



# PolarisMesh开发者例会

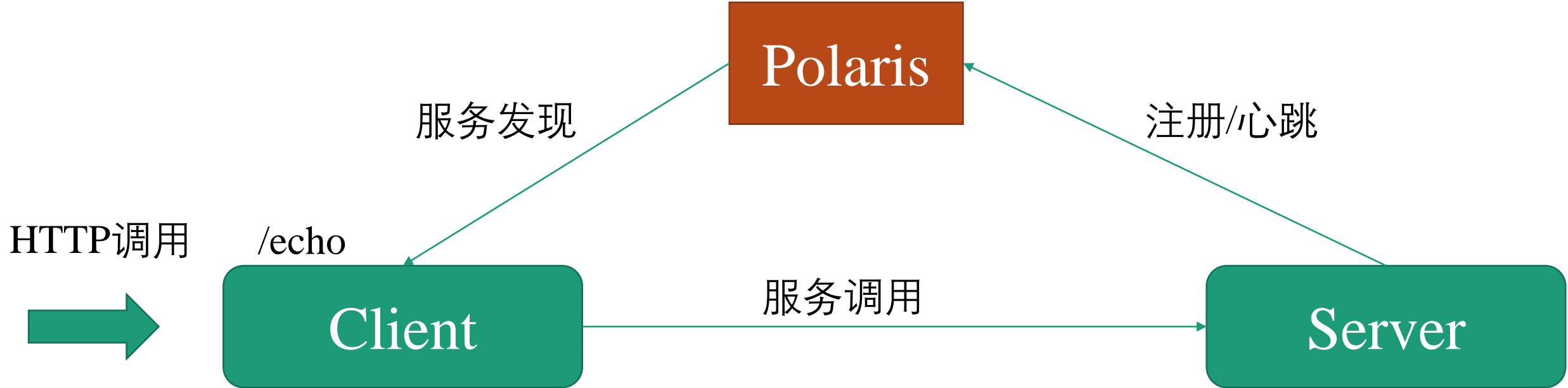
2021-12-14

- 开源生态对接进展介绍
- 文档及样例进展介绍
- 12月已发布版本介绍
- v1.4.0版本进展介绍

# 开源生态对接进展介绍

分类	组件库	负责人	服务注册	心跳上报	服务发现	动态路由	负载均衡	故障熔断	服务限流
gRPC	gRPC-Go-Polaris	andrewshan	已完成	已完成	已完成	已完成	已完成	已完成	未完成
	gRPC-Java-Polaris	li-xiao-shuang	已完成	已完成	已完成	未完成	未完成	未完成	未完成
	gRPC-Cpp-Polaris	jason	未完成	未完成	未完成	未完成	未完成	未完成	未完成
	gRPC-NodeJS-Polaris	Jeff zuo	未完成	未完成	未完成	未完成	未完成	未完成	未完成
dubbo	dubbo-go	chuntaojun	已完成	已完成	已完成	未完成	未完成	未完成	未完成
spring-boot	spring-boot-polaris	chuntaojun	已完成	已完成	已完成	未完成	未完成	未完成	未完成
sofa-rpc	registry-polaris	ZLBer	未完成	未完成	未完成	未完成	未完成	未完成	未完成
goframe	gogf/polaris	houseme	已完成	已完成	未完成	未完成	未完成	未完成	未完成
kitex	registry-polaris	liu-song	未完成	未完成	未完成	未完成	未完成	未完成	未完成

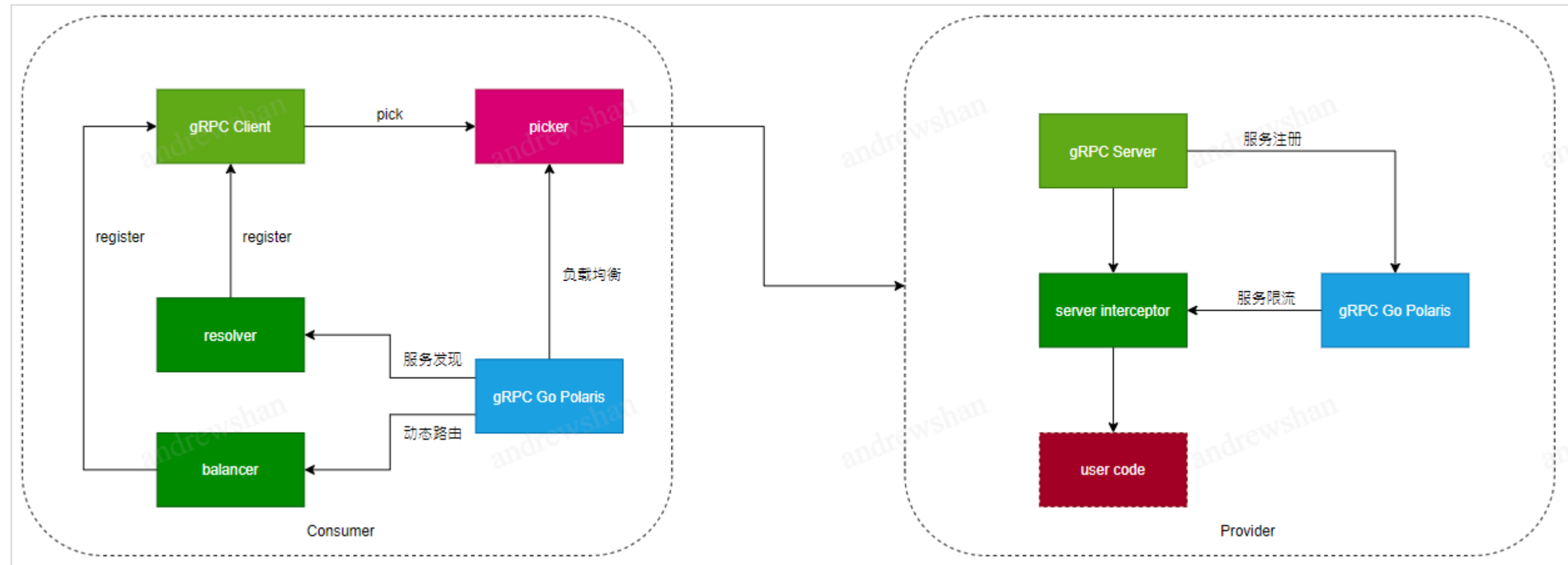
<https://github.com/polarismesh/polaris/issues/163>



```
func main() {
    //grpc客户端连接获取
    ctx, cancel := context.WithCancel(context.Background())
    defer cancel()
    conn, err := polaris.DialContext(ctx, target: "polaris://EchoService", polaris.WithGRPCDialOptions(grpc.WithInsecure()))
    if err != nil {
        log.Fatal(err)
    }
    defer conn.Close()
    echoClient := pb.NewEchoServerClient(conn)

    indexHandler := func(w http.ResponseWriter, r *http.Request) {
        err := r.ParseForm()
        if nil != err {
            log.Printf("fail to parse request form: #{err}\n")
            w.WriteHeader(statusCode: 500)
            _, _ = w.Write([]byte(err.Error()))
            return
        }
        values := r.Form["value"]
        log.Printf("receive value is %s\n", values)
        var value string
        if len(values) > 0 {
            value = values[0]
        }
        resp, err := echoClient.Echo(ctx, &pb.EchoRequest{Value: value})
        log.Printf("send message, resp (%v), err(%v)", resp, err)
        if nil != err {
            w.WriteHeader(statusCode: 500)
            _, _ = w.Write([]byte(err.Error()))
            return
        }
        w.WriteHeader(statusCode: 200)
        _, _ = w.Write([]byte(resp.GetValue()))
    }
    http.HandleFunc(pattern: "/echo", indexHandler)
    if err := http.ListenAndServe(fmt.Sprintf(":%d", listenPort), handler: nil); nil != err {
        log.Fatal(err)
    }
}
```

```
func main() {
    srv := polaris.NewServer(polaris.WithServerApplication(application: "EchoService"))
    pb.RegisterEchoServerServer(srv.GRPCServer(), &EchoService{})
    // 监听端口
    address := "0.0.0.0:0"
    listen, err := net.Listen(network: "tcp", address)
    if err != nil {
        log.Fatalf("Failed to addr #{address}: #{err}")
    }
    err = srv.Serve(listen)
    if nil != err {
        log.Fatal(err)
    }
}
```



resolver: 扩展resolver接口, 对接polaris获取进行服务发现, 并监听变更事件

balancer: 扩展balancer接口, 对grpc连接池进行管理, 按节点路由过滤结果进行连接的新增剔除

picker: 扩展picker接口, 根据用户的请求头内容信息, 对健康连接进行负载均衡

server: 封装GRPC Server的声明周期, 进行服务注册/心跳上报以及反注册操作

interceptor: 扩展interceptor, 实现限流配额获取功能

# 文档及样例进展介绍



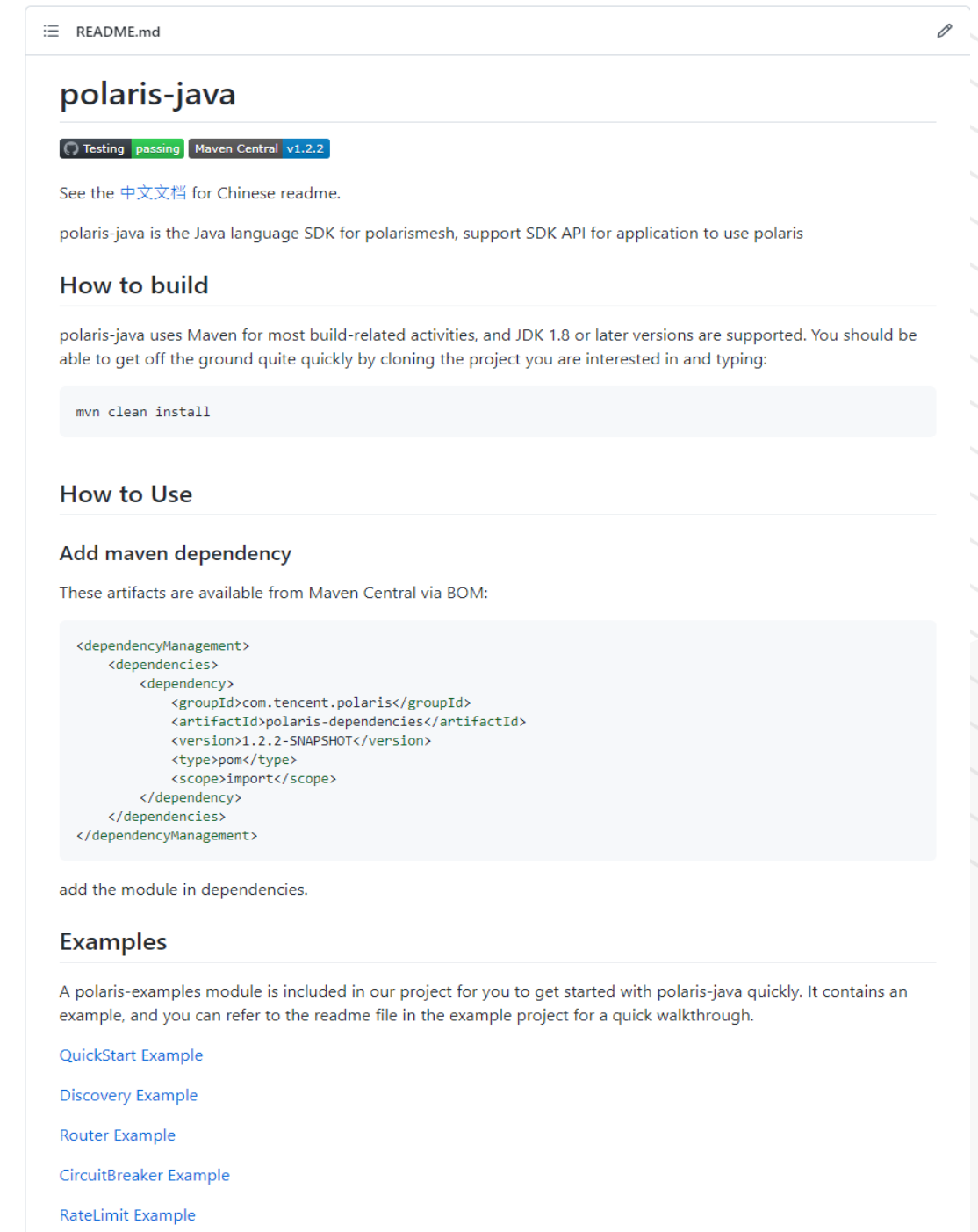
# 整体文档样例进展

分类	文档名称	负责人	样例	文档
服务端部署文档	单机版安装	andrewshan	不涉及	已完成
	集群版安装	andrewshan	不涉及	已完成
	配置说明	andrewshan	不涉及	未完成
服务端接口文档	命名空间管理	andrewshan	不涉及	已完成
	服务管理	andrewshan	不涉及	已完成
	实例管理	andrewshan	不涉及	已完成
	路由规则管理	andrewshan	不涉及	未完成
	熔断规则管理	andrewshan	不涉及	未完成
	限流规则管理	andrewshan	不涉及	未完成
	服务发现	andrewshan	不涉及	未完成
服务端开发文档	架构模型介绍		不涉及	未完成
	数据模型介绍		不涉及	未完成
	插件开发手册		不涉及	未完成
Java SDK	快速入门	andrewshan	已完成	已完成
	使用指南		未完成	未完成
	开发手册		未完成	未完成
Go SDK	快速入门	chuntaojun	已完成	已完成
	使用指南			
	开发手册			
PHP SDK	快速入门	chuntaojun	已完成	已完成
	使用指南	chuntaojun	已完成	已完成
	开发手册			



1. 需要提供中英文README文档，中文以-zh结尾
2. 中英文文档可以通过链接相互跳转
3. 顶层README需要提供样例链接
4. 样例至少需要包含quickstart快速入门

<https://github.com/polarismesh/polaris-java>



# 12月已发布版本介绍

## 已发布版本

1.3.0发布了2个版本，解决了以下问题：

1. 单机版使用的问题
2. 集群版安装问题

提供了2个功能的预览版本：

1. 命名空间管理
2. 服务自动创建

<https://github.com/polarismesh/polaris/releases>

v1.3.2 Latest

Bug FIX

- dirty instances left while config boltdb store mode

Assets 11

v1.3.1

Feature

- [Preview]console support namespace management

Enhancement

- [Preview]support service auto creation

Bug FIX

- cluster install error when self registering

Assets 11

# v1.4.0版本进展介绍

<https://github.com/polarismesh/polaris/milestone/5>

模块	描述	issue	进展
服务注册	支持服务自动注册	<a href="https://github.com/polarismesh/polaris/issues/136">https://github.com/polarismesh/polaris/issues/136</a>	已完成
控制台	支持命名空间创建	<a href="https://github.com/polarismesh/polaris-console/issues/8">https://github.com/polarismesh/polaris-console/issues/8</a>	已完成
安装部署	支持Helmchart部署	<a href="https://github.com/polarismesh/polaris/issues/161">https://github.com/polarismesh/polaris/issues/161</a>	开发中
可运维	日志支持按模块分文件打印	<a href="https://github.com/polarismesh/polaris/issues/119">https://github.com/polarismesh/polaris/issues/119</a>	已完成
可运维	支持服务事件日志打印	<a href="https://github.com/polarismesh/polaris/issues/147">https://github.com/polarismesh/polaris/issues/147</a>	已完成
缓存管理	支持命名空间删除后清理连带资源	<a href="https://github.com/polarismesh/polaris/issues/157">https://github.com/polarismesh/polaris/issues/157</a>	开发中
服务限流	支持分布式限流	<a href="https://github.com/polarismesh/polaris/issues/184">https://github.com/polarismesh/polaris/issues/184</a>	未开始
ServiceMesh	XDS支持熔断规则下发	<a href="https://github.com/polarismesh/polaris/issues/185">https://github.com/polarismesh/polaris/issues/185</a>	PR中
	XDS支持限流规则下发	<a href="https://github.com/polarismesh/polaris/issues/186">https://github.com/polarismesh/polaris/issues/186</a>	未认领

欢迎大家继续认领，争取成为committer

# Q&A