

方正飞鸿云构件业务开发平台 技术白皮书

Founder Fix BPMCS

方正国际软件有限公司



1	目录	
2	概述	4
2.1	什么是 Fix BPMCS	4
2.2	软件公司会遇到的问题	4
2.2.1	大中型企业应用的开发	4
2.2.2	公司的技术积累	4
2.3	Fix BPMCS 客户价值	5
2.3.1	管控能力	5
2.3.2	运营能力	5
3	Fix BPMCS 的特点	6
3.1	集成的、可编译、可调试的开发环境	6
3.2	支持标准 BPMN2.0 的工作流体系	6
3.3	丰富的 Ajax 控件, 实现与 C/S 应用相媲美的高交互 WEB 应用	6
3.4	具有丰富的样板程序, 能有效降低开发者学习曲线	6
3.5	优秀的性能体现满足复杂的运行环境	6
3.6	完整的业务基础框架, 完善的组织架构及权限体系	6
4	Fix BPMCS 系统架构	6
4.1	开发运行环境	7
4.1.1	硬件	7
4.1.2	开发技术标准	7
4.1.3	软件	7
4.2	应用系统运行环境	7
4.3	关键技术	8
5	Fix BPMCS 的主要功能	9
5.1	复合式 Web 应用	9
5.2	建模设计	9
5.2.1	业务对象建模	9
5.2.2	服务建模	9
5.2.3	业务视图建模	10
5.2.4	业务流程建模	11
5.3	设计器	12
5.3.1	平台功能分类管理	12
5.3.2	团队协作开发管理	12
5.3.3	建模工具	13
5.3.4	表单设计工具	14
5.3.5	工作流设计工具	15
5.4	流程引擎	15
5.5	安全、权限	16
5.5.1	功能权限	16
5.5.2	数据权限	16
5.5.3	字段权限	17
5.6	缓存体系	17
5.7	集群支持	17
5.7.1	Session 策略	18



5.7.2 高效率.....18



1. 概述

2.1. 什么是 Fix BPMCS

Fix BPMCS 是集技术开发平台、业务流程平台、灵动应用平台为一体的 BPM 软件开发平台。旨在为企业提供强大的信息化系统基础架构，增强各系统之间的集成性，大幅缩减软件项目的开发成本。

平台其中包括：应用开发框架、Web 应用引擎、流程引擎、建模设计工具等，它综合了 SOA 体系架构和 BPMN2.0 国际标准，为提升开发效率、降低维护复杂度、规范开发过程打下了结实的基础。主要定位于企业管理软件的快速实现。

Fix BPMCS 平台继承了 Fix BPMES 平台快速灵活的特点，并在此基础上增强了系统的扩展性和灵活性，支持各种集群、多数据库以及多 Web 服务器的部署环境。设计器基于 Eclipse 平台图形化开发，方便了代码的编写与调试。

使用其的目的是通过应用服务组件化的开发模式，让快速开发及部署的理念融入到每个企业，让本软件安装到每个程序员的桌面并为企业服务，使软件开发的速度更快、成本更低、门槛更低、让企业信息系统更简单，进而推动企业信息化进程。

2.2. 软件公司会遇到的问题

2.2.1. 大中型企业应用的开发

- 功能点多而复杂，难以管理与复用
- 开发维护周期长，人员变动，新人熟悉环境成本高
- 前端页面开发费时而又难统一
- 性能要求高，需要完善的缓存与集群方案
- 多系统集成

2.2.2. 公司的技术积累

- 没有统一的技术平台，各个项目平台孤立，项目最终的技术水平参差不齐。



- 项目的维护严重依赖成员对特定框架的熟悉程度，往往 A 项目里面的 a 程序员无法维护 B 项目的功能

2.3. Fix BPMCS 客户价值

2.3.1. 管控能力

- 稳健的系统框架，免去搭建新框架的风险
- 众多的既有功能和模块，大大提高开发效率
- 规范的开发模式，保证应用开发过程统一
- 完善的学习文档，更快的新人上手时间
- 强大前端组件框架，快速拖拽配置开发
- 成熟的缓存及集群解决方案

2.3.2. 运营能力

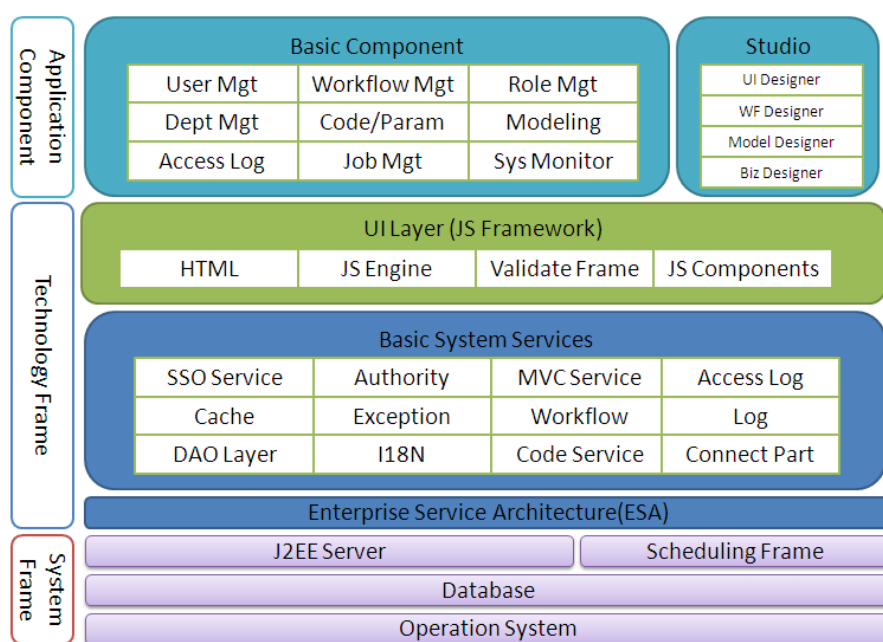
- 统一的技术框架，公司各个项目在技术层面都能保持较高的水准
- 机动性更强的开发团队，任何开发人员只要熟悉了业务，基于平台的各个项目都能采用标准化的实现方法进行开发和维护



2. Fix BPMCS 的特点

- 3.1. 集成的、可编译、可调试的开发环境
- 3.2. 支持标准 BPMN2.0 的工作流体系
- 3.3. 丰富的 Ajax 控件，实现与 C/S 应用相媲美的高交互 WEB 应用
- 3.4. 具有丰富的样板程序，能有效降低开发者学习曲线
- 3.5. 优秀的性能体现满足复杂的运行环境
- 3.6. 完整的业务基础框架，完善的组织架构及权限体系

3. Fix BPMCS 系统架构



4.1. 开发运行环境

4.1.1. 硬件

Fix BPMCS 基于 Eclipse Helios 平台开发，推荐以下配置

- Inter(R) Core(TM)2 Duo CPU
- 4G 系统内存
- 700M 可用空间

4.1.2. 开发技术标准

- BPMN2.0
- JavaEE
- AJAX
- Spring
- OGNL
- IBatis
- JQuery
- ExtJS

4.1.3. 软件

- windows 操作系统（xp,2003,vista,win7）
- 数据库服务器，（sqlserver 2005+， oracle 9i+）

4.2. 应用系统运行环境

Fix BPMCS 平台支持广泛的操作系统和应用技术，构建在平台上的业务系统具有最广泛的技术兼容性和可扩展性：

- 操作系统支持
 - ◇ 支持网络操作系统：NT，UNIX，Linux 等。



- ✧ 支持客户端：WindowsXP 及更高版本，浏览器 Chrome 10、Firefox 6、IE8.0 及更高版本。
- 数据库支持
 - ✧ 支持如下数据库接口：ODBC，JDBC。
 - ✧ 支持 Oracle 9i +、SQL Server 2005+、Mysql 5+、Db2 9.1+。
- 分布计算和服务支持
 - ✧ 支持如下分布计算和服务技术：DCOM，EJB（J2EE）。
 - ✧ 支持如下分布计算支撑环境：BEA WebLogic、IBM WebSphere 等。
- Web 服务支持
 - ✧ 支持如下 Web 服务支持技术：Serverlet。
 - ✧ 支持如下主流 Web 服务器产品：Tomcat、BEA WebLogic、IBM WebSphere 等。
- 网络和应用协议
 - ✧ 支持多种应用协议，包括：
 - ✧ 通讯协议：HTTP、TCP/IP
 - ✧ 邮件协议：SMTP，POP3
 - ✧ 应用协议：DCOM、EJBIIOP、EJBRMI

4.3. 关键技术

- 基于动词可执行表达式的面向服务框架(SOA)。
- 基于 Spring 事务机制特别封装的可跨数据源服务级声明式事务。
- 基于业务对象模型的模块。
- 高适应性的权限结构包括功能权限、数据权限、字段权限，分别控制功能点级、行级、列级。
- 基于 Eclipse 的 RPC 应用开发
- 丰富的 UI 组件并具有可积累特性，同时组件可以方便集成到平台中
- 附件存储多样化（数据库【分表或分库】、FTP、本地目录）
- 自定义工具栏和功能服务接口
- 可忽略上下文环境服务器执行命令
- 混合式缓存体系，动态生成字节码，提高平台性能
- 流程采用 BPMN2.0 标准,支持全面泳道图展示



4. Fix BPMCS 的主要功能

5.1. 复合式 Web 应用

使用复合式 Web 应用程序是能够将一个主站点拆分为多个独立的模块，每个模块负责主站点中一个目录及其所有内容。多个开发人员分别独立负责不同的模块而不会相互干涉。

- 让业务逻辑开发与 Web 开发相互独立，这样可以集中精力开发特定领域的模块
- 使用一致的框架体系，提高开发出来的产品项目质量
- 通过架构师和开发者的共同努力提高生产力，降低开发时间

5.2. 建模设计

5.2.1 业务对象建模

平台抽象出一套业务系统的模型，使用户通过业务模型完成模块的建立和修改，所有应用的基础就是业务对象，业务对象包括其属性、服务、权限控制、数据权限、关联流程。同时业务对象也有自己的列表和展示页面，可以是一个单表、多表联合查询、存储过程，或者一段程序代码。

5.2.2 服务建模

产品是基于 SOA 的平台，保证了粗颗粒度的业务组合。每个颗粒就是一个服务，所有前台与后台的交互都是服务，所有外部对内部的调用也是服务。

5.2.2.1. 可执行表达式

表达式在组合的时候有一定规则，表达式是由“动词+空格+内容”的公式组成。动词决定了使用的适配器，每个适配器识别的内容各不相同，在无法满足开发需求的情况下也可以自行添加适配器和动词来进行扩展。



5.2.2.2. 服务的作用域

- 业务对象的服务
- 模块的服务
- 工程的服务

5.2.3 业务视图建模

即为数据显示页面业务功能配置，包含字段验证、toolbar 按钮、列表页面定义等管理。

5.2.3.1. 基础验证

主要针对业务对象的验证配置，如长度控制、格式校验(JS、server、common)。

5.2.3.2. 列配置

定义对列表页面显示的字段信息列，每列可配置是否显示、显示格式、统计方式。

5.2.3.3. 关联视图

数据关联显示配置，点击列表上一条数据时，让其显示关联对象的列表或表单。

5.2.3.4. UIScript

处理页面样式改变配置和当前对象页面加载时执行的 JS 脚本代码。

5.2.3.5. Toolbar

toolbar 按钮是数据列表和表单中重要的部分。按钮不但负担起中转的作用，还起到功能权限关联的作用。



5.2.4 业务流程建模

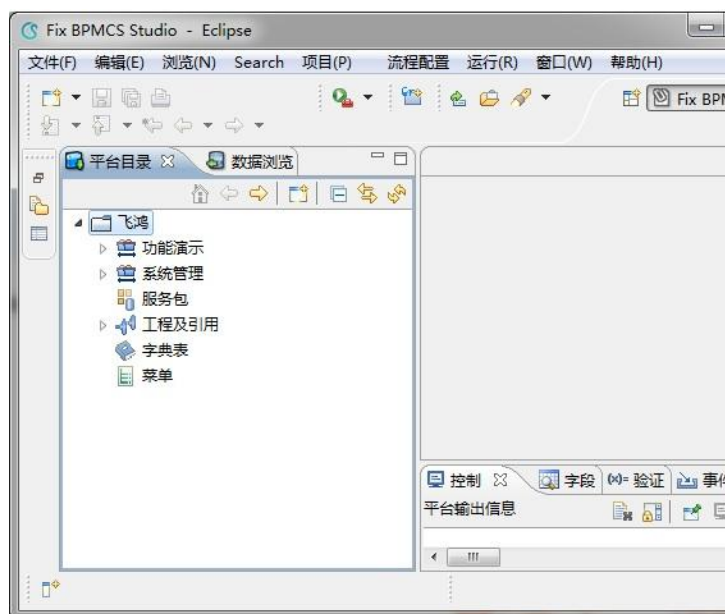
业务流程建模是由一组可视化图形元素构成。这些元素让我们很容易开发一个简单的，为大多数业务分析人员更熟悉的流程图。开发 BPMN 的一个动机就是创建一套简单的机制来创建业务流程模式，与此同时，还要能够应付业务流程内在的复杂性。BPMN 四种基本元素是：

- Flow Objects
- Connecting Objects
- Swimlanes
- Artifacts



5.3. 设计器

5.3.1. 平台功能分类管理



5.3.2. 团队协作开发管理

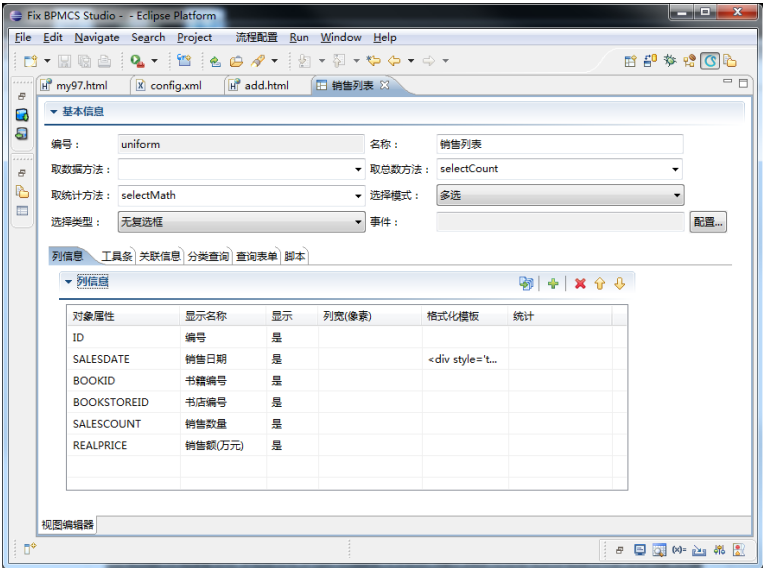
平台集成 SVN 版本控制系统，用 SVN 管理对文件和目录进行检查数据修改的历史，允许文件恢复到早期版本。SVN 可以满足各种企业 VPN 的要求，通过为公司内部网络、远程和移动用户、分支机构和合作伙伴提供基于 Internet 的安全连接。

5.3.3. 建模工具

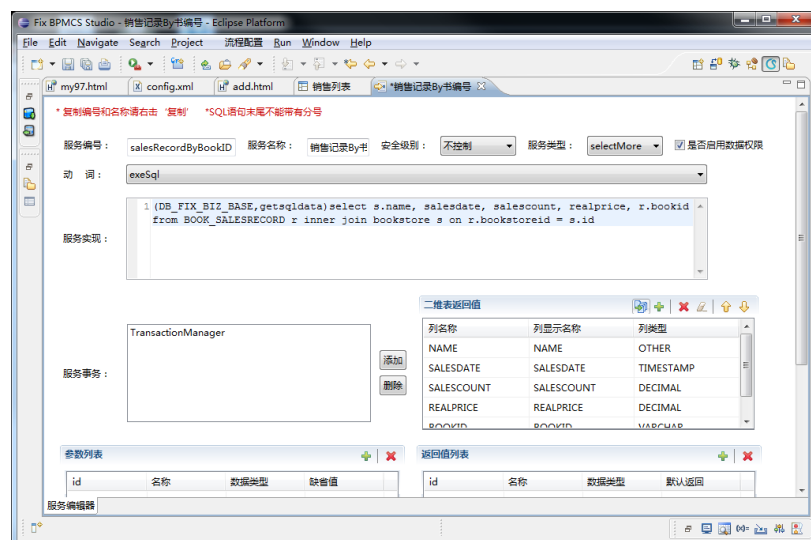
5.3.3.1. 业务对象设计



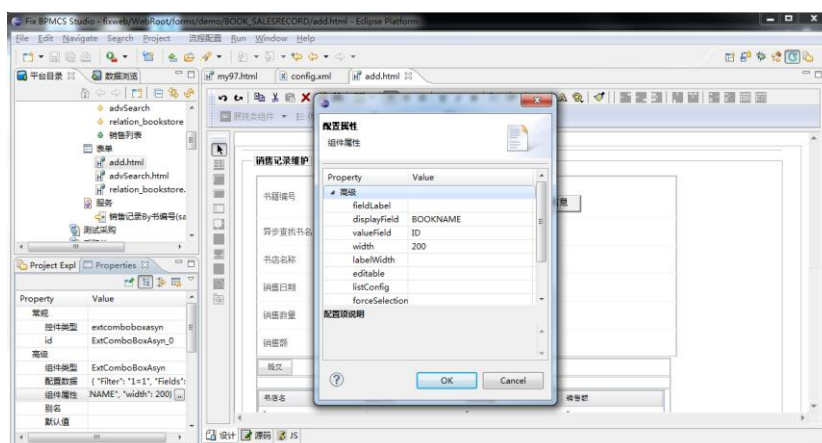
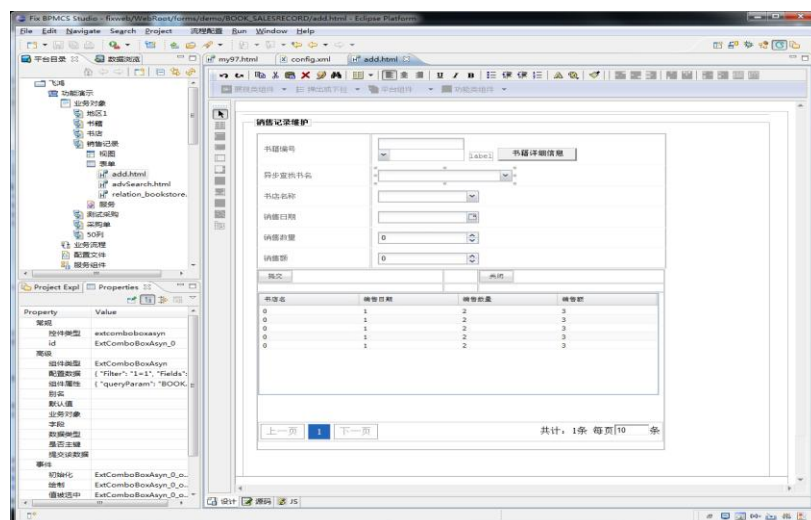
5.3.3.2. 视图建模设计



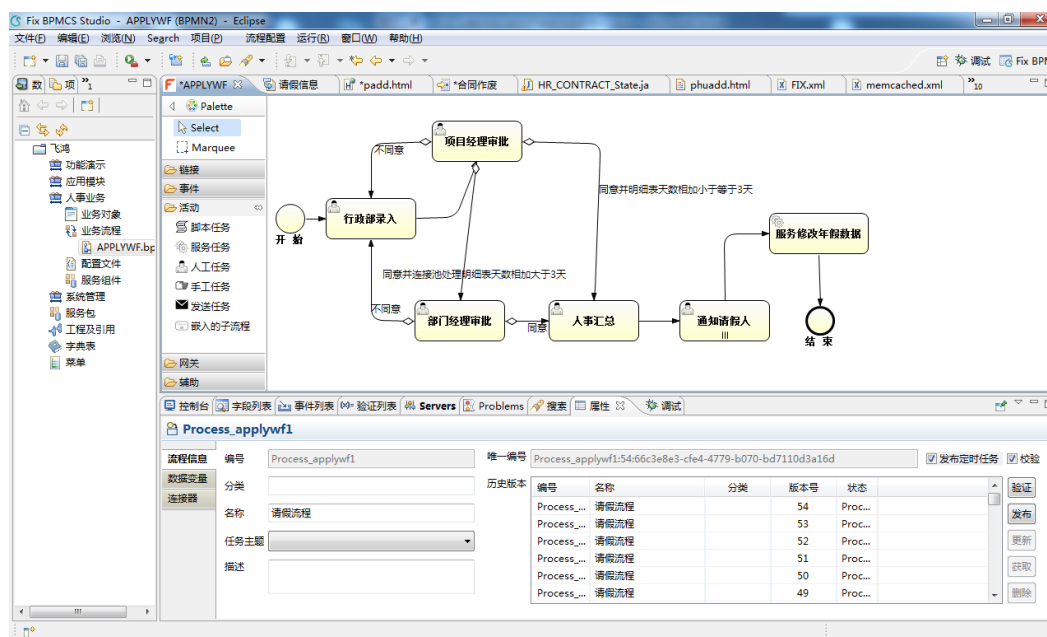
5.3.3.3. 服务建模设计



5.3.4. 表单设计工具



5.3.5. workflow 设计工具



5.4. 流程引擎

业务流程图由一系列的图形化元素组成。这些元素简化了模型的开发，且业务分析者看上去非常熟悉。开发 BPMN 的动力就是为了在创建业务流程模型时提供一个简单的机制，同时又能够处理来自业务流程的复杂性。要处理这两个矛盾的需求的方法就是将标记的图形化方面组织分类为特定的类别。

工作流引擎是与对象模型紧密配合的专用流程引擎。

平台的对象引擎实现了业务信息的表达以及相对简单的独立操作。工作流支持各种模式的流程控制，不仅支持顺序流程的流转，而且还支持分支、并发、循环、子过程等。

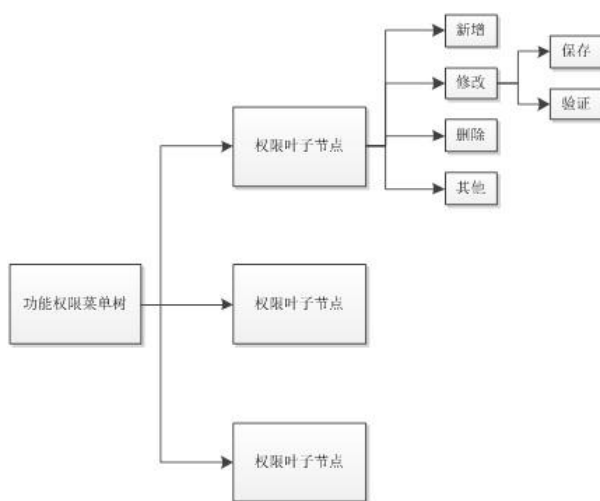
在分支上可以定义条件，实现按条件自动流转，条件转移之间还可设置逻辑关系，同时满足流转条件的分支还可通过人工路由方式决定流转方向；在并发流转中，多个活动节点可以同时激活；在某些活动节点上，也可以通过创建子过程来完成任务。在节点活动中，流程引擎还加入符合中国管理要求的审批模式支持，如单人单审、多人单审、多人并审、多人串审等。流程支持各种干预方式，如回收、回退等。支持各种定时超时任务，如超时发送提醒信息或超时直接转向等流程引擎支持多种参与人方式，不仅支持静态的用户、角色、组织节点，还支持动态的组织角色及基于条件筛选的用户，任务还支持授权委托给用户的代理人进行处理。

5.5. 安全、权限

5.5.1. 功能权限

功能权限是当一个用户登录时，可以访问所拥有的权限对应的功能，同时他所作的操作都会在后台被验证，如果访问无权限的功能，系统后台会拦截，防止连接地址盗用。

5.5.1.1. 结构图



5.5.1.2. 权限树

权限树由菜单树和视图模块构成：

- 菜单树构成：由菜单组成，菜单将组成权限树中到“权限叶子节点”的部分。菜单里 `menuId` 将成为赋权的 `permissionId`。在系统启动时将加载菜单到内存中。同时将所有内容复制一份将其保存到 `permission` 里。
- 视图模块构成：视图模块是平台中用于定义业务对象的增删改查的功能。

5.5.2. 数据权限

数据权限即对数据记录集权限控制，考虑到安全性数据的筛选在后台处理完成。数据权限的核心就是 `sql` 语句注入过滤。

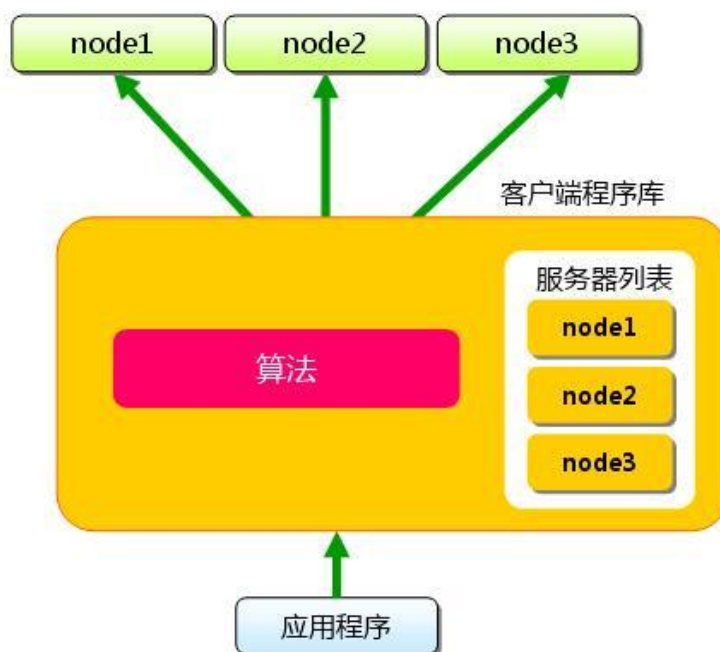
5.5.3. 字段权限

字段权限是对字段进行权限控制，涉及到每行数据的每一列值，如果设置列为敏感字段，只有拥有权限才能看到这列数据，反之在浏览列数据时以“****”代替。

5.6. 缓存体系

平台使用多级缓存，为保证效率的高效和集群环境下进行数据同步，建立二级缓存机制，一级缓存用于虚拟机本地最高效，二级缓存用于数据交换，效率不及一级缓存但是可以保证安全性和同步性。从而保证整个缓存的高效、安全、同步。

缓存是一整套体系，对内完全对自己负责，有自己完成信息同步和存取；对外提供保存和获取接口。



5.7. 集群支持

集群的方式其实就是多个服务器进行负载均衡，其本质上的行为就是请求分发，把大压力多并发分担到多台机器上。支持集群要实现：

- 完善的 session 策略
- 高效率应用执行

5.7.1. Session 策略

Session 的策略包括三部分 session 开始，用户登录，session 销毁：

- Session 开始：session 开始时就在系统里生产该 session 的唯一标识，该标识如果用户登录就作为该用户的单点登录 sso 来使用。
- 用户登录：用户登录时会将用户的所有数据保存在本地 session，同时以 SSOID 为标识在缓存服务器上建立一组对象，这样保证如果一个用户在多个点同时登录时呈可控状态。
- Session 销毁：在 session 结束时固定的销毁在缓存服务器上的数据，以保证不会产生垃圾数据，如果用户没有登录则没有这部分数据。

5.7.2. 高效率

- 无 JSP：我们平台没有 jsp 的存在，后台只依靠一个 servlet 来处理请求，请求的格式要求有固定格式，这样对服务器省下了大量的流量压力和翻译执行 JSP 的时间。
- 高效率反射：对于可配置的框架来说，反射几乎是不可或缺的代码。但是在中枢部位大量使用反射是不高效的行为，所以我们使用 CGLIB 的方式创建动态代码，这样相对于反射有很大的性能提高。
- 运行消耗的转移：对于运行来说有时候消耗是不可避免的，这时候我们的做法是把消耗转移，比如在启动时完成各种初始化来保证其运行时效率。
- 大量运用缓存：权限、定义数据等等在运行时不会频繁更改的内容都会以缓存的方式存在，同时缓存接口对外开发，业务数据也可以使用缓存来优化自己的功能。

