

梦坊国际教育

SAAS 多租户站群技术方案

文档修订记录

修改日期	修改章节	修改类型	修改描述	修改人	审核人	版本号
2014.06.05		A		卢海伟、毛伟		0.1
2014.06.18		M		毛伟		0.2

修改类型分为 **A** – ADDED（增加） **M** – MODIFIED（修改） **D** – DELETED（删除）

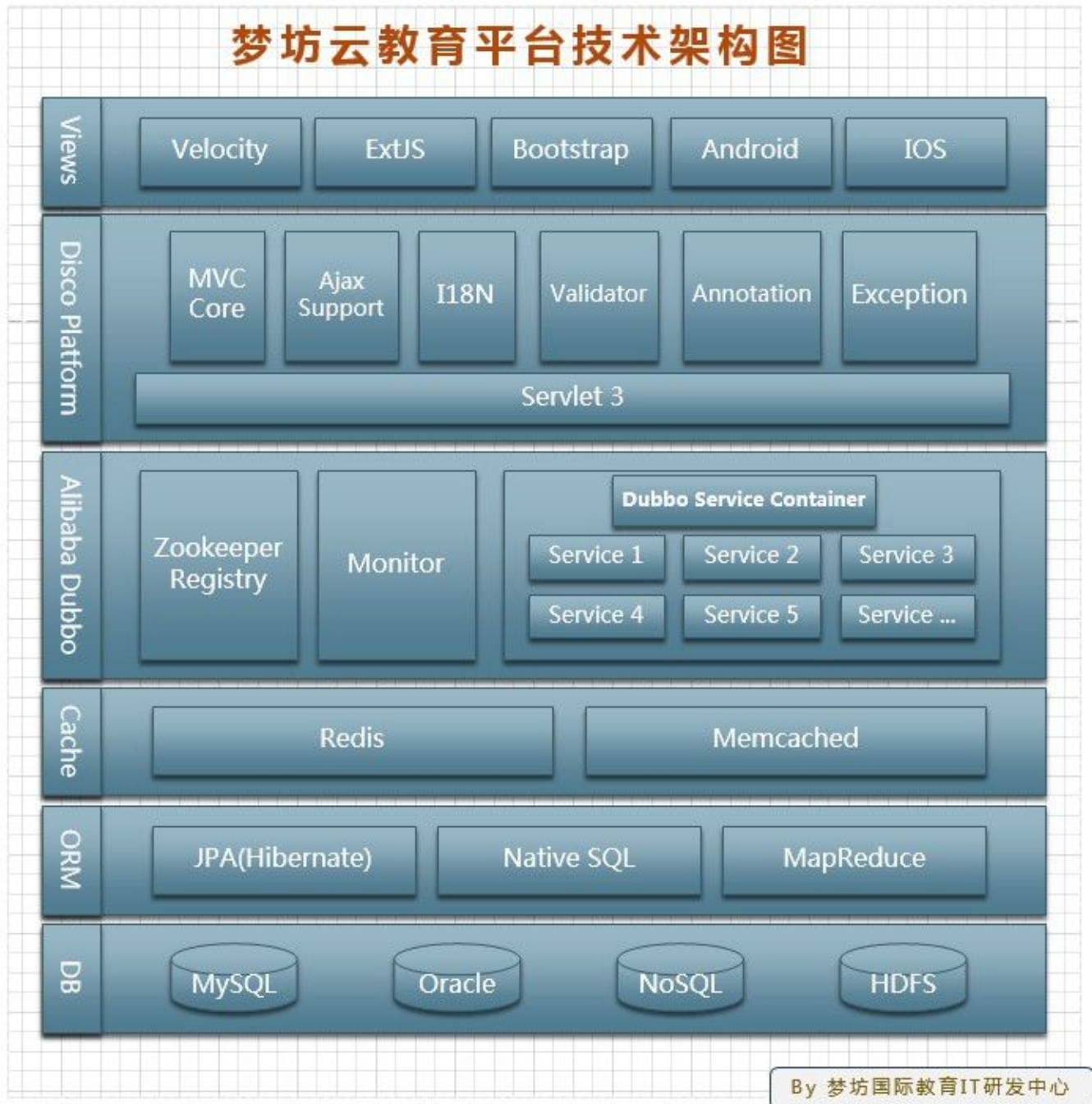
目 录

1 引言.....	1
2 技术架构体系.....	2
3 平台服务部署方案.....	3
4 数据隔离方案.....	4
4.1 结构化数据隔离.....	4
4.2 文件隔离.....	5
4.3 缓存隔离.....	7
5 资源请求方案.....	8
5.1 系统资源 URI 结构说明.....	8
5.2 资源分类.....	8
5.3 非受控资源： 代码、公共 JS、图片等。	9
5.4 受控资源.....	10
5.5 集成资源： 自媒体、活动大赛、网校、班级圈等.....	10
5.6 开发视图支持.....	10
6 配置文件管理方案.....	11
7 系统初始化方案.....	11
7.1 系统部署.....	11
7.2 系统部署步骤说明.....	13
7.3 租户开通（初始化）	13

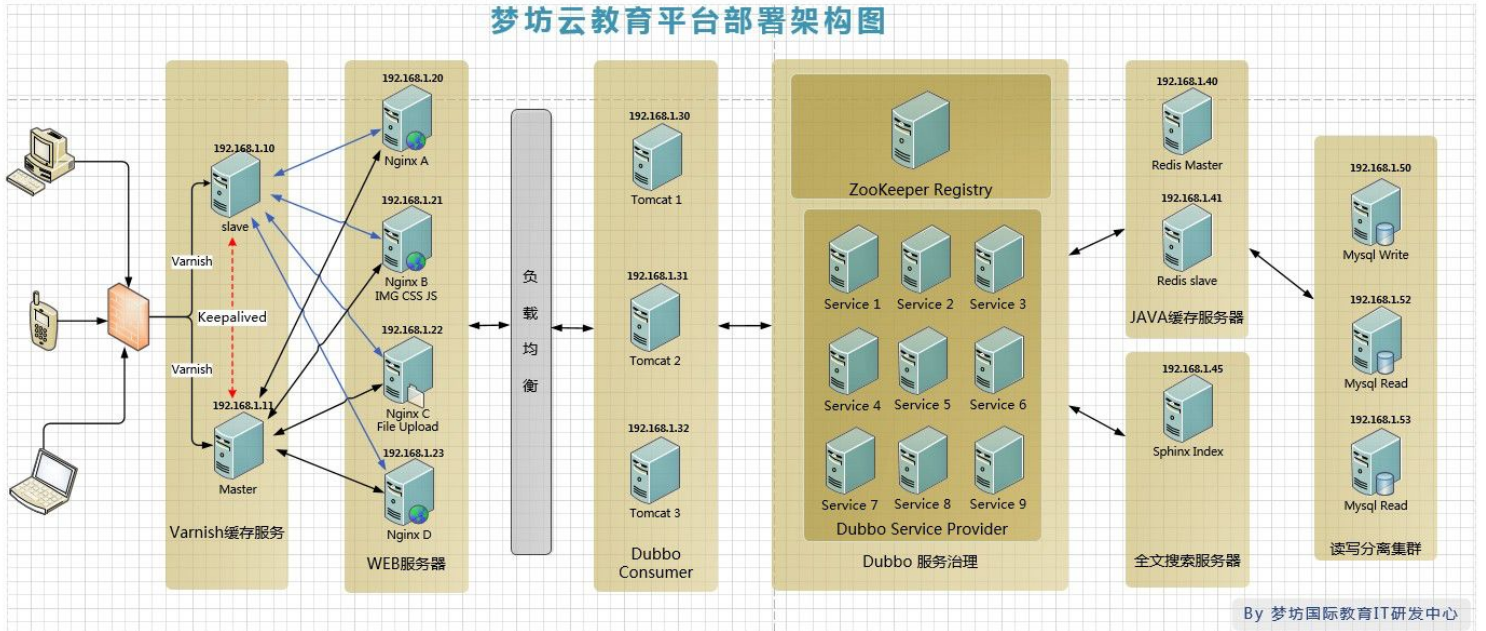
1 引言

梦坊国际教育的产品架构采用了基于单一共享应用程序实例的多租户模型,本文主要说明了在这种模式中多租户资源共享与隔离机制。并针对对资源分类的基础上,描述了不同资源的请求方案。

2 技术架构体系

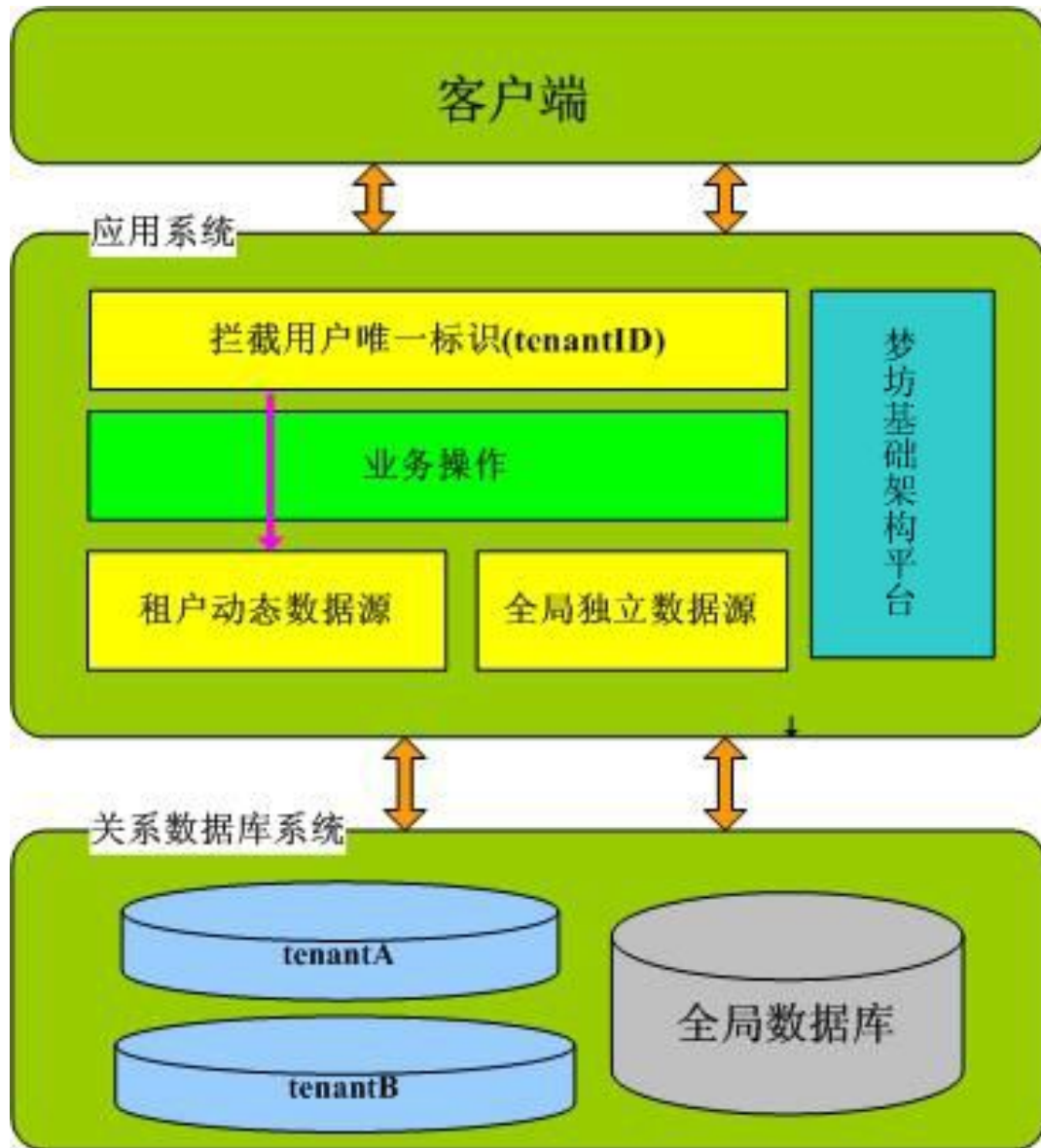


3 平台服务部署方案



4 数据隔离方案

4.1 结构化数据隔离

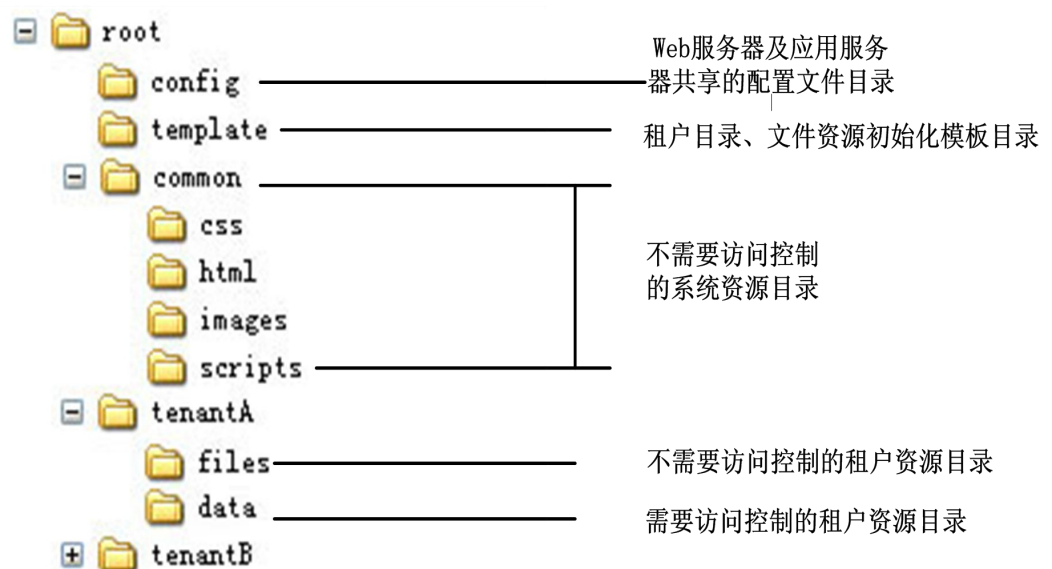


如上图所示,数据库的多租户支持采用数据库模式隔离的方式,每个租户具备自己独立的数据库模式,一个租户不能访问其它租户数据库模式中的数据,它们的数据是严格隔离的。用户成功登录后,持久化过滤器会根据请求的域名确定租户 ID,并保存为当前处理线程的

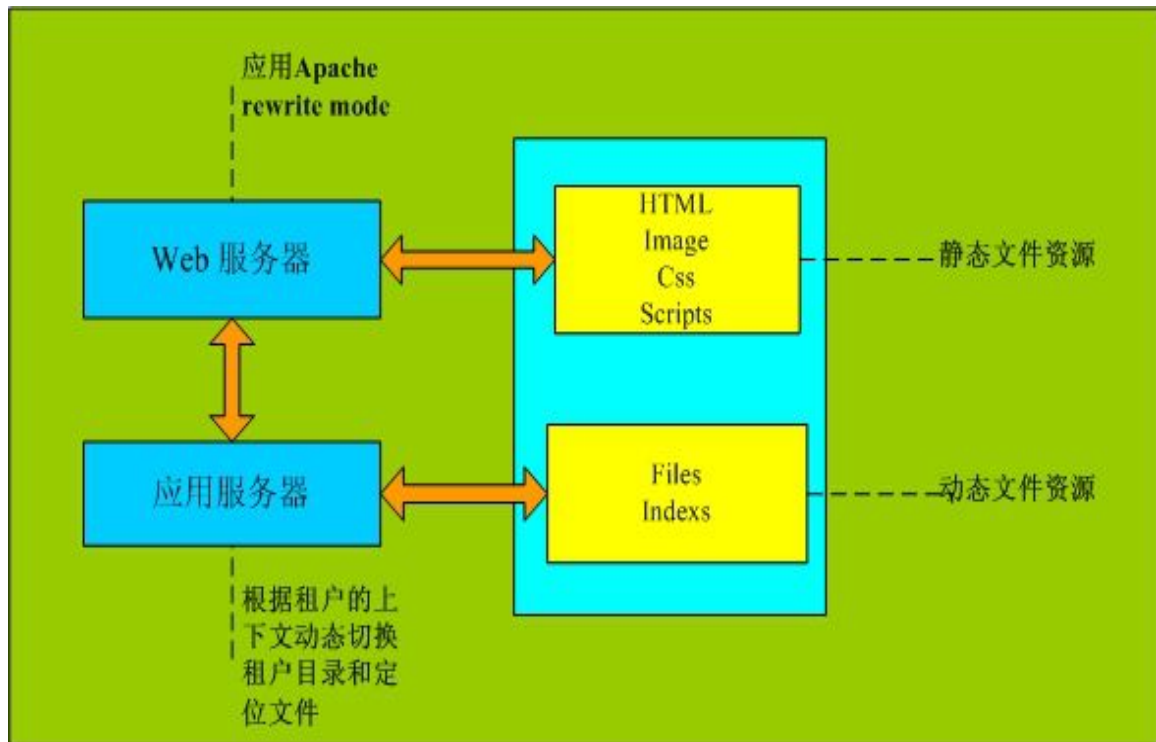
线程变量。当通过租户动态数据源访问数据库时，在返回数据库链接前，将根据当前线程变量的值切换所访问的数据库模式。系统中非动态数据源将不做这种切换，全局独立数据源将固定访问系统的全局数据库。

4.2 文件隔离

系统中多租户文件资源的隔离策略是采用目录隔离方式。如下图所示：



系统目录分为配置文件目录（config）、模板文件(template)目录及资源文件目录，配置文件目录放置 vhost.properties 文件，模板文件目录放置租户初始化的目录结构模板及文件资源。对资源目录而言，从控制属性上可以划分两类资源：有访问控制的资源及无访问控制的资源；从资源隔离角度可以分为：系统级资源（对所有租户都可见）及租户级资源（只对某个租户可见）。根据文件的资源属性及类别不同，系统将采用不同的访问策略，具体说明如下：



对不需要访问控制的文件资源，系统将通过前端的 Web 服务器直接提供，而需要访问控制的文件资源则由后端应用服务器来提供。具体策略是：web 服务器接收到资源请求后，根据资源类别来判定该资源是否需要访问控制，如果不需要控制，就将文件资源映射成本地文件，并直接返回给客户端。否则，就将请求代理到后台应用服务器来处理。

系统中不需要访问控制的资源在 URI 编码中需要遵循前缀规范：

scripts/* 系统级脚本文件；

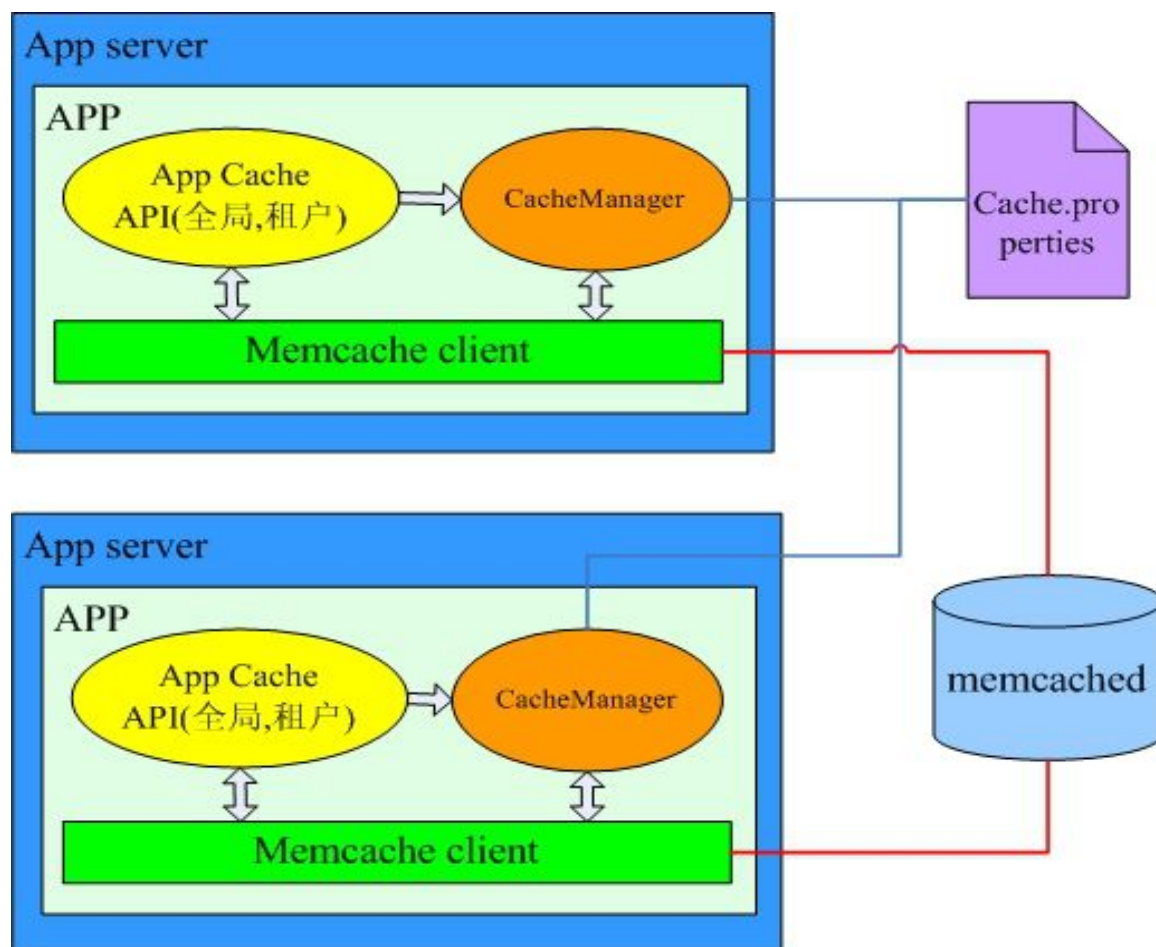
pages/* 系统级静态页面文件；

css/* 系统级样式文件；

images/* 系统级图片资源；

files/* 租户级资源文件。

4.3 缓存隔离



缓存隔离主要解决多租户缓存在集中式分布缓存中的数据隔离问题。具体策略是采用键值空间隔离来实现数据隔离。由于缓存中数据的访问方式是基于键值查找的，视图上可以理解为一个 HashMap，所以只要不同租户的数据在键值空间上是隔离的，就能够实现他们数据的隔离。

根据数据归属地不同，通用分为系统级缓存数据（对所有租户都可见）及租户级缓存数据（只对某个租户可见）。这两类数据缓存键值的编码方式是：系统级数据的键值以 common 开头，租户级数据的键值以租户 ID 开头。系统实现上封装了系统数据访问接口及租户数据访问接口，用户直接调用即可完成相应数据的访问。

5 资源请求方案

5.1 系统资源 URI 结构说明

URI 构成：

http(https)://域名/应用名/资源名

域名：为租户分配的产品域名，具体规则如下

- 1、对于梦坊国际教育的平台系统资源：租户名.梦坊国际教育的平台域名（例如 tenantA.sg.com）；
- 2、对于集成系统资源：租户名.集成产品名.梦坊国际教育的平台域名（例如：tenantA.ip.sg.com）。

应用名：系统部署的应用名（例如：kmengfangdm/restful）。

资源名：1、系统级文件：目录文件名（images/*；scripts/*；pages/*；css/*）。

2、租户级文件：目录文件名（files/*）

3、服务：服务单元名/操作名（例如：order/list）。

上述 URI 编码方式是为了区分资源的不同属性，例如：资源的系统归属（梦坊国际教育的平台或 IP）、范围归属（租户或系统）以及资源类别等，并以此驱动系统的资源定位。

5.2 资源分类

访问方式上可将系统资源分为三类：

非受控资源：不需要访问控制的资源，有前端 Web 服务器直接返回给客户端；

受控资源：由应用服务器返回，并进行访问控制；

集成资源：集成的第三方产品上的资源。

其中对非受控资源由分成两子类：

非受控公共资源：存储在系统级文件目录里 (common)；

非受控租户资源：存储在租户级文件目录里 (例如 tenantA)。

具体的处理策略如下所述。

5.3 非受控资源：代码、公共 JS、图片等。

✧ 非受控公共资源：系统级页面、脚本、样式及图片。

这类资源主要存放在系统目录：

common\pages;

common\scripts;

common\css;

common\images 。

URI 编码方案为：http(https)://域名/应用名/目录文件名。

遇到 URI 标识中目录文件名以[html/ ; scripts/ ; css/ ; images/]为前缀的资源时，

前端 Web 服务器将其映射成系统目录中本地文件直接返回。

✧ 非受控租户资源：租户级页面、脚本、样式及图片。

这类资源包括自定义表单 JS 文件，租户标识图片等

这类资源主要存放在租户目录中，例如：

tenantA\files

URI 编码方案为：http(https)://域名/应用名/目录文件名。

遇到 URI 标识中目录文件名以[files/]为前缀的资源时，前端 Web 服务器将其映

射成租户目录中的本地文件直接返回。

5.4 受控资源

除上述描述的 URI 之外，其它 URI 都将直接代理到后台应用服务器处理，它们表示了一种受控的资源

前面描述的三种资源主要是对应用系统而言，不包括第三方的产品而言。在客户端编码是直接利用相对 URI 访问就可。

5.5 集成资源：自媒体、活动大赛、网校、班级圈等

对于集成的第三方产品而言，由于它们采用不同的域名访问。所以不能采用相对 URI 的访问策略，必须拼接完整的 URI。具体业务规范等业务平台需求切入进来再进一步完善。

5.6 开发视图支持

由于根据访问资源属性的不同开发人员需要根据需要拼接相应的 URI。而这种拼接可能发生在客户端，也可能发生在服务器端，所以在框架层面给出相应支持。

在客户端提供一个单例实现，包含：

- 1、获取产品访问域名的方法（例如：`get_domain_name(im)`返回访问 IM 系统的域名）；
- 2、获得部署应用名的方法（例如：`get_application_context()`返回“`kmengfangdm/restful`”）；

用户用这些方法完成访问资源 URI 的拼接。

在服务器端需要提供相同的支持，以完成服务器端的 URI 拼接。

由于根据运营、部署情况的不同，上述 URI 中的构成因子可能会发生变化。通过上述

方案，就可以将这种变化封装起来，开发人员代码只依赖与这些接口。而系统可以在底层维护这些可变因子，当变化产生时，不会影响到已经建设的代码。

具体的接口，当确定后将补充到开发规范中。

6 配置文件管理方案

配置文件从影响到模块范围分可以分为全局与服务单元两个级别，前者影响对整体应用都起作用，后者只对具体服务单元有效。

配置文件从可见性分为系统级及租户级两种：前者对整个系统（所有租户）都是可见的，后者只对具体租户可见。

配置文件的可编辑期可以分为：开发期、部署期及运行期。

系统配置的具体描述请见附件文件《配置文件说明.xls》。

7 系统初始化方案

7.1 系统部署

梦坊国际教育的平台系统采用分布式部署结构，Web 服务器、应用服务器及数据库服务器均采用集群方式。系统程序代码、资源文件以及数据文件将部署在 NAS 目录中，顶级目录划分为：

deploy：程序代码目录，部署应用的 war 包文件；

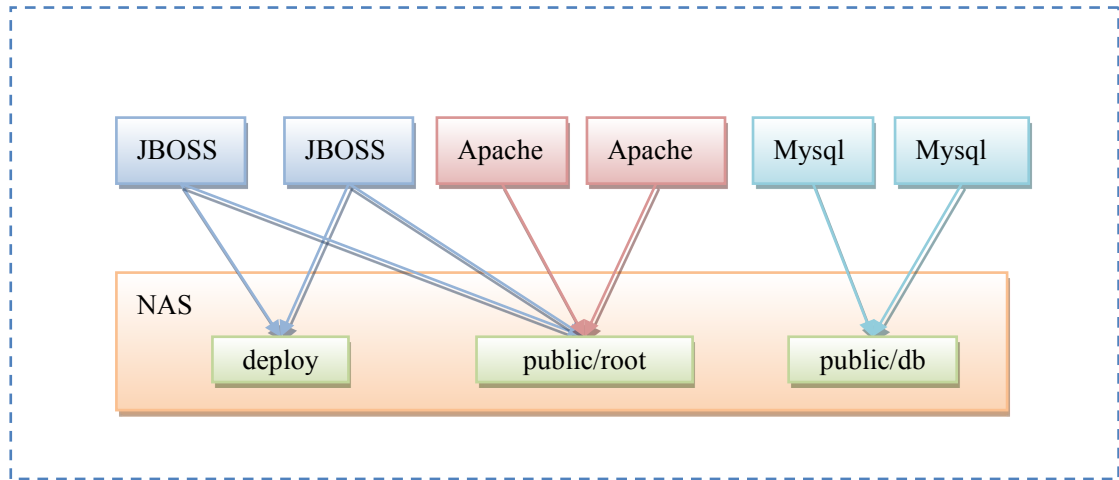
public/root：资源及数据文件目录；

public/db：数据库文件目录；

其中 deploy 目录将挂接到集群中所有应用服务器节点上，public/root 目录将挂接到

集群中所有 web 服务器及应用服务器节点中，而 public/db 目录将挂接到集群中所有数据

库服务器节点中。如下图所示：



NAS 目录说明：

deploy

-War (Web 应用程序)

public/root

-config (公共配置文件夹)

-vhost.properties

-dmap.perperities

-template (租户资源模板文件夹)

-...

-common (公共资源文件夹)

-pages/

-scripts/

-images/

-css/

-TA (租户文件夹)

-files/

-data/

-TN (租户文件夹)

-...

public/db

-data (数据库文件夹)

7.2 系统部署步骤说明

1、 按照 web 服务器集群、应用服务器集群及数据库服务器集群，安装 NAS 并按照上述方案，创建并挂接 NAS 目录。

2、 向 NAS 目录中拷贝相应资源 (war -> deploy , 公共资源 -> public/common , 共享配置文件 -> public/root/config , 租户初始化目录模板及文件资源 -> public/template)

3、 修改部署期需要变更的配置文件。

最后就是讲应用挂接到应用服务器节点的 Jboss 上。

7.3 租户开通 (初始化)

系统部署并运行后，当开通租户时运营平台发送指令到后端 ERP 梦坊国际教育平台，部署在梦坊国际教育平台的运营嵌入模块接受到指令后，调用持久引擎，持久引擎创建租户

相关数据库、文件、配置文件等，会完成下列动作：

主要有数据库初始化与文件初始化两部分组成。

数据库初始化过程中，需要根据开通租户的语言选择初始化相应语言的字典表。整个初始化过程会分成结构初始化与数据初始化两个部分，前者主要是执行系统结构的 DDL 语句完成数据库表结构的初始化，后者则需要根据选择租户选择语言的不同选择相应语言的 DML 语句完成表中数据的初始化，具体描述如下：

- 1、读取两个 sql 脚本文件，文件的位置可通过 conf_directory.properties 文件配置；
- 2、执行数据库及表的定义脚本：db_ddl.sql；
- 3、根据租户的语言选择，执行数据初始化脚本，文件名称定义为：

db_dml_<language>.sql

其中<language>为语言编码，英文为：en 中文为：dn

例如：语言为英文的数据初始化脚本文件为：db_dml_en.sql.

ddl 文件名称为:db_ddl.sql

dml 文件名称为:db_dml_<langauge>.sql 比如：db_dml_en.sql

在执行 dml 脚本时，需要分析替换脚本中的 uuid 后执行。

文件初始化同样分为结构初始化与数据初始化大部分，前者会为开通租户拷贝初始的目录结构，后者需要为租户拷贝初始化资源，其中需要根据租户的语言选择拷贝不同的语言文件，并且需要根据用户运营参数的选择（例如：租户配额），更新相应的配置文件，具体说明如下：

- 1、初始化租户目录，为租户拷贝文件目录结构，板的根目录可在 conf_directory.properties 中配置；

2、初始化配置文件，主要有两个文件: vhost.properties 及 dsmap.properties，用于多租户路由的配置信息记录。文件名的存放位置可通过 conf_directory.properties 文件配置；

3、初始化配额文件：

数据库和文件初始配额记录在一个文件中：租户名.properties

租户文件已用空间 文件名：租户名_file.properties

租户数据库已用空间 文件：租户名_db.properties

文件名的存放位置可通过 conf_directory.properties 文件配置,