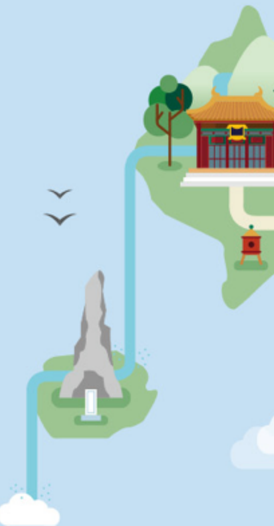
Go 微服务架构 - 来自 Java/Spring 开发者视角

Nash Tsai, Technical Expert @ 阿里云



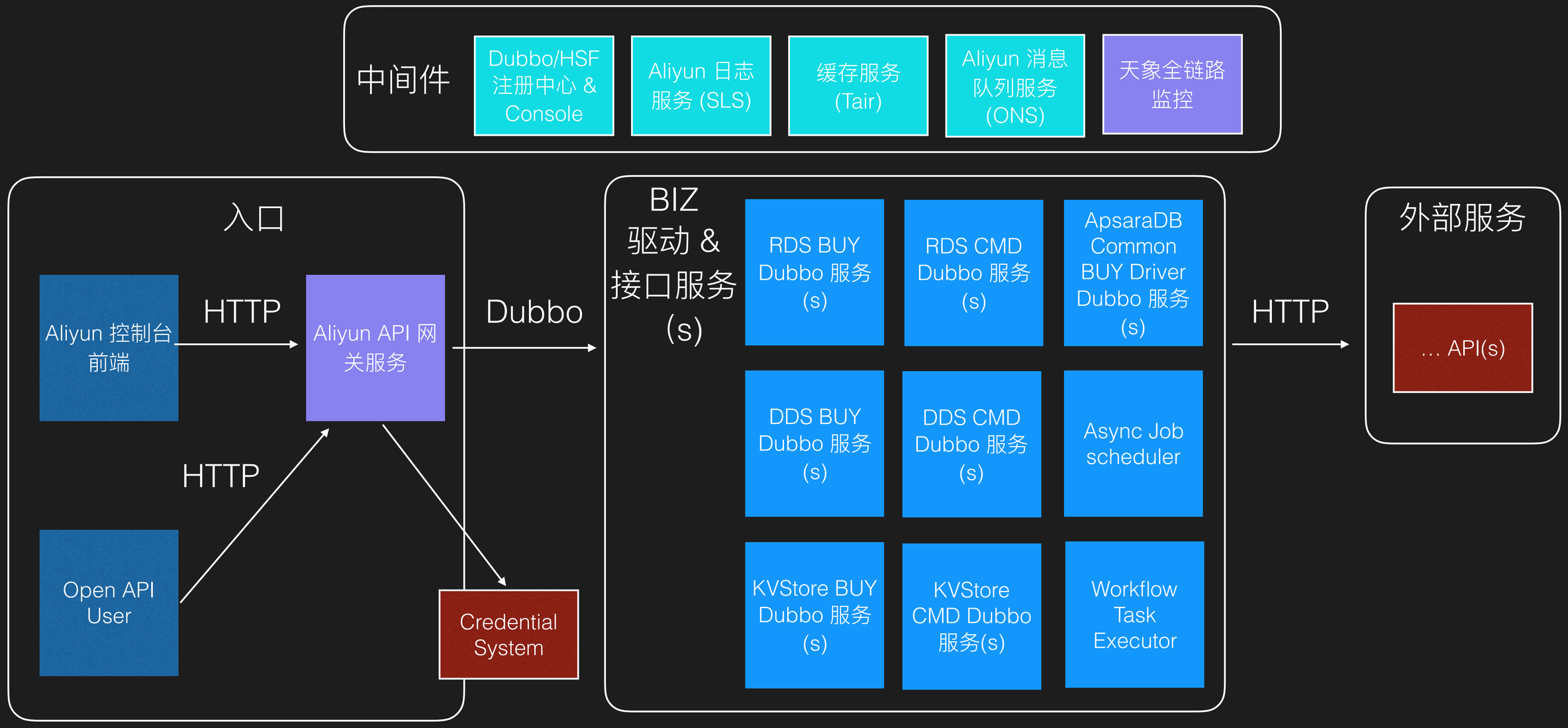
Nash Tsai（聪心）

- 技术专家 @ 阿里云-飞天8部-数据库组
- OpenAPI services architecture
- Software Engineering (~10y experience) & DevOps

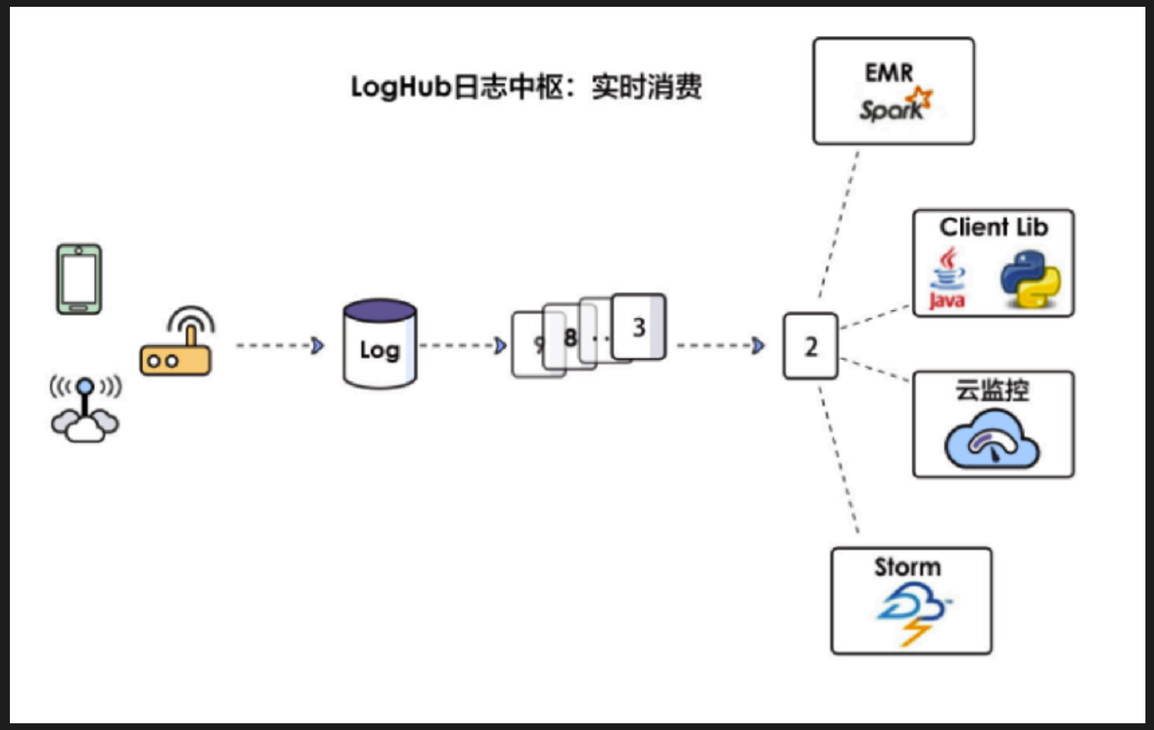
Agenda

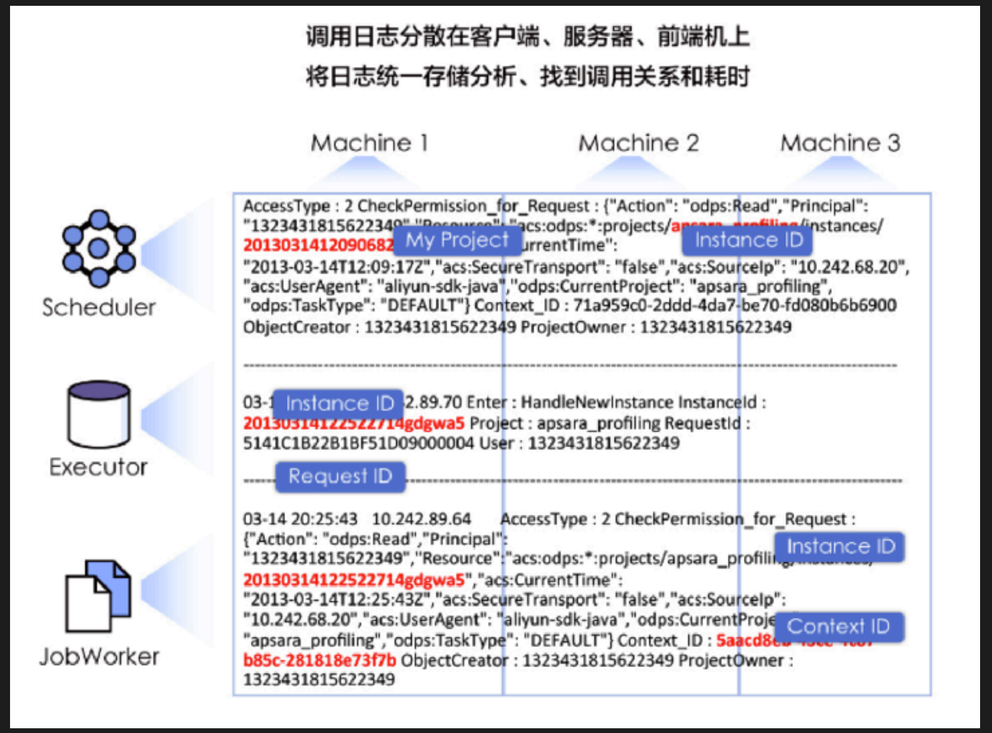
- 背景 - 云产品前端架构
- Micro-services complexity at 阿里云
- Profound of Java vs. Golang (Spring vs. Go tooling)
- Introduction of gPRC and Go kit
- Micro-services best practices

背景-云产品前端架构

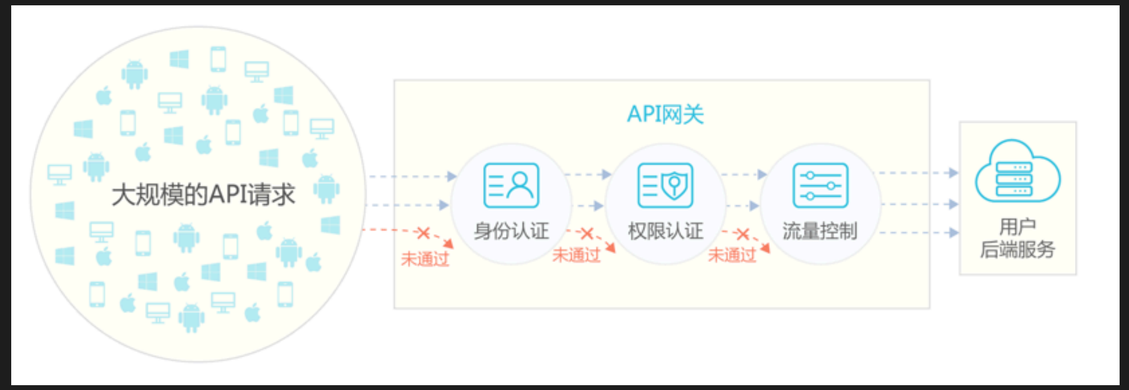

中间件
Dubbo/HSF 注册中心 & Console
Aliyun 日志服务 (SLS)
缓存服务 (Tair)
Aliyun 消息队列服务 (ONS)
天象全链路监控
入口
Aliyun 控制台前端
HTTP
Aliyun API 网关服务
HTTP
Open API User
Credential System
Dubbo
BIZ 驱动 & 接口服务 (s)
RDS BUY Dubbo 服务 (s)
RDS CMD Dubbo 服务 (s)
ApsaraDB Common BUY Driver Dubbo 服务 (s)
DDS BUY Dubbo 服务 (s)
DDS CMD Dubbo 服务 (s)
Async Job scheduler
KVStore BUY Dubbo 服务 (s)
KVStore CMD Dubbo 服务(s)
Workflow Task Executor
HTTP
外部服务
… API(s)

阿里云日志服务

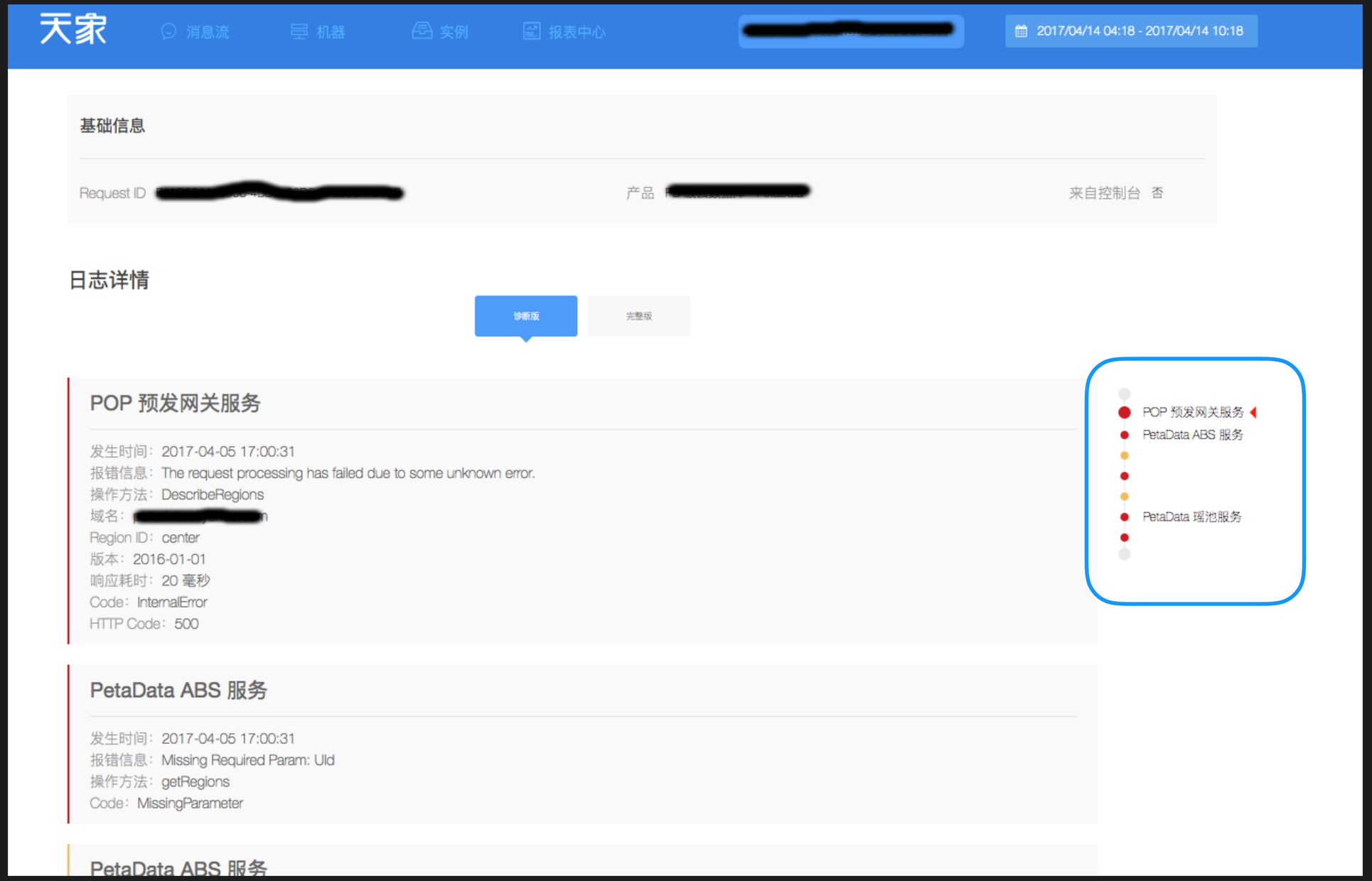

LogHub日志中枢：实时消费
EMR
Spark
Client Lib
Java
云监控
Storm
Log
9
8
3
2


调用日志分散在客户端、服务器、前端机上
将日志统一存储分析、找到调用关系和耗时
Machine 1
Machine 2
Machine 3
Scheduler
Executor
JobWorker
My Project
Instance ID
Request ID
Context ID

阿里云 API 网关服务


大规模的API请求
API网关
身份认证
权限认证
流量控制
未通过
未通过
未通过
用户
后端服务

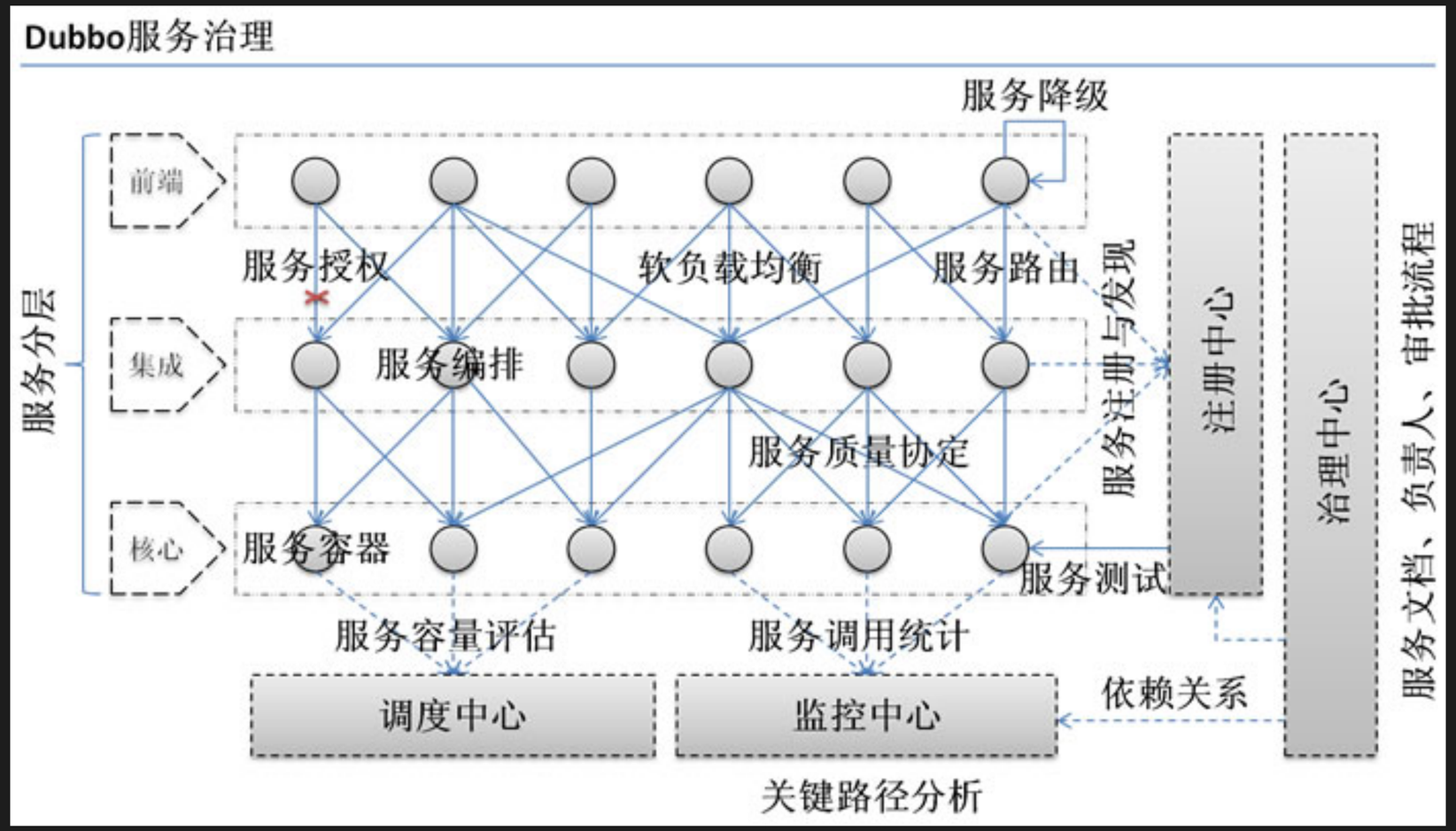

API请求
API网关
phone=18612345678
phone⇌a
参数检验
参数转换
a=18612345678
phone=abcd
用户
后端服务
参数名：a
参数描述：手机号
正则规则：11位数字


天家
消息流
机器
实例
报表中心
[redacted]
2017/04/14 04:18 - 2017/04/14 10:18
基础信息
Request ID [redacted]
产品 [redacted]
来自控制台 否
日志详情
诊断版
完整版
POP 预发网关服务
发生时间：2017-04-05 17:00:31
报错信息：The request processing has failed due to some unknown error.
操作方法：DescribeRegions
域名：[redacted]
Region ID：center
版本：2016-01-01
响应耗时：20 毫秒
Code：InternalError
HTTP Code：500
PetaData ABS 服务
发生时间：2017-04-05 17:00:31
报错信息：Missing Required Param: Uid
操作方法：getRegions
Code：MissingParameter
PetaData ABS 服务
POP 预发网关服务
PetaData ABS 服务
PetaData 瑶池服务

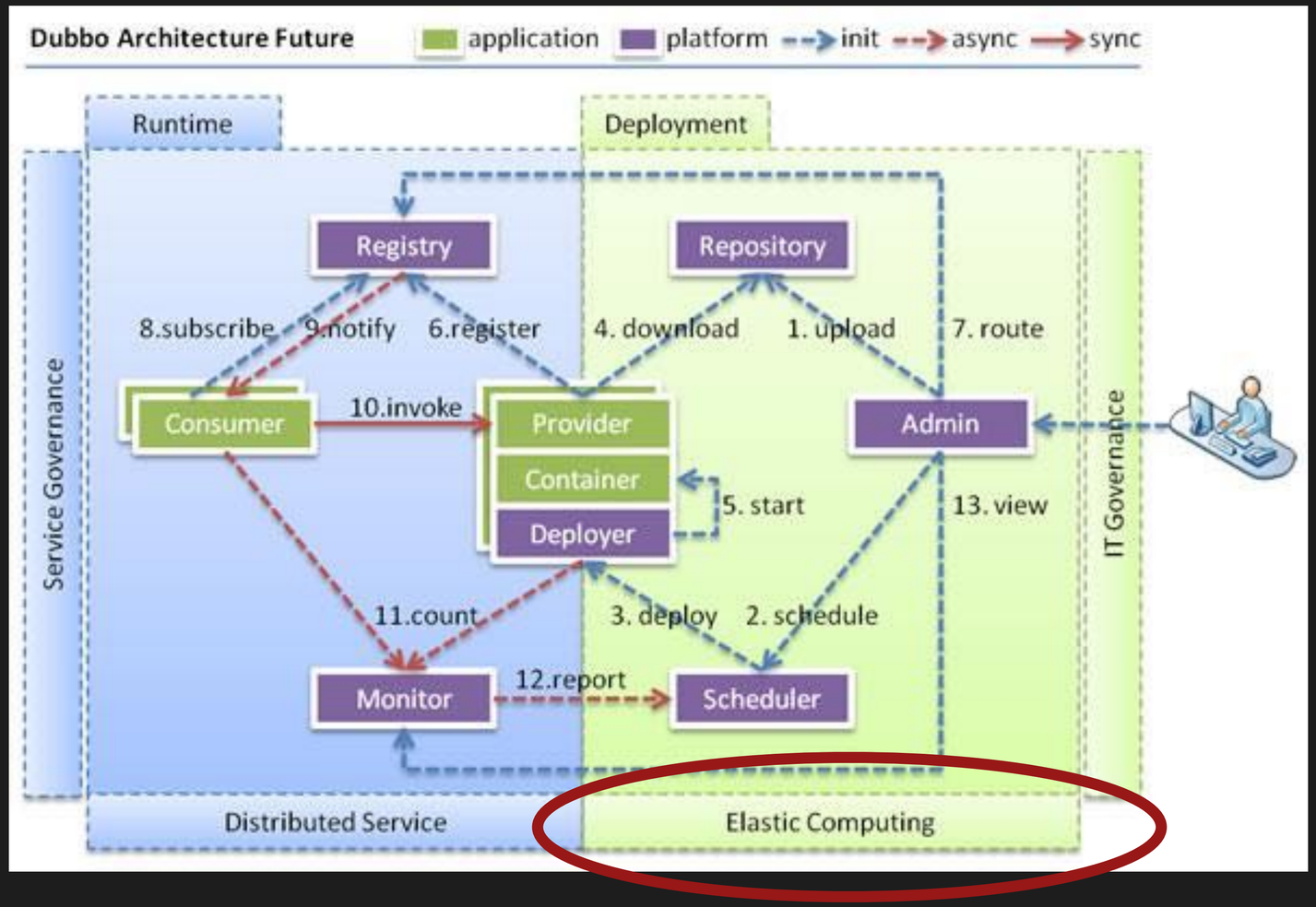
Dubbo background

- 分布式 RPC 框架
  - Fast, highly scalable and industrial proven (2K+ nodes providing 3B+ request/day, adopted by Alibaba, JD, 当当网)
- Play nice with Java Spring application (J2EE)
- Features (好用的):
  - 服务动态注册 & 服务发现
  - SOA 服务治理
  - 软负载均衡
  - 熔断、服务降级

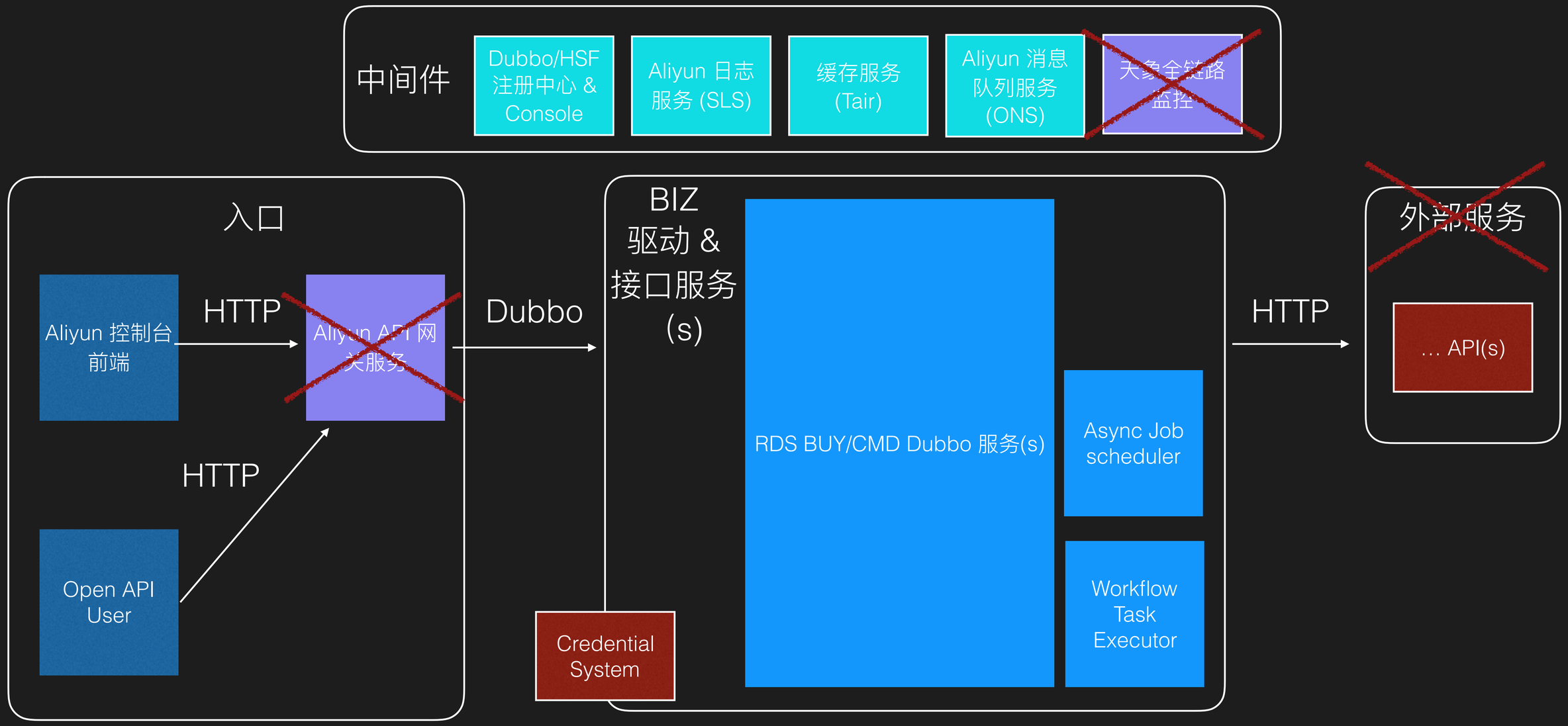
Dubbo 服务治理


Dubbo服务治理
服务降级
前端
集成
核心
服务分层
服务授权
软负载均衡
服务路由
服务编排
服务质量协定
服务容器
服务注册与发现
服务测试
注册中心
治理中心
服务文档、负责人、审批流程
服务容量评估
服务调用统计
调度中心
监控中心
依赖关系
关键路径分析

Dubbo 部署到可服务化



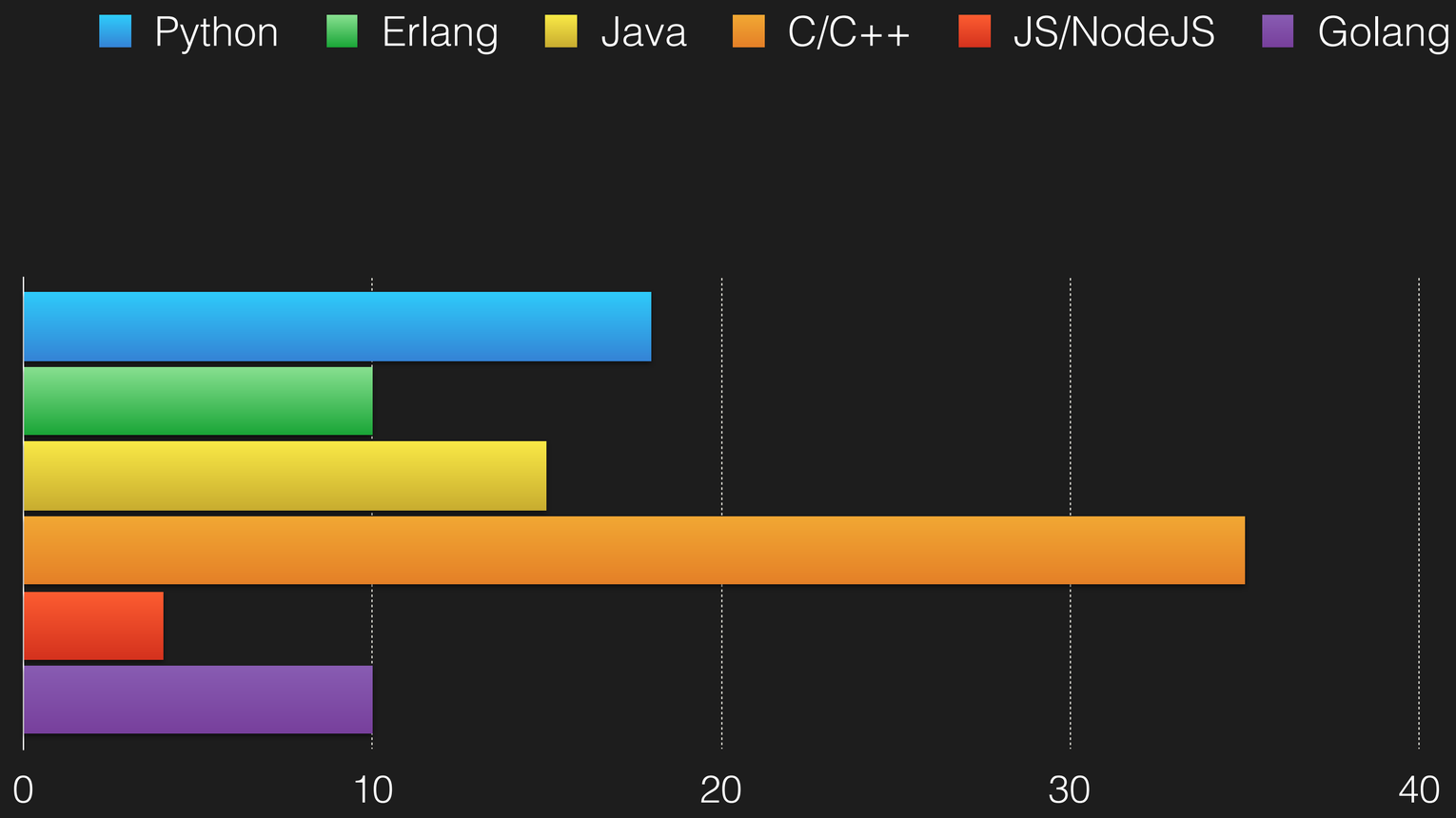
背景-云产品前端架构（老）


中间件
Dubbo/HSF 注册中心 & Console
Aliyun 日志服务 (SLS)
缓存服务 (Tair)
Aliyun 消息队列服务 (ONS)
天象全链路监控
入口
Aliyun 控制台前端
HTTP
Aliyun API 网关服务
HTTP
Open API User
Dubbo
BIZ 驱动 & 接口服务 (s)
RDS BUY/CMD Dubbo 服务(s)
Async Job scheduler
Workflow Task Executor
Credential System
HTTP
外部服务
… API(s)

Micro-services complexity at 阿里云

- Why Micro-services?
  - Team is too large
  - product velocity
- Many Micro-services inherited problems are solved with Dubbo and Spring (Cloud) Framework, however:
  - Testing is still HARD!
  - DevOps culture?
  - Security?
  - Distributed Tracing?
  - Huge payload (Dubbo specific)

Distributed map for programming languages among the team



Java is “SO DYNAMIC”!

```
18          at java.lang.reflect.Method.invoke(Method.java:497) ~[na:1.8.0_66]
19          at
org.springframework.boot.loader.MainMethodRunner.run(MainMethodRunner.java:48)
[exploded/:na]
20          at org.springframework.boot.loader.Launcher.launch(Launcher.java:87)
[exploded/:na]
21          at org.springframework.boot.loader.Launcher.launch(Launcher.java:50)
[exploded/:na]
22          at
org.springframework.boot.loader.JarLauncher.main(JarLauncher.java:51)
[exploded/:na]
23 Caused by: java.lang.NoSuchMethodError:
javax.servlet.ServletContext.addServlet(Ljava/lang/String;Ljavax/servlet/Servl
et;)Ljavax/servlet/ServletRegistration$Dynamic;
```

Java is “MAGICAL!” (这是什么鬼？)

```
@Service
public class MagicalClass {

   @Autowire
   private Bean bean;

   @Resource
   private DAO resoureceDao;

   ...
}
```

Java is “MAGICAL!” (这是什么鬼？)

```
@Aspect
public class MagicalClassAspect {

   @Before("getSomethingPointcut()")
   public void loggingAdvice() {
      System.out.println(
         "Executing loggingAdvice on "
         + "getSomething()");
   }
}
```

Profound of Java vs. Golang (Spring vs. Go Tooling)

- Java:
  - No all Java Applications are Spring Application (and not all Java developers are Spring Developers)
  - Spring is BIG (Spring 2~4), and too much magics happening
  - Dubbo's IDL is a Java interface class
  - JVM is a memory hog (0.5~6GB per micro-service JVM)
- vs. Golang
  - Simple, Elegant (i.e., defer vs finally) and forced to bundle 3rd part source codes
  - Go tooling:
    - go test/go test -bench & go tool <pprof/vet/cover/…>
    - go-torch (by Uber)
  - Memory (<=0.5GB per application container)

Introduction of gRPC (grpc.io)

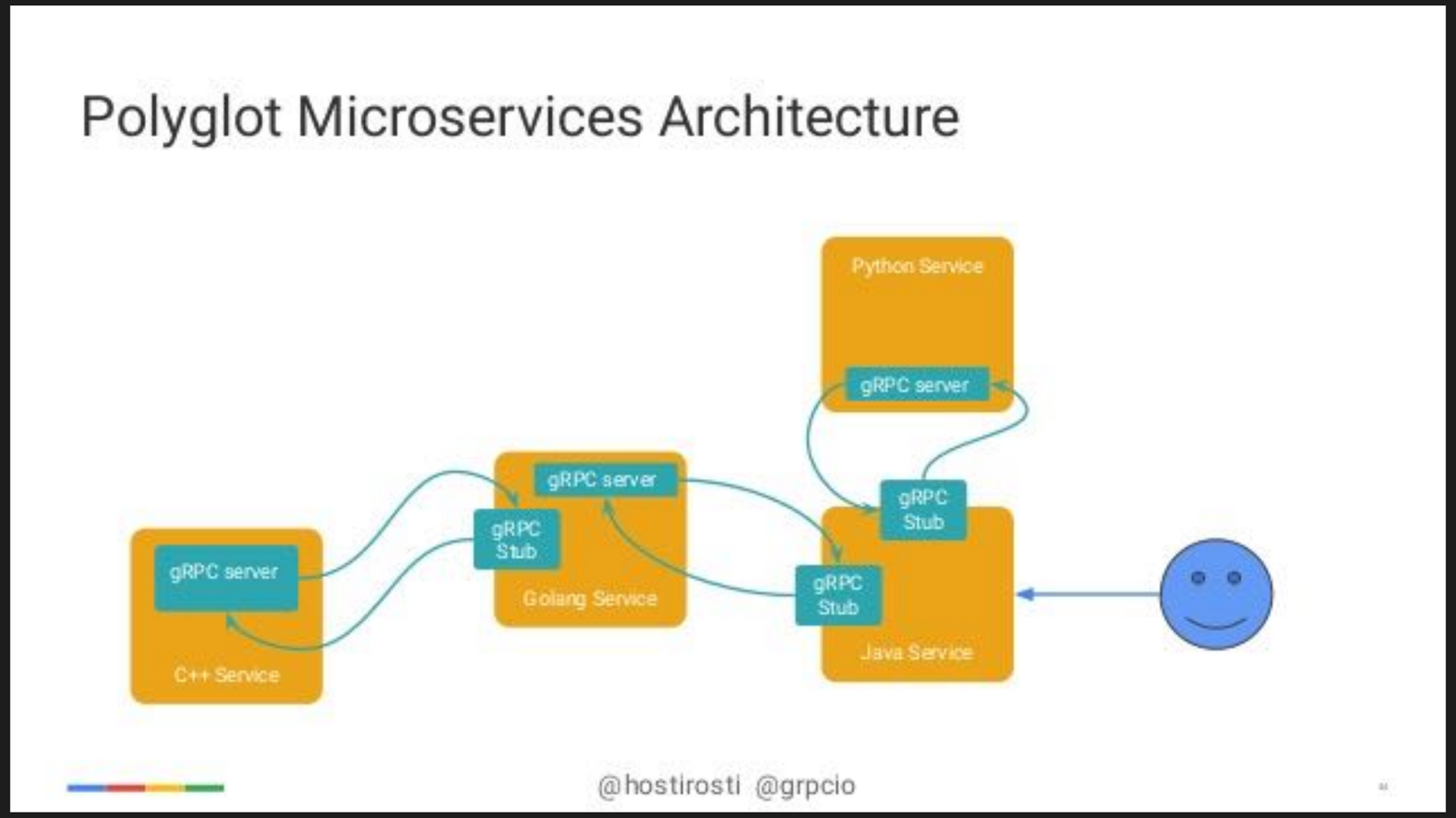
- Open sourced version of Google “Stubby RPC”
- IDL for the service APIs
- “HTTP/2” & “Bi-Directional streaming”!
- Working with Protobuf3
- generated both client and server in 9 languages, officially (others with C language binding are available)

Officially Supported Platforms

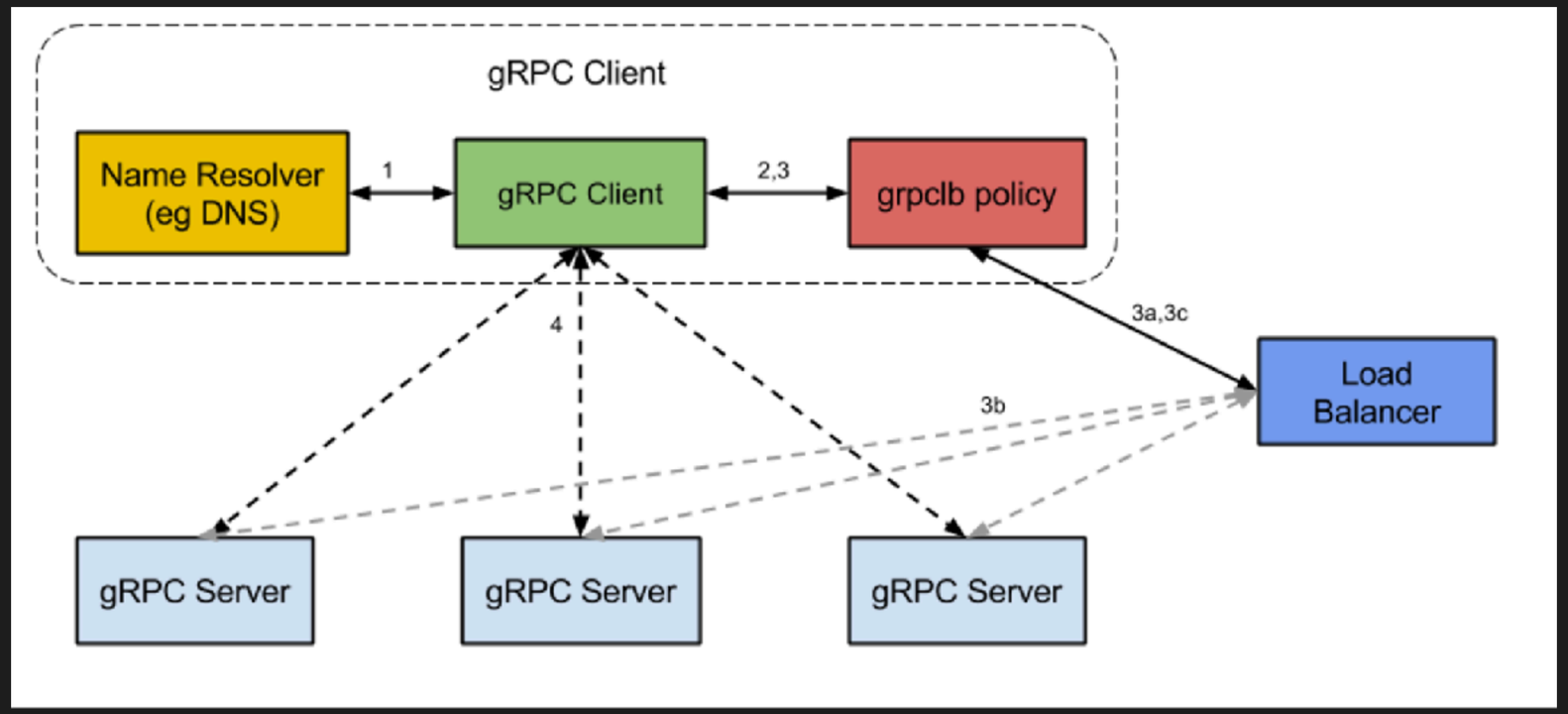
Language	Platform	Compiler
C/C++	Linux	GCC 4.4 GCC 4.6 GCC 5.3 Clang 3.5 Clang 3.6 Clang 3.7
C/C++	Windows 7+	Visual Studio 2013+
C#	Windows 7+ Linux Mac	.NET Core, .NET 4.5+ .NET Core, Mono 4+ .NET Core, Mono 4+
Node.js	Windows/Linux/Mac	Node v4+
PHP *	Linux/Mac	PHP 5.5+ and PHP 7.0+
Ruby	Windows/Linux/Mac	
Python	Windows/Linux/Mac	Python 2.7 and Python 3.4+
Go	Windows/Linux/Mac	Go 1.5+
Java	Windows/Linux/Mac	JDK 8 recommended. Gingerbread+ for Android

** still in beta*

gRCP polyglot Micro-services Architecture (cited @hostirosti)

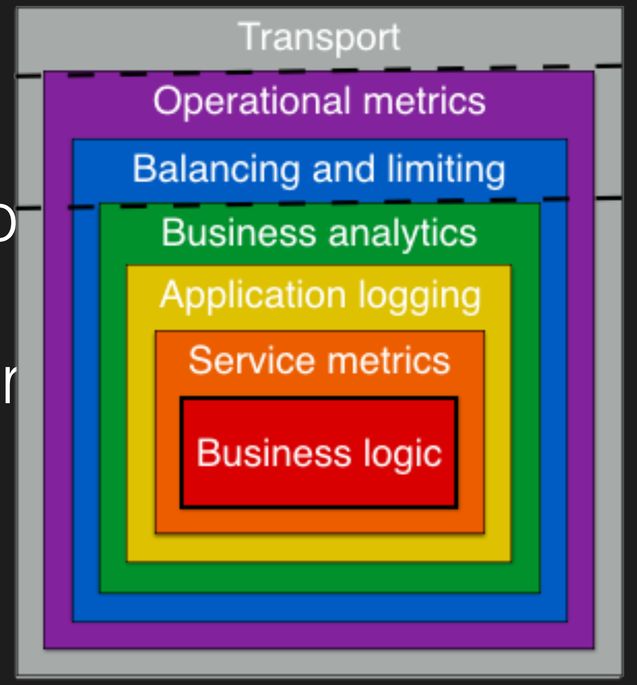


gRPC LB


gRPC Client
Name Resolver
(eg DNS)
1
gRPC Client
2,3
grpclb policy
3a,3c
Load
Balancer
4
3b
gRPC Server
gRPC Server
gRPC Server

Introduction of Go kit (gokit.io)

- A toolkit for micro-services
- Aim for make Go a first-class citizen fo
- It's something that's comparable to our services



Dubbo vs. Go kit

	Dubbo & Spring	Go kit
Service Discovery & LB	Dubbo Registry & Dubbo Subscriber	github.com/go-kit/kit/sd/(zk/consul/etcd/dnssrv/lb) google.golang.org/grpc/naming (lack of structured versioning)
“Structured” Logging	Log4j/Slf4j	github.com/go-kit/kit/log
Metrics	Spring Actuator (many others)	github.com/go-kit/kit/metrics
Circuit Breaker	Dubbo/Netflix Hystrix	github.com/go-kit/kit/circuitbreaker
Transports	HTTP(JSON)/Dubbo/(gRPC)	github.com/go-kit/kit/transport/(grpc/http/httprp)
Caching layer	Dubbo/Spring Cache	-
Distributed Tracing	ELK/(天象全链路)	github.com/go-kit/kit/tracing (OpenTracing project)

Micro-services best practices

- Design with “Single” domain in mind (DB)
- Strong DevOps culture - CI/CD
- Logging, Metrics and Tracing
  - Logging Options - Aliyun Logging Service/Apache Kafka/ELK
  - A trace ID to co-relate all the requests that’s been made
- Transactional requests with idempotence handling in mind/Eventual Consistency
- Think twice if you need to propagate your requests to a number of micro-services request in “parallel”
- Provider services governance and versioning

Micro-services best practices (cont.)

- Circuit Breaker/Fallbacks
- Multi-region cluster/failover
- Employ Container/Docker technologies (DevOps)
  - docker-compose
  - swarm
  - k8s
- Be very careful when introduce a whole new set of framework/library (shoot yourself in the foot)
- SIMPLE is the BEST

Resources

- Dubbo - http://dubbo.io/
- gRPC - http://www.grpc.io/
- Go kit - http://gokit.io/
- 阿里云 RDS - https://www.aliyun.com/product/rds
- 阿里云 API 网关 - https://www.aliyun.com/product/apigateway
- 阿里云 日志服务 - https://www.aliyun.com/product/sls
- 阿里云 EDAS - https://www.aliyun.com/product/edas