权限=规则+区域

考虑将规则表和权限表分开，权限表中存规则表的ID，而不是同一张表。

AreaAuthority，AccessRule，AuthorityAreaRelation

也就是说同样一个规则可以和不同区域关联起来成为不同的权限设置。然后相同的区域可以和不同的规则关联起来成为不同的权限设置。

区域1，区域2，区域3

规则1，规则2

权限1=规则1+区域1

权限2=规则2+区域2

权限3=规则1+区域2

权限4=规则1+（区域1&区域2）

权限5=规则2+（区域1&区域2&区域3）

是否影响到子区域？

因为规则中有很多属性，如果不将规则和权限分开，规则的复用就比较麻烦了。

权限中的区域的父子关系怎么处理？

1. 子区域在父区域的内部
   1. 能进入子区域，不能进入父区域。
   2. 能进入父区域，不能进入子区域。
2. 子区域不在父区域的内部
3. 子区域不完全在父区域的内部

权限：多个不同的区域进出规则的集合。

一个规则可以属于多个不同的权限。

权限1：规则1(可以进入区域1)+规则2(不能进入区域2)+规则3(不能离开区域3)

权限2：规则1(可以进入区域1)+规则4(可以进入区域2)

最终用于绑定的是权限。

判断时，根据权限获取规则列表，判断规则，发送告警。

操作界面的功能上无论是权限还是规则都是能够复制的，复制并修改，这样子创建新的会比较方便。

权限组，多个不同的权限的集合，分组用，暂时不具有业务相关的功能。

可以考虑将权限组绑定到一个卡组上。

一个卡组上可以有多个权限，这些权限独立判断。

可能发生几个权限相互矛盾。一个权限可以进入，一个权限不能进入的情况，最好能识别出来，当时就算是不能识别出来也没关系，因为不影响告警的发出。

要不要有规则组呢？有了规则组的话，整理上比较方便，属于某一个组，而且，需要的话可以在创建权限时选择一个组，就添加了组内的所有规则。