# CEC4J设计思路

**主要实体：**

**Peer（对等体）**：以套接字（ip地址+端口）形式划分

**Group（组群）**：Peer可以属于不同的组群，每个组群可以拥有多个Peer，每个Peer维护一个所在组群的列表

**Service（服务）**：分为提供者和接受者，提供者维护一个接受者列表，接受者需要有这一服务的处理函数

**Pipe（管道（单向和多向））**：把对等体之间的连接抽象成一个管道，定义管道的几种状态和监听的事件

**Message（消息）**：消息封装在xml中然后通过管道进行发送和接受

**对应的通告：（通告采用定时发送方式，全部采用UDP的方式发送，每接收到通告就刷新对应的心跳值）**

**对等体通告**：定时向**已知的对等体**发送你还存活的通告

**组群通告**：定时向周围人发送你属于的组群

**服务通告**：提供者服务通告：告送其他人我提供这个服务

接受者服务通告：告送服务提供者我接受这个服务

**管道事件：**

建立

收到数据

异常断开：对方peer到达心跳值产生异常断开事件

正常断开

**心跳：**

Peer设定每隔一段时间触发一次心跳，来保持自己组群列表中的peer都是存活的，心跳是对应对等体通告而言的，整个系统都依赖对等体存活

Jxta初始化Peer之后会让其默认在一个组群中，这对于整个互联网来说是不可能的

**组群的组织方式：**

对等体可以创建组群，对等体可以查询另一对等体所在的组群，然后以那个对等体为中间人（采用自我介绍的形式：从那个对等体得到该组群的成员列表，然后发送通告）加入该组群

**本地化：**

没有本地化对等体下线，下一次登录需要重新进行一次所有操作，

本地化把对等体信息序列化到本地，当下一次初始化的时候加载

**对应对等体的状态：**

存活：

睡眠：

死亡：判定死亡的两种方式：对等体宣布自己死亡、对等体长时间不上线