# 金融中间件spider架构及关键功能介绍

## 中间件目标

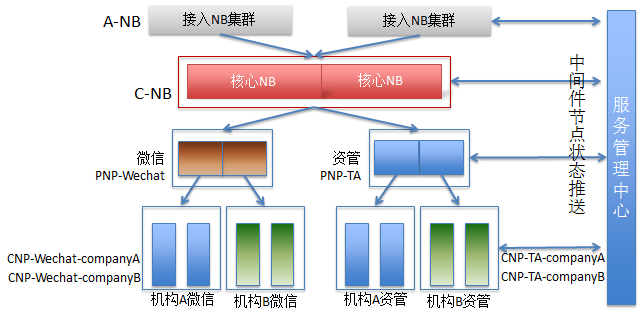
金融中间件需要满足以下要求：

* 能够根据请求包里面的功能号、子系统号进行静态路由；
* 支持多租户。能够根据请求包里面的机构号、产品（系统）号、版本号进行动态路由；  
  saas架构的理想模式就是所有的业务实现均原生支持多租户。  
  如果做不到原生支持多租户，按照一个租户一个runtime，也需要动态能够设置某个runtime提供对哪个客户的支持，这肯定不能让运维动态更改配置文件，他只需要中心化指定即可。即使能做到，也需要支持动态能够设置某个app提供对哪个客户的支持，因为一个app仅能够提供对有限个客户的支持，且mysql也不可能做到一个库支持所有。反之，如果做不到原生支持多租户，则app必须支持多数据源切换以及动态管理，因为通常一个客户通常难以大到需要一个app去支持的规模。
* 支持服务器集群中的各节点负载均衡（根据模式）；
* 支持基于中心化管理的大型部署和独立管理的中小型部署；
* 支持自动重连到断开的服务器节点；
* 支持中心化剔除已删除的服务器节点；
* 支持中心化增加服务器节点；
* 支持中心化删除服务器集群；
* 支持中心化增加服务器集群；
* 一个可执行组件，支持任何时候节点（Work Node，WN）作为代理服务器（Node Broker，NB）(ANB:access node broker；CNB:core node broker）或处理服务器（Node Processor，NP）（PNP（产品NP），CNP（机构NP）），或从开发角度分为业务服务处理器（Business Node Processor，BSNP）、微服务处理器（MSNP,Micro Servicce Node Processor）；
* 基于TCP/IP协议，支持客户端语言无关；
* 支持连接亲和性（没有亲和性，就不是一个好的负载均衡实现）；
* 节点间技术上对等，即允许同时满足A是B的客户端，B是A的客户端，只要保证路由不形成死循环即可；
* 支持报文加密；
* 能够自动校验报文是否被篡改；
* 支持客户端认证功能；
* 支持报文压缩；
* 支持是否启用服务端功能；
* 能够中心化监控整个私募服务系统中所有中间件的运行状况；
* 支持基于Spring IOC注解方式的Java原生式远程服务调用；
* 最小化节点本地（集中化）配置；
* 支持多种序列化机制；
* 支持List泛型；
* 兼容原LDPack格式打包的C++客户端至17年底，但（对于新的C#和B/S系统，使用JSON序列化）
* 支持客户端异步请求与回调（主要用于定时任务和跑批场景）；
* 支持服务可靠执行（可保证请求收到后不会丢失且肯定会按照提交的顺序（这是个复杂的策略，决定了并发性）执行，后续版本会考虑支持回调；
* 兼容原错误号；
* 消息主推；
* 灵活的路由，任何服务都可以发送到任何集群（暂不支持到具体的节点）。

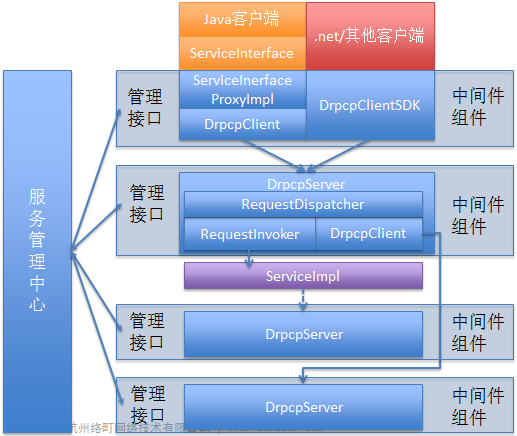
下述为Spider特性支持列表：

|  |  |  |
| --- | --- | --- |
| 特性 | 起始支持版本 | 已自测 |
| 根据请求包里面的功能号、子系统号进行静态路由； | 1.0.0 | 是 |
| 根据请求包里面的机构号、产品号、版本号进行动态路由； | 1.0.0 | 是 |
| 服务器集群中的各节点负载均衡； | 1.0.0 | 是 |
| 配置负载均衡策略 | 1.0.1 |  |
| 支持基于中心化管理的大型部署和独立管理的中小型部署； | 1.0.0 | 是 |
| 自动重连到断开的服务器节点； | 1.0.0 | 是 |
| 支持中心化剔除已删除的服务器节点； | 1.0.0 |  |
| 支持中心化增加服务器节点； | 1.0.0 |  |
| 支持中心化删除服务器集群； | 1.0.0 |  |
| 支持中心化增加服务器集群； | 1.0.0 |  |
| 一个可执行组件，支持任何时候节点（Work Node，WN）从运行角度作为下列角色：   * 代理服务器（Node Broker，NB）(ANB:access  node broker；CNB:core node broker） * 处理服务器（Node Processor，NP）（PNP（产品NP），CNP（机构NP））；   或从开发角度分为：   * 业务服务处理器（Business Node Processor，BSNP）； * 微服务处理器（MSNP,Micro Servicce Node Processor） | 1.0.0 | 是 |
| 基于TCP/IP协议，支持客户端语言无关； | 1.0.0 | 是 |
| 节点间技术上对等，即允许同时满足A是B的客户端，B是A的客户端，只要保证路由不形成死循环即可； | 1.0.0 |  |
| 支持报文加密； | 1.0.0 | 是 |
| 自动校验报文是否被篡改； | 1.0.0 | 是 |
| 支持报文压缩； | 1.0.0 | 是 |
| 支持是否启用服务端功能； | 1.0.0 | 是 |
| 能够中心化监控整个私募服务系统中所有中间件的运行状况； | 1.0.0 |  |
| 独立管理模式下能够查看本节点运行状态 | 1.0.0 | 是 |
| 基于Spring IOC注解方式的Java原生式远程服务调用； | 1.0.0 | 是 |
| 最小化节点本地（集中化）配置； | 1.0.0 |  |
| 支持客户端设置异步请求及回调； | 1.0.1 |  |
| 支持服务端可信模式； | 1.0.0 | 是 |
| 支持客户端认证； | 1.0.0 | 是 |
| 支持不同字符集配置； | 1.0.0 | 是 |
| 兼容原LD服务注解； | 1.0.0 | 是 |
| 兼容原LDPack打包适配 | 1.0.0 | 是 |
| 支持List泛型 | 1.0.1 | 是 |

## 私募服务平台中间件架构



## 金融中间件内部请求处理流程



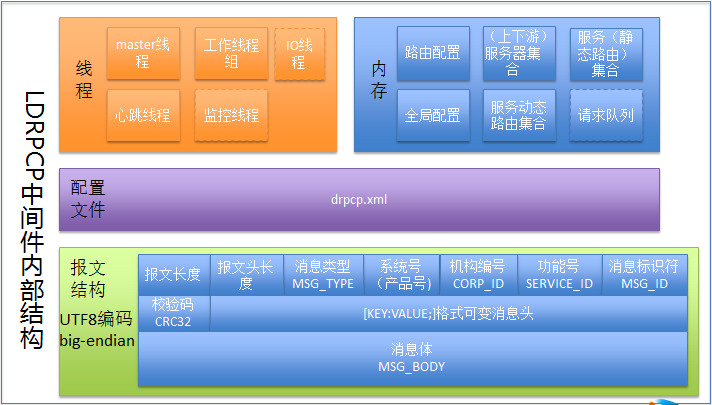
作为路由角色时，spider将运行于纯异步模式，1.0.8。

作为LS/AS角色时，spider将运行于同步模式，1.0.8。

当客户端和服务端运行在相同的主机时，将从NIO/EPOLL模式自动切换为UnixDomainSocket模式，1.0.8。

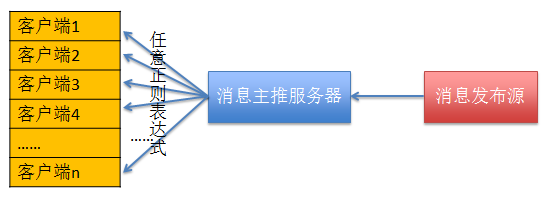
## 金融中间件内部结构

本图仅供参考，随着版本的演进，内部组件和结构可能会发生变化，实际以概要设计说明为准。



## 消息主推

基于对在络町中间件平台集成自行开发、RabbitMQ、Redis这三者实现消息主推机制的对比研究，建议消息主推和中间件本身独立提供，并使用Redis/Rabbitmq作为消息主推的实现机制。其结构如下所示：



计划在1.0.5版本（2017年1月），spider会提供原生的高可靠性主推实现。

## 主要中间件管理流程

## 新增中间件节点（NB或NP）流程



## 新增产品NP集群流程



## 新增机构NP集群流程



## 删除中间件（NB或NP）节点流程



## 删除产品NP集群流程



## 删除机构NP集群流程

