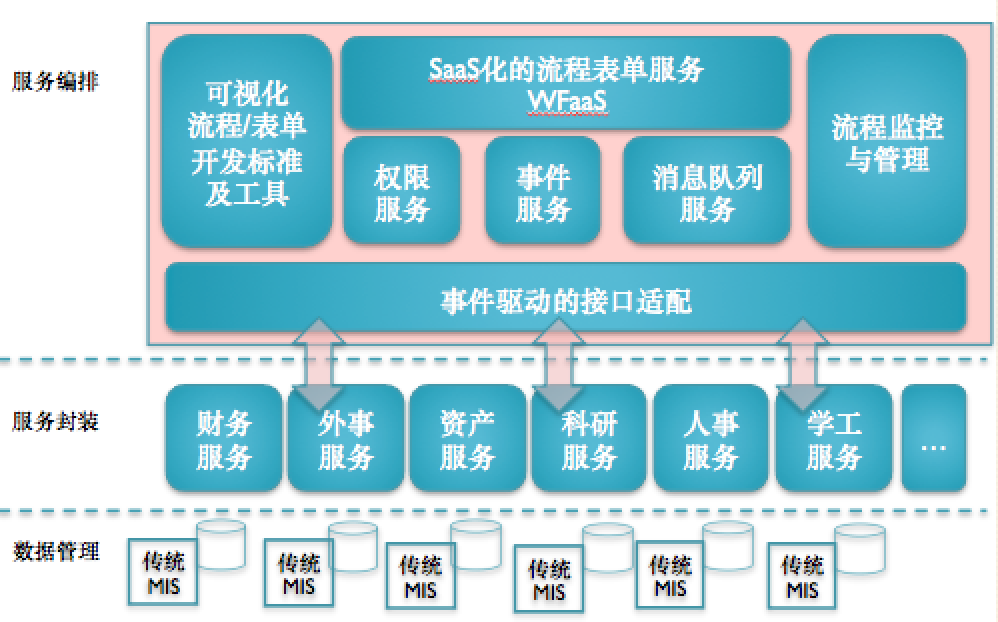
基于服务框架的整个信息化应用体系包括四个层面，如下图示意：



1）应用／微应用层。是面向用户的最终的服务组装的方式和结果。包括若干通用模型应用和涵盖个性化应用的应用市场机制。在这个层面服务组装的机制是非常简洁、清晰、易于理解、易于维护管理的。体现的互联网环境下微服务架构的一个重要设计理念。在这个层面需要“流程应用”、“数据应用”、“协作应用”这三个通用型应用来解决相关的可标准化服务的展示。“应用市场”机制解决的是完全个性化、开放式的应用及其管理。

2）服务编排。这个层面被设计为面向用户的交互服务的实现点。编排的对象是下层的系统服务，编排的方式包括标准化的流程模型编排、标准化的数据模型编排、完全个性化的信息服务编排。这三种编排方式的组合能够有效实现互联网高度个性化需求和企业计算高度标准化要求的平衡，同时这三者的组合基本能够覆盖所有智慧校园的服务编排形式。

流程服务编排采用WFaaS架构，通过从服务开发、流程开发、表单开发到面用户的交互界面的端到端完整的流程服务编排与交付。考虑到对遗留系统的支持，WFaaS编排的服务可以包括REST封装服务，也可以直接支持数据库服务编排。WFaaS涵盖了传统的表单引擎、流程引擎、消息队列等核心机制，把中间件提升为一个完整的云端服务编排。如下图示意：



数据服务的编排采用操作型数据表单及多维度BI展示框架进行数据服务编排，分别面向数据查询类服务和统计分析／决策支持类服务。考虑到数据服务，特别是统计分析／决策支持服务的特性，数据服务编排既需要面向REST标准的统一API封装的服务，也需要考虑面向ODS／DW／DM直接对接的服务编排。

如果新的需求，未来可继续扩展更多标准化的模型编排。

3）服务封装。这个层面是基于RESTful标准的系统间交互服务的封装与实现。这个层面分成三个主要部分：业务服务、通用服务、服务支撑。

* + 业务服务。指业务功能相关的服务封装。随着业务功能信息化的覆盖会不断扩展
  + 通用服务。与业务功能无直接依赖，属于智慧校园的通用服务封装。随着技术和应用场景的发展会不断扩展。
  + 服务支撑。微服务架构的重要底层支撑。应采用新一代虚拟化技术，实现自动化部署、持续集成、高效监控运维等重要的运维能力。是对上面两层的用户交互服务及系统交互服务的直接支撑。是微服务架构得以发展和应对问题的重要基础。

4）数据管理。这个层面包括两个部分，一是传统的MIS／ERP系统所在的层面。在微服务架构下，这些管理模块都可以是独立系统、独立数据库。二是数据集成部分。微服务架构将不在依赖重量级企业总线，通过ODS＋ETL实现被大幅控制的操作数据集成，通过ETL＋DW/DM实现数据／大数据的管理，以及数据分析模型的建模与展现。通过第三层的封装后可以支撑第二层的数据服务编排。数据集成部分的ODS／DW／DM，不仅仅指数据库，而是包括围绕ODS／DW/DM数据模型的建模、抽取、转换、装载、多维度组合、报表产生等完整的BI支持。

与四个业务层面正交的是业务无关的服务管理维度，包括：

* + 安全管理维度；
  + 性能及可靠性管理维度；

所有四个层面的服务，以及服务管理维度的机制，都可以在符合统一规范的基础上根据需要不断扩展。这个架构能够有效利用微服务架构的优势，弥补传统单体架构和SOA架构的缺陷，易于理解和管控，同时具备足够的表达能力和扩展性，可以基本覆盖所有的智慧校园信息服务场景。

## 服务调度的模型和实现

### 3.5.1业务层面的服务调度

微服务架构下，面向业务的服务调度被规约为两个主要机制：

1）基于端到端的业务流程服务 WFaaS的编排

* + 承载主要的业务／流程逻辑
  + 将人机交互／系统交互整合起来
    - 人机交互——表单交互与服务编排的对应
    - 系统交互——业务流程（BP）与服务编排的对应
  + 完全可视化——表单／流程一体化的可视化编排
  + 完全开放——语言无关、工具无关

1. 基于消息队列 MQ的编排
   * 高性能、高效率、高实时性的局部业务逻辑
   * 承载持久化、一致性、可靠性等非业务要求
   * 替代重量级的ESB

两者的有效组合及微服务架构的顶层设计，将大幅度降低对ESB机制的依赖，传统重量级的ESB机制将逐渐退化为ETL，主要用于数据集成。数据在源头上被WFaaS／MQ有效控制，进一步降低ETL数据集成需求。

### 3.5.2业务无关的服务调度

业务无关的服务调度包括两大部分功能：

运维管理：

* 部署、注册、管理、发现、监控
  + 自动化部署
  + 集中注册、发布、管理
  + 监控、服务的关联与智能发现

服务交付；

* API Gateway
  + 集群式运行
  + 流量分发
  + 协议转换
  + 权限控制
  + 缓存管理
  + 统计日志

## 服务调度的支撑机制

### 3.6.1服务的底层封装

智慧校园涉及的业务范围极其广泛，其对应的服务数量也非常可观。微服务架构能以清晰、可方便扩展、可方便维护的方式组合大量的服务，但当服务的数量达到一定的数量，并且需要做性能和可靠性的冗余的情况下，服务需要更好的底层封装技术来应对管理和调度的复杂性。

底层服务封装的要求包括：

* 封装的颗粒度更细，面向服务而不仅是虚拟机
* 自动化部署
* 持续集成
* 快速启停、复制、集中化管理和监控

针对这些要求，目前IaaS层的虚拟机VM技术在服务数量达到一定程度后会明显无法应对，需要新一代基于容器技术的支持。

### 3.6.2性能可靠性管理

微服务架构将整个服务框架更为扁平化，从数据到系统服务到用户交互服务，进行了一体化的整合。服务框架将成为智慧校园的重要基础。因此，全面的性能和可靠性管理成了一个既迫切需要又可行的管控维度。

在微服务框架下，性能和可靠性（故障）管理可以被统一的用分布式架构来实现。通过引入端到端的完整性能管理解决方案，实现从用户终端、应用服务、服务编排、业务服务、通用服务、容器支撑、硬件基础设施的完整性能（可靠性）分析监控机制，从而能够实现完整、全面的用户体验、服务能力的精细控制。

在统一的性能可靠性管理机制的指导下，各层面可具体实现特定的性能可靠性优化。

### 3.6.3安全管理

服务框架范围内的安全管理是基于整个信息化环境的安全体系设计，成为整个安全体系的一个细化的组成部分。服务框架范围内的安全体系将和整个信息化环境的基础安全机制（包括防火墙、入侵检测、访问控制、网络隔离等等各种手段）进行有效组合，以实现最终的安全性。

服务框架范围内的安全管理机制包括：

* 管理实现层面：
  + 其安全机制主要采用传统的IaaS、数据库、应用系统的安全机制。
* 服务封装层面：
  + 基于服务容器管理（CaaS）的安全机制解决系统服务的隔离与安全性
  + 基于HTTPS标准解决传输层面的安全
  + 基于OAuth标准解决数据开放授权及访问控制安全性
  + 可以选择通过使用API Gateway机制，实现安全策略的集中交付
* 服务编排层面：
  + 基于流程服务（WFaaS）／数据服务（BIaaS）的安全机制解决面向用户的交互服务的隔离与安全性。
  + 基于HTTPS标准解决传输层面的安全
  + 基于OAuth标准解决数据开放授权及访问控制安全性
* 应用／微应用层面：
  + 基于SaaS层面的安全机制解决应用隔离和租户的隔离与安全性
  + 基于SAML等认证标准，以及其他辅助身份认证手段，实现用户隔离及安全性
  + 基于OAuth标准实现基于用户授权的资源／数据授权管理
* 探索在整个服务框架体系内实现一定程度的入侵容忍

安全是个相对的概念，已有的信息基础设施安全体系，加上上述的分层安全机制，依然并不能完全防止极端安全事件及恶意入侵的发生。智慧校园服务框架是面向全体师生甚至面向公众的开放平台，因此应当进一步考虑在发生安全问题时，如何限制损害范围，使得入侵事件局限在部分系统、部分模块、部分数据之中，降低整体系统破坏及整体数据丢失的可能性，即系统应具备一定的入侵容忍度。

上述的微服务架构把整个服务框架约束在一个可控制的标准化架构和标准化建设方式之中，因此，一定程度的入侵容忍机制成为可行。

在整个服务调度架构中将在上下两层之间，再加入一层防护，该层防护是分布式的，不形成单点故障，同时屏蔽核心数据。通过该层防护，层与层之间的交互方式可单独设计，不仅提高对核心数据的保护层厚度，同时，即使上层某一台服务器完全被攻破，攻击者也只能侵入部分系统及部分数据，不会直接获取到核心数据的访问入口。

# 系统设计方案

## 建设内容

大学服务中心的概念与目前市场上流行的面向师生服务的理念是完全对应的。

大学服务中心在各面向业务部门的管理系统和终端用户界面之间增加了一层面向服务的机制。这一层可以屏蔽下层业务管理系统的复杂性和逻辑，以统一的方式面向终端用户。用户不需要了解最终的服务由哪几个系统实现，实现真正的教职工一站式服务体验。

• 大学服务中心的流程中心面向用户可以提高体验，降低操作的学习成本。

• 面向开发人员而言，”一张表”工程中的流程完全是通过标准化的流程开发工具完成，降低开发的学习成本，同时大幅度提高标准化和流程质量，使得流程开发服务成为软件服务供应链中的一个标准化环节，这是新一轮信息平台建设的核心目标之一。

项目的建设内容主要包括两大方面：

1）统一的表单流程服务开发与运行平台

该平台包括以下核心功能：

• 全功能的表单开发工具与处理引擎，可以可视化、零代码地开发任意类型的表单并实现高效率、高交互的处理

• 一体化的流程开发工具与处理引擎，实现与全功能表单的无缝融合、一体化开发与运行。实现流程逻辑设计与程序代码开发高度分离，业务流程咨询师与开发人员可以独立工作、高度融合

• 实现与全校各业务系统的标准化融合。通过高度易用的标准化接口体系，实现统一流程服务与各业务管理系统的无缝融合。以完备的接口体系实现业务流程功能的无限制扩展

• 实现统一的师生”一张表”工程服务中心，并与数字校园门户无缝融合，提供统一的流程服务体验

• 实现全校流程的统一管理、监控、优化分析功能

2）基于统一的表单流程服务平台开发符合需要学校管理服务需要的业务流程

基于流程平台，建议可以在以下领域建设一站式业务流程：

• 教学服务领域

在教学管理领域存在大量面向师生的服务可以用流程的方式进行建设，包括各类课程管理流程、学位服务流程、奖学金及各类奖项申请、迎新离校服务等等。

• 行政管理及公共服务领域

行政管理信息化是业务流程数量最大的领域，包括人力资源类流程、出入境管理等外事流程、请假用印等办公类流程、资产管理类流程、科研管理类流程等等

• 财务服务领域

标准的财务系统解决的是会计管理功能，而在学校内存在大量与财务相关的服务可以用流程信息化来提高效率。包括：预算制定、师生报销预约、与科研管理的整合、与资产管理的整合、与住宿、学费等等各项服务功能的整合。这些流程是高度个性化并不断改进的，最适合用灵活的流程平台建设，对标准化的财务系统是很好的补充。能够是财务系统专注于专业的财务管理，而把个性化的服务由更适合的平台来实现。

## 标准规范建设内容

通过本项目建设，在学校内形成以下重要的信息化标准规范：

1）可视化表单的开发与运行标准

使用Office Infopath编制任意类型表单

内置高性能表单渲染引擎：

——将Office表单全自动渲染为交互式的web网页

——无需任何其他开发环境或开发工具

2）可视化的业务流程开发与运行标准

全功能的表单引擎与可视化的流程引擎无缝整合

——用流程驱动所有表单的处理逻辑

——可以实现从表单绘制、流程配置、流程运行的零代码开发

——实现业务逻辑设计与程序代码开发高度分离，业务咨询师与开发人员可以独立工作、高度融合

3）标准化的流程服务（WFaaS）开放API体系

用标准化的Messenger事件接口模型描述所有流程扩展逻辑

可以通过标准化的方式实现业务流程与管理系统的无缝整合

## 基本数据描述结构与逻辑关系

**表单数据结构：**

* 表单存储采用可扩展的XML数据结构
* 表单数据传输采用JSON数据结构

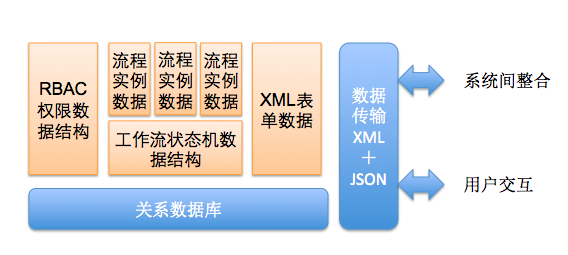
**流程数据结构：**

* 采用标准的工作流状态机对应的数据结构。

**权限数据结构：**

* 基于标准的RBAC模型设计。支持部门、岗位的二维数据结构。

数据逻辑关系总体设计如下图示意：



## WFaaS总体架构

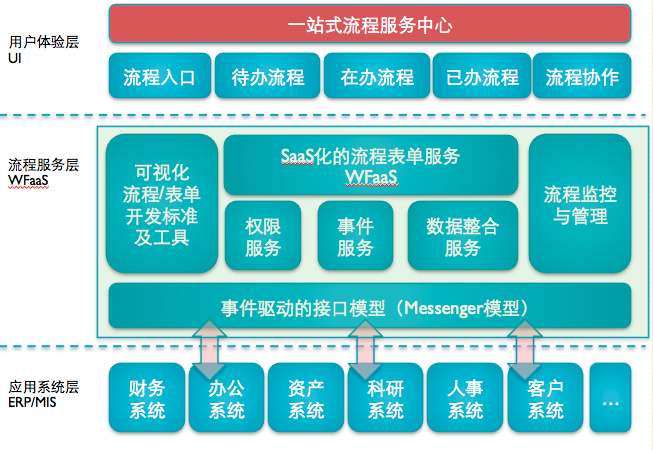
WFaaS是传统的BPM向SaaS模型的演进，也是科探SaaS开放平台产品线的核心部分之一。

1）建设独立BPM层。将面向用户的服务流程从应用系统中抽取出来移植到标准的BPM层中。将分散的流程开发模式变为标准化、可管理、跨系统的流程开发

2）建设统一的UI和API层。流程服务完全屏蔽应用系统，用户对所有流程拥有近乎同样的使用体验，不需关心来自哪个或哪几个应用领域

BPM＋统一的UI／API＝WFaaS

WFaaS的总体架构如下图示意：



该架构下的主要功能模块包括：

|  |  |
| --- | --- |
| **模块名称** | **功能描述** |
| 大学服务中心服务中心（及下属所有功能） | 师生及各级管理者用户对所有流程操作的入口，屏蔽了各种后台系统的复杂性，不再需要知道专业系统的存在，师生通过统一的流程入口和待办／已办流程操作方式，实现各种信息服务和办事流程。 |
| SaaS化流程表单服务 | 对表单编辑器生成的表单上传至平台后进行解析并实现HTML／JS／CSS渲染，并实现与用户的数据交互；实现工作流引擎，实现流程编辑器生成的业务流程的严格执行；将表单数据与工作流流程进行整合，实现工作流执行过程中对表单数据的读写操作。 |
| 可视化流程表单开发工具 | 实现可视化的工作流及表单的开发，形成系统标准的工作流描述结构和表单描述结构，交由SaaS化的流程表单服务进行处理 |
| 流程监控与管理 | 支持多级权限下不同角色对流程的运行过程进行监控与管理；对流程运行的全过程进行数据采集，制定分析指标，对所有业务流程进行优化分析，提供决策支持。 |
| 事件驱动接口模型 | 通过标准的事件驱动模型实现流程平台与各现有业务系统、数据库的整合；支持在流程运行的各个环节触发不同类型的事件并实现数据交互；提供基于Java／.net的事件模型开发包 |
| 权限服务 | 基于RBAC模型实现完整的权限管理，包括岗位／角色管理，部门管理，岗位与部门的二维结构，基于岗位／角色的授权管理等等。 |
| 事件服务 | 通过实时的消息／事件队列机制，实现分布式系统间的实时、高并发事件交互。保障PIM系统的各模块之间高效率通信 |
| 数据整合服务 | 实现了各业务管理系统间的流程整合，其过程中需要支持各种类型的跨系统或者共享数据库与流程平台间的数据整合，特别是大数据量的数据整合，需要有专门的机制保障这些数据整合的效率和可靠性 |

## 可视化的流程设计

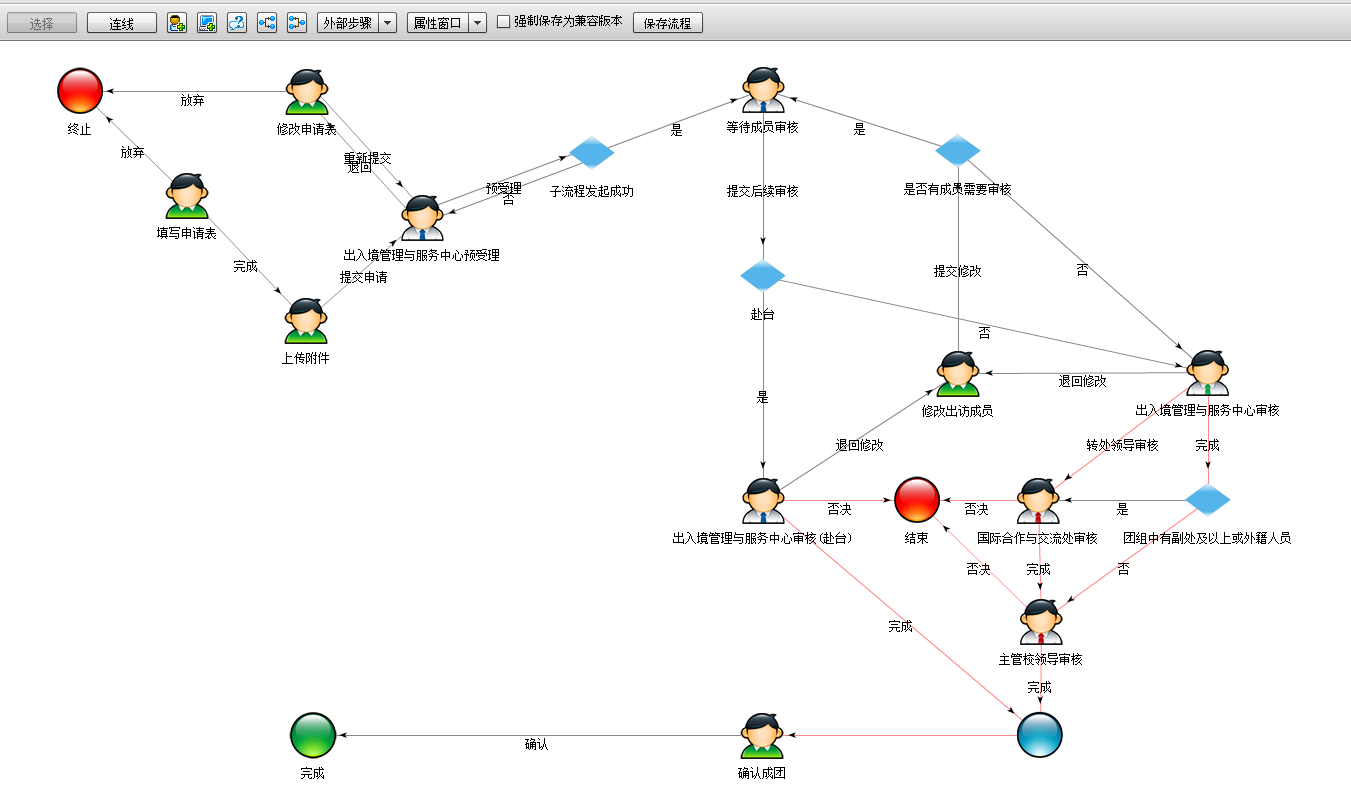
流程设计采用完全基于web的可视化设计技术。工作流程的开发完全在“云端”进行，即开发者访问部署在用户环境中的InfoPlus私有云服务，即可在浏览器环境中完成所有的开发工作。

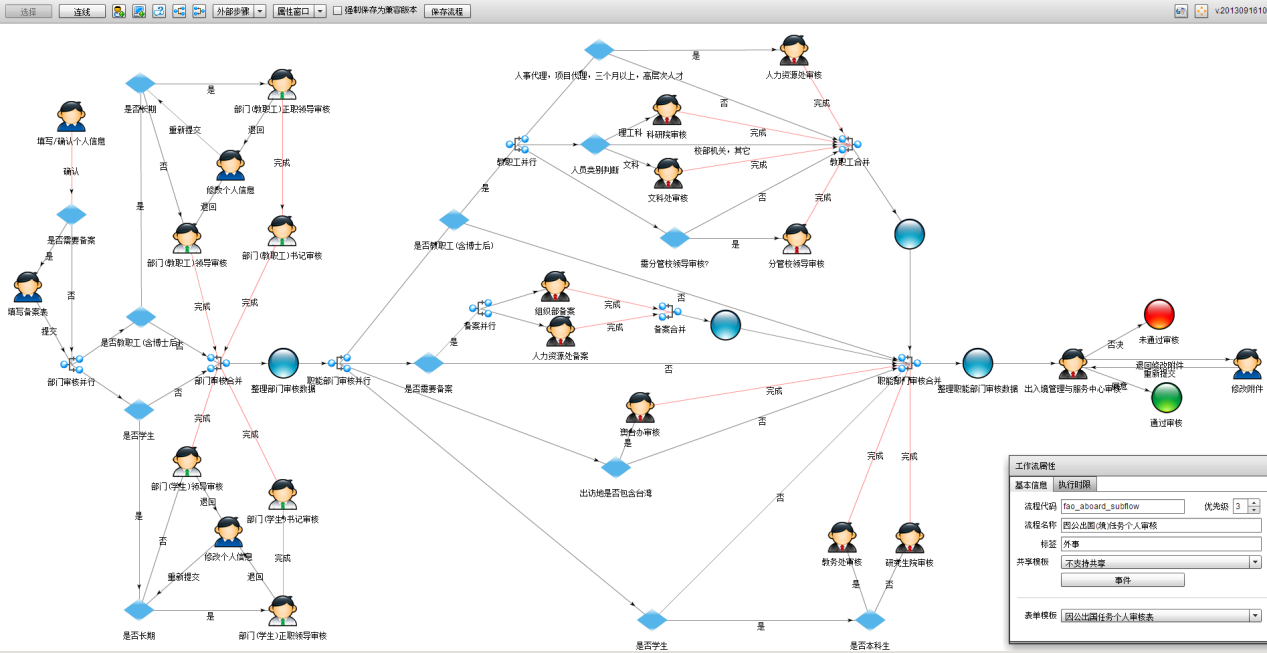
工作流模型的开发实现了完全可视化，即通过类似MS Visio的拖曳式流程状态模型编辑工具进行流程编辑。流程状态模型包括7个基本元素，通过拖曳式编辑可以形成所有类型的工作流模型。七项基本元素包括：

* + 人工交互节点
  + 自动处理节点
  + 外部页面节点
  + 条件判断节点
  + 并行开始节点
  + 并行合并节点
  + 连线——状态迁移动作

Macintosh HD:Users:mac:Desktop:屏幕快照 2014-09-15 下午5.01.34.png

可视化编辑环境如下图。编辑环境除了构建流程模型图外，还可以对流程的上述7项基本元素进行各种属性配置。





## 可视化的表单设计

表单设计采用基于Office Infopath的可视化表单技术，开发者只需掌握Office排版技巧，实现了最低的学习曲线。

系统实现了效率极高的解析引擎，将office表单自动转化为交互式的web页面，无需一行代码，即可实现全功能的表单设计。这一技术全面领先于业界相关的流程产品。

* 表单、流程的制作方式与word／visio几乎吻合，引擎将其直接转换为web系统，完全所见即所得；
* 可视化的开发工具使得流程咨询顾问得以用最精确最易理解的方式定义流程需求，并与代码工程师轻松协作；

设计时界面与运行时界面高度一致：

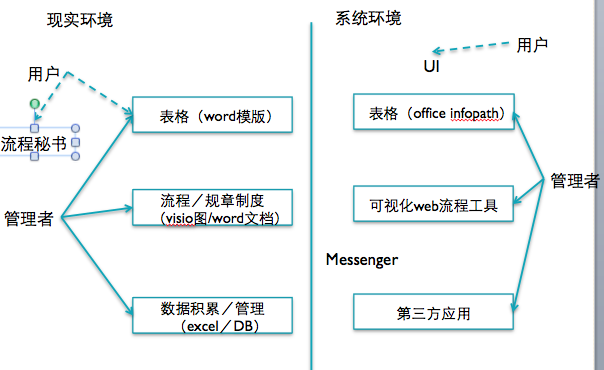




## 标准的事件驱动模型

表单和流程在WFaaS中得到了无缝融合，同时通过标准的事件驱动模型——信使模型Messenger，实现流程与外部应用和数据库之间的交互。

信使模型实现了流程系统模型与实际的业务工作模型的一一对应，简介而容易理解。



## 开放性与云端化

在互联网时代，开放性是学校信息基础架构得以长期成功发展的重要前提条件。

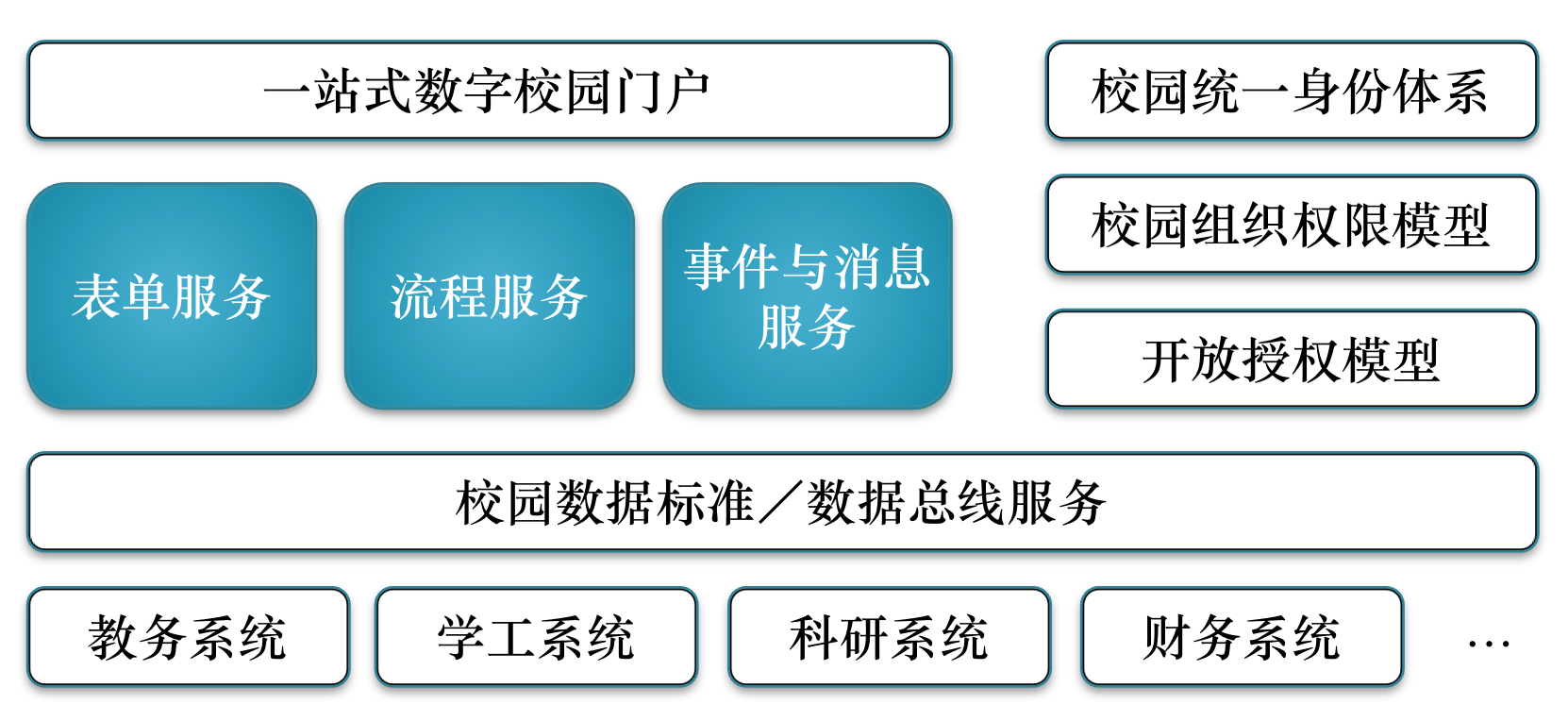
科探Infoplus流程平台是一个云服务平台还不仅是一个开发工具。科探InfoPlus流程云服务平台在开放性和云端化方面具有以下重要特性：

* 平台是分布式的，可以实现异构系统的对接
* 平台是云端化的，即平台具有一体化的基于web开发、配置、运行、测试环境，无需安装特定的开发工具／插件。基于云端的开发、测试一体化环境使得用户可以得到类似Google GAE这样的PaaS开发体验。
* 基于平台的流程开发支持各种语言、各种开发工具对流程进行扩展，至少支持java／.net／BeanShell脚本。以实现足够的开放度
* 不限制流程处理代码的开发工具。最大限度降低开发的特殊性和对开发人员的培训门槛
* 能够实现与java/.net开发的各种应用系统与流程的对接
* 分布式部署架构，流程和处理代码均可以独立部署，分别升级改进
* 平台升级后，已开发的流程可以实现平滑迁移，完全不需要重新编译
* 能否做到无缝升级

# 系统核心特点与优势

## 与数字校园完善对接“即插即用”

方案的整体设计深入考虑了数字校园完整信息架构的需求和技术特点，在长期积累的数字校园整体架构设计经验和实践经验的基础上，平台内置实现与市场主流数字校园基础服务平台产品的融合对接，实现面向数字校园的“即插即用”。如下图示意：



* 与统一身份体系的对接

平台内置实现与市场主流的CAS、SAML等国际通用身份认证标准的对接。此外，针对高校市场大部分主流统一身份认证产品的内置对接支持。

* 与组织架构岗位权限的对接

基于Infoplus独有的“三元组”架构，实现与各种应用场景有效兼容的组织架构与岗位权限对接。

InfoPlus所需的组织架构与岗位信息，包括三个实体和一个三元组关系，如：

1.用户（User）

2.部门（Dept），部门存在父子关系

3.岗位（Post），有时候也可以看成用户类型，或身份，或角色

4.三元组（UserId，DeptId，PostId），即三者的关系表

所有的用户，均落在部门、岗位构成的矩阵的交点上，即：

用户A是软件学院（Dept）的教师（Post）

或用户B是教务处（Dept）的外事秘书（Post）

或用户C是机械学院（Dept）的研究生（Post）

对于上述的实体和关系，InfoPlus通过对外部接口定期轮询实现简洁、无外部依赖的同步机制。如果的确需要更高的实时性，则使用消息队列和用户定制接口的方式实现快速对接。已在数十个高校场景中有效验证。

* 与OAuth开放授权体系的对接

内置完善的互联网主流授权体系OAuth的协议支持。内置支持OAuth协议参数配置，快速对接。所有API均受OAuth协议保护，基于RESTful标准实现，有效开放。

* 与数据总线／数据标准的对接

与业界主流的ODI、BizTalk等ESB中间件产品均在众多场景中实现了有效对接，能够为用户提供现成方案，实现与用户已有数据管理方案的对接。

* 与数字校园门户的对接

支持完整的开放API，所有本方案内置的服务中心功能均基于开放API实现。开放API能够为第三方的数字校园门户提供完整的对接支持。已实现在众多场景中与不同供应商的web门户、移动门户，在不同层面实现的对接案例。

* 与应用系统的对接

基于事件驱动的标准化接口体系，实现与各类应用系统的有效整合。参见6.4节。已在数十所高校，实现了数以百计的流程服务与业务系统的功能对接。

## 深度优化的校园服务可视化开发

本方案基于WFaaS理念，从用户体验到开发体验，实现了一个完整的流程服务开发、运行、维护、管理的一体化云服务。在一体化服务中对整个流程服务开发过程进行了深度优化，通过可视化的配置模式，实现了大量的信息服务开发优化。包括：

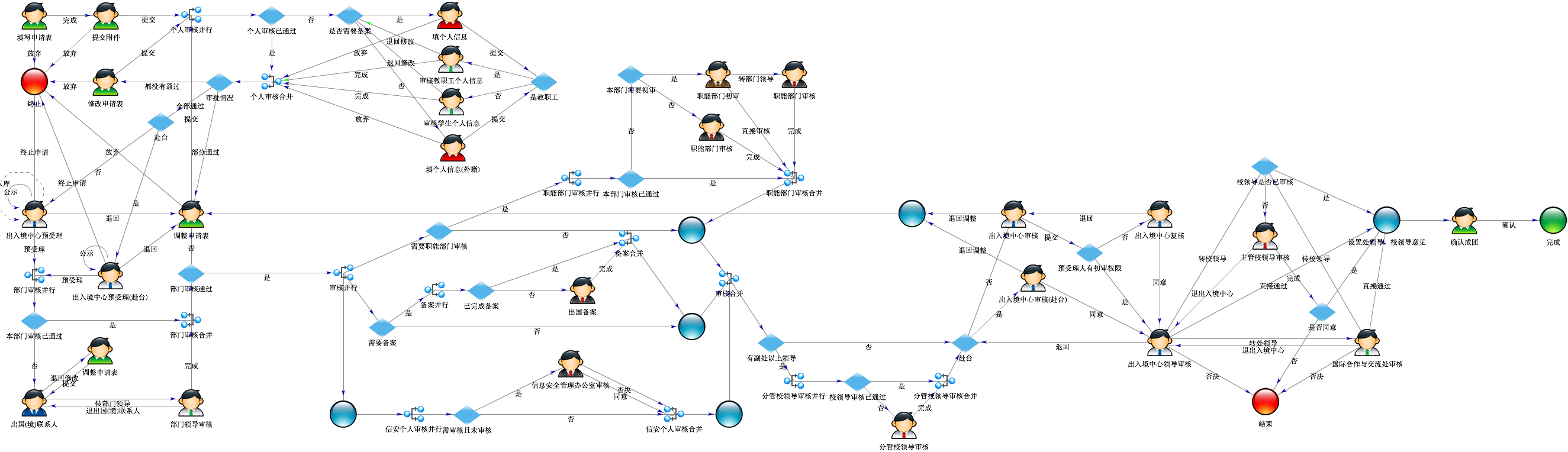
* 用户交互页面的可视化：

用户交互界面通过表单定义，表单的开发与传统的程序语言／HTML标签语言完全无关。采用基于Office的表单开发工具，能够有效的实现现有Office文档自动导入并转化成XSN开放表单定义标准。通过表单引擎自动渲染为可执行的web表单页面。整个过程完全可视化、所见皆所得，且无需任何代码介入。参见6.2



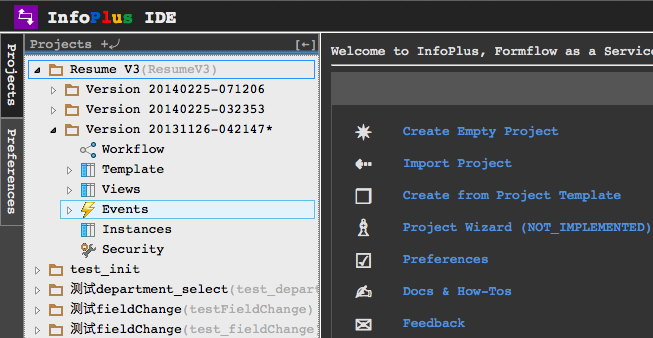
* 流程模型的可视化开发：

基于标准的流程状态机模型，实现完全可视化的业务流程开发。业务流程的可视化和表单可视化无缝对接，能够实现表单数据对流程状态的直接驱动，且配置过程无需代码接入。通过事件驱动方式整合的流程处理代码也完全无具体的开发语言开发工具依赖。参见6.1



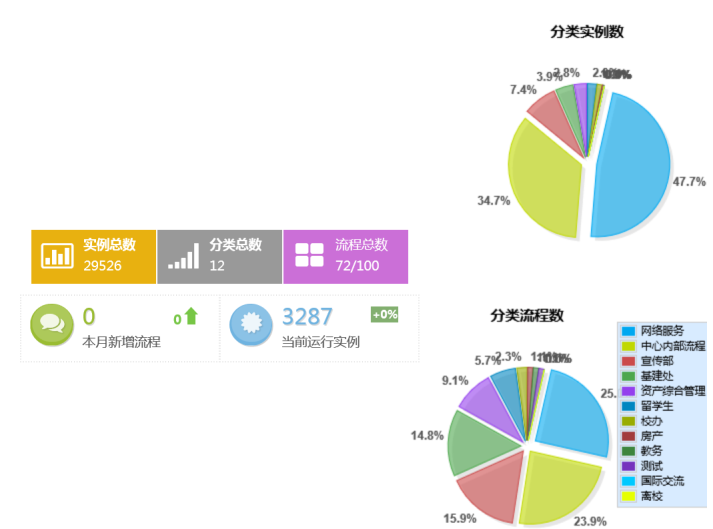
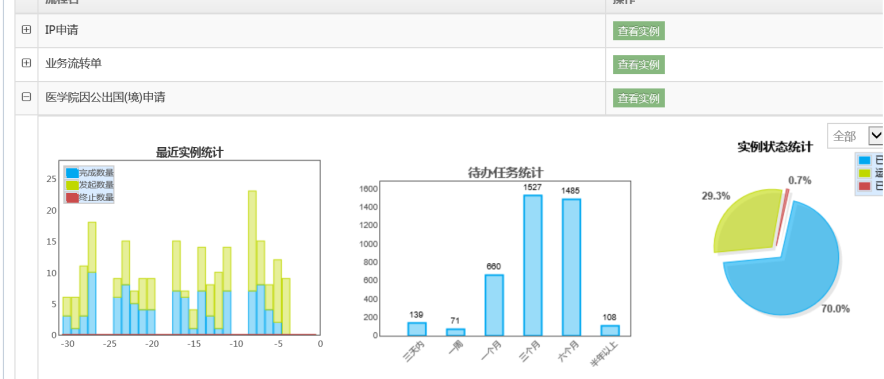
* 基于云端IDE的集成式开发：

表单、流程的开发集成在一个完整的基于web的云端开发环境中，支持大量配置项，实现各种流程服务场景。同时集成开发环境提供清晰易懂的表单流程版本管理。流程编辑器内置的Beanshell动态脚本、正则表达式支持，能够使开发者在可视化环境中直接操纵表单数据并影响流程执行状态，无需编译、发布等传统的程序开发过程。



* 流程服务监管的可视化：

基于Dashboard模式实现统一的流程运行管理视图，对所有流程服务的运行情况实现有效的可视化管理与监控。



本方案提供的经过深度优化的可视化信息服务开发模式，经过众多高校累的一站式师生服务中心多年的开发运行实践，证明了具备非常高的流程服务开发效率和质量，同时在长期、可持续维护方面，相比传统的流程中间件开发模式具备明显的优势。

## 无缝融合的移动应用环境支持

本方案的架构中针对互联网服务的终端开放性和多样性，设计了独有的多视图架构。通过一个统一的后台服务开发架构，解决多终端、多视角、多语言等复杂场景应用需求。

基于Infoplus IDE能够实现一个流程、一套数据，根据不同的流程节点、不同的用户终端、不同的语言进行组合，轻松应对异构复杂的互联网环境

方案提供多种移动应用适配场景，各种场景可无缝组合、切换：

* 现有的iOS／Android原生大学服务中心应用app，无缝实现与后台流程逻辑的对接。
* 提供基于移动终端浏览器的HTML5版本的大学服务中心，支持所有主流的移动浏览器兼容访问，实现完整服务中心功能
* 提供微信企业号／服务号支持，将完整的大学服务中心功能整合到用户的微信企业号或服务号功能中
* 提供基于桌面浏览器的HTML5版本的大学服务中心，支持包括IE 8及以上版本、Firefox、Chrome、safari等所有主流的桌面操作系统浏览器兼容访问，实现完整服务中心功能
* 系统后台能够自动检测用户终端大小、语言设置等参数，通过多视图架构推送合理的视图，实现一套后台，各种终端的支持。
* 系统后台的所有API均支持OAuth2.0标准，支持基于用户的安全的开放授权，适合移动应用场景，且和目前主流的互联网／移动互联网授权协议兼容，能够有效提高移动应用开发的标准化程度和安全性。

多视图架构不仅解决流程服务的终端适配问题，而且能够有效降低移动应用场景开发的难度和工作量。一项服务发布成移动的“微应用”，只需定制一个新的视图即可，其他的处理交给平台解决。基于本方案，在后台流程逻辑开发完成的基础上，面向移动终端、移动应用的功能迁移完全无需程序代码介入，能够以“天”为单位实现大量移动功能的迁移，且适配各种终端使用场景。

多视图架构还能够有效应用于多语言场景，无需修改代码即可基于平台即可实现流程服务国际化（多语言）界面的发布和运行。系统会自动根据终端语言设置进行语言界面适配。

## 程序语言无关的完全开放性

大学信息服务环境不同与企业信息服务环境的一个最重要的特点是开放度：大学庞大的信息服务需求以及不断变化的师生群体，决定了不可能将各种信息服务的开发统一到一种开放语言及其工具体系中。

大学的信息服务环境必然需要具备高度的异构性、开放性和兼容度。

基于本方案的业务流程开发，不限定程序开发语言和开发工具。完全的开放性。代码开发者可使用任意开发语言、开发工具，无需安装开发插件。特别的，为主流的Java／.net框架提供成熟易用的SDK。流程编辑器内置的Beanshell动态脚本和正则表达式支持，能够使开发者在可视化环境中直接操纵表单数据并影响流程执行状态，无需编译、发布等传统的程序开发过程。

流程所需的所有代码均是事件驱动的标准化“微服务”程序。核心是跨平台的HTTP协议，系统可为相关应用系统提供共享数据访问服务，为第三方软件及二次开发提供基于OAuth和RESTful标准的服务接口；

在已有案例中，能够支持包括Java、.Net、PHP、Python、Ruby的各种语言进行完整的流程开发

# 系统主要功能特性

## 全功能流程引擎

### 6.1.1可视化工作流编辑/Visual Workflow Editor

工作流模型的开发实现了完全可视化，即通过类似MS Visio的拖曳式流程状态模型编辑工具进行流程编辑。流程状态模型包括7个基本元素，通过拖曳式编辑可以形成所有类型的工作流模型。七项基本元素包括：

* + 人工交互节点
  + 自动处理节点
  + 外部页面节点
  + 条件判断节点
  + 并行开始节点
  + 并行合并节点
  + 连线——状态迁移动作

支持上述元素的任意组合，系统支持流程图有效性自动编译检测，开发过程中如流程图存在逻辑错误，会自动提示。

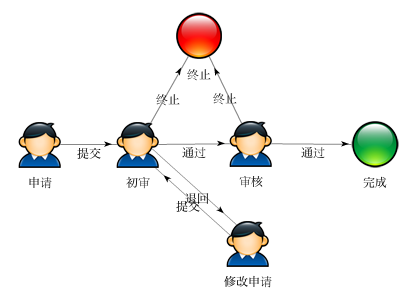
支持丰富的流程节点／动作的属性配置，支持工作流优先级。支持基于标签的工作流管理。

infoplus支持完整的流程模型：人工节点、自动节点、条件节点、子流程、并行与归并等各种流程要素及其任意嵌套；支持包括多人任意、多人抢占、指定执行、会签、传阅等常用模式；支持撤回、评价、催办、通知、过期、中止等常用操作；以及上述功能的各种组合

支持混合流程：业务流程引擎完全支持人工参与流程和自动处理流程。

可视化过程监控，跟踪活动状态，发现流程错误时可人工干预。

与权限体系实现无缝整合：平台提供完善的组织架构管理，支持多维权限模型。支持任意级别的组织架构管理，支持基于角色／岗位的权限体系RBAC，支持分级授权管理，支持权限体系与学校统一身份体系的无缝对接，支持权限体系与流程表单数据的无缝对接。



### 6.1.2岗位-步骤级基本权限控制/Posts-Step Authorization

支持丰富的流程权限控制，基于岗位／流程状态的组合控制，即可以在任意流程状态配置可以访问的岗位组合。

对于可以访问的用户，还可以精确配置表单中每一个变量在不同流程状态下的不同的增、删、读、写功能。从而能够方便地实现不同的流程状态下的用户的不同表单字段访问权限与可见性。



### 6.1.3执行人限定-部门限定的高级权限控制

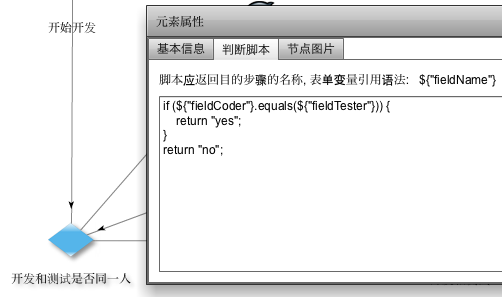
在复杂的流程场景下，可以对每一个流程步骤下的最终经手人进行精确、灵活、方便的限定，从而最大限度地实现流程以合乎真实逻辑的方式进行流转。



### 6.1.4Shell脚本支持/Shell Scripting

支持在流程编辑中直接编写脚本对表单数据进行存取，包括：条件节点中基于表单数据进行的条件快速计算；自动节点中对于表单数据进行操作；

脚本可以直接访问流程数据，无需编译，系统直接解析执行，是最高效的数据操作方法。



### 6.1.5实例和步骤级别超时

可以针对流程实例以及流程步骤，精确控制流程超时时间。

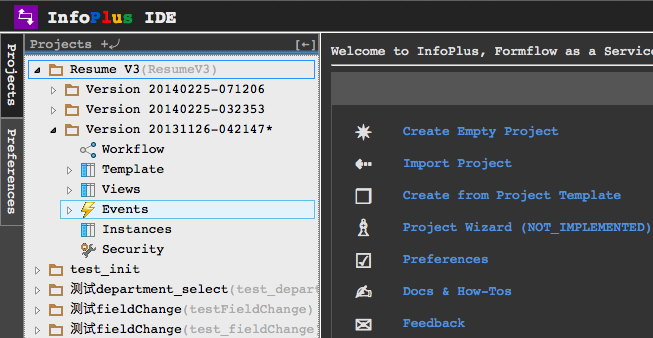
### 6.1.6灵活的选人方式

支持在流程状态迁移时，实现足够灵活地对下一个流转步骤地具体操作人的指定。



### 6.1.7工作流版本控制

工作流程均支持版本管理，可以边运行边升级，新旧版本可以同时运行，支持新旧版本数据无错误升级

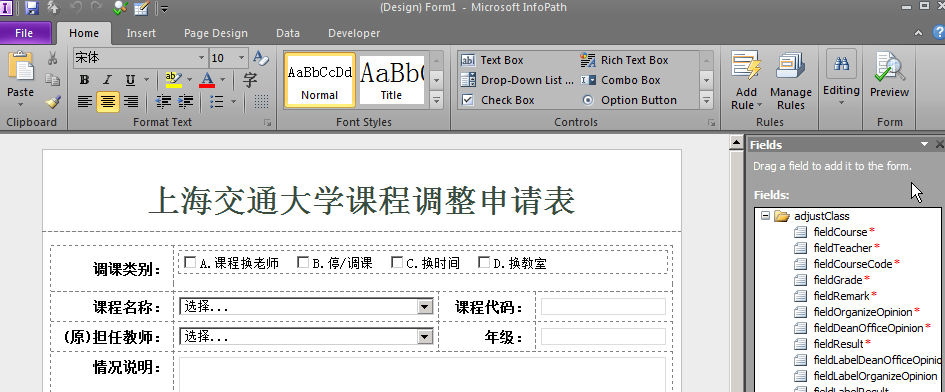


## 全功能表单引擎

### 6.2.1可视化编辑

表单设计方式：支持可视化建模，提供丰富的组件库图形化设计表单，系统支持所见即所得的拖曳式表单设计，使用体验完全类似ms office常用体验。支持word、excel粘贴后修改。

兼容Microsoft InfoPath 2007－2012



### 6.2.2字段级功能设置

支持表单数据定义、控件类型定义，正则表达式验证、代码字段级联、通过公式配置表单间管理。支持条件样式。同时，支持完整的事件触发代码，配合事件响应代码可以完成任意类型的表单数据验证。

支持字段初始值、字段级验证、代码表绑定等。



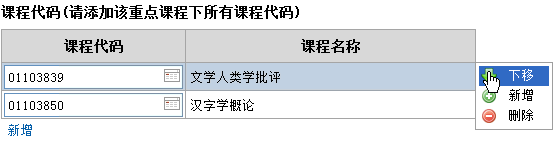
### 6.2.3字段-步骤级权限控制

支持角色／组织架构的矩阵式模型，权限配置的颗粒度可以细化到每个流程、每个节点、每个字段的增、删、读（可见）、写权限。



### 6.2.4可嵌套的重复字段、重复节

支持以最简易的office编辑方式，定义表单中的动态数据结构，支持重复表，以及任意控件组合而成的重复节。



### 6.2.5表单打印功能

系统支持方便的打印配置：只需在流程指定哪里步骤需要打印，系统自动实现打印页面的渲染，所见即所得。且无需任何代码开发，无需任何报表开发工具。

如需要特定的打印要求，可以通过与定义表单同样的方式，在Office InfoPath中定义一个打印视图即可。

### 6.2.6丰富的控件渲染类型

支持常见数据类型控件、人员院系，附件上传、图片框、事件日期、感知框、隐藏字段、帮助、备注等。



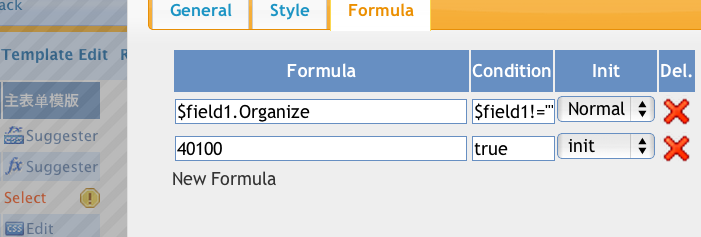
### 6.2.7基于正则表达式表单验证

系统支持通过正则表达式，快速定义表单中某一数据字段的验证方式。系统会自动解析正则表达式并实现前端快速验证。无需任何程序编码。



### 6.2.8条件公式和条件样式

系统支持丰富的字段间的关联关系，配置方法简单：在任意字段发生变化时，系统会自动触发执行一个开发者配置的条件计算。如条件满足即可对其他字段的数据和样式进行修改。这是快速开发界面友好的表单的重要手段。



### 6.2.9表单变量

内置、外部、字段三种基本变量，普遍应用到消息接口、工作流设置、表单字段设置、条件样式和公式之中。

### 6.2.10表单版本控制

表单支持版本管理，可以边运行边升级，新旧版本可以同时运行，支持新旧版本数据无错误升级

## 代码表（Data Exchange）

### 6.3.1内部代码表

通过数据同步、手工输入或导入

### 6.3.2外部代码表

运行时通过标准接口获取

### 6.3.3代码表的首字母拼音感知检索

代码表通过字段级别绑定



## 事件驱动的标准化接口体系

### 6.4.1应用-工作流多对多的消息接口

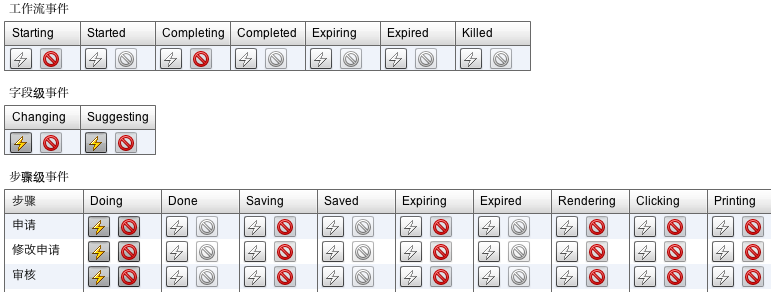
系统支持流程平台与任意业务应用系统的对接。通过部署平台SDK并注册后，可以在开发环境中快速定义向某个应用发送流程执行中的事件。



### 6.4.2贯穿表单实例的整个生命期

多达16种不同的事件接口。

* 7个发生在实例生命期
* 7个发生在步骤生命期
* 2个发生在界面交互过程中



### 6.4.3应用端数据干预

在复杂的应用场景下，应用端通过标准的消息接口在流程创建、步骤渲染、数据提交等进行数据或界面的修改。应用端数据干预能力，几乎赋予了流程平台无限的数据操作能力。

### 6.4.4应用端表单验证

除了上述的自动配置前端验证外，在复杂场景，及高安全性要求场景中，可以通过触发应用端表单验证事件，实现验证较为复杂的业务逻辑。

### 6.4.5应用端流程干预

在复杂的应用场景下，可以通过SDK调用，实现在业务应用中干预流程的挂起、取消、中止、超时、手动步骤的自动完成等。

### 6.4.6应用端数据持久

由于具有完备的事件触发，应用可以随时获得最新的表单数据并作持久化。

## 流程与表单驱动的大学服务中心门户

所有表单／工作流在面向用户时使用的是任务驱动模型，即通过任务的发起、推送、办理、跟踪、归档、查询、评价等工作，实现所有的工作流的运行。对于用户而言，只需要掌握任务中心的使用方式，就可以使用所用的工作流程。这是标准化的流程模型和流程开发运行规范给用户带来的重要价值：通过一种使用体验，实现无限的服务内容。标准化的任务中心，几乎可以做到每一个新流程都是可以免除用户的操作培训，最大限度了降低了用户学习使用的成本。

任务中心主要包括5大基础功能：

* 服务大厅
* 待办任务
* 进行中任务
* 已结束任务
* 任务评价

### 6.5.1服务大厅

服务大厅是所有流程的办理入口。

用户在这里可以找到到自己所能办理的所有流程。系统通过流程的权限定义可以自动计算出每个用户可以办理的流程，并推送到他的服务大厅。

服务大厅对所有可办理的流程按照合理的分类进行组织，主要包括：

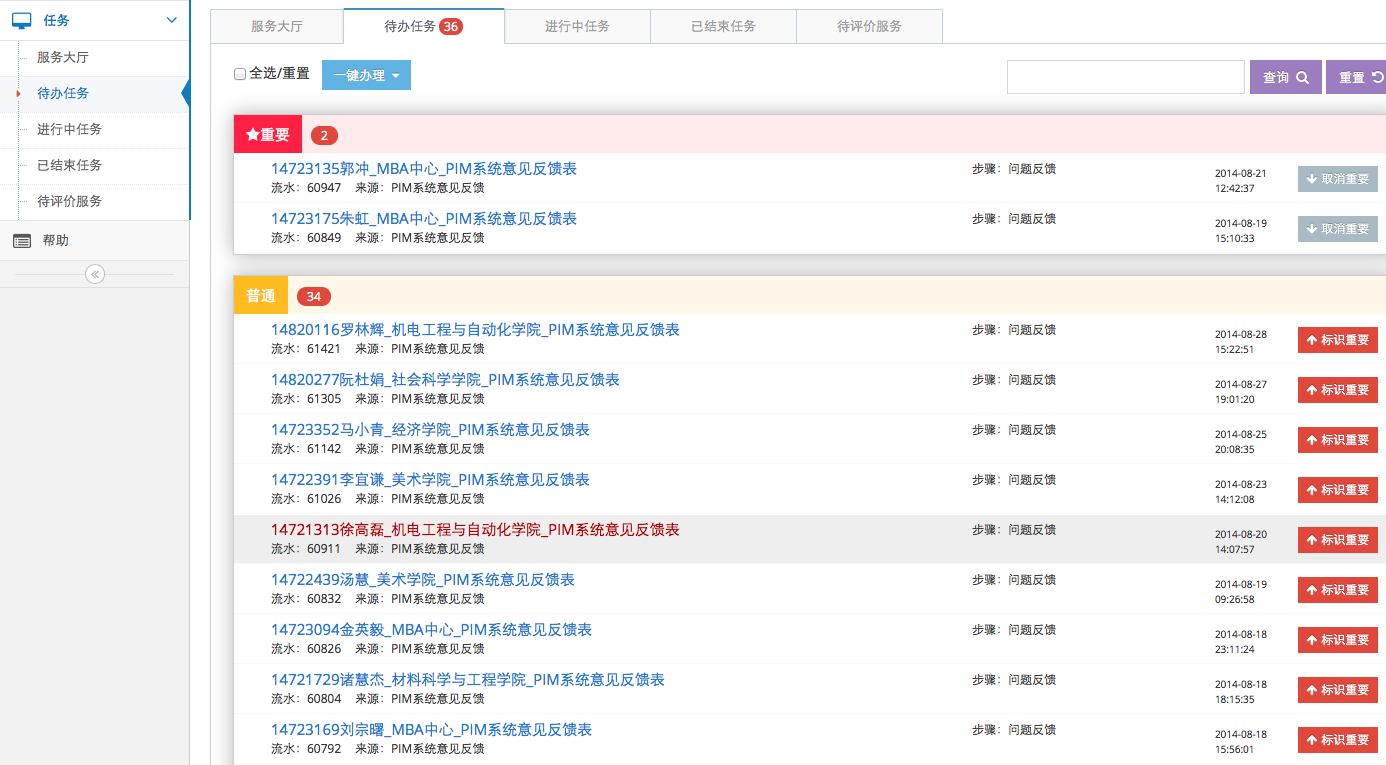
* 学校根据需要（包括时间、业务需求、紧急程度等）推荐给用户的流程
* 用户可以自行定义的自己的常用流程入口
* 按分类进行检索的全部流程



服务大厅的界面可以根据用户的要求进行个性化修改。

### 6.5.2待办任务

每个流程运行过程中，当前的流程状态下所需要参与流程交互的用户会在待办任务的列表中看到需要他处理的流程任务。比如，审批人会在需要其审批的时候在待办任务中看到这个流程，点击流程链接即可看到审批页面。



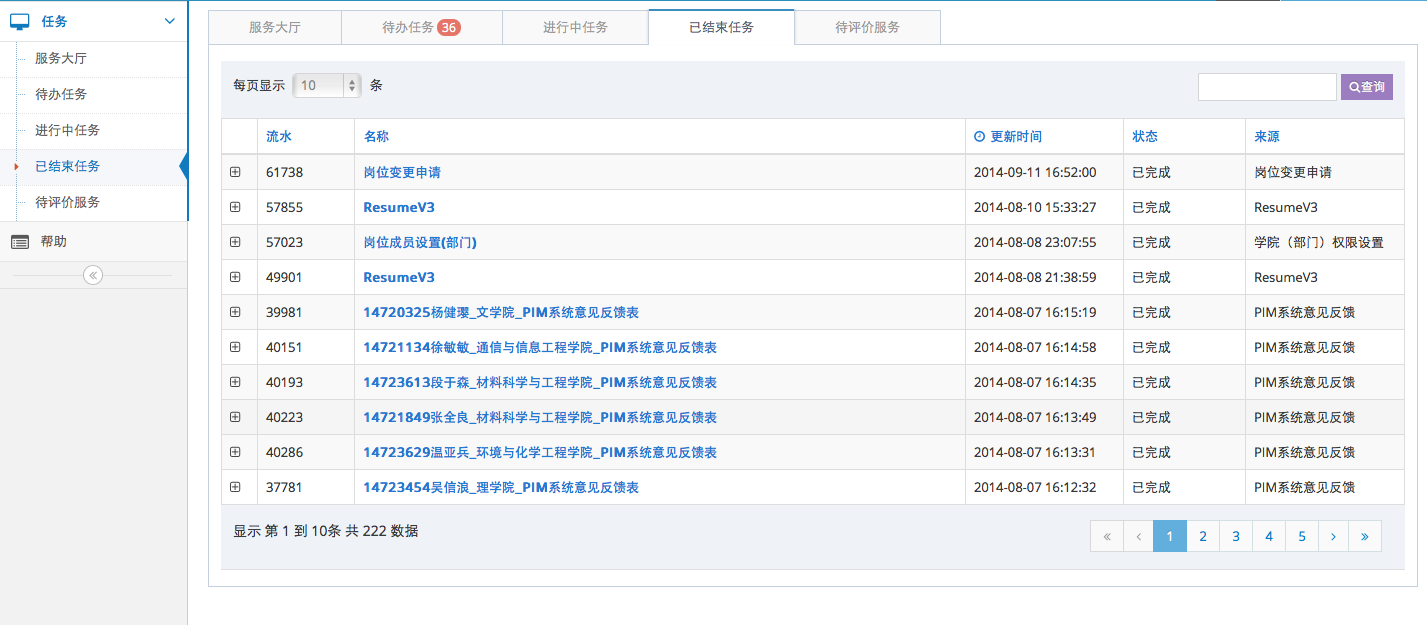
### 6.5.2进行中任务

每个流程运行过程中，流程发起人或者参与人可以对这个流程的进行状态进行跟踪，以提高用户体验。



### 6.5.3已完成任务

已完成的任务会被归档，用户在已完成用户列表中可以查询到所有历史上存在的表单的原始数据。



### 6.5.4待评价任务

系统支持每个使用某个流程的申请人，可以在完成后对这个流程服务做出评价，相关数据会在今后的管理KPI中得到统计。



## 统一的流程监控管理与分析

流程监控管理中心可以提供重要的对于流程的干预功能，以及基于流程运行数据进行的统计分析功能。

流程干预功能主要用于流程的管理员对流程的异常情况或者一些需要特定干预的情况下，对流程／表单进行高权限的强制修改。这是对于流程的长期运营必不可少的功能。系统支持可视化的监控。



流程统计分析主要用于对全校流程运行的整体情况进行统计，以便于未来的流程优化工作。相关功能需要根据用户的需求进行定制。



## 数据管理中心

基于流程产生的数据，本方案还将提供完整的数据管理中心开发框架。数据管理中心开发框架是基于主流的J2EE下SSH开发框架，结合主流的管理信息系统界面框架，实现对各种数据模型的有效管理功能。

### 6.7.1数据模型的自动持久化

Infoplus平台支持在流程任意节点通过一个简单配置项，即可实现流程数据与业务数据（外部数据结构）的映射和自动持久化——无需程序代码介入。

支持外部数据源配置

支持按字段自动生成数据库表

支持流程字段与外部字段的自动匹配／手工绑定

根据配置自动持久化数据，并支持spring的事务和回滚，即，任何情况的异常或messenger取消，可保障引擎库和Binding库同时回滚

### 6.7.2数据管理功能界面的快速开发

基于本方案提供的数据管理中心开发框架，能够快速开发各种数据模型的增、删、改、查、统计报表界面。如下图示意：









## 服务中心的移动支持

1） 提供原生的服务中心移动应用：产品提供现成的一站式大学服务中心的移动终端支持，包括Android／iOS操作系统的原生服务中心移动应用

2） 提供基于HTML的服务中心移动网页版：兼容Android／iOS操作系统的主流移动浏览器。

3） 提供基于微信的服务中心移动应用场景：服务中心实现与微信应用平台的对接，用户可以用微信企业号／服务号做为服务中心的移动访问方式，访问服务中心的各项功能。

## 系统接口设计

接口分为两组，一组提供对工作流定义的维护，用于创建、删除工作流定义和修改工作流定义的状态。另一组提供对工作流实例的维护。所有接口均返回一个Map对象作为返回值，下表列出了该对象的公共属性。

|  |  |  |
| --- | --- | --- |
| 键 | 类型 | 说明 |
| success | 布尔型 | true -> 调用成功，false -> 调用失败 |
| errorMsg | 字符串 | 错误信息或者失败原因，仅在失败时才设置 |

不同的接口方法还可能包含更多的属性，参考各个方法的说明。

1、工作流定义

1)创建工作流

Map<String, Object> createWorkflowDefinition(String guiDefinition)

调用参数:

|  |  |
| --- | --- |
| 参数 | 说明 |
| guiDefinition | 工作流定义的xml表达 |

返回附加属性：

|  |  |  |
| --- | --- | --- |
| 键 | 类型 | 说明 |
| id | 字符串 | 新创建的工作流的唯一ID，该ID用来唯一标识该工作流 |

说明：该接口提供给工作流编辑器保存新创建的工作流。新创建的工作流默认为停用状态。

2)修改工作流

Map<String, Object> updateWorkflowDefinition(String id, String guiDefinition)

调用参数:

|  |  |
| --- | --- |
| 参数 | 说明 |
| id | 工作流ID |
| guiDefinition | 工作流定义的xml表达 |

返回附加属性：

|  |  |  |
| --- | --- | --- |
| 键 | 类型 | 说明 |
| id | 字符串 | 保存后的工作流的唯一ID，该ID用来唯一标识该工作流 |

说明：该接口提供给工作流编辑器保存修改后的工作流。如果被修改的工作流没有产生过实例，或者其拓扑结构未发生改变，返回的id等于传入的id。否则，原工作流将被标识为废弃并停用，新的工作流将被创建并返回其id。新的工作流和原工作流的代码相同。

3)获取工作流数据

Map<String, Object> getWorkflowDefinition(String id)

调用参数:

|  |  |
| --- | --- |
| 参数 | 说明 |
| id | 工作流ID |

返回附加属性：

|  |  |  |
| --- | --- | --- |
| 键 | 类型 | 说明 |
| name | 字符串 | 工作流名称 |
| definition | 字符串 | 工作流定义的xml定义 |
| guiDefinition | 字符串 | 工作流定义图形的xml表达 |
| form | 字符串 | 工作流绑定的表单id |
| obsolete | 布尔型 | 工作流定义是否废弃 |
| code | 字符串 | 工作流定义的代码 |
| priority | 整数 | 工作流定义的预设优先级 |

说明：该方法给工作流编辑器提供了接口来获取已有的工作流数据。

4)删除工作流

Map<String, Object> deleteWorkflowDefinition(String id)

调用参数:

|  |  |
| --- | --- |
| 参数 | 说明 |
| id | 工作流ID |

返回附加属性：无

说明：该方法删除指定的工作流。如果工作流已经产生过实例，删除会失败。

5)启用工作流

Map<String, Object> enableWorkflowDefinition(String id)

调用参数:

|  |  |
| --- | --- |
| 参数 | 说明 |
| id | 工作流ID |

返回附加属性：无

说明：该方法将指定的工作流置于启用状态。只有处于启用状态的工作流才能被发起。启用标记为废弃的工作流会失败。

6)停用工作流

Map<String, Object> disableWorkflowDefinition(String id)

调用参数:

|  |  |
| --- | --- |
| 参数 | 说明 |
| id | 工作流ID |

返回附加属性：无

说明：该方法将指定的工作流置于停用状态。处于停用状态的工作流不能被发起，但是不影响已发起的工作流继续运行。

2、工作流实例

1) 发起工作流

Map<String, Object> initialize(String user, String code)

|  |  |
| --- | --- |
| 参数 | 说明 |
| user | 调用用户 |
| code | 工作流定义的代码 |

返回附加属性：

|  |  |  |
| --- | --- | --- |
| 键 | 类型 | 说明 |
| instance | long | 成功发起的工作流实例ID |

2) 获取指定工作流实例中等待用户处理的Step

Map<String, Object> getAvailableSteps(String user, long instance)

|  |  |
| --- | --- |
| 参数 | 说明 |
| user | 调用用户 |
| instance | 工作流实例ID |

返回附加属性：

|  |  |  |
| --- | --- | --- |
| 键 | 类型 | 说明 |
| result | List | List的每项为一个Map对象，代表一个Step   |  |  |  | | --- | --- | --- | | 键 | 类型 | 说明 | | id | 整数 | Step ID | | name | 字符串 | Step名称 | |

说明：该方法返回指定的工作流实例中，正在等待调用用户处理的Step。对于存在并行流程的工作流，可能有多个Step处于等待状态，一般应该由不同的用户进行处理，但是工作流的设计并不阻止同一个用户对多个Step负责。UI应该将所有等待该用户处理的Step提供给用户，由用户选择先处理哪个。

3)获取用户处理指定的工作流实例的某个Step时的表单字段权限和可以选择的Action

Map<String, Object> getFieldsPermissionAndAvailableActions(String user, long instance, int step)

|  |  |
| --- | --- |
| 参数 | 说明 |
| user | 调用用户 |
| instance | 工作流实例ID |
| step | Step ID |

返回附加属性：

|  |  |  |
| --- | --- | --- |
| 键 | 类型 | 说明 |
| fields | List | List的每项为一个Map对象，代表一个表单字段   |  |  |  | | --- | --- | --- | | 键 | 类型 | 说明 | | id |  | 字段ID | | permission | 字符串 | 权限：r->可读，w->可写 | |
| result | List | List的每项为一个Map对象，代表一个Action   |  |  |  | | --- | --- | --- | | 键 | 类型 | 说明 | | id | 整数 | Action ID | | name | 字符串 | Action名称 | |

说明：该方法返回指定处理工作流实例的某个Step时，调用用户对表单字段的权限和可以选择的Action。

4) 查询用户以指定action处理工作流时的下一步Step和相应权限

Map<String, Object> getNextStepsAndRoles(String user, long id, int action)

|  |  |
| --- | --- |
| 参数 | 说明 |
| user | 调用用户 |
| instance | 工作流实例ID |
| action | Action ID |

返回附加属性：

|  |  |  |
| --- | --- | --- |
| 键 | 类型 | 说明 |
| steps | List | List的每项为一个Map对象，代表一个Step   |  |  |  | | --- | --- | --- | | 键 | 类型 | 说明 | | id | 整数 | Step ID | | name | 字符串 | Step Name | | roles | 字符串数组 | 可操作该Step的角色列表 | |

说明：当用户选择了某个action对工作流进行处理时，调用该方法来确定哪些Step将新增为当前Step，以及每个Step对应的操作角色。在稍后调用doAction进行工作流实际处理时，需要为每个Step提供一个具体的用户。如果该方法返回一个空的List，则意味着调用doAction后工作流将结束。

5)为用户处理指定的工作流实例

Map<String, Object> doAction(String user, long instance, int action, Map<Integer, String>nextUsers, Map<String, Object> properties)

|  |  |
| --- | --- |
| 参数 | 说明 |
| user | 调用用户 |
| instance | 工作流实例ID |
| action | Action ID |
| nextUsers | 完成action后负责处理下一步Step的用户，因为可能存在多个下一步，该参数为一个Map，键为下一步Step的ID，值为负责的用户ID |
| properties | 其它附加数据，如表单数据 |

返回附加属性：无

6) 挂起指定的工作流实例

Map<String, Object> suspendWorkflowInstance(String user, long instance, String reason)

|  |  |
| --- | --- |
| 参数 | 说明 |
| user | 调用用户 |
| instance | 工作流实例ID |
| reason | 挂起原因 |

返回附加属性：无

说明：该方法将指定的工作流实例挂起，挂起的工作流实例将停止工作直至其被恢复。

7) 恢复指定的工作流实例

Map<String, Object> resumeWorkflowInstance(String user, long instance)

|  |  |
| --- | --- |
| 参数 | 说明 |
| user | 调用用户 |
| instance | 工作流实例ID |

返回附加属性：无

说明：该方法将恢复之前被挂起的工作流实例。

8)中止指定的工作流实例

Map<String, Object> killWorkflowInstance(String user, long instance)

|  |  |
| --- | --- |
| 参数 | 说明 |
| user | 调用用户 |
| instance | 工作流实例ID |

返回附加属性：无

说明：该方法将中止一个工作中的工作流实例，该操作不可逆。

## 流程开发规范

本节描述了在目前的框架下，对开发过程中的一些标准定义和规范化要求。随着系统的不断升级这些规范也会不断完善。

### 6.8.1表单开发

控件种类与数据类型

目前支持的数据类型： string anyURI XHTML integer double boolean date User Organize Code

|  |  |
| --- | --- |
| **数据类型** | **说明** |
| string | 字符 |
| anyURI | URI |
| XHTML | 大文本 |
| integer | 整型 |
| double | 浮点型 |
| boolean | 布尔型 |
| date | 日期，Unix时间戳,记录了1970年1月1日（00:00:00 GMT）以来的秒数 |
| User | jaccount用户 |
| Organize | 组织机构 |
| Code | 代码表 |

**条件样式与条件公式**

**条件样式:**

**规则**为{style:CSS,condition:BOOL-EXPRESSION},其**含义**是当condition满足时候，该字段用CSS里的样式显示

CSS可以有多个,以;分割，设置一个style时候可以写成styleName:styleValue形式，如果要去掉一个以前动态设置的style可以写成styleName:null或者styleName:的形式，BOOL-EXPRESSION是一个布尔表达式，可以用内置函数，可以用表单字段，表单字段以$fieldName表示，布尔表达式不写代表总是true

例如：希望文本框(fieldEdit)大于等于80时候另一个标签控件(fieldLabel)显示绿色字体，小于60时候显示红色字体，其余时候显示控件原来的颜色，那么需要在fieldLabel上定义3条style(在render type编辑界面上):

{style:"color:red",condition:"$fieldEdit<60"}

{style:"color:null",condition:"$fieldEdit>=60 && $fieldEdit<80"}

{style:"color:green",condition:"$fieldEdit>=80"}

**条件公式:**

**规则**为{formula:EXPRESSION,condition:BOOL-EXPRESSION,type:TYPE},其中TYPE目前有"change"和"init"两种，init类型的公式只用于重复节和重复表中的字段新增初始化时候使用

TYPE=="change" 公式的**含义**是公式或者条件中任何涉及到的字段发生变化时，如果condition满足，该字段用EXPRESSION计算出来的值显示

TYPE=="init" 公式的**含义**是重复表或者节中新增一行或者一段时，如果condition满足，该字段用EXPRESSION计算出来的值作为初值

例1：希望选择课程(表单字段名为fieldCourse，类型为Code,CodeTable名为Course)的时候，带出教师(fieldTeacher)信息到表单上显示，需要在fieldTeacher上定义一条公式

{formula:"$fieldCourse.teacher",condition:"",type:"change"}

其中$fieldCourse.teacher代表当前所选fieldCourse课程的属性信息中的teacher字段，condition不写代表不用判断条件

例2：希望对重复表的某个数量字段(fieldCount)求和，把求和结果显示在重复表外面的合计字段(fieldSum),需要在fieldSum上定义一条公式

{formula:"sum($fieldCount)",condition:"",type:"change"}

其中sum是内置的求和函数

例3：希望重复表中的字段fieldCountry新增时候自动带出上一行的country值，需要在fieldCountry上定义一条公式

{formula:"$fieldCountry[$rowIndex-1]",condition:"",type:"init"}

其中$rowIndex代表了重复表中当前操作的行号，新增时候就是增出来的这行行号，$rowIndex是context里面一个特殊的变量，只在init公式里会出现

**表达式解析**

1. 支持的数学运算符包括 + - \* / %
2. 支持的逻辑运算符包括 && || !
3. 支持的比较运算符 > >= < <= == !=
4. 支持的常量包括 字符串(以单引号括起来)，数字(由0-9.组成)，布尔值(true和false)
5. 支持的求变量的属性，数组值以及变量中的函数值，其中数组下标可以用表达式计算，例如a={c:1,d:[1,2,3],e:function(){return true;}} 则$a.c==1 ， $a.d[$c]==2 ， $a.e()返回true
6. 支持的内置函数包括如下(有待扩展)
   1. sum(value) 对value数组进行求和
   2. avg(value) 对value数组进行求平均
   3. count(value) 对value数组求数量
   4. toFixed(number,fix) 对number的第fix位进行四舍五入
   5. getBirthday(id) 根据身份证号取生日
   6. dateSpan(dateFrom, dateTo) 计算dateFrom和dateTo之间的天数
   7. concat(value, gapString, property) 拼接value，以gapString为间隔，如果value不是string而是一个对象，那么可以用property指定拼接对象的某个属性

**其它注意点:**

1. 只有支持onFieldRefer的控件才能设置条件样式和条件公式
2. 条件公式支持联动，例如fieldA的变动引起fieldB的变动，而fieldB的变动又引起fieldC的变动，则在表单上改变A时会同时引起B和C的变动
3. 条件公式的联动可以有环路，例如设置A->B B->C C->A(箭头代表前者引起后者的变动),则改变B会引起C的变化，C的变化引起A的变化，但A的变化不会改变B，即不会循环变动
4. 对于Code、User、Organize数据类型的字段(fieldA)
   1. 如果别的字段(fieldB)想根据该字段的值变化，则需要设置fieldB的公式为{formula:"$fieldA.value",condition:""}
   2. 如果fieldB想根据fieldA的显示的值变化，则需要设置fieldB的公式为{formula:"$fieldA.displayValue",condition:""}
   3. 如果fieldB也同样为Code、User、Organize并且想跟随fieldA变动，则设置fieldB公式为{formula:"$fieldA",condition:""}
   4. 如果fieldA具有属性attrC，fieldB想根据该属性变化，则设置fieldB公式为{formula:"$fieldA.attrC",condition:""}
   5. 如果fieldB也同样为Code、User、Organize且想根据fieldA的属性（例如teacher）变化，那么引起其变化的属性值必须有相应的\_name字段支持，即必须fieldA同时具有属性teacher,teacher\_name,设置fieldB公式{formula:"$fieldA.teacher",condition:""}，则fieldB的显示值和实际值都根据fieldA的teacher变化，如果fieldA只有teacher属性，但是没有teacher\_name属性，那么在上例中fieldB只会改变其值，但是显示值会为空

**表单控件设置注意点**

1. Remark 将大文本框长宽都设置为16
2. Suggester 如果数据类型是Code,需要将代码表选择好
   1. 如果要让suggester控件支持onFieldChange事件，将字段的event设置为1024
   2. 如果要让suggester控件支持联动，可以设置字段的parent为另一个字段
3. Anchor 在infopath里选择文本控件，在字段初值上设置连接地址；在settings里的displayText设置链接显示文字；在settings的windowMode设置显示模式，可选择六种模式（1 本窗口模式 2 本窗口非模式 3 新窗口对话框 4 新窗口模式 5 新窗口非模式 6 本窗口打开）；在settings的windowHeight设置窗口高度，在windowWidth设置窗口宽度。
4. Hidden 可以调整其大小为0
5. 如果要让控件支持onLocalFieldChange事件，要将字段的map填写好，格式是XXX=fieldxxx,YYY=fieldyyy（其中XXX是代码表项扩展信息的名字(也可以是self或者selfDisplayValue)，fieldxxx是字段名字，多个字段用逗号分隔）
6. Select 只支持Embedded类型的代码表

**表单字段设置注意点**

**infopath表单域字段命名规范**

1. 字段名称命名必须以字母开头，且可由字母和数字组成
2. 所有域字段名称以field开头，驼峰格式
3. 所有重复字段的组以group开头，驼峰格式

**infopath表单设计注意事项**

1. repeatTable 表头必须有且只能有一行
2. repeatTable 每行上的弹出菜单有时和重复表的右端稍离开一些，可以稍调整一下repeatTable的宽度
3. repeatTable的新增按钮没有和重复表的左端对齐，可以将重复表放到一个表格中
4. 表格 行高的调整需要调表格padding，否则firefox,chrome行高度不会增高
5. 表格千万不要有太多行，可以拆成多个表格，分开画
6. 如果希望表格行是左或者右对齐的，请选择对齐方式，否则ie下都会居中，虽然infopath里看上去是左对齐的
7. 文字不要直接从word复制过来，可以先复制到notepad，再copy
8. 重复节的高度设为自动(清空高度)
9. 文件控件请设置宽度为固定像素值，百分比宽度不支持
10. view的名字决定了view显示的次序，对于打印，每个view前自动分页
11. 如果需要在repeatSection中支持分页(即每个section前自动分页)，该repeatSection必须是view的顶级元素，即必须删除infoplus自动创建的最外层table，然后直接插入repeatSection

### 6.8.2数据代码表开发

* 代码表本身，是三层结构的（一对多）：CodeTable => Code => Attribute
* 属性是大小写敏感的。
* 属性的引用，可以用${foo.bar}的形式，如果是纯属性的使用，可以省略${}
* 所有代码(code)，均有两个默认属性：

用户(User)代码表属性：

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Type** | **Description** |
| userCode | string | 该用户的学号或工号 |
| organizeCode | string | 该用户所属的组织机构的代码 |
| organizeName | string | 该用户所属的组织机构的名称 |

1. 组织机构(Organize)属性：

|  |  |  |
| --- | --- | --- |
| **Attribute Name** | **Type** | **Description** |
| topOrganizeCode | string | 该组织机构所属的顶级组织机构的代码 |
| topOrganizeName | string | 该组织机构所属的顶级组织机构的名称 |

### 6.8.3事件驱动开发

共16个可监听的事件：

* 7个发生在实例生命期
* 7个发生在步骤生命期
* 2个发生在界面交互过程中

|  |  |  |
| --- | --- | --- |
| **EventType** | **Type** | **典型用例** |
| INSTANCE\_STARTING | 实例级 | 检查是否可发起;赋予初始化数据 |
| INSTANCE\_STARTED | 实例级 | 持久化流程流水号等信息;关联第三方数据实体 |
| INSTANCE\_COMPLETING | 实例级 | 此事件无效，原理上InfoPlus无法准确取得，请勿使用 |
| INSTANCE\_COMPLETED | 实例级 | 持久化表单数据 |
| INSTANCE\_KILLED | 实例级 | 通知用户;处理第三方相关的业务逻辑 |
| STEP\_EXPIRING | 实例级 | 干预超时行为：延期/忽略/终止 |
| STEP\_EXPIRED | 实例级 | 通知用户;处理第三方相关的业务逻辑 |
| STEP\_RENDERING | 步骤级 | 修改表单界面数据;追加代码表内容配合Select使用 |
| STEP\_RENDERED | 步骤级 | 暂不支持 |
| ACTION\_CLICKING | 步骤级 | 验证表单数据 注：比ACTION\_DOING要早，但不能保障验证后不被用户恶意修改 |
| ACTION\_SAVING | 步骤级 | 验证表单数据给出错误提示 |
| ACTION\_SAVED | 步骤级 | 无 |
| ACTION\_DOING | 步骤级 | 验证表单数据给出错误提示;更新表单数据 |
| ACTION\_DONE | 步骤级 | 处理第三方相关的业务逻辑 |
| FIELD\_CHANGING | 字段级 | 引起界面其他字段联动。 注：需配合字段的Event掩码配置一起使用 |
| FIELD\_SUGGESTING | 字段级 | 引起界面其他字段联动。 注：需配合字段的Event掩码配置一起使用 |
| ECHO | 系统级 | 无 |

# 安全体系设计

## 安全体系设计原则

安全体系建设是一个系统工程，网络系统安全体系建设应按照“统一规划、统筹安排，统一标准、相互配套”的原则进行，采用先进的“平台化”建设思想，避免重复投入、重复建设，充分考虑整体和局部的利益，坚持近期目标与长期数字化校园建设目标相结合。在进行网络系统安全方案设计、规划时，应遵循以下原则：

·需求、风险、代价平衡的原则

·综合性、整体性原则

·一致性原则

·易操作性原则

·适应性、灵活性原则

·多重保护原则

·可评价性原则

## 安全体系解决方案

### 7.2.1物理层安全技术方案

1、环境安全

对系统所在环境的安全保护，如区域保护和灾难保护；（参见国家标准GB50173－93《电子计算机机房设计规范》、国标GB2887－89《计算站场地技术条件》、GB9361－88《计算站场地安全要求》。

2、设备安全

设备安全主要包括设备的防盗、防毁、防电磁信息辐射泄漏、防止线路截获、抗电磁干扰及电源保护等；设备冗余备份；通过严格管理及提高员工的整体安全意识来实现。

3、媒体安全

包括媒体数据的安全及媒体本身的安全。显然，为保证信息网络系统的物理安全，除在网络规划和场地、环境等要求之外，还要防止系统信息在空间的扩散。计算机系统通过电磁辐射使信息被截获而失密的案例已经很多，在理论和技术支持下的验证工作也证实这种截取距离在几百甚至可达千米的复原显示技术给计算机系统信息的保密工作带来了极大的危害。为了防止系统中的信息在空间上的扩散，通常是在物理上采取一定的防护措施，来减少或干扰扩散出去的空间信号。

### 7.2.2网络层安全技术方案

1、防火墙安全技术建议

在网络的边界都应安装防火墙，并需要实施相应的安全策略控制。另外，根据对外提供信息查询等服务的要求，为了控制对关键服务器的授权访问控制，建议把对外公开服务器集合起来划分为一个专门的服务器子网，设置防火墙策略来保护对它们的访问。

2、数据传输安全建议

为保证数据传输的机密性和完整性，同时对拨号用户接入采用强身份认证，建议在网络中采用安全VP系统，对于移动用户安装VPN客户端软件。

VPN应具有以下功能：

1）信息透明加解密功能：支持网络IP数据包的机密性保护。网络密码机串联在以太网中，凡是流经的IP报文无一例外地都要受到它的分析和检查，根据需要进行加解密和认证处理。信息加解密功能支持商务系统专有的业务服务，及WWW、FTP、SMTP、Telnet等基于TCP/IP的服务。

2）信息认证功能：支持IP 数据包的完整性保护。流过的IP 报文在被加解密的同时，还要进行认证处理，由加密方在每个报文之后都自动附有认证码，其他人无法伪造，在接收方对该认证码进行验证，保证了信息的完整性和不可篡改性。

3）防火墙功能：支持网络访问控制机制，防止外部非法用户攻击。在操作系统底层直接实现报文包过滤技术、IP 地址伪装技术（NAT），并与信息加密、认证机制无缝结合。可以保证局域网的边界安全，防止了通常的类似于IP 地址欺骗的攻击。

4）远程分布式集中统一管理功能

5）安全审计及告警功能：支持对网络非法访问操作的审计和自动告警。VPN网关对流过的报文进行动态过滤分析，根据网络安全审计策略，自动调度审计进程，进行审计记录，产生审计报告，并以多种方式，如语音、电子邮件等方式对非法事件进行实时告警，以便安全管理员在第一时间了解情况，做出正确的应对措施，以尽可能的将非法事件造成的损失降低至最小。

### 7.2.3系统层安全技术方案

1、操作系统安全技术

操作系统是所有计算机终端、工作站和服务器等正常运行的基础，操作系统的安全十分重要。目前的商用操作系统主要有IBM AIX、Linux、AS/400、OS/390、SUN Solaris、HP Unix、Windows 98/XP、Windows 2000/2003等。针对操作系统应用环境对安全要求的不同，电子商务系统对操作系统的不同适用范围作如下要求：关键的服务器和工作站（如数据库服务器、WWW 服务器、代理服务器、电子邮件服务器、病毒服务器、DHCP主域服务器、备份服务器和网管工作站）应该采用服务器版本的操作系统。典型的有：SUN Solaris、HP Unix、Linux、Windows 2000 Server。

2、操作系统安全管理

操作系统因为设计和版本的问题，存在许多的安全漏洞；同时因为在使用中安全设置不当，也会增加安全漏洞，带来安全隐患。在没有其它更高安全级别的商用操作系统可供选择的情况下，安全关键在于操作系统的安全管理。

为了加强操作系统的安全管理，要从物理安全、登录安全、用户安全、文件系统和打印机安全、注册表安全、RAS安全、数据安全、各应用系统安全等方面制定强化安全的措施。

3、数据库安全技术

目前的商用数据库管理系统主要有MS SQL Sever、Oracle、Sybase、DB2 等。数据库管理系统应具有如下能力：

1）自主访问控制（DAC）：DAC 用来决定用户是否有权访问数据库对象；

2）验证：保证只有授权的合法用户才能注册和访问；

3）授权：对不同的用户访问数据库授予不同的权限；

4）审计：监视各用户对数据库施加的动作。

### 7.2.4应用层安全技术方案

根据网络的业务和服务，我们采用加密技术、防病毒技术、以及对各种应用服务的安全性增强配置服务来保障网络系统在应用层的安全。

1、防病毒技术

目前主要采用病毒防范系统解决病毒查找、清杀问题。根据网络结构和计算机分布情况，病毒防范系统的安装实施要求为：能够配置成分布式运行和集中管理，由防病毒代理和防病毒服务器端组成。防病毒客户端安装在系统的关键主机中，如关键服务器、工作站和网管终端。在防病毒服务器端能够交互式地操作防病毒客户端进行病毒扫描和清杀，设定病毒防范策略。能够从多层次进行病毒防范，第一层工作站、第二层服务器、第三层网关都能有相应的防毒软件提供完整的、全面的防病毒保护。以下是全方位防毒系统所需提供的应对策略：

1）来自系统外部（互联网或外网）的病毒入侵

这是目前病毒进入最多的途径。因此在与外部连接的网关处进行病毒拦截是效率最高，耗费资源最少的措施。可以使进入内部系统的病毒数量大为减少。但很明显，它只能阻挡来自外部病毒的入侵。

2）病毒集散地之一，网络邮件/群件系统

如果网络内采用了自己的邮件/群件系统实施办公和信息自动化，那么一旦有某个用户感染了病毒，通过邮件方式该病毒将以几何级数在网络内迅速传播，并且很快会导致邮件系统负荷过大而瘫痪。因此在邮件系统上部署防病毒也显得尤为重要。

3）病毒集散地之二，文件服务器

文件资源共享是网络提供的基本功能。文件服务器大大提高了资源的重复利用率，并且能对信息进行长期有效的存储和保护。但是一旦服务器本身感染了病毒，就会对所有的访问者构成威胁。因此文件服务器也需要设置防病毒保护。

4）最终用户

病毒最后的入侵途径就是最终的桌面用户。由于网络共享的便利性，某个感染病毒的桌面机可能随时会感染其它的机器，或是被种上了黑客程序而向外传送机密文件。因此在网络内对所有的客户机进行防毒控制也很有必要。

5）内容保护

随着病毒所采用的技术日趋复杂，单纯依靠病毒码和被动的文件分析技术往往造成防病毒的响应时间过长。为了能够在第一时间主动的阻止新型病毒的入侵，在防病毒系统中附加内容过滤和保护功能就显得十分重要。例如，由于目前邮件系统的使用异常方便，造成了用户很容易在不经意间将重要的、机密的或是不当的信息通过邮件发送出去；另一方面，来自互联网上的垃圾邮件也到处都是，导致用户需花大量的精力和时间去处理，降低了工作效率。因此对往来的邮件内容进行过滤也很重要。

6）集中管理

一个缺少管理的系统就是一个无效的系统。对于一个大型网络来说，部署的防毒系统将十分复杂和庞大。尤其在各网点在地域上分离的情况下，通过一个监控中心对整个系统内的防毒服务和情况进行管理和维护显得十分重要。这样可以大大降低维护人员的数量和维护成本，并且缩短了升级、维护系统的响应时间。

防毒系统的最大特点是需要不断的升级和更新防毒软件，以应对新产生的各类病毒。因此确保各点的防毒软件集中进行部署、升级和监控也是有效防毒的重要一环。

2、服务器安全性增强

服务器安全性增强的内容包括：Web服务器自身安全、Web服务器参数配置、Web服务器权限配置、Web服务器配置安全、网络信息加密配置、Web应用代码审计。

## 数据备份方案

实施过程中分应用服务器和备份服务器，应用服务器提供应用访问服务。备份服务器为当应用服务器在不用状态下可以很快的启动备份服务器并提供服务，或者当应用服务器上的数据被破坏后可以通过备份服务器上的数据进行恢复。

备份的方法如下

1、软件提供自动定期增量备份的策略；

2、提供系统手工完全备份的方法。

其他可供选择的备份方式有：独立数据存储，磁带机，光盘刻录等。

## 安全管理方案建议

1、安全体系建设规范

本期项目涉及的业务系统建设整网安全需要一套统一的安全体系建设规范，此规范应结合系统应用的实际情况制定。

2、安全组织体系建设

实施安全应管理先行，安全组织体系的建设势在必行。应由一个主管领导，网络管理员，安全操作员等人员组成安全工作小组。主管领导应领导安全体系的建设实施，在安全实施过程中取得相关部门的配合。领导整个部门不断提高系统的安全等级。网络管理员应具有丰富的网络知识和实际经验，熟悉本地网络结构，能够制定技术实施策略。安全操作员负责安全系统的具体实施。

3、安全管理制度建设

其具体工作包括：确定该系统的安全等级；根据确定的安全等级，确定安全管理的范围；制定安全管理制度；

完整的安全管理制度必须包括以下几个方面：

1）人员安全管理制度

2）操作安全管理制度

3）场地与设施安全管理制度

4）设备安全使用管理制度

5）操作系统和数据库安全管理制度

6）运行日志安全管理

7）备份安全管理

8）异常情况管理

9）系统安全恢复管理

10）安全软件版本管理制度

11）技术文档安全管理制度

12）应急管理制度

13）审计管理制度

14）运行维护安全规定

15）对系统安全状况的定期评估策略

16）技术文档媒体报废管理制度

4、安全管理手段

安全技术管理体系是实施安全管理制度的技术手段，是安全管理智能化、程序化、自动化的技术保障。安全技术管理对OSI 的各层进行综合技术管理。网络安全管理，主要对网络安全体系的防火墙、入侵检测系统等网络安全设备进行管理；应用安全管理，主要对网络安全体系的应用安全系统进行管理，如用户认证系统的管理、病毒防范系统的管理。

# 方案可行性及案例验证

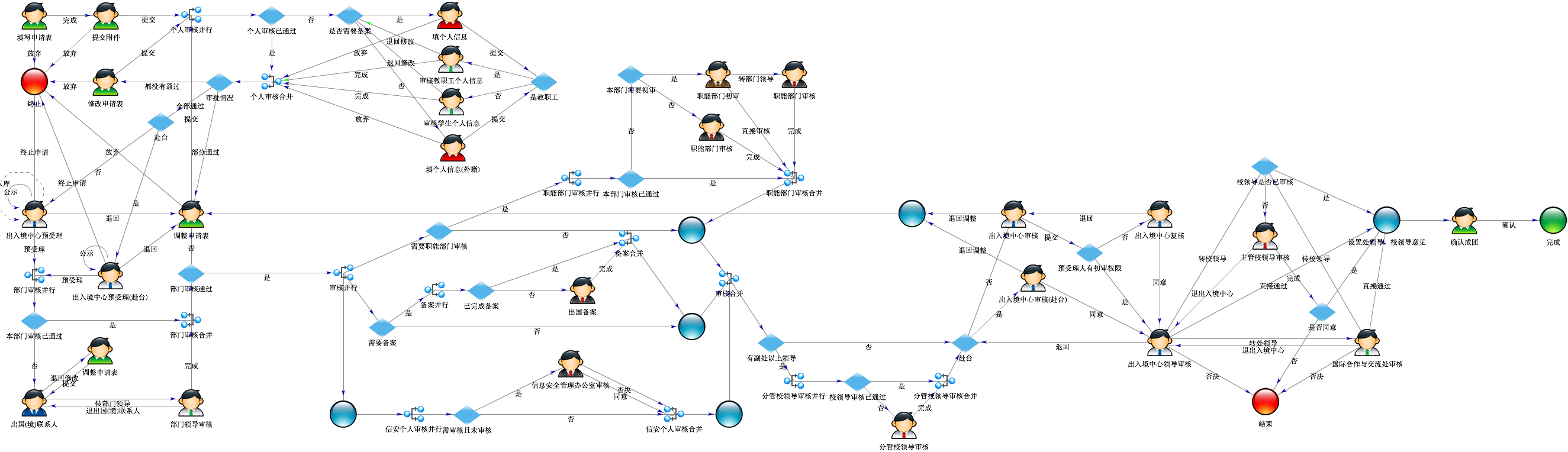
本项目方案在包括上海交通大学、北京大学、哈尔滨工程大学、上海大学、西安交通大学、华中师范大学、华中农业大学、江南大学、长安大学等众多985/211重点高校中得到了长期验证，开发了大量真实用于实际应用场景的流程服务。

此外，在前期在中山大学实际运行环境和用户需求中，进行了完整的部署、测试、开发、应用和人员培训验证，涵盖了包括桌面、移动终端的应用体验设计，业务流程开发验证等等，取得了确定的验证效果，充分证明了方案的可行性。

部分案例举例如下：

## 上海交通大学

到目前位置，已经有70余项流程在Infoplus的环境中开发和运行，很多流程已经经历了2－3年的长期考验与不断改进。典型的案例是学校的出入境审核流程，经过2年的不断运行、改进，流程已经覆盖了几乎所有的复杂运行场景，如下图示意：





该流程每年处理数千个初入境团队的申请与处理过程，经历了高强度和复杂处理过程的考验。充分验证了InfoPlus的灵活表达能力、高可靠性、高处理能力。Infoplus在上海交通大学的应用还在持续扩展中。

## 上海大学的应用案例

上海大学是上海学生人数最多、规模最大的211大学。

上海大学制定了完整的基于流程化管理理念的新一轮信息化建设规划。科探的Infoplus平台在竞争中脱颖而出，被上海大学采纳为流程基础设施，构建上海大学的流程与智能管理系统（PIM系统）。

上海大学自2014年5月采用了Infoplus平台，由于平台的低学习曲线、低开发成本、快速开发迭代能力，已经在2个多月中开发了近40个流程。其中6个流程正式上线。

正式上线的流程中包括了2014年本科生、研究生迎新、暑期住宿申请等涉及范围很广，使用强度很大的流程。经过努力，Infoplus配置的迎新流程经历的上万学生的考验，并且支持了诸如电话抢号等高强度应用的考验。进一步验证了Infoplus的能力。

此外，在这个项目中充分体现Infoplus在开发效率、学习曲线方面的优势。上海大学信息办5位老师，经过短暂培训后已经能够独立、高效地承担流程开发工作。科探提供了技术支持服务。2个月40个流程开发的效率是明显高于传统代码开发方式的，并且得到了“流程建设中开发已经不再是负担，而业务梳理业务推进工作成为工作重点”，真正体现了信息化的优势。 

截至2014年11月底已经累计开发了63个不同的网上流转业务流程，包括：

|  |
| --- |
| 上海大学\_ACM大赛报名 |
| 上海大学\_上海大学本科生迎新 |
| 上海大学\_上海大学毕业生离校工作流程 |
| 上海大学\_上海大学研究生迎新 |
| 人事业务\_教职工工作签到 |
| 人事处\_上海大学教职工请(销)假申请 |
| 信息办\_PIM系统岗位权限设置 |
| 信息办\_PIM项目组内部工作协作表 |
| 信息办\_学院(部门)权限设置 |
| 信息办\_岗位变更申请 |
| 信息服务\_PIM系统意见反馈 |
| 信息服务\_VPN开通申请 |
| 信息服务\_信息化工程建设申请 |
| 信息服务\_信息化系统建设申请 |
| 信息服务\_数据库资环申请 |
| 信息服务\_联系方式登记 |
| 信息服务\_虚拟服务器申请 |
| 信息服务\_部门邮箱申请 |
| 外事业务\_上海大学教职工因公出访计划审批 |
| 外事业务\_上海大学教职工因公出访计划申请 |
| 学生事务中心\_上海大学"绿色通道"申请 |
| 学生事务中心\_上海大学团组织关系转接登记 |
| 学生事务中心\_上海大学学生家庭经济情况调查 |
| 学生事务中心\_上海大学学生社团注册登记 |
| 学生事务中心\_上海大学设摊及张贴宣传等校园活动审批 |
| 学生事务中心\_学生缓考申请 |
| 学生事务中心\_社团更名申请 |
| 学生事务中心\_社团更改挂靠单位申请 |
| 学生事务中心\_社团更改类别申请 |
| 学生事务中心\_社团证补办申请 |
| 学生事务中心\_社团负责人换届 |
| 学生事务中心\_音乐学院场馆使用申请 |
| 学生事务中心\_音乐学院服装借用申请 |
| 学生事务中心\_音乐学院设备借用申请 |
| 校办\_上海大学主页重要活动通知发布申请 |
| 校办\_上海大学伟长楼国际会议中心场地租地申请 |
| 校办\_上海大学合同(协议)签订审批 |
| 校办\_上海大学外收文处理 |
| 校办\_上海大学校内发文申请 |
| 校办\_上海大学校内请示 |
| 校办\_上海大学用印申请 |
| 校办\_上海大学邀请校行政领导出席会议活动申请 |
| 校医院\_新生病史采集 |
| 社区\_学生假期住宿申请 |
| 社区\_学生辅导员公寓住宿申请 |
| 社区\_跨校区住宿申请 |
| 采购机会 |
| 温哥华学院\_上海温哥华学院入学 |
| 钱学伟学院招生申请表 |
| 钱学伟学院招生专家评分 |
| 房产处\_上海大学教师公寓租用申请 |
| 房产处\_新职工支取住房补贴款申请 |
| 武保处\_上海大学学生在住证明申请 |
| 武保处\_居住证办理登记审批 |
| 学生事务中心\_上海大学毕业生补报生源申请 |
| 学生事务中心\_上海大学职业咨询预约登记 |
| 学生事务中心\_意向外省市就业毕业生登记 |
| 校办\_上海大学校内设摊及张贴宣传活动申请 |
| 社区\_上海大学学生提前报到住宿申请 |
| 社区\_外单位研究生住宿申请 |
| 社区\_上海大学学生住宿申请 |
| 组织部\_干部任免审批 |

该案例充分证明了本方案的有效性以及科探在流程开发领域的实施能力。

## 哈尔滨工程大学

自2014年9月起，采用与本项目相同的技术方案和流程开发方式，在哈尔滨工程大学实施了多个数字校园管理业务流程。包括：

学生住宿申请(ZhuSuShengQing)

信息化处事项处理(XXHCSXCL)

信息化处办公用品申领表(BanGongYongPinShenLin)

信息化处发文处理(XXHCFaWenChuLi)

信息化处收文处理(XXHCShouWenChuLi)

助学贷款申请(ZhuXueDaiKuanShengQing)

助学金申请(ZhuXueJinShengQing)

勤工助学岗位申请(QinGongZhuXueShengQing)

奖学金申请(JiangXueJinShengQing)

应聘申请(ZhaoPinShenQing)

教职工年度考核申请(NianDuKaoHe)

数据上报(ShuJuCaiJi)

学生调宿申请(TiaoSuShengQing)

贫困生申请(PingKunShenShengQing)

学生退宿申请(TuiSuShengQing)