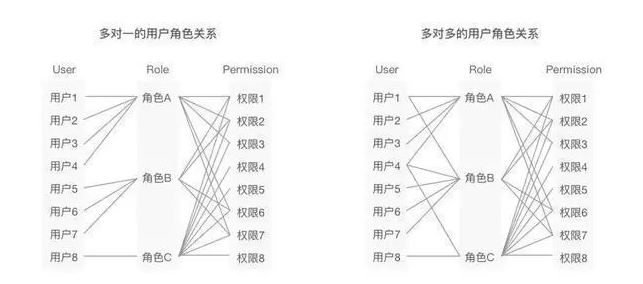
**中浙高铁系统权限功能的设计**

RBAC（Role-Based Access Control ）基于角色的访问控制。

1. **角色**

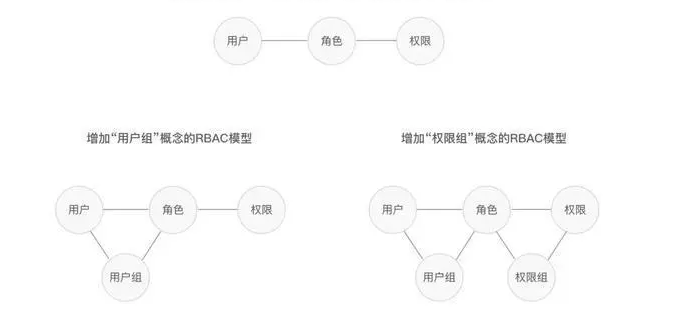
用户与角色可为多对一或多对多的关系，当一个用户的角色为多对多时，当前用户的权限是多个角色的并集。此时只需要为角色赋予权限，能够大大减轻管理负担，同时将用户与权限解耦，提供更大的灵活性，同时提高了整个系统设计的容错能力。



1. **引入用户组**

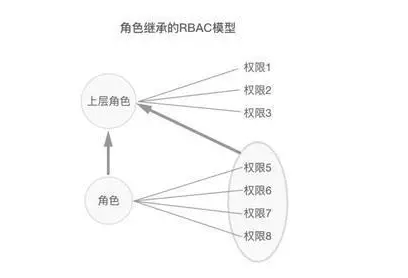
作为大型的平台，如果用户数量较大，新增角色时，需要为大量用户分配新的角色，工作量巨大，此时可以引入用户组的概念，将这些用户拉到同一个用户组中，然后对整个用户组进行角色的指定，这就大大减少了角色分配的工作量。

同理如果权限较多时也会存在一样的问题，对角色进行权限设置时也需要大量的操作，此时可以考虑引入权限组的概念，将关联性较强的权限大包成组赋予角色，从而减少赋值时的工作量，现实中权限组的使用相对较少，因为系统中的权限一般来讲是有限的。需要注意的是即使有用户组或权限组的存在，也可以允许用户或权限与角色直接关联，这个可以视具体业务情况而定。

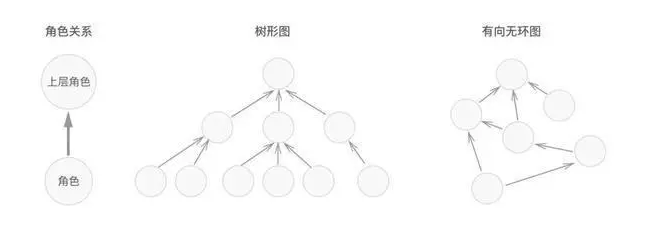


1. **角色继承的RBAC模型**

在一个业务场景中，如果角色需区分：设计主管、设计组长、设计成员，并且管理方式为向下兼容时，则需使用角色继承的RBAC模型。上层角色继承下层角色的全部权限，且可额外赋予权限（临时代理权限等）。



此时除了对角色进行定义，还需要管理角色间的关系，通过关系来体现角色的层级关系，从而达到继承权限的效果。角色的继承关系主要有两种：树形图和有向无环图。



继承关系常常来源于内部组织结构，此时常将角色与组织结构进行关联达到继承角色模型的效果。如赵先生，其角色是“三级团队负责人”，与其并列的小组中有多个“三级团队负责人”的角色，但依附于左侧的组织结构树，各级负责人仅有查看和操作自己下属子节点的权限。

1. **限制的RBAC模型**

在中浙高铁系统中，部分角色可能是需要隔离的、不允许被同时赋予一个人的。跟大家熟知的“不能既是‘运动员’又是‘裁判员’ ”一个道理。

因此，对于众多角色中的一组，只能是单选的关系，但多组角色之间可以共同存在。如下图中，一个用户可以既为设计师又为管理员，但在设计师角色组中仅能被赋予一个角色，在管理员角色组中也仅能被赋予一个角色。



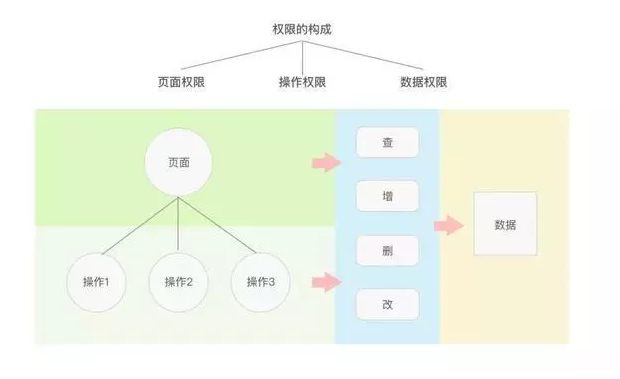
此外，限制还有可能是数量上的，比如一个产品组中必须有且只有一个管理员，不允许删除或再分配管理员角色，仅允许将负责人角色变更。



1. **权限的拆分与设计**

通过RBAC模型已经能够很好的搭建起用户、角色与权限之间的关系了。但具体是什么样的关系，以及“权限”这个抽象的概念具体如何规划？这些都需要分析清楚才能进一步设计出完善的权限系统。

首先需要知道，一般系统的权限由页面、操作和数据构成。页面与操作相互关联，必须拥有页面权限，才能分配该页面下对应的操作权限。数据可被增删改查。



因此，在设计之初我们就需要考虑到未来可能区分角色的地方，尽量解耦、模块化。对于技术来说，每一个页面模块、每一个操作都最好使用独立的接口。对于设计来说，需要保障所有角色因为权限而屏蔽掉部分操作和数据后，页面和流程仍能体验流畅。保证初期设计支持后，配置权限时，还需要注意以下几点：

1. **确定是否支持前端配置**

如果角色和权限相对固定，则一般将角色与权限的关系可以写在后台，改动时需要后端变更且重新上线。这种情况适用于公司内部系统等只有一个使用主体的系统。

如果需要自定义角色或者每个角色在不同使用者的场景下有不同的权限，则需要将角色的定义、角色与权限之间的配置体现在“前端用户配置页面”。这种情况适用于有频繁变动的自定义角色权限，和有租户体系的B2B系统。

1. **以基本单元拆分，以业务逻辑配置**

一般可将每个对象的“增、删、改、查”各自作为一个基本的权限单元。如在“人员管理”中，查看人员列表、添加人员、删除人员、编辑人员信息最好拆分为4个权限单元。在技术和设计上，能尽量做到解耦和模块化。

但是在业务层面有些操作却是一体的。这些不能拆开的权限在“前端用户配置页面”中建议打包成一个整体提供配置。例如：如果我们确定在系统的现有和未来业务中，仅分为普通成员有“人员管理”的查看权限，管理员有操作权限，则可将“增、删、改”三个基本权限单位合并为“操作”权限进行配置。

1. **页面权限优先于操作和数据权限**

必须配置了页面模块权限后，才能配置当前页面模块下具体的操作权限，以及页面模块的数据展示权限。

1. 查看权限优先于增删改权限

正常情况下，一定要先能查看某个模块或操作，其它的增删改操作才有意义。因此在设计时，应在获取查看权限前限制其它权限的配置，或者配置其它权限时默认赋予查看权限。

1. **角色与权限的多种关系**

角色与权限的关系不仅是单纯“是/否关系”，还包括以某种限制进行操作，和以某种程度访问数据。例如在“人员管理”中：

数据范围：用户拥有查看人员列表的权限，但仅能查看自己所在的团队；数据边界限制（上限等）：添加人员时不能超过20个等。数据字段：HR能查看人员列表中包括职级、薪资等字段，其它角色仅能查看姓名邮箱等字段；

1. **角色与权限的设计表达**

在传达一个系统的权限设计规则时，设计师常常习惯用主观最直接的方式表达想法，如用“当……时，就……”的句式来表达。但一个平台中涉及的权限规则是非常多的，当通篇以这样的形式描述时，表达对象将很难理解。

正确的描述方式：更清晰的是基于开发的语言，和技术模型的结果进行表达。将各角色与权限单元绘制成网格，每个交叉点网格中描述该角色与权限的数据关系和限制。



1. **后台数据的访问控制及服务控制**

因为系统采用高性能的前后端剥离的技术框架；通常前端页面的浏览或操作需要通过后台的API接口获取数据，所以需要特别注意后台的数据访问应加入通过数据访问的逻辑判断模块，并能及时响应错误输出。

1. **隐形的admin（超级管理员）**

在可自定义角色和权限的系统中，一般需要预留一个admin角色来进行系统的初始配置，用于添加首批的业务人员和配置基本的角色。

1. **初始权限的赋予**

对于允许用户自行加入的系统，需要设定一至多个默认的角色，有时可以是仅有最基础权限的“游客”角色。

初始权限还可以与用户既有的某些数据字段进行关联，如添加用户时获取到用户的岗位为“设计师”，则直接赋予“设计师”角色的权限。

1. **人员管理中对自己的处理**

在人员管理中，管理员角色处理自己时需要额外注意。因为如果修改或删除了自己角色后，可能导致系统没有管理角色，从而无法添加其他成员和正常运行。设计时可添加判断，当自己为唯一管理角色时，禁止编辑和删除。

1. **无页面权限的提示**

虽然可以通过页面权限限制直接隐藏当前用户没有权限的页面，但不能排除用户获取到权限外的url地址。当用户意外访问到没有权限的页面时务必提供“无权限”的提示，避免用户认为系统bug。