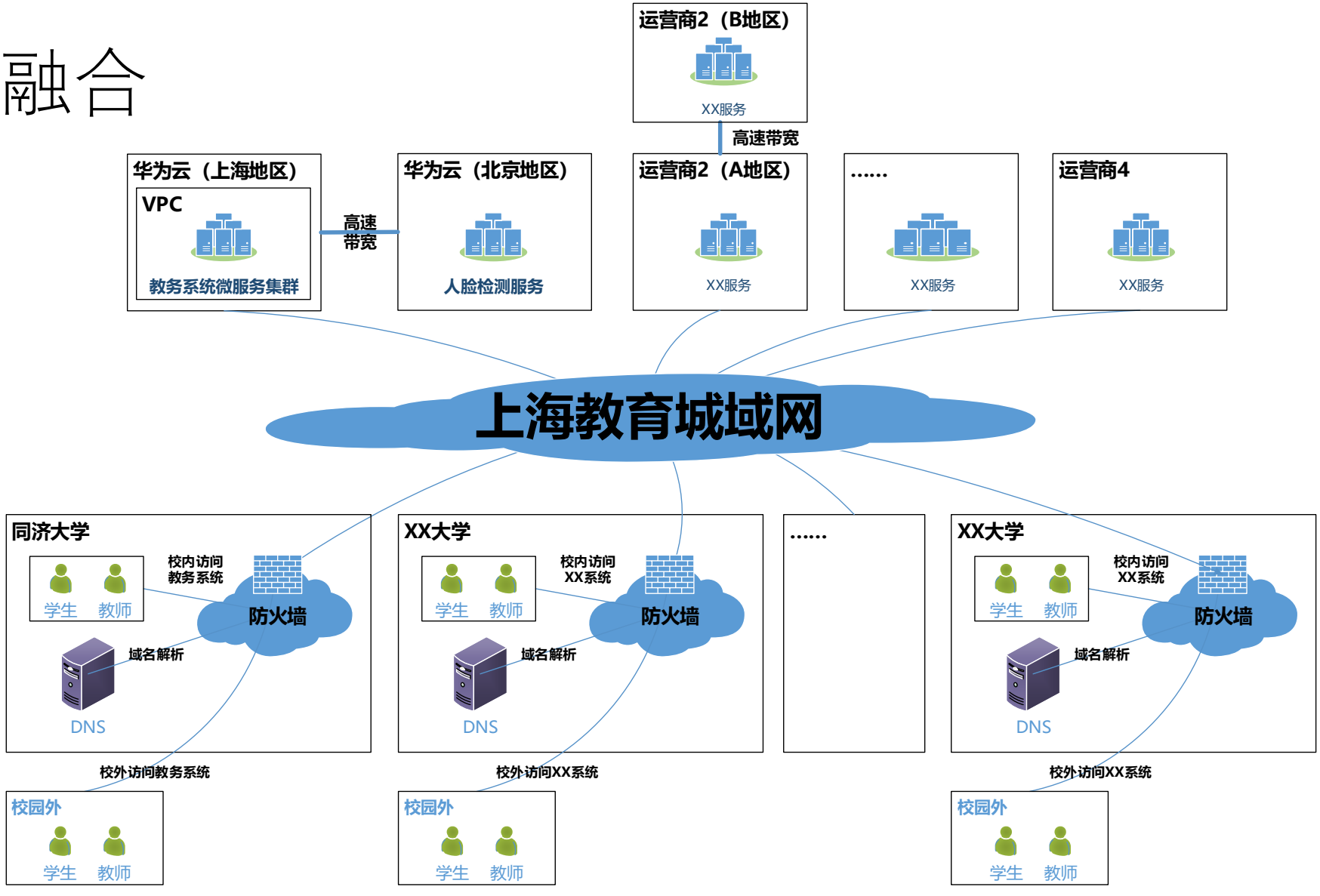
TONGJI UNIVERSITY

基于云网融合的云原生
本研一体化教务信息系统

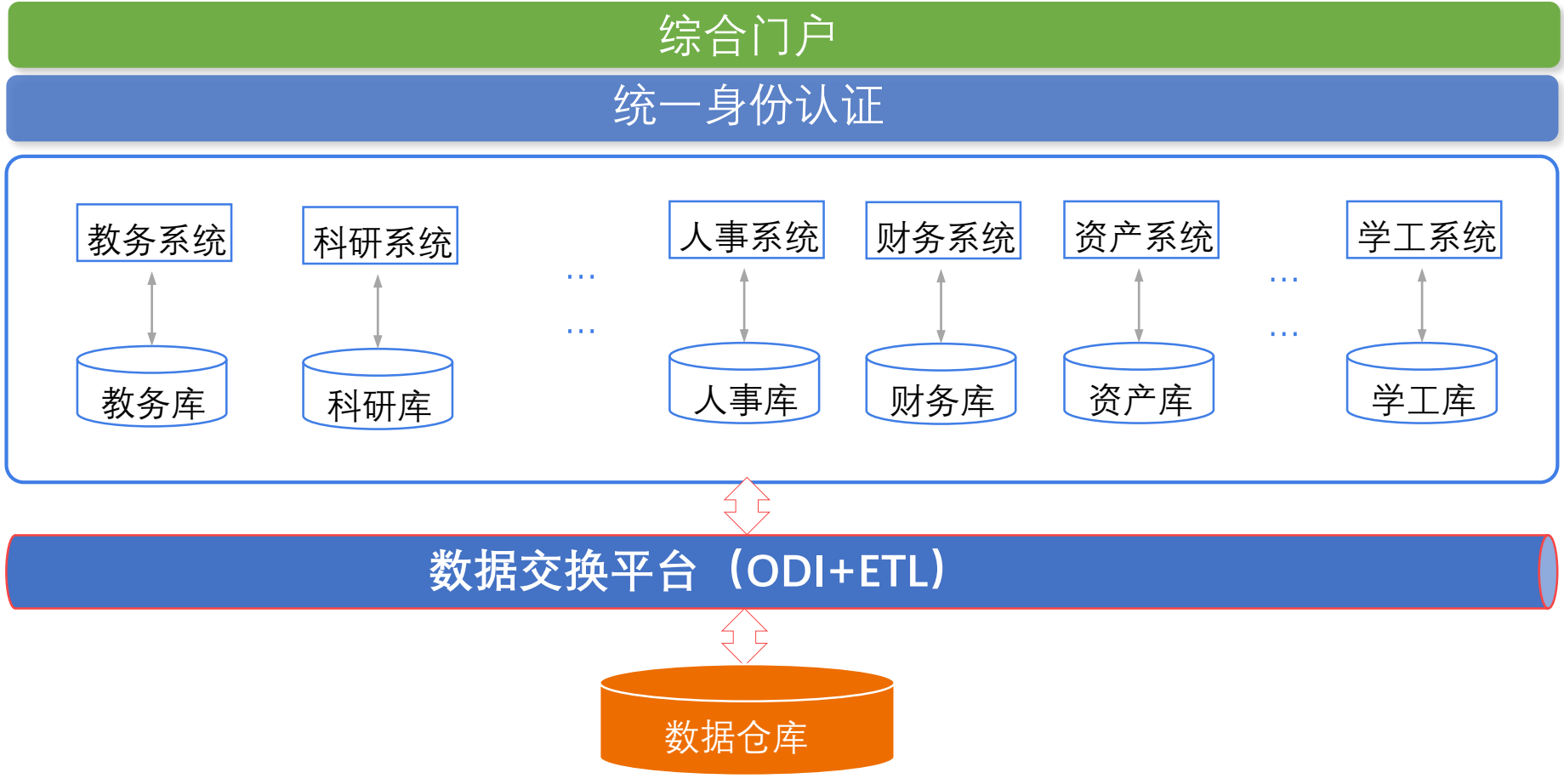
许维胜

同济大学信息化办公室　大数据与网络安全研究中心

云网融合


运营商2（B地区）
XX服务
高速带宽
华为云（上海地区）
VPC
教务系统微服务集群
高速带宽
华为云（北京地区）
人脸检测服务
运营商2（A地区）
XX服务
……
XX服务
运营商4
XX服务
上海教育城域网
同济大学
学生
教师
校内访问教务系统
防火墙
域名解析
DNS
校外访问教务系统
校园外
学生
教师
XX大学
学生
教师
校内访问XX系统
防火墙
域名解析
DNS
校外访问XX系统
校园外
学生
教师
……
XX大学
学生
教师
校内访问XX系统
防火墙
域名解析
DNS
校外访问XX系统
校园外
学生
教师

烟囱式业务系统的集成架构



同济大学目前的业务系统架构

应用架构变迁

第一代：单块架构

- 紧耦合
- 系统复杂、错综交互，牵一发而动全身
- 重复制造轮子：OS、DB、Middleware
- 完全封闭的架构

第二代：SOA架构

- 松耦合
- 通常通过ESB进行系统集成
- 有状态
- 大团队：100~200人
- TTM: 1年、半年、月
- 集中式、计划内停机扩容

第三代：微服务架构

- 解耦
- 小团队：2 Pizza Team
- TTM: 按天、周进行升级发布
- DevOps: CI, CD, 全自动化
- 可扩展性：自动弹性伸缩
- 高可用：升级、扩容不中断业务

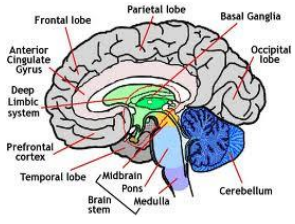
应用向CloudNative演进，微服务是CloudNative的事实标准

PAAS日趋成熟，微服务是大趋势，Docker等容器技术能更好支撑微服务

同济大学面临的业务挑战

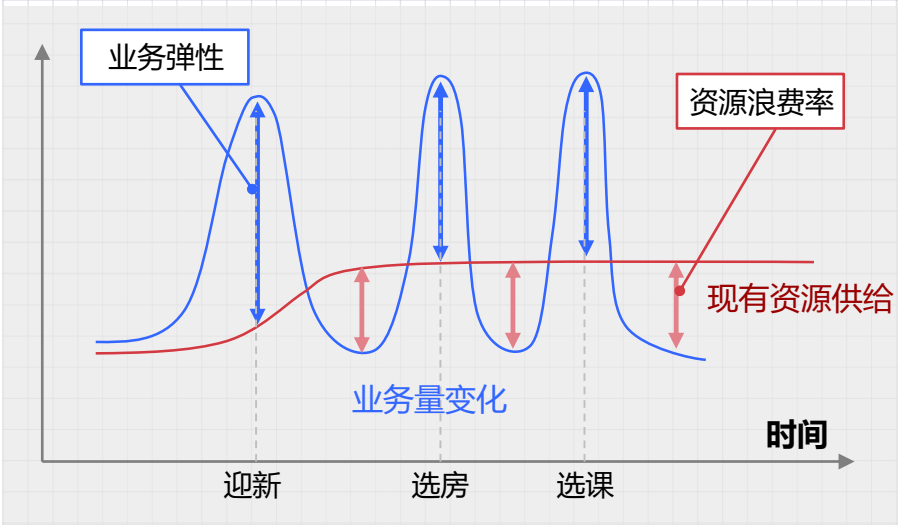
传统架构不可演进

高校业务变革期



- 传统架构，不支持高可靠和高并发需求。
- 业务过多耦合，导致新增、修改功能引发业务系统故障。

- 老的系统无法满足十三五期间同济大学一流学科竞争力提升与拔尖创新人才培养业务需求，新的模块化本研一体化的系统设计迫在眉睫。

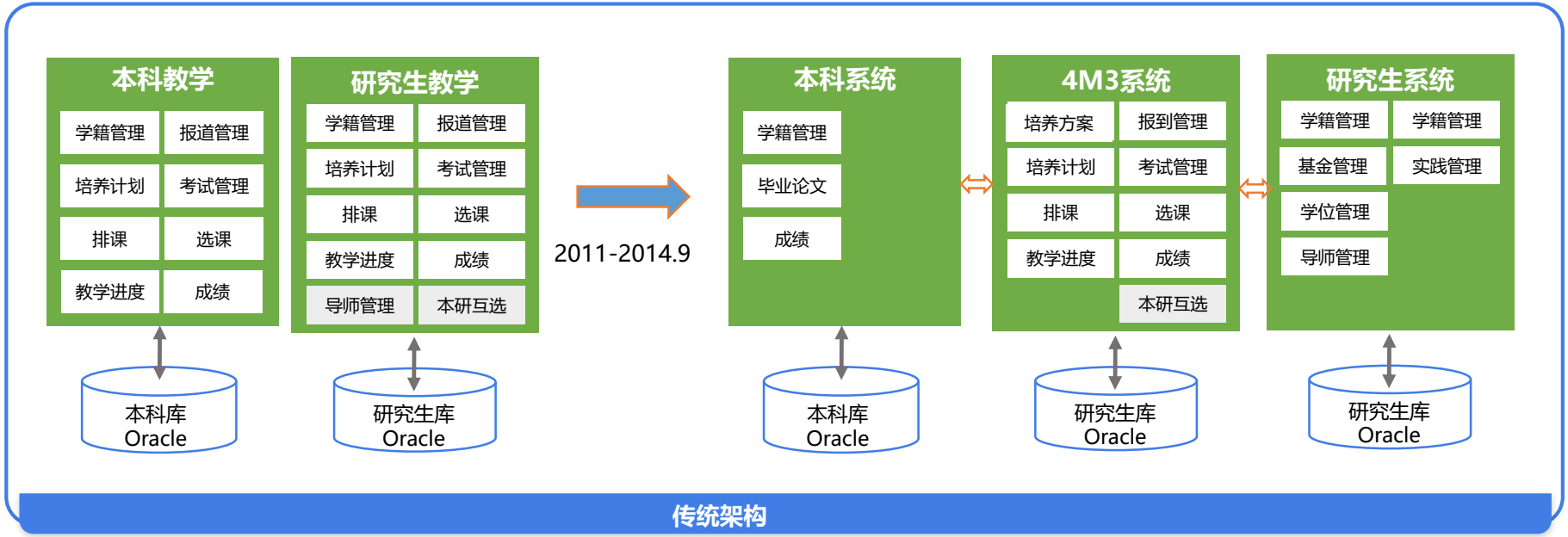


业务弹性和基础设施资源供给之间的矛盾：

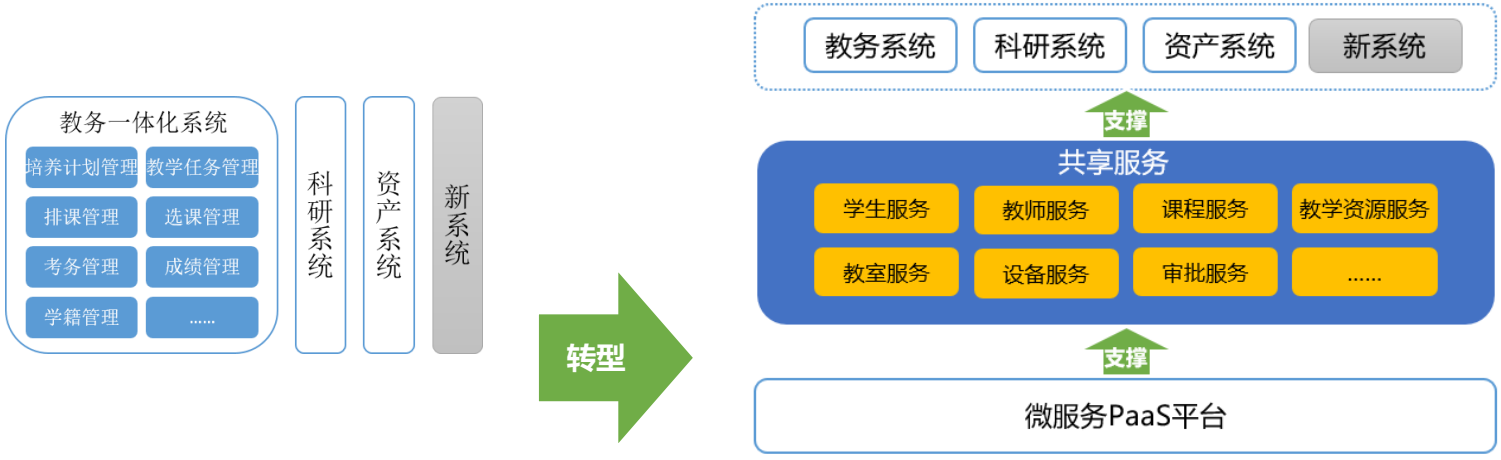
- 高校业务的周期性特点非常强，高峰期的业务量和低谷期需求差异在10倍以上，如何最小化由于资源供给与业务需求之间不匹配带来的业务失败率与资源浪费率？

- **高教行业信息化不断深入普及，所带来的业务变化、业务量和数据量快速增长，传统IT系统架构在柔性和扩展能力上不能满足要求。**
- **本科教学和研究生教学是两套独立系统，业务功能重叠高，重复建设。**
- **传统巨石应用，应用功能更新迭代难，ISV厂商替代难。**

教务系统改造努力：一体化4M3



从传统IT架构向互联网架构转型



原有方案：传统IT架构——生产工具。仅以实现业务功能为目的。

- 创新困难，更新周期长
- 数据孤岛
- 弹性不足，扩展性差
- 无法持续运营和业务沉淀

新型微服务方案：互联网架构——生产力。以核心能力服务化和数据为中心，以支持创新为目标，快速响应需求。

- 以开放的业务能力快速响应需求，灵活定制服务
- 数据实时打通，基于实时数据提供准确服务
- 系统任意服务能力线性扩展，应对用户数量不确定性
- 持续的业务沉淀能力，共享服务开放运营建设生态

本研教务管理系统现状

本科生教务管理系统

研究生教务管理系统

本研一体化教务管理系统

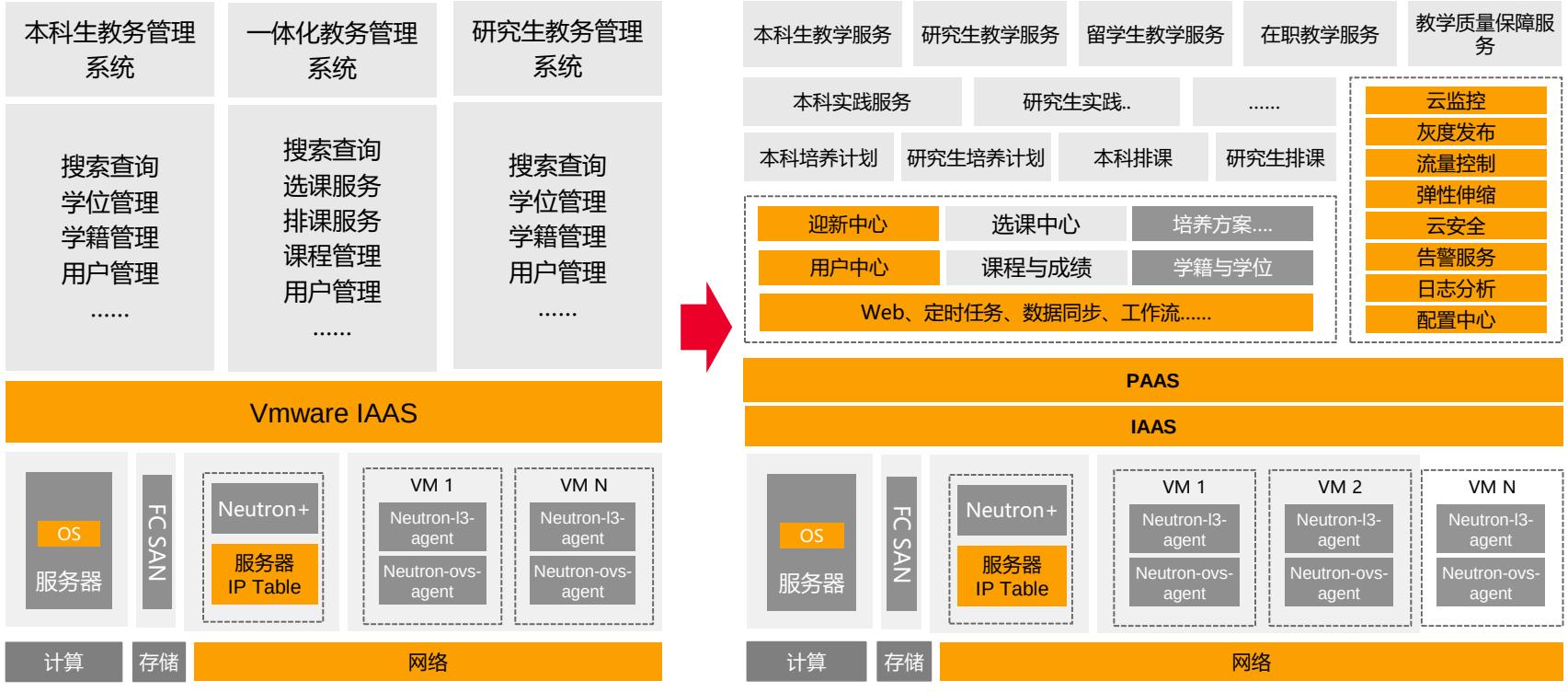
4m3.tongji.edu.cn

新本研一体化教务管理信息系统
(1.tongji.edu.cn)

- 2011年立项建设，2014年9月上线。因域名“4m3”被戏称“四毛三”（4年本科，3年博士，中间m代表硕士，学制可变；体现本硕博贯穿培养理念）。
- 年代久，架构老， 多次升级，衍生出很多模块和小系统。
- 业务耦合度高、稳定性差、bug多，界面不友好、浏览器不兼容、选课高并发时就崩溃。
- 原拟合并成一个系统，结果成为3个系统都在运行。

- 2018年底招标，2019.1.17签订合同，计划开发时间1年，验收后免费维保1年。
- 目前已上线本研迎新报到模块、学堂导师管理模块、本科生微专业/辅修、语言强化班报名/录取/缴费功能；非全日制研究生教务管理主体功能。
- 部分功能随着学校培养改革，有变更或增加。

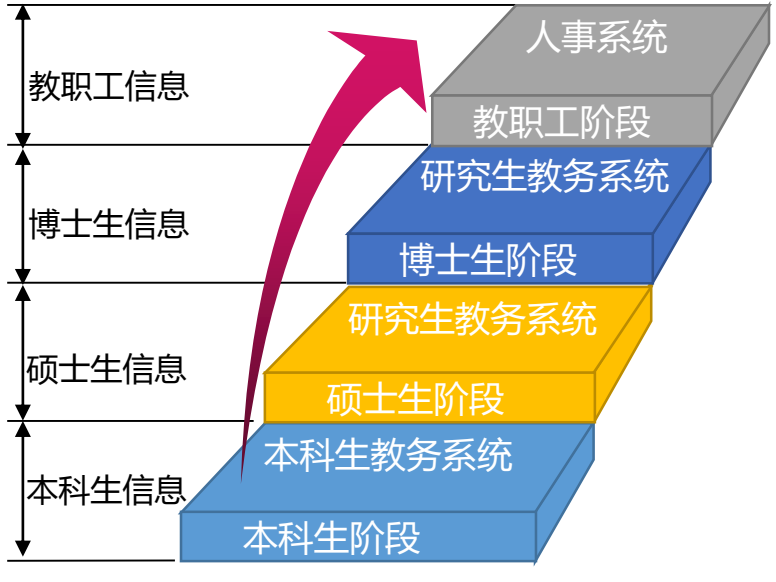
同济大学教学管理信息系统微服务改造探索



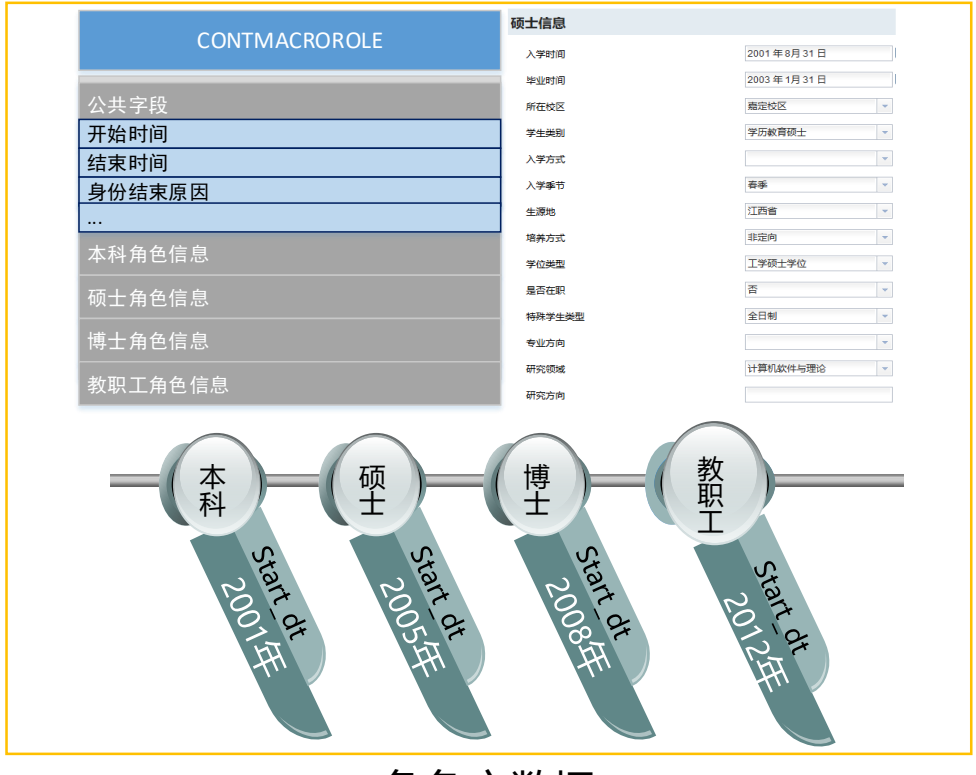
一体化理念

- 一体化学生信息模型，本硕博贯通培养

人员角色模型



高校人员角色特点



角色主数据

同济大学 TONGJI UNIVERSITY | 教学管理信息系统

搜索相关信息 | 李海霞 | 博士生 | 中文

工作台 | 我的课表

- 博士研究生 学号:1810920
  - 我的培养方案
  - 我的培养计划
  - 学生选课
    - 我的课表
  - 我的考试
  - 我的成绩
  - 本研互选课程
  - 论文盲审申请
  - 我要申请答辩
  - 我的学位要求
- 硕士研究生 学号:1531710
  - 培养方案查询
  - 我的培养计划
  - 我的课表
  - 我的成绩
  - 本研互选课程
  - 我的毕业论文
  - 我的学位信息
- 本科生 学号:1153535
  - 我的入学考试信息
  - 培养方案查询
  - 我的培养计划
  - 我的课表
  - 我的成绩
  - 本研互选课程
  - 我的辅修微专业
  - 我的毕业设计

节次/周次	周一	周二	周三	周四	周五	周六	周日
第一节次		数据库技术及应用(010328) 吴冰	张雪伟（1012111）历史建筑形制工艺（101001）(1-9,315室)				
第二节次							
第三节次			周明浩（1012111）历史建筑形制工艺（101001）(1-9,315室)				数据库技术及应用(010328) 吴冰
第四节次							
第五节次			李贞 历史建筑形制与(1-23152...				
第六节次							
第七节次	数据库技术及应用(010328) 吴冰						
第八节次							
第九节次							
第十节次							
第十一节次							
第十二节次							

基本信息

姓名：李海霞　学号：1234003　年级：2018　学院：交通运输工程学院

专业：交通运输工程　培养层次：博士　入学时间：2018.09　学制：4.0年

导师：张敏　培养方案：交通运输专业培养方案

我的个人培养计划

导出个人计划　制定培养计划

学分完成情况

*（统计信息仅供参考，以实际修读完成情况为准）

课程性质	学位课	选修课	必修环节	补修课
已完成学分	3/27	3/27	0/15	0/15
学分小计	5/15/15	5/15/15	0/15/15	0/15/15
学分总计	6/60/60			

学分格式：已完成学分/个人计划要求学分/培养方案要求学分

该专业培养方案总学分要求19学分，学位课8学分；个人培养计划总学分20学分，已获13学分，其中学位课8学分，专业非学位课4学分，必修环节1学分

桥梁技术与工程专业个人培养计划

课程性质	课程编码	课程名称	学分	学时	开课学期	开课学院	是否必修	备注	选课情况	成绩	完成情况
学位课	226061	全日制专业学位硕士英语	3.0	36	18-19(1)	外语学院	是		已选	免修	已完成
	226062	中国特色社会主义理论与实践	3.0	36	18-19(1)	人文学院	是		已选	90	已完成
	226063	交通运输工程学	3.0	36	18-19(1)	交通运输工程学院	是	5选2	已选	90	已完成
	226064	专业外语	3.0	36	18-19(1)	交通运输工程学院	是		未选		未完成
	226065	交通统计分析	3.0	36	18-19(1)	交通运输工程学院	是		未选		未完成
选修课	226066	交通管理学	3.0	36	18-19(1)	交通运输工程学院	是		未选		未完成
	226067	最优化方法	3.0	36	18-19(1)	交通运输工程学院	是		未选		未完成
必修环节	226068	学术讲座	3.0	36	18-19(1)	交通运输工程学院	是		未选		未完成
补修课	226068	工程力学	3.0	36	18-19(1)	交通运输工程学院	是		未选		未完成
总计			27.0	324							

博士生成绩单

姓名	李海霞	性别	女	出生日期	1990.05.15
学号	1234003	国籍	中国	攻读学位	工程博士学位
学制	4年	入学年月	2018.09	指导教师	张敏
学院	交通运输工程学院	学科/专业	交通运输工程		

学位课	学分	成绩	修读学期	备注
全日制专业学位硕士英语	3	免修	2018 秋	免修免考
中国特色社会主义理论与实践	3	80	2018 秋	补考
交通运输工程学	3	90	2018 秋	重修
专业外语	3	80	2018 秋	学分互认
交通统计分析	3	90	2019 春	学分认定
选修课				
交通管理学	3	62	2019 春	
最优化方法	3	68	2019 春	
必修环节				
学术讲座	3	65	2019 春	
补修课				
工程力学	3	65	2019 春	

应修学分	32	平均成绩	76.6	学位课总学分	16
已修学分	35	平均成绩	3.24	学位课总学分	80.4

制表人　王芳　　主管院长　黄雨

同济大学研究生院
打印日期：2019.01.01
(2016版)

支撑系统

智慧校园生态系统：数据仓库与数据治理

（架构先进、功能丰富、安全稳定）

2013.12 咨询、预研
2014.4 一期启动
2014.8 数仓平台上线
2014.10 一期验收
2015.1 二期启动
2015.6 二期验收

如今：
扩大入仓数据源
迁移到大数据平台
完善数据服务平台
提供数据服务
. . .

- 收集50多个业务系统
- 调研入仓35个业务系统
- 涉及22个业务部门
- 召开100多场调研会议
- 对接20余个开发公司
- 1123张数据表入仓
- 数据总量达164G

主要成果：

1. 先后上线 9 个子系统

2. 人员信息阶段性整合

263471 人次 → 230146 人

12.64% OFF

3. 数据服务（160余次）

人事考核、各学院考核、个税申报、工程认证等

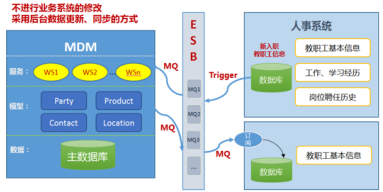
智慧校园数据生态系统：主数据平台

（高校首例、平台先进、功能强大）

2015.10 平台采购招标
2015.12 熟悉平台
2016.9 人员模型
2017.3 ESB模板流
2017.6 数据加载
2017.7 开发管理平台

如今：
完善主数据模型
基于主数据应用开发
丰富平台功能
专项经费/退管办/妇委等系统
提供教职工主数据
. . .

主数据与现有系统交互示例

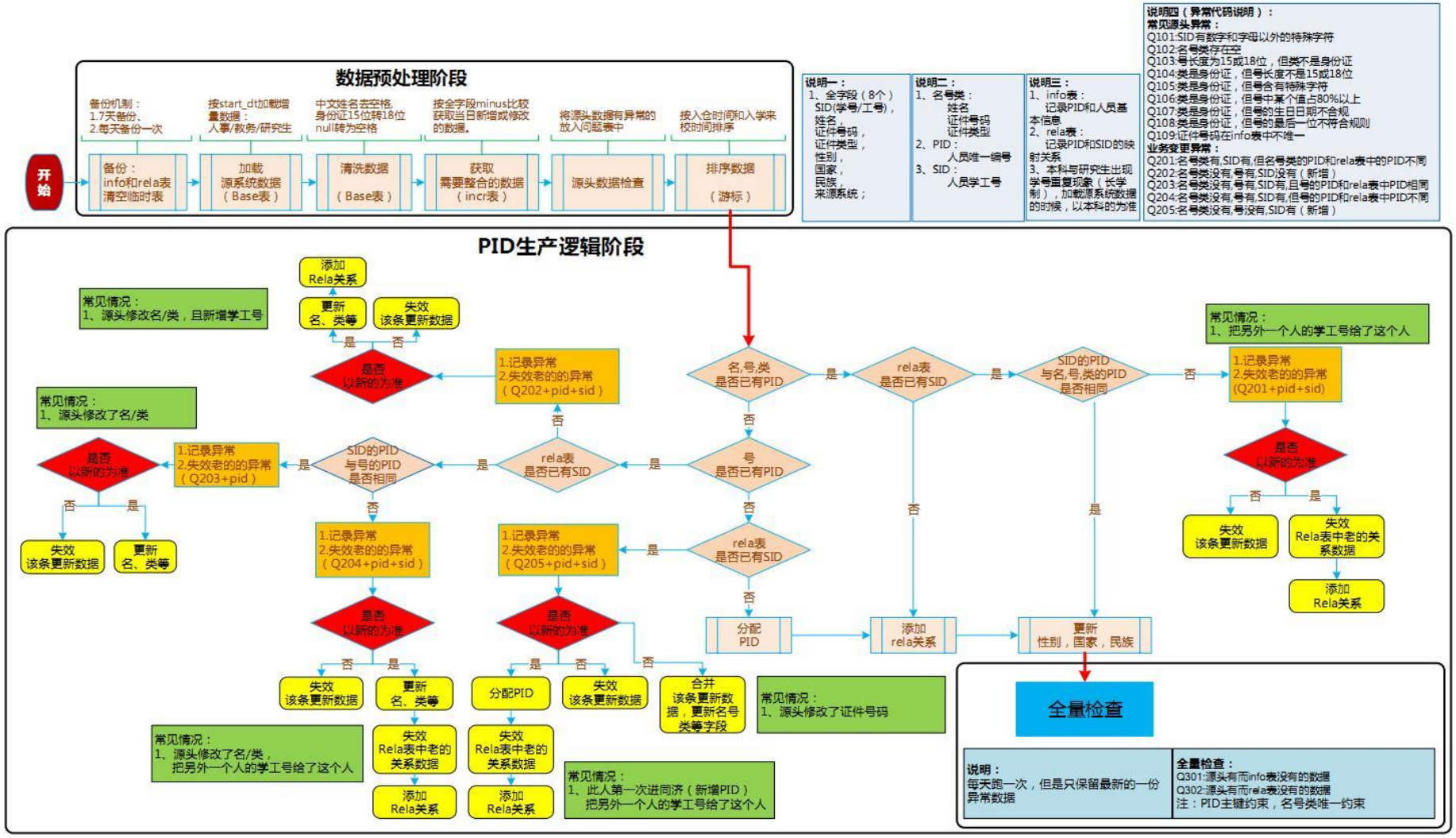


主数据是数据管控与治理的核心平台，让数据真正可用！

主数据管理四大要素

主数据的质量决定着数据可用，关系着主数据的命运！在将业务系统数据加载到主数据库时，发现了近30种数据问题，逐一与源头部门沟通、解决。

业务系统	数据问题	已处理	无法处理
人事系统	7	6	1
研究生系统	22	21	1
教务系统	7	2	5


数据预处理阶段
开始
备份机制：
1.7天备份、
2.每天备份一次
备份：
info和rela表
清空临时表
按start_dt加载增量数据：
人事/教务/研究生
加载
源系统数据
（Base表）
中文姓名去空格，
身份证15位转18位
null转为空格
清洗数据
（Base表）
按全字段minus比较获取当日新增或修改的数据。
获取
需要整合的数据
（incr表）
将源头数据有异常的放入问题表中
源头数据检查
按入仓时间和入学来校时间排序
排序数据
（游标）
说明一：
1、全字段（8个）
SID(学号/工号)，
姓名，
证件号码，
证件类型，
性别，
国家，
民族，
来源系统；
说明二：
1、名号类：
姓名
证件号码
证件类型
2、PID：
人员唯一编号
3、SID：
人员学工号
说明三：
1、info表：
记录PID和人员基本信息
2、rela表：
记录PID和SID的映射关系
3、本科与研究生出现学号重复现象（长学制），加载源系统数据的时候，以本科的为准
说明四（异常代码说明）：
常见源头异常：
Q101:SID有数字和字母以外的特殊字符
Q102:名号类存在空
Q103:号长度为15或18位，但类不是身份证
Q104:类是身份证，但号长度不是15或18位
Q105:类是身份证，但号含有特殊字符
Q106:类是身份证，但号中某个值占80%以上
Q107:类是身份证，但号的生日日期不合规
Q108:类是身份证，但号的最后一位不符合规则
Q109:证件号码在info表中不唯一
业务变更异常：
Q201:名号类有,SID有,但名号类的PID和rela表中的PID不同
Q202:名号类没有,号有,SID没有（新增）
Q203:名号类没有,号有,SID有,且号的PID和rela表中PID相同
Q204:名号类没有,号有,SID有,但号的PID和rela表中PID不同
Q205:名号类没有,号没有,SID有（新增）
PID生产逻辑阶段
名,号,类
是否已有PID
是
rela表
是否已有SID
是
SID的PID
与名,号,类的PID
是否相同
否
1.记录异常
2.失效老的的异常
(Q201+pid+sid)
是否
以新的为准
否
是
失效
该条更新数据
失效
Rela表中老的关系数据
添加
Rela关系
常见情况：
1、把另外一个人的学工号给了这个人
否
号
是否已有PID
否
rela表
是否已有SID
是
否
分配
PID
添加
rela关系
更新
性别，国家，民族
否
是
rela表
是否已有SID
是
否
1.记录异常
2.失效老的的异常
（Q205+pid+sid）
是否
以新的为准
是
否
否
分配PID
失效
该条更新数据
合并
该条更新数据，更新名号类等字段
常见情况：
1、源头修改了证件号码
失效
Rela表中老的关系数据
添加
Rela关系
常见情况：
1、此人第一次进同济（新增PID）
把另外一个人的学工号给了这个人
1.记录异常
2.失效老的的异常
（Q202+pid+sid）
否
rela表
是否已有SID
是
SID的PID
与号的PID
是否相同
是
否
1.记录异常
2.失效老的的异常
（Q203+pid）
是否
以新的为准
否
是
失效
该条更新数据
更新
名、类等
常见情况：
1、源头修改了名/类
1.记录异常
2.失效老的的异常
（Q204+pid+sid）
是否
以新的为准
否
是
失效
该条更新数据
更新
名、类等
失效
Rela表中老的关系数据
添加
Rela关系
常见情况：
1、源头修改了名/类，
把另外一个人的学工号给了这个人
是否
以新的为准
是
否
更新
名、类等
失效
该条更新数据
常见情况：
1、源头修改名/类，且新增学工号
全量检查
说明：
每天跑一次，但是只保留最新的一份异常数据
全量检查：
Q301:源头有而info表没有的数据
Q302:源头有而rela表没有的数据
注：PID主键约束，名号类唯一约束

人员信息整合

整合前：

教职工：26768
本科生：105621
研究生：131082

整合结果

身份数	人数（人）
6个身份（如：本-硕-博-博士后-派遣-在编）	2
5个身份（如：本-硕-博-博士后-在编）	49
4个身份（如：本-硕-博-在编）	391
3个身份（如：本-硕-博、本-硕-派遣、硕-博-在编）	2214
2个身份	24923
1个身份	202567

人事系统（派遣、兼职等）

教务系统（从97级开始）

研究生系统（从98级开始）

按姓名、性别、身份证号等鉴别

同济人信息库

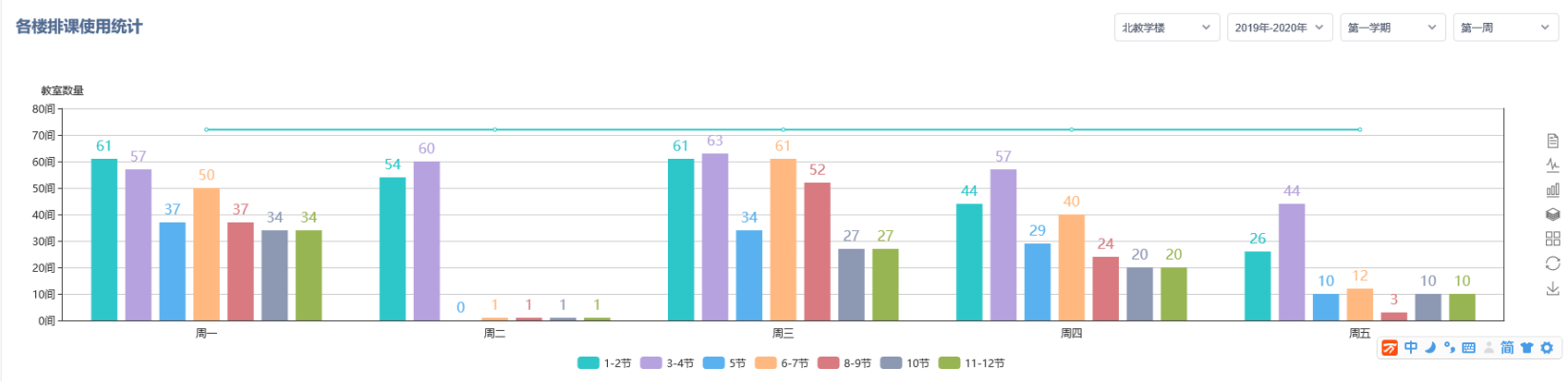
12.64% OFF

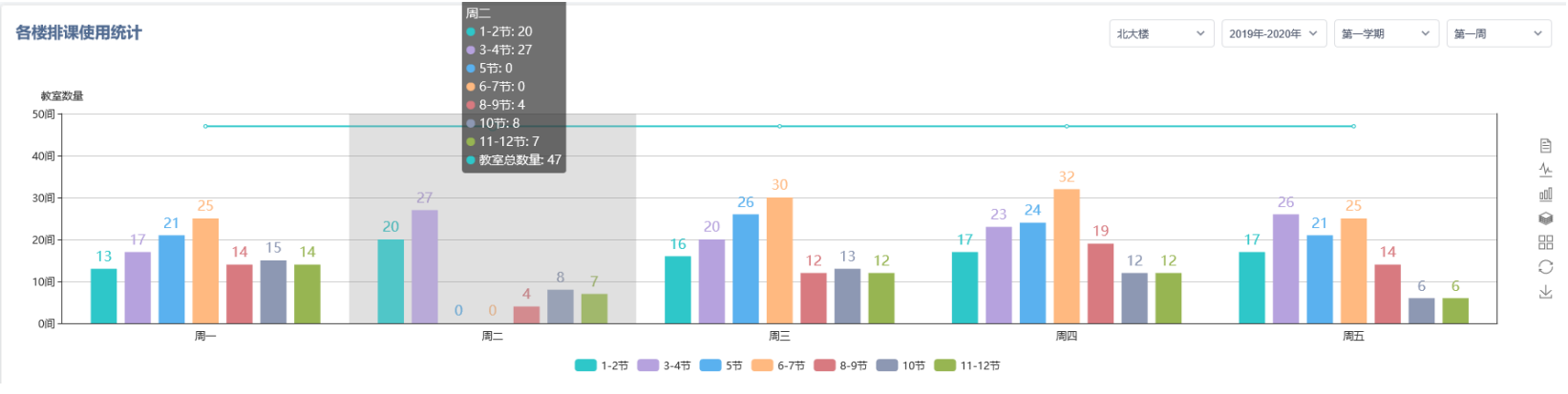
263471人次 → **230146人**

数据统计时间：2019-09-23

一体化理念

- 资源配置一体，提高资源利用率





一体化理念

- 知识体系一体化，本硕课程互选互认
  - ✓可以保研的本科生，在本科阶段可以选修硕士课程，入学后计入学分
  - ✓硕士研究生或博士研究生，导师可以要求其补修本科或硕士课程

本科生 学号:1153535
- 我的入学考试信息
- 培养方案查询
- 我的培养计划
- 我的课表
- 我的成绩
  - 本研互选课程
- 我的辅修微专业
- 我的毕业设计
- 我的毕业设计

博士研究生
- 课程查询
- 培养方案查询
- 讲座查询

硕士研究生
- 课程查询
- 培养方案查询
- 讲座查询

筛选条件 收起

重置 筛选

学期：2014-2015学年 第2学期　课程代码：　课程名称：

开课学院：请选择　课程性质：请选择　审核状态：请选择

本研互选申请

添加 删除

课程代码	课程名称	开课学院	课程性质	学分	考核方式	申请理由	驳回原因	审核状态	操作
1200130	偏微分方程数值解	建筑与城市规划学院	公共课	3.0	考试	已保研,申请提前修读		审核通过	
1200143	艺术造型（2）	艺术与传媒学院	公共课	3.0	考试	已保研,申请提前修读		审核通过	
1200168	传播学基础	艺术与传媒学院	公共课	3.0	考试	已保研,申请提前修读		审核通过	

共5条 20条/页 1 前往 1 页

选型过程——高校系统架构现状比较

架构名称	特点	缺点	使用范围	产品（平台）	主流开发商
单块架构	采用面向对象设计，按应用系统的***应用功能***（如用户界面层、控制层、逻辑层、数据层）横向分层；整个应用运行在***单个进程***	需求变更困难,维护成本高,扩展性差,可靠性差	二十多年前就被提出，目前高校中**仍占主流**	基于J2EE或MVC等开发平台	
SOA架构	是一种粗粒度、松耦合服务架构，服务之间通过简单、精确定义接口进行通讯，不涉及底层编程接口和通讯模型。解决了单块架构的问题	靠总线来整合集成大量单一庞大的应用，导致难以维护	十多年前被提出，已有不少学校在实践		
微服务架构	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，中立性，语言无关，协议跨平台	难以解决事务一致性、管控、集成和部署复杂	近几年提出来的一种全新系统架构，个别学校在探索		

- 微服务从头实施比较困难，因为微服务架构需要配合完整的服务生命周期管理框架（自动化工具及DevOps平台），帮助完成服务部署、监控、运维和治理。中小企业由于自身技术实力等问题，难以从头搭建微服务平台，即使成功搭建，也耗时耗力，且可靠性、扩展性低，所带来的收益可能不及付出的成本，后期的扩展、运维成本甚至不降反增。
- 因此，借助**微服务PaaS平台**能够极大减少前期基础设施建设所带来的成本和复杂性，便于直接进行业务开发，借助平台自动化运维，提高系统后期的扩展性和灵活性。

微服务架构特性——Martin Fowler

- **通过服务来实现组件：**将应用运行在不同的进程中，单一服务的局部变化只需重新部署对应的服务进程。 服务之间的调用是跨进程的，设计时必须定义清晰的边界和职责。
- **按业务能力来划分服务与组织团队：**康威定律（Conway ‘s law）指出任何设计系统的组织，最终产生的设计等同于组织之内、之间的沟通结构。 系统架构与组织沟通结构匹配。
- **服务即产品：**让开发团队负责整个产品的全部生命周期。 You build it, you run it. 开发团队对软件在生产环境的运行负全部责任，让服务的开发者与服务的使用者（客户）形成每天的交流反馈，来自直接客户端的反馈有助于开发者提升服务的质量。
- **聪明的端点和哑管道：**服务作为智能终端，所有的业务逻辑在服务内部处理，而服务间的通信尽可能的轻量化，不添加任何额外的业务规则。
- **分散的治理和数据管理：**可以针对不同的业务服务特征选择不同的技术平台或产品，有针对性的解决具体的业务问题。
- **基础设施自动化：**一系列的多进程服务，意味着开发、调试、测试、集成、监控和发布的复杂度都会相应增大。 必须要有合适的自动化基础设施来支持微服务架构模式，否则开发、运维成本将大大增加。
- **考虑到失败的设计：**任何时刻对服务的调用都可能因为服务方不可用导致失败，这就要求服务的消费方需要优雅的处理此类错误。
- **进化的设计：**发送时要保守，接收时要开放。按照伯斯塔尔法则的思想来设计服务调用，发送的数据要更保守，意味着最小化的传送必要的信息，接收时更开放意味着要最大限度的容忍信息的兼容性。 多余的信息不认识可以忽略，而不应该拒绝或抛出错误。

SOA与微服务架构区别

层次	传统SOA	微服务实践
架构层	ESB ,BPEL, SCA等	• 去中心化 • 同步调用通过服务发现机制直接调用，无中心转发，无消息格式转换 • 异步调用通过不感知内容的消息服务集群
协议层	Web Service, Soap, WSDL	拥抱简单透明的轻量级协议，http REST, JSON. （有些场景也可以使用XML转换格式）
应用层	• OLTP中间件 • Weblogic, Websphere	• 弱化作为中间层的传统中间件的作用，一切皆服务，由独立的服务提供实现 • 无语言和框架绑定，倾向于更小，启动更快的语言或者框架，golang，node.js… • 如果使用java语言倾向于jetty，spring这些轻量级技术
数据层	• 全局事务 • 商业关系型数据库如Oracle	• 一般无全局事务和强一致性要求 • 开源数据库mysql, prostegsql, NoSql • 看业务场景和开发人员的能力
适用场景	• 数据有强一致性要求 • 集成多供应商、异构协议的商业应用	• 数据最终一致性场景 • 基于开源的产品或者构架 • Scalability很重要

传统SOA重型框架，各个模块紧耦合，升级周期长、成本高，不能满足产品需求快速变化的场景。

微服务带来的好处与挑战

服务模块的边界更清晰：微服务强调模块化结构（REST接口调用），这对大型团队非常重要

支持独立部署：简单服务更易部署，由于服务是自治的，出现问题之后不会引起系统崩溃

允许技术多样性：有了微服务，你可以混合使用多种编程语言、开发框架和数据存储技术

Martin Fowler

01 July 2015

Microservices provide benefits...

- Strong Module Boundaries: Microservices reinforce modular structure, which is particularly important for larger teams.
- Independent Deployment: Simple services are easier to deploy, and since they are autonomous, are less likely to cause system failures when they go wrong.
- Technology Diversity: With microservices you can mix multiple languages, development frameworks and data-storage technologies.

...but come with costs

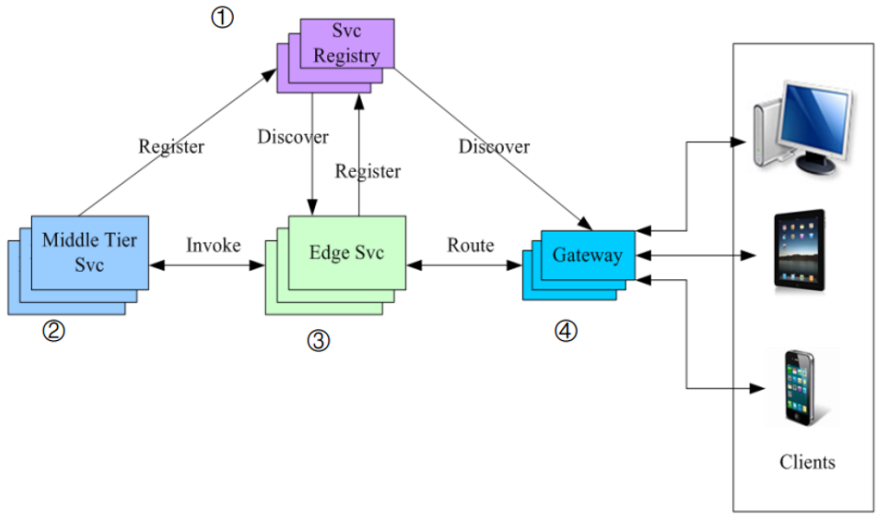
- Distribution: Distributed systems are harder to program, since remote calls are slow and are always at risk of failure.
- Eventual Consistency: Maintaining strong consistency is extremely difficult for a distributed system, which means everyone has to manage eventual consistency.
- Operational Complexity: You need a mature operations team to manage lots of services, which are being redeployed regularly.

分布式编程难度大、有风险：分布式系统编程难度更大，远程调用更慢且总存在失败风险

需处理分布式系统的一致性：对于分布式系统来说，保持一致性非常困难，意味着大家都要处理最终一致性

增加运维复杂性：需要一个成熟的运维团队（机制）来管理大量需要频繁部署的服务

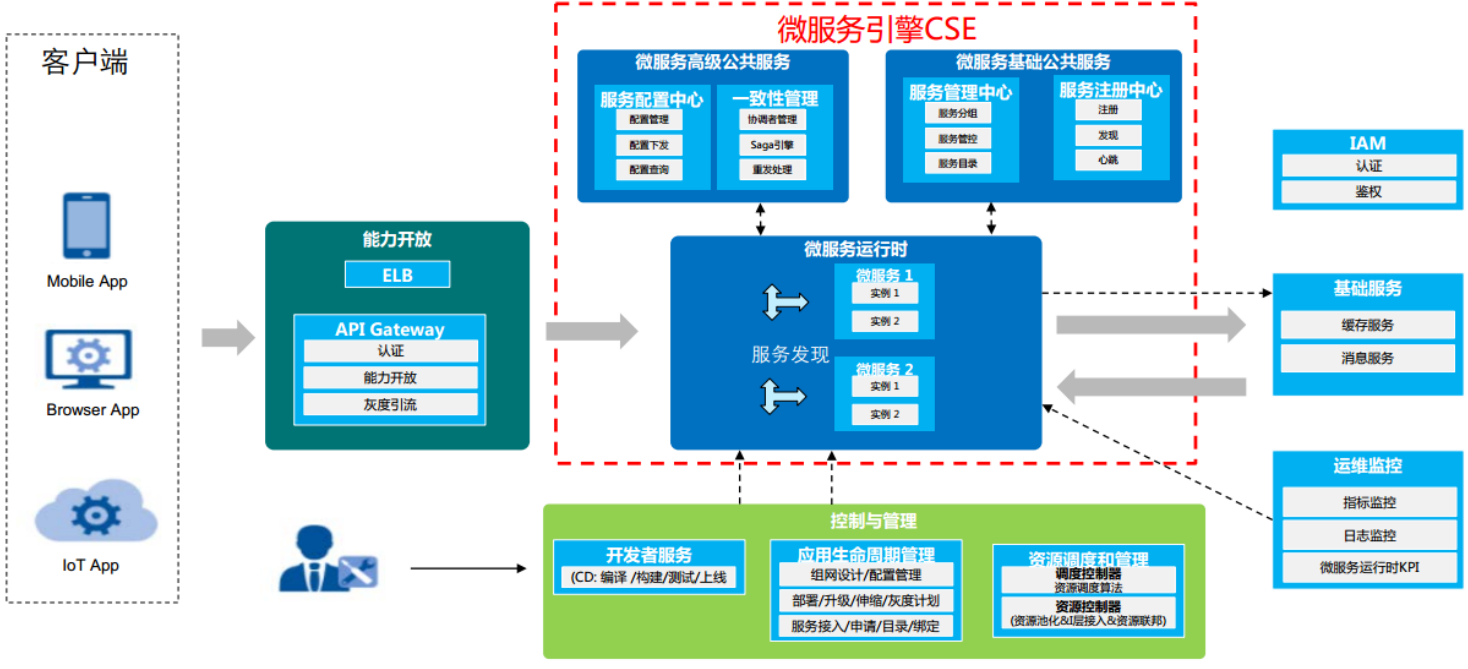
微服务架构核心组件（简化模型）



- **服务注册：**为服务提供方提供服务地址注册服务，并接受服务提供方定期上报心跳以表明服务的存活状态；为服务消费方的LB组件提供目标服务地址列表查询服务。
- **后端服务：**也称中间层服务Middle Tier Service，提供具体业务服务。
- **前端服务：**也称边缘服务Edge Service，前端服务的作用是对后端业务服务做必要的聚合和裁剪后暴露给外部不同的设备，如PC，Pad或者Phone。
- **服务网关(Service Gateway)：**网关是连接企业内部和外部系统的一道门。
- **服务发现：**后端服务启动时，会将地址信息注册到服务注册表，前端服务通过查询服务注册表，就可以发现然后调用后端服务。前端服务启动时，也会将地址信息注册到服务注册表，这样网关通过查询服务注册表，就可以将请求路由到目标前端服务。

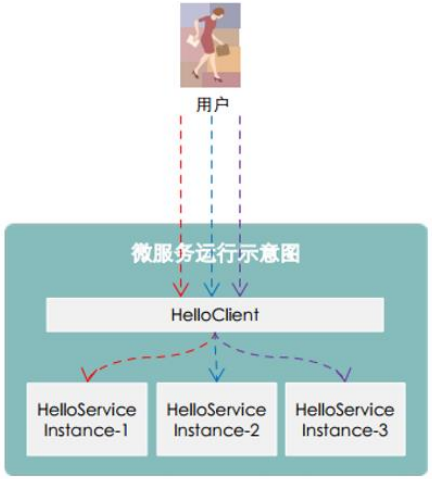
PaaS平台 – 微服务云应用平台

支持基于微服务的应用开发、治理、部署及运维监控的全生命周期管理，并提供大规模容器集群管理及中间件服务等平台能力；提供微服务框架，兼容主流开源生态，帮助快速构建基于微服务架构的分布式应用。



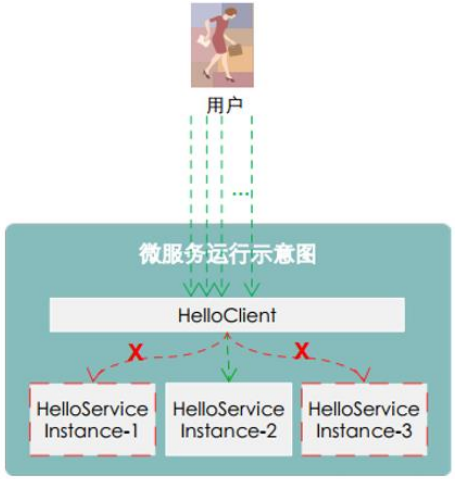
PaaS平台 – 微服务云应用平台

微服务治理，保障应用运行不断服



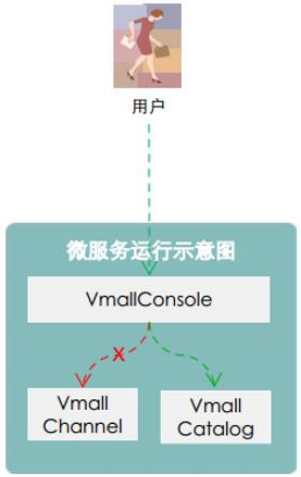
自动负载均衡

可选策略：顺序、随机、本地、粘滞、基于实例负载、基于权重）



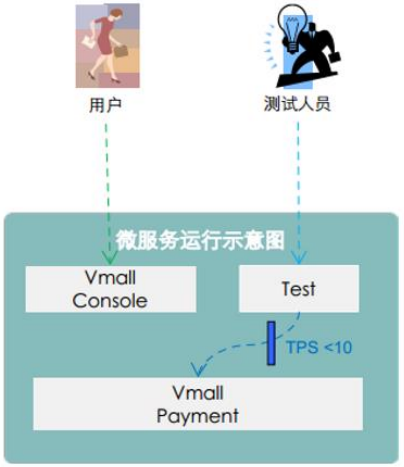
自动容错

任意实例Down掉(至少保留一个实例)，不影响正常业务



服务降级

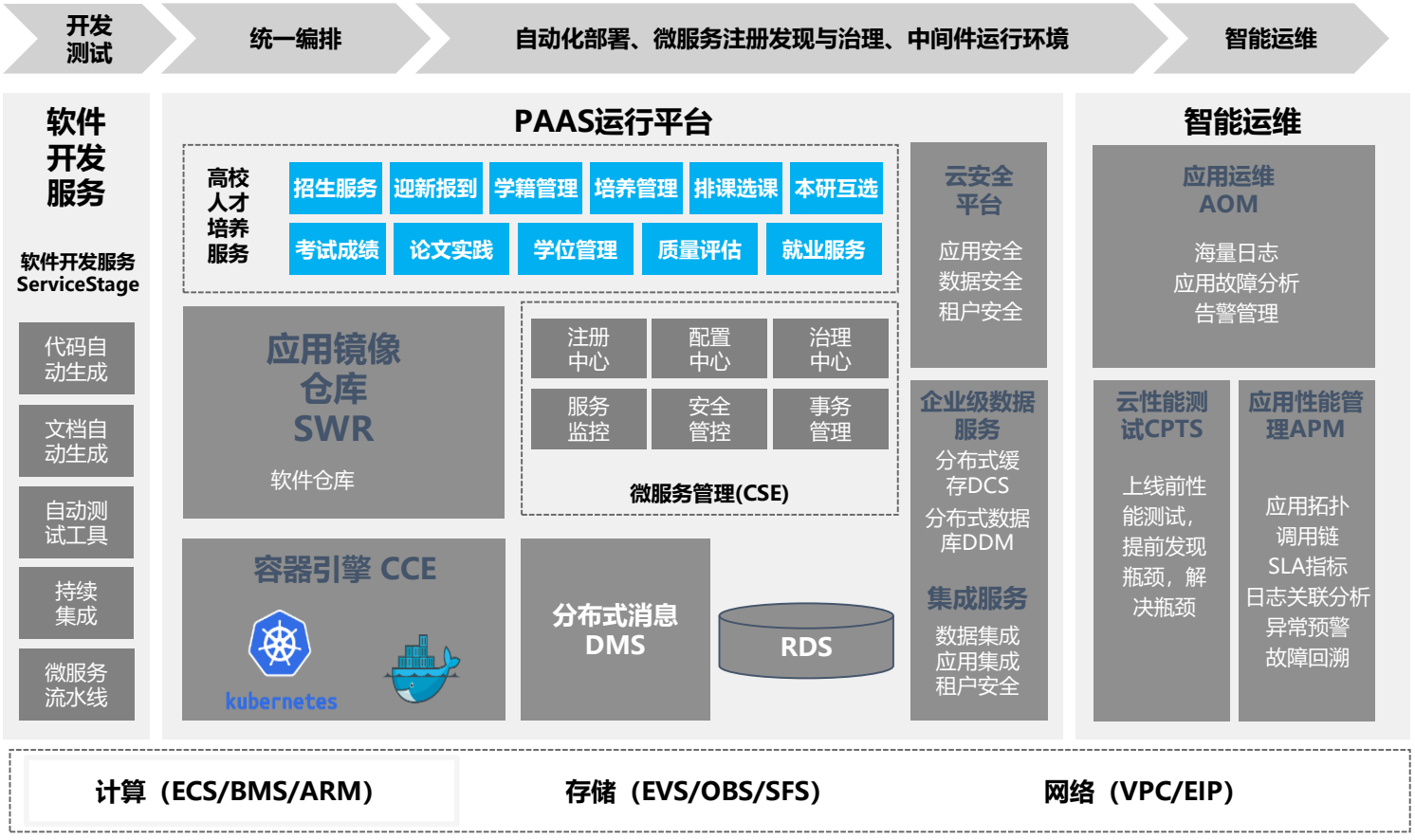
关闭非核心业务，保证核心业务可用



服务限流

请求超过处理能力，按策略丢弃，保证已接纳请求正常处理

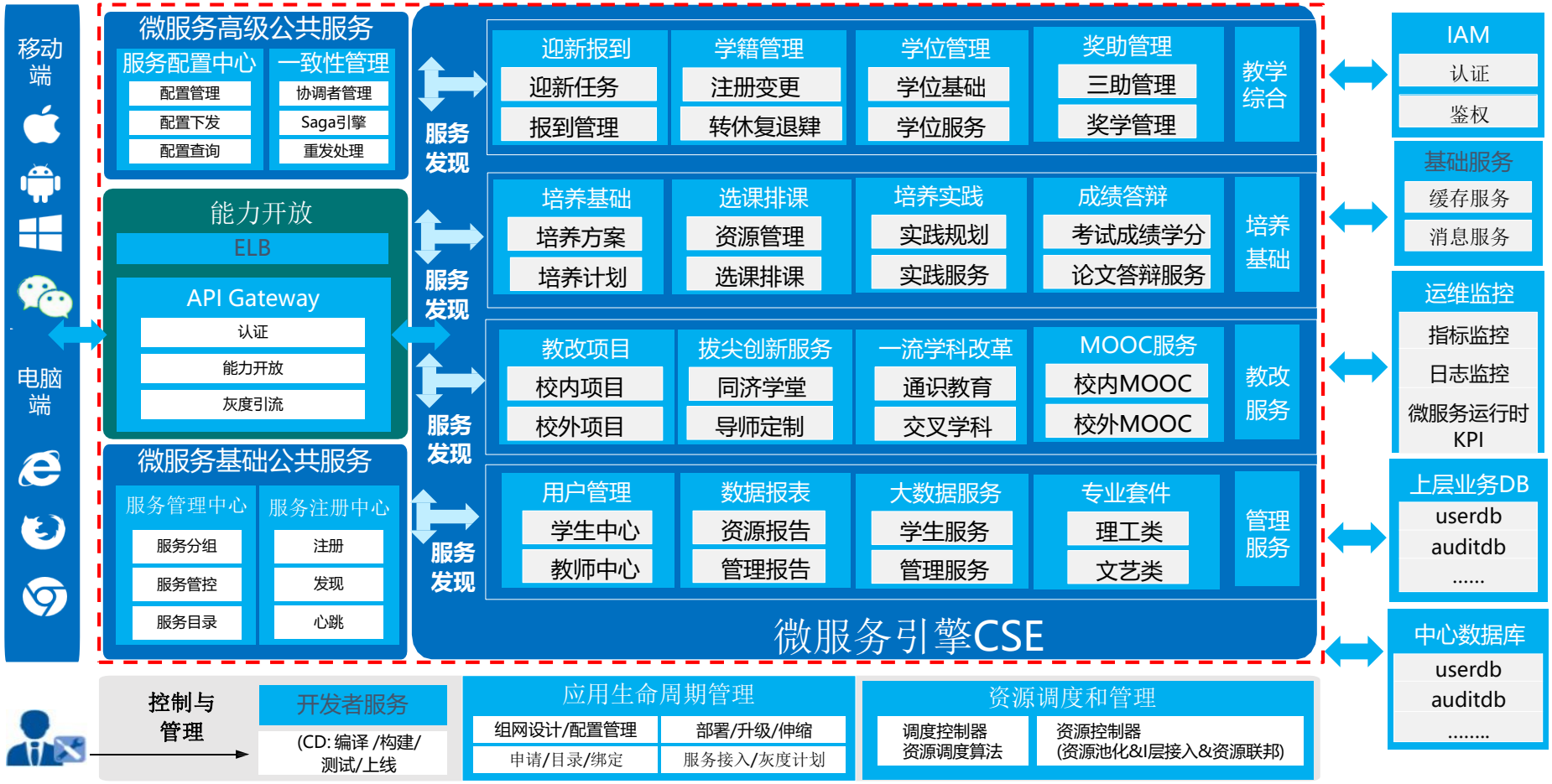
同济大学教学管理系统技术中台的PaaS平台



优势：

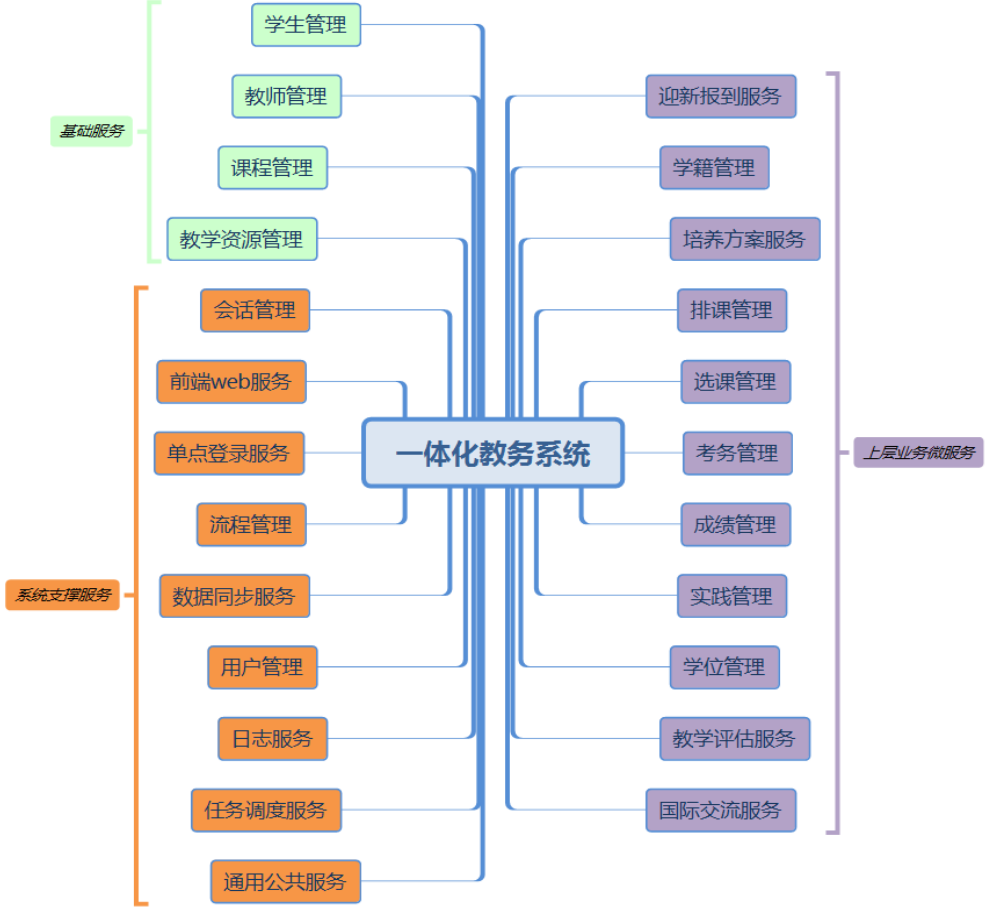
- 解决信息孤岛问题，提升工作效率；
- 一站式应用生命周期管理；
- 应用弹性伸缩，从容面对流量高峰；
- 保障业务高安全、高可靠的运行；
- 技术领先，业务模块可以拆分部署上线；

同济大学本研一体化教务管理系统的微服务设计规划

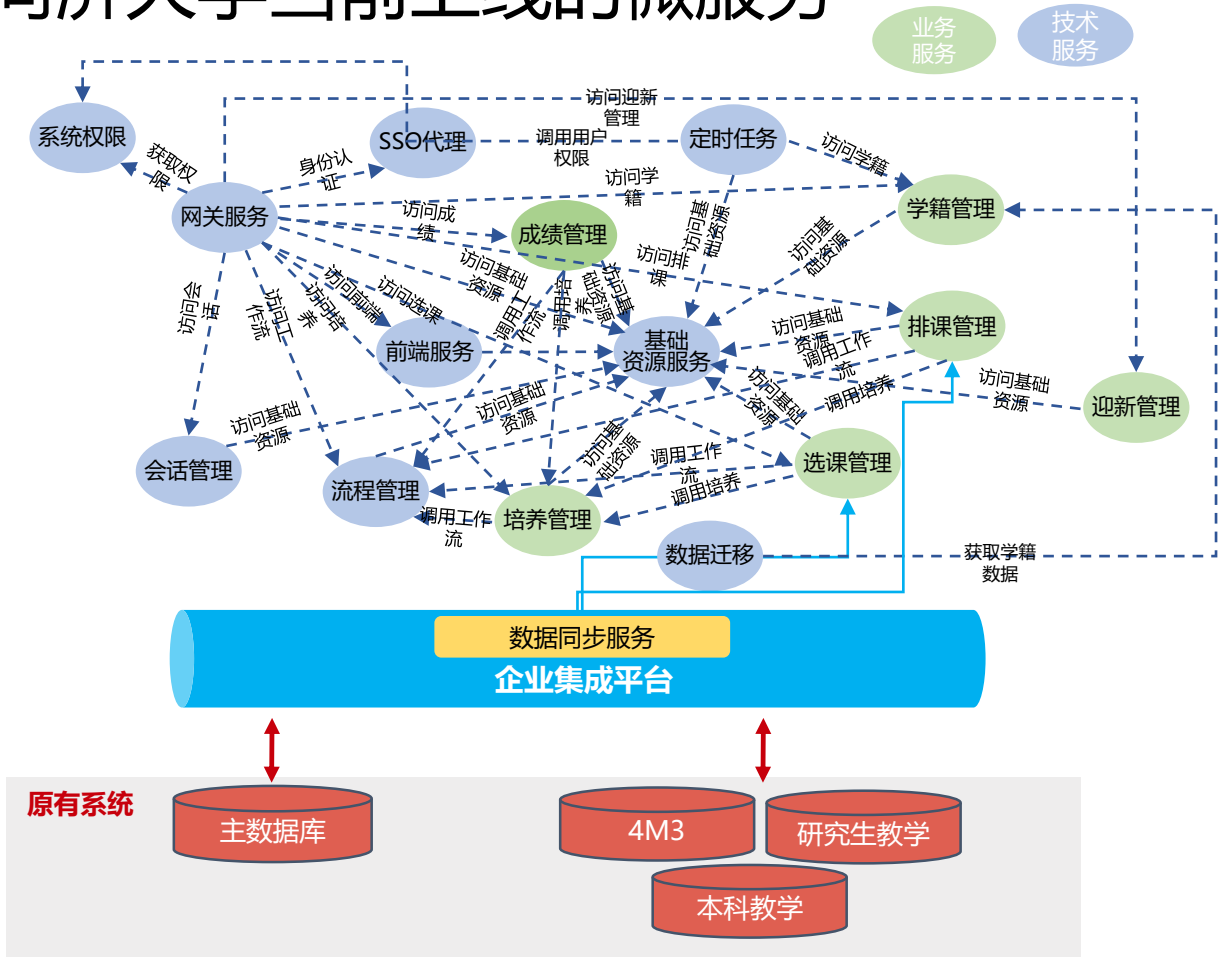


同济大学教学管理系统微服务划分

- 重新梳理系统业务流程，以“服务”视角重组和整合业务场景。
- 按照微服务划分原则对一体化教务系统进行服务划分。



同济大学当前上线的微服务



- eduweb-product2
- baserssservice
- scoremanagement
- coursearrangementservice
- courseelection
- cultureservice
- workflow
- gatewayservice
- transformer
- cron-service
- welcomeservice
- oiosaml-proxy-online
- studentservice
- userservice
- sessionservice

同济大学 TONGJI UNIVERSITY 教学管理信息系统

学号或工号搜索 切换用户 中文 您好, admin(admin1) 普通研究生

学籍信息查询

资源管理

课程管理
- 课程申请
- 课程查询
- 课程维护
- 课程审批
- 外国语课程

专业管理
- 本国科学学位学科专业
- 本国专业学位学科专业
- 研究生科学学位专业
- 研究生专业学位专业

教室管理
- 基本信息管理
- 教室借用人管理
- 查看空闲教室
- 教室借用申请
- 教室借用审核
- 教室使用申请
- 教室使用情况报表
- 教室使用率统计

教师管理
- 校内教师管理
- 校内教师查询
- 校内授课资格申请
- 校内授课资格审批
- 校外教师申报
- 校外教师审核
- 校外教师申请
- 在职教师统计
- 教师管理设置

中期考核

- 考核规则设置
- 考核时间设置
- 考核名单管理
- 考核申请模板
- 考核批次维护
- 中期考核工作申请
- 考核审批
- 考核全过程画像
- 考核成绩录入
- 考核结果查询

消息中心

- 未读消息

系统管理

基础信息
- 审计日志
- 参数配置
- 部门管理
- 流程监控
- 校历监控
- 作息时间管理
- 节假日管理
- 数据字典管理
- 公共导出
- 系统配置

系统消息
- 系统公告

权限管理
- 用户监控
- 用户组监控
- 角色管理
- 权限管理
- 菜单管理

成绩管理

研究生
- 成绩帐号设置
- 成绩录入开关
- 课程成绩管理
- 学生成绩维护
- 学生成绩变更
- 课程成绩变更
- 课程成绩变更审核
- 学生成绩变更审核
- 学分认定开关设置
- 校内学分认定申请
- 校外学分认定申请
- 校内学分认定审核
- 校外学分认定审核
- 校内学分认定指定审核
- 校外学分认定指定审核
- 补录考核成绩录入
- 必修环节成绩管理
- 不及格成绩管理
- 我的成绩查看
- 学生成绩信息导出
- 学生成绩打印
- 平均成绩与绩点
- 学生离校成绩查询
- 成绩操作日志
- 学院成绩录入汇总
- 成绩录入汇总

我的待办

- 我的申请
- 我的审批
- 未审批
- 已审批

培养方案

研究生
- 培养方案模板
- 培养方案规则
- 课程分类管理
- 培养方案开关设置
- 培养方案管理
- 批量修改培养方案
- 培养方案审批
- 培养方案查询
- 我的培养方案

培养计划

研究生
- 培养方案关联
- 培养方案关联/取消关联记录
- 培养计划管理
- 我的培养计划
- 跨院系选课设置
- 学生外国语设置
- 计划厂版权限
- 计划课课统计
- 批量修改培养计划

教学进度

教学进度设置
- 教学进度录入监控

教学进度管理
- 编辑教学进度

教学进度录入
- 录入进度列表

教学进度审核
- 教学进度审核
- 管理审核进度教师

教学进度查询
- 查询教学进度

学生管理

学生信息管理
- 审核权限设置
- 审核权限配置
- 学生信息管理
- 信息变更管理
- 信息变更日志
- 学生导师指定

缴费管理
- 在校应收费设置
- 应缴费名单维护

注册管理
- 注册权限设置
- 注册管理
- 在校生注册
- 学籍关系维护

学生离校管理
- 学生离校名单管理
- 离校日志查询

研究生选课

选课设置
- 选课轮次管理
- 可选名单管理
- 添加可选课程

学生选课

报表管理
- 点名册
- 学生课表
- 教师课表
- 选课日志
- 选课结果统计
- 代理选课

选课管理
- 选课维护设置
- 教学班容量管理
- 选课结果处理
- 学生课程维护
- 未选课学生名单

重修选课
- 重修选课设置
- 重修选课门数上限

免修免考
- 免修申请/审核开关
- 免修申请条件
- 免修申请审核
- 免修申请统计

迎新管理

任务管理
- 任务信息
- 任务统计报表

报到管理
- 报到权限管理
- 报到大屏

人脸识别
- 人脸识别客户端
- 照片采集
- 学生报到状态

入学分级考试管理
- 考试分级设置
- 考试名单管理
- 考试成绩管理
- 我的入学考试信息

排课管理

研究生
- 排课参数配置
- 开课权限管理
- 开课管理
- 批量开班
- 排课权限管理
- 研究生排课
- 排课汇总
- 排课汇总统计
- 短期课课
- 短期课课审批
- 长期课课
- 长期课课审批

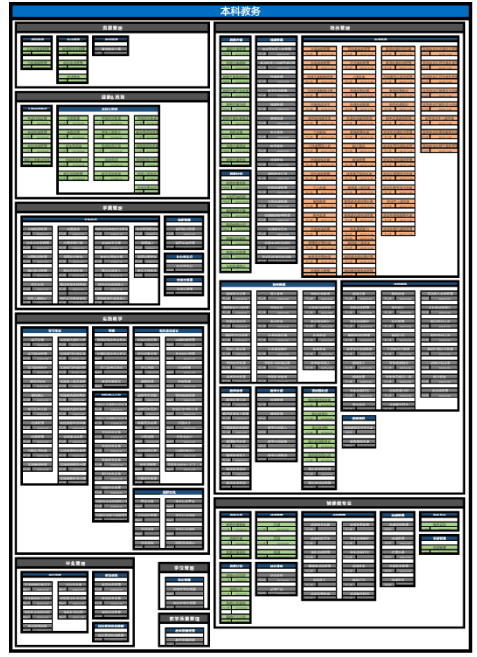
论文选题

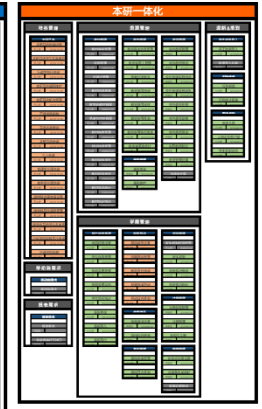
选题设置
- 选题规则设置
- 选题时间设置

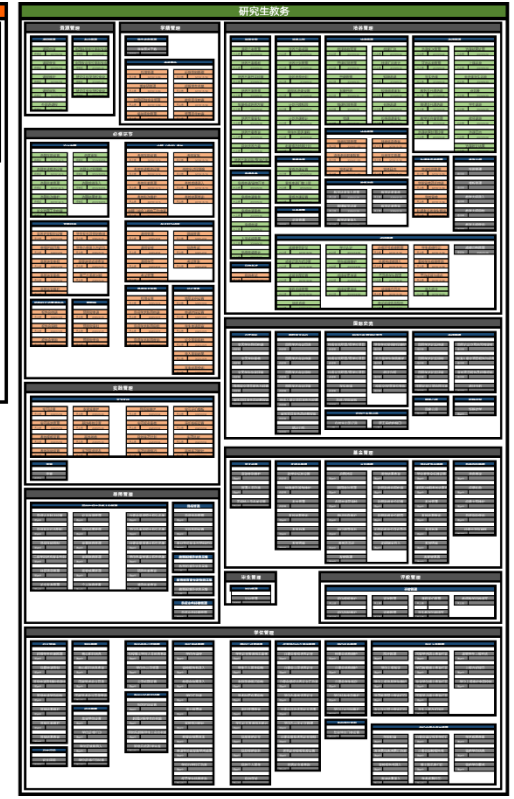
选题申请模板设置
- 选题申请模板列表

- 选题名单管理
- 选题名单管理
- 选题批次维护
- 论文选题工作申请

选题审批
- 选题全过程画像
- 选题成绩录入
- 选题结果查询

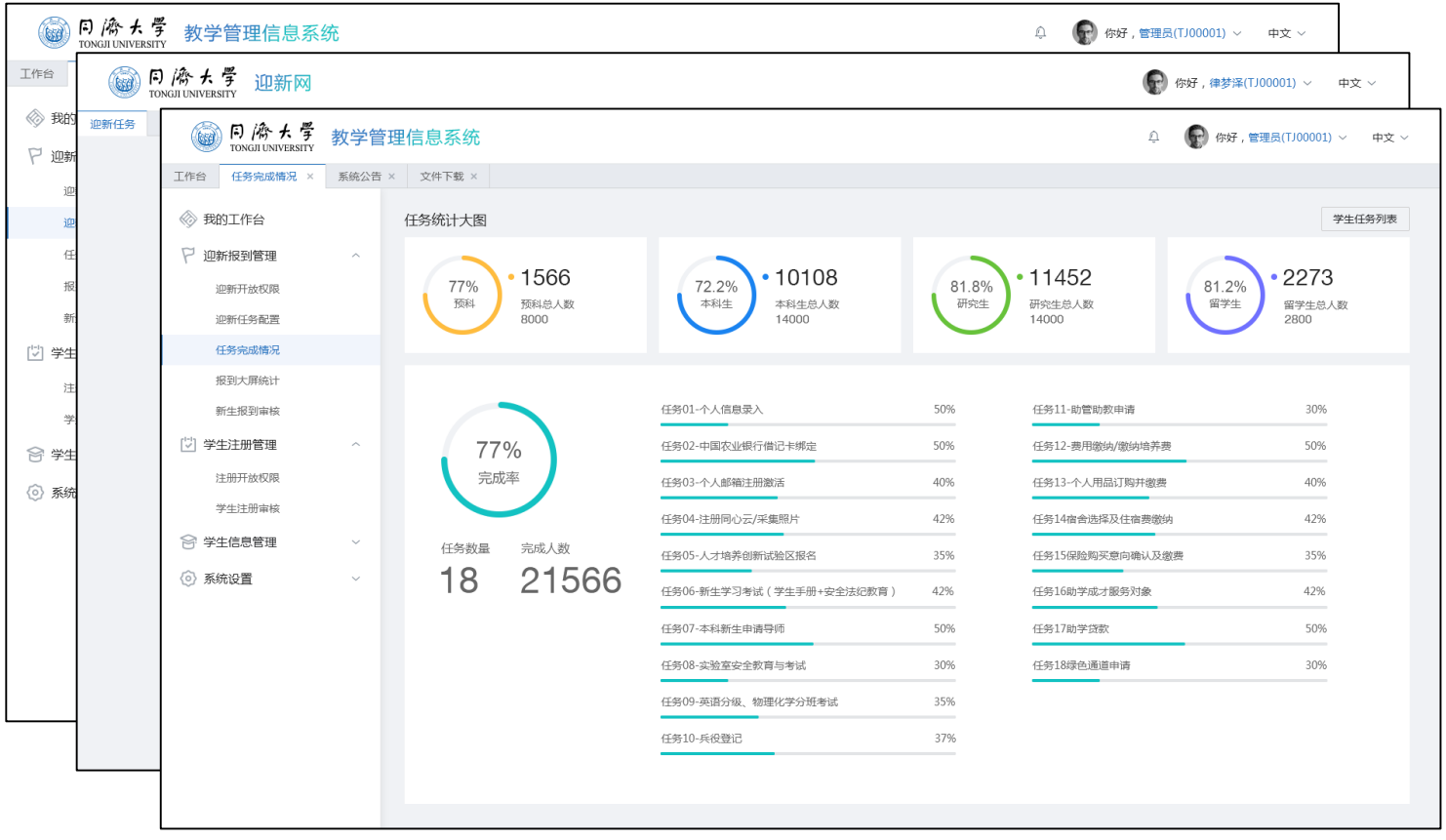






同济大学教学管理系统 - 迎新任务

迎新报到服务：新生入学前需完成迎新任务



迎新任务统计

同济大学教学管理系统 － 新生报到

迎新报到服务：新生入学现场报到

实时大屏展示

TONGJI
UNIVERSITY

感谢倾听，谢谢！