



快手 Node.js 监控排障服务的架构与实现





周鸿轩

2019 年加入快手，目前在前端基建团队从事 KNode 与 Node.js 基础组件的相关工作。

曾就职于百度，从事部门内探索业务和基础服务的研发。

<https://github.com/alienzhou> 

alienzhou16@163.com 

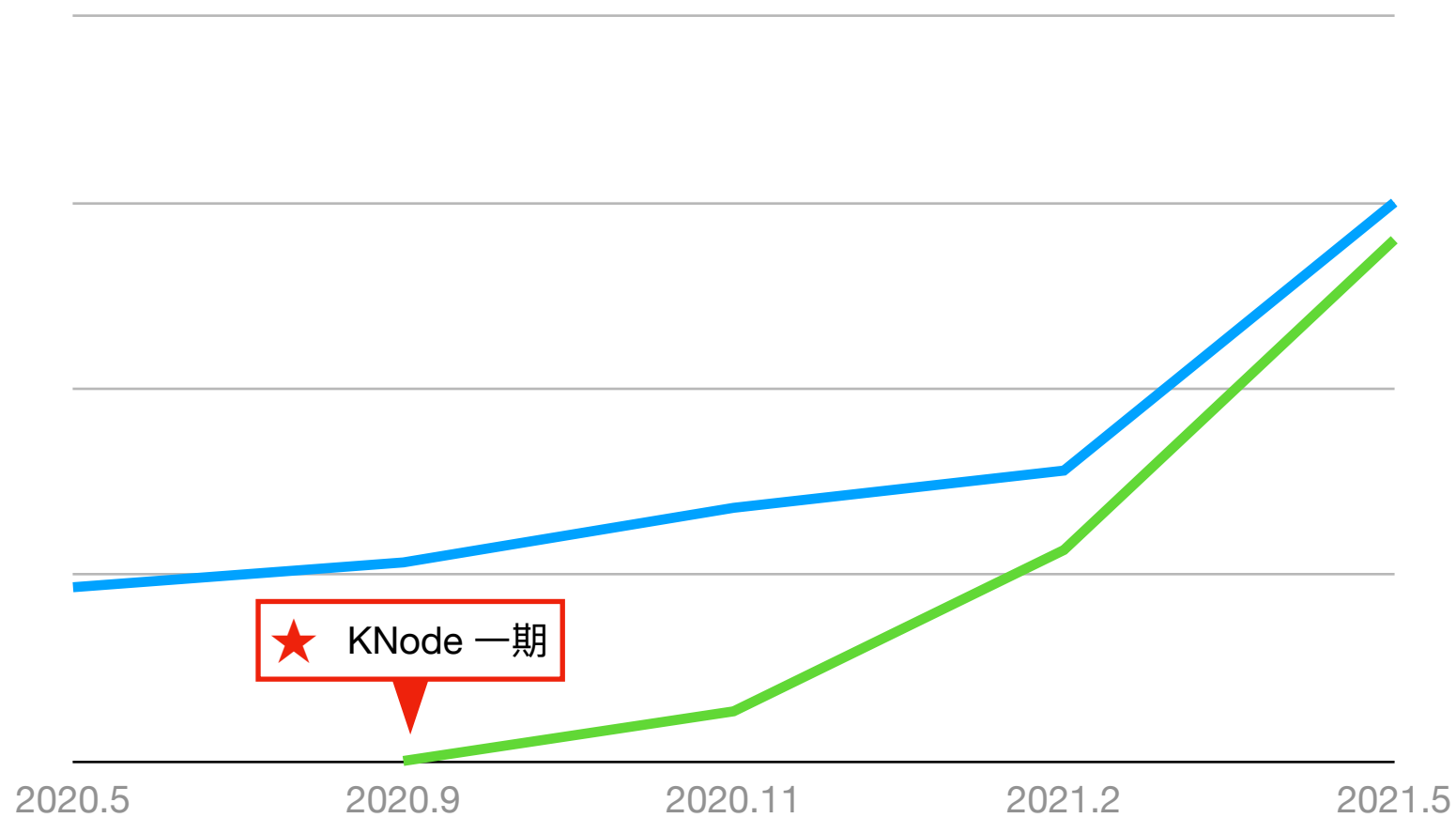
AlienZHOU 知乎

- ☒ 快手 KNode 的发展背景
- ☐ Node.js 运行时采集方案的选型
- ☐ 运行时状态采集的技术实现
- ☐ KNode 服务数据链路 with 整体架构
- ☐ 未来展望

270₊

100_%

95_{%+}



Node.js 服务的数量的快速增长

分布式、微服务等模式对服务链条上
各环节的 **监控能力** 提出了 **更高要求**



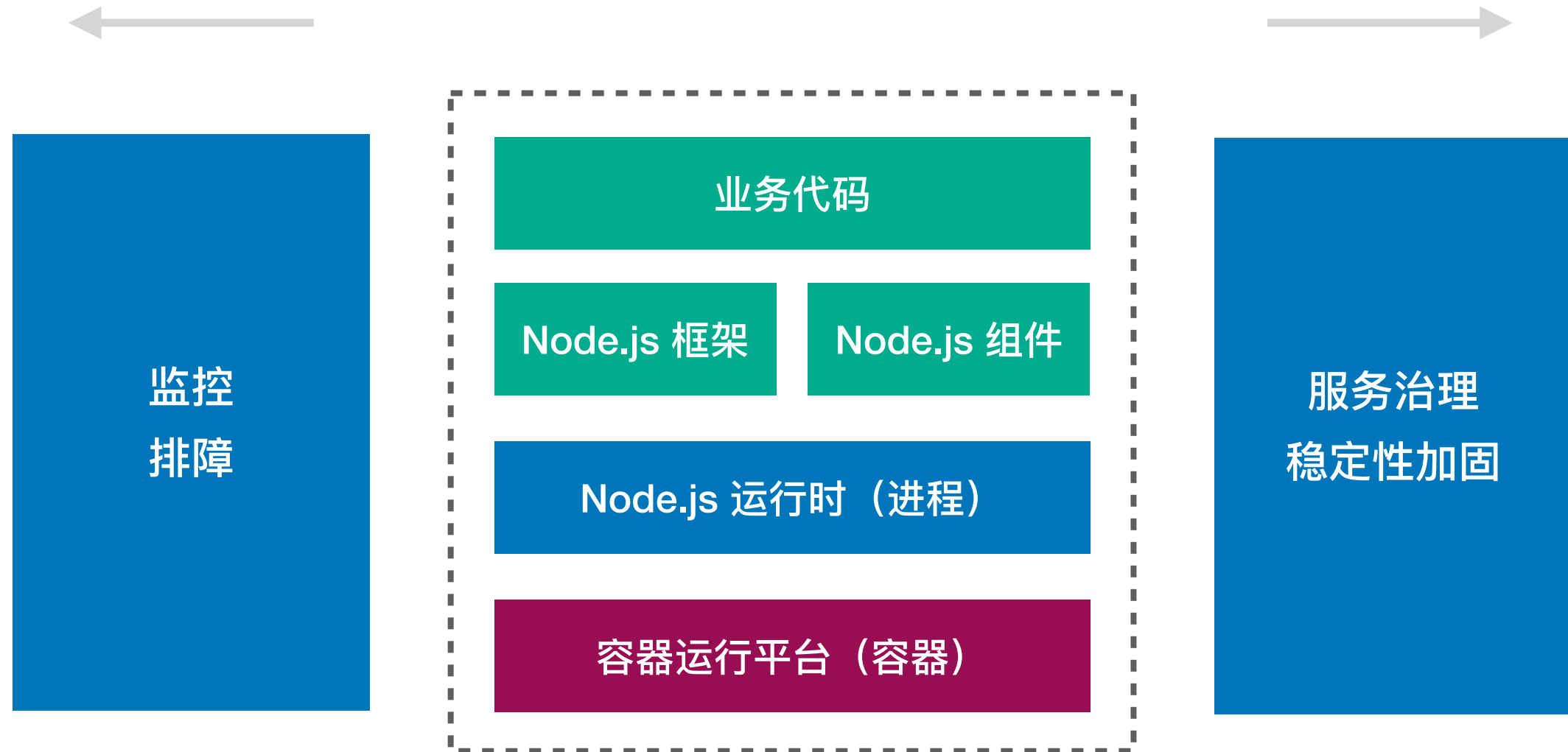
Node.js

Node.js 服务需要

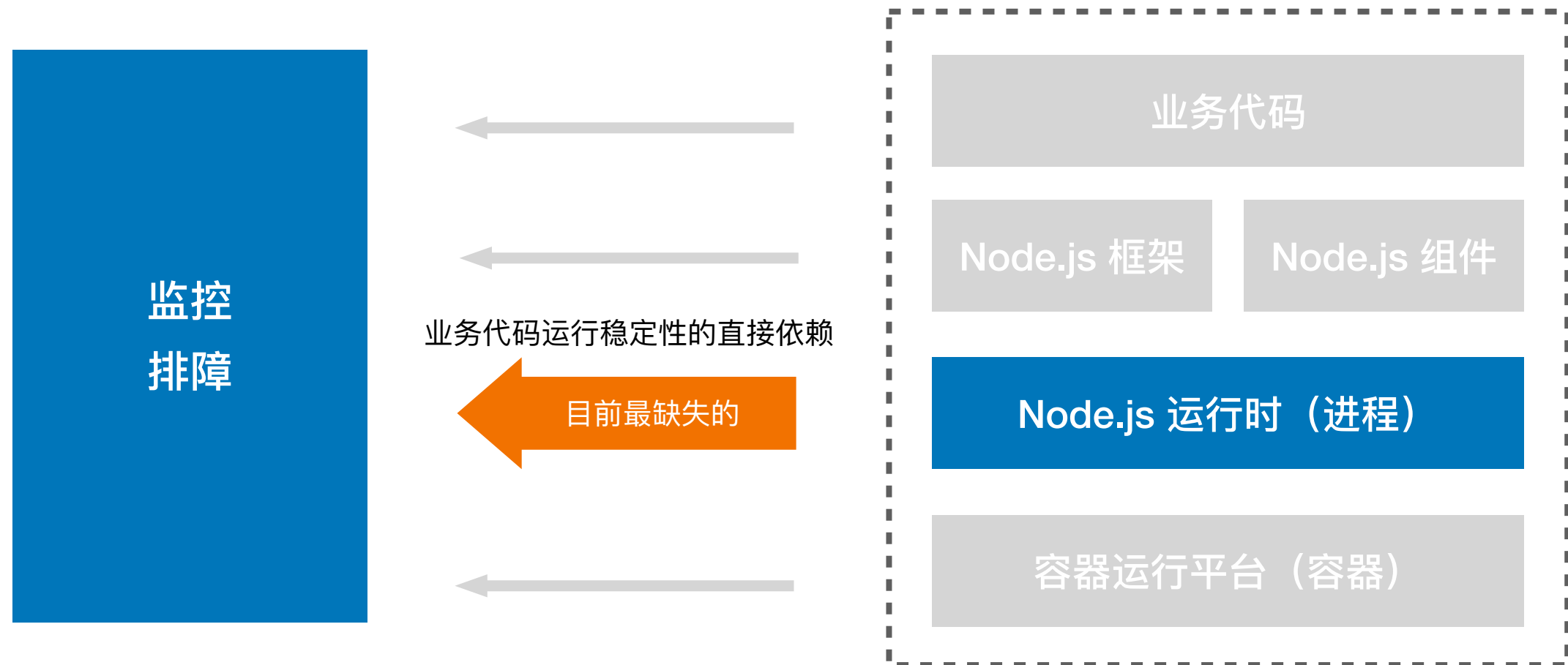
哪些监控与保障?

快手 KNode 发展的背景

Node.js 服务需要哪些监控与保障?



Node.js 服务需要哪些监控与保障?



Node.js 服务需要哪些监控与保障?



内存
泄漏

服务
不响应

CPU
高耗

- ▶ 有没有 Node.js 进程的堆内存 / CPU / Event Loop 等详细监控?
- ▶ 能不能做堆内存 / CPU / Event Loop 等问题的排查?
- ▶ 排查线上问题 快不快, 能否保留现场?

Node.js 服务需要哪些监控与保障?

内存

服务

CPU

对 Node.js 进程内部的运行状态

「一无所知」

- ▶ 能不能 做内存 / CPU / Event Loop 的问题排查?
- ▶ 排查线上问题 快不快, 能否保留现场?

Node.js 服务需要哪些监控与保障?



KNode

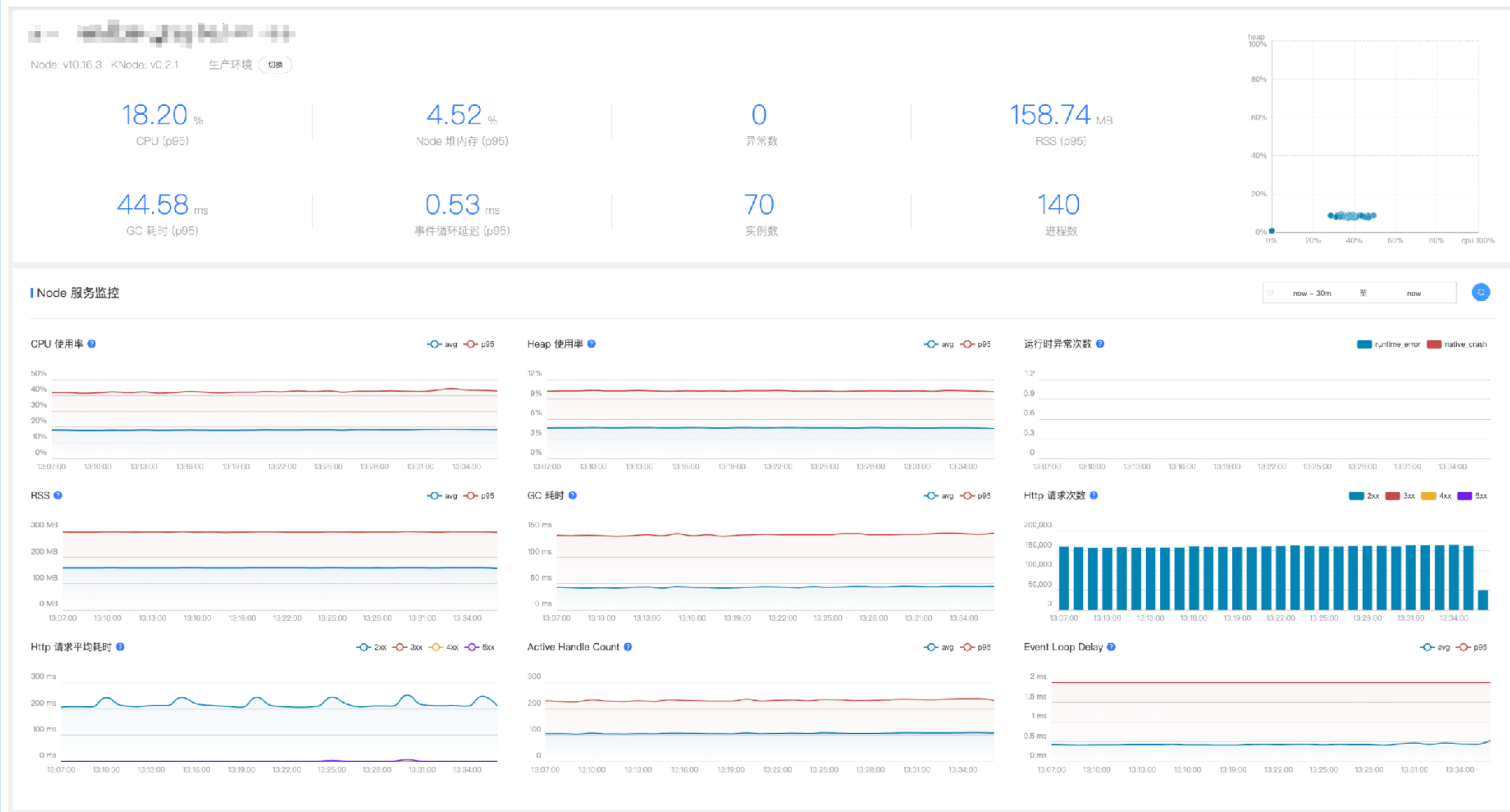
Node.js 运行时维度

进程的基础监控与排障服务

- 定制的 Node.js 运行时 (Node.js 源码定制)
- 业务无感知的监控数据链路 with 存储
- 易用的监控分析平台
-

快手 KNode 发展的背景

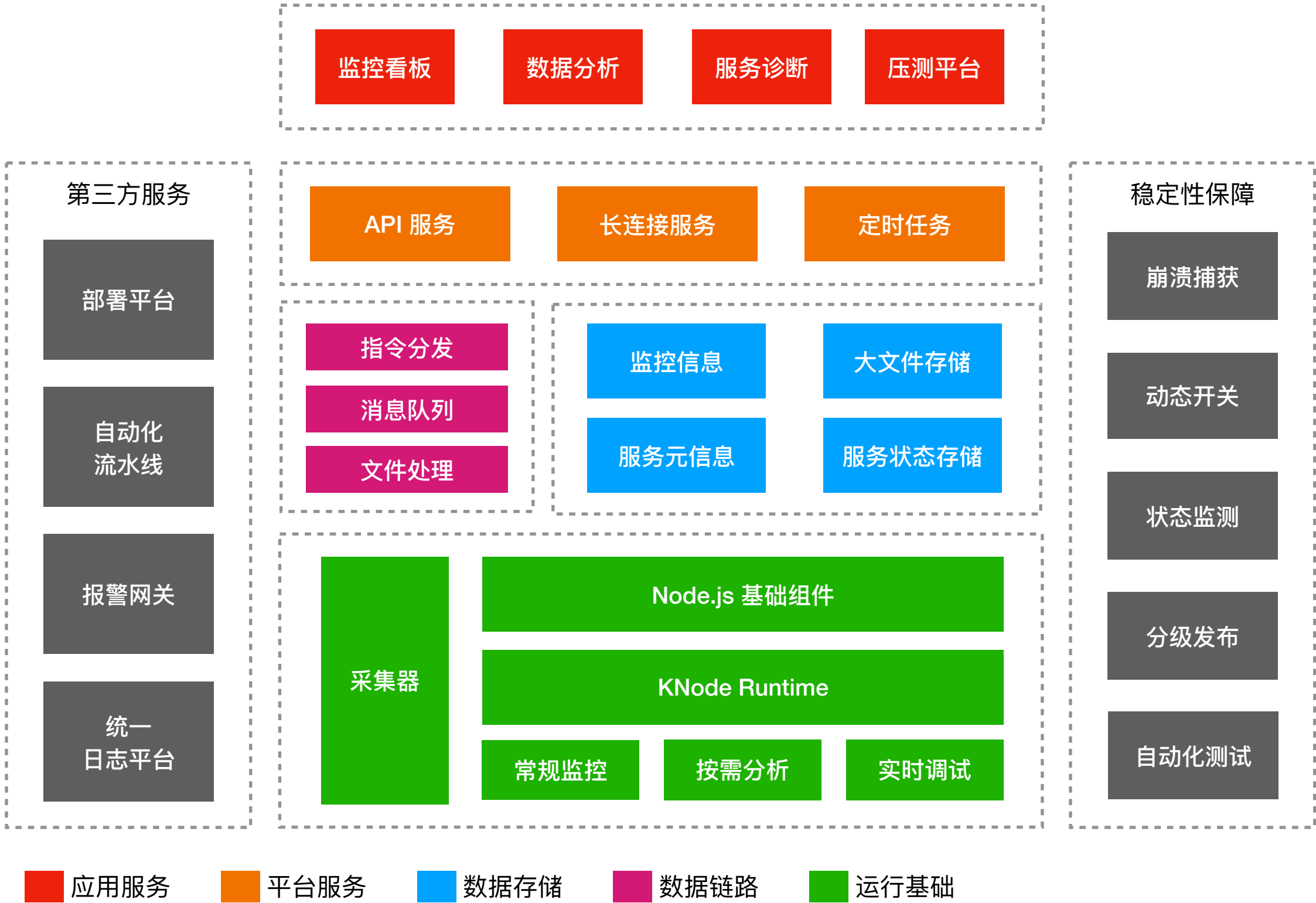
Node.js 服务需要哪些监控与保障?



快手 KNode 发展的背景



Node.js 服务需要哪些监控与保障?



小结：在 Node.js 服务快速增长的情况下，
Node.js 进程维度的基础监控 成为亟待补
齐的短板。

- ☒ 快手 KNode 的发展背景
- ☒ Node.js 运行时采集方案的选型
- ☐ 运行时信息采集的技术实现
- ☐ KNode 整体服务架构
- ☐ 未来展望

一个监控排障服务

由哪些部分组成？

一个监控排障服务由哪些部分组成？



服务分层架构

一个监控排障服务由哪些部分组成？

Node.js 运行时信息的 **采集**
是其监控排障能力的 **核心点**

应用层

监控平台、诊断报告

服务层

API 服务、定时任务等

数据层

数据链路、数据存储

采集层

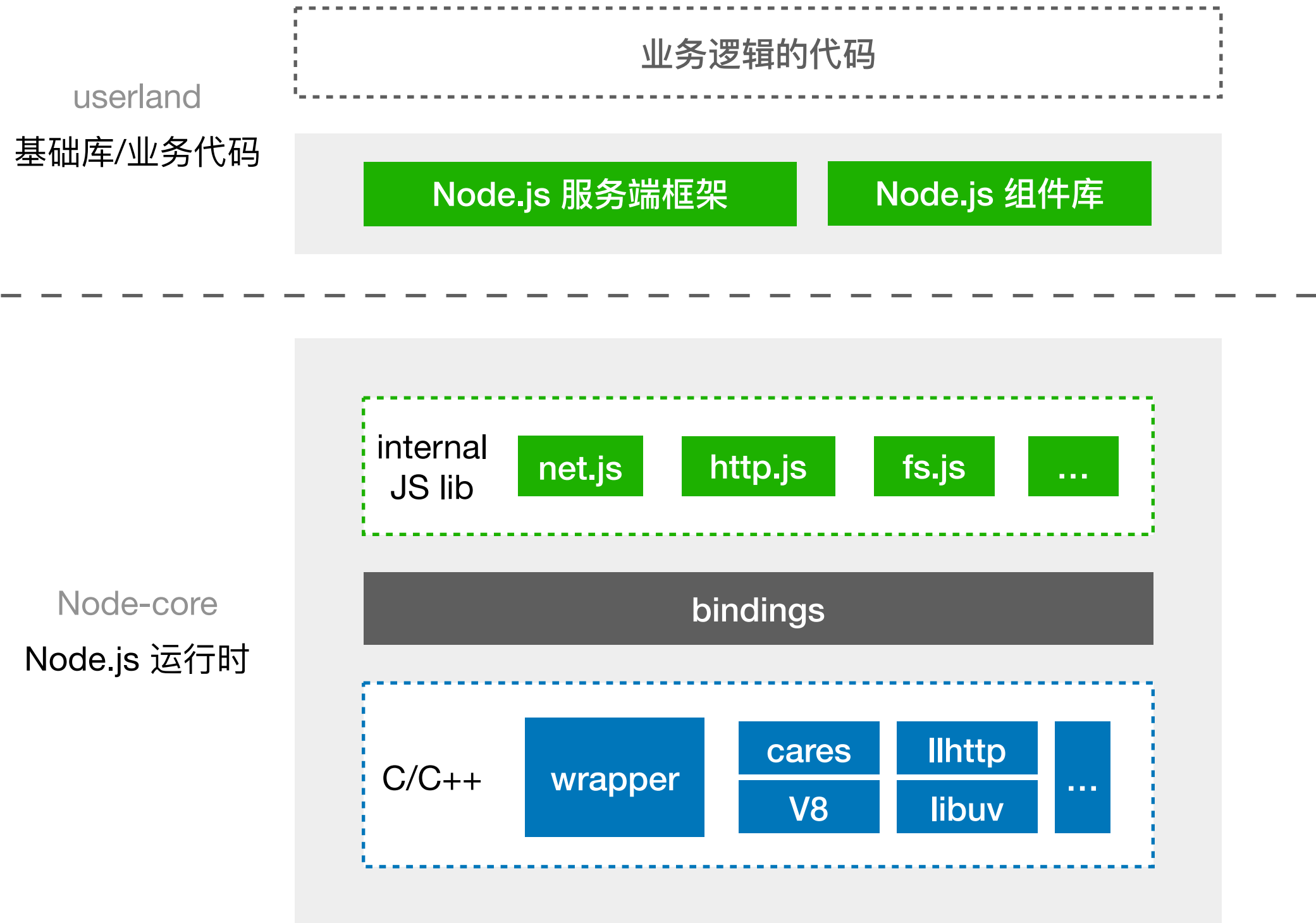
数据生产、数据采集

如何采集

Node.js 运行时的内部状态和信息？

运行时状态采集的技术实现

如何采集 Node.js 运行时的内部状态和信息？



Node-core = Runtime = 运行时

如何采集 Node.js 运行时的内部状态和信息？

在哪个层级来实现采集？

- 1、使用 Node.js 暴露到 JavaScript 层的 API
- 2、使用 C/C++ 的 public API，实现在 add-on 层面
- 3、直接做在 runtime 中，定制 Node.js 源码
- 4、基于操作系统提供的能力获取进程状态信息

如何采集 Node.js 运行时的内部状态和信息？

业务如何接入？

引入第三方模块 或 **runtime 内置✓**

Node.js 运行时采集方案选型

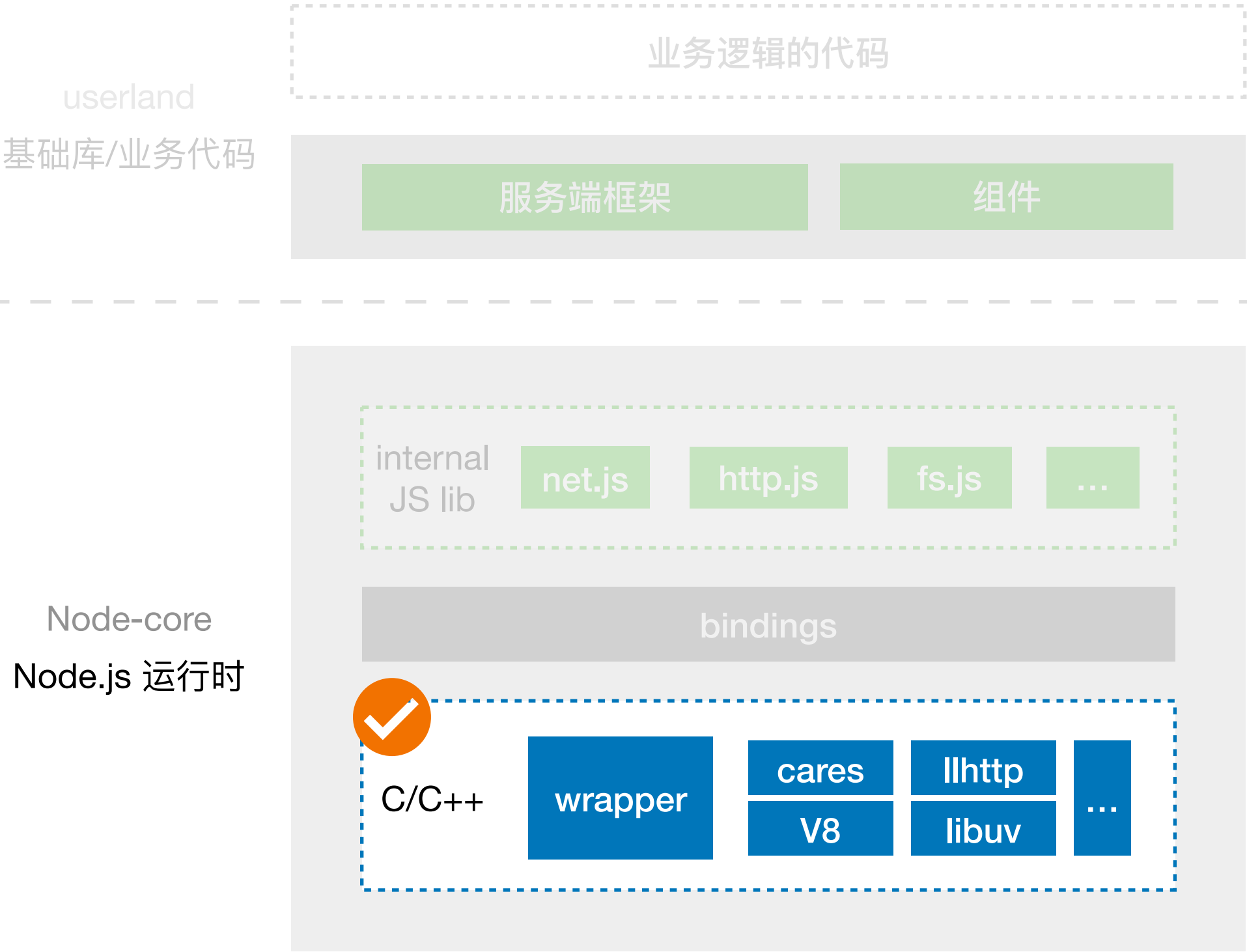


如何采集 Node.js 运行时的内部状态和信息？

高 > 一般 > 低 > 无

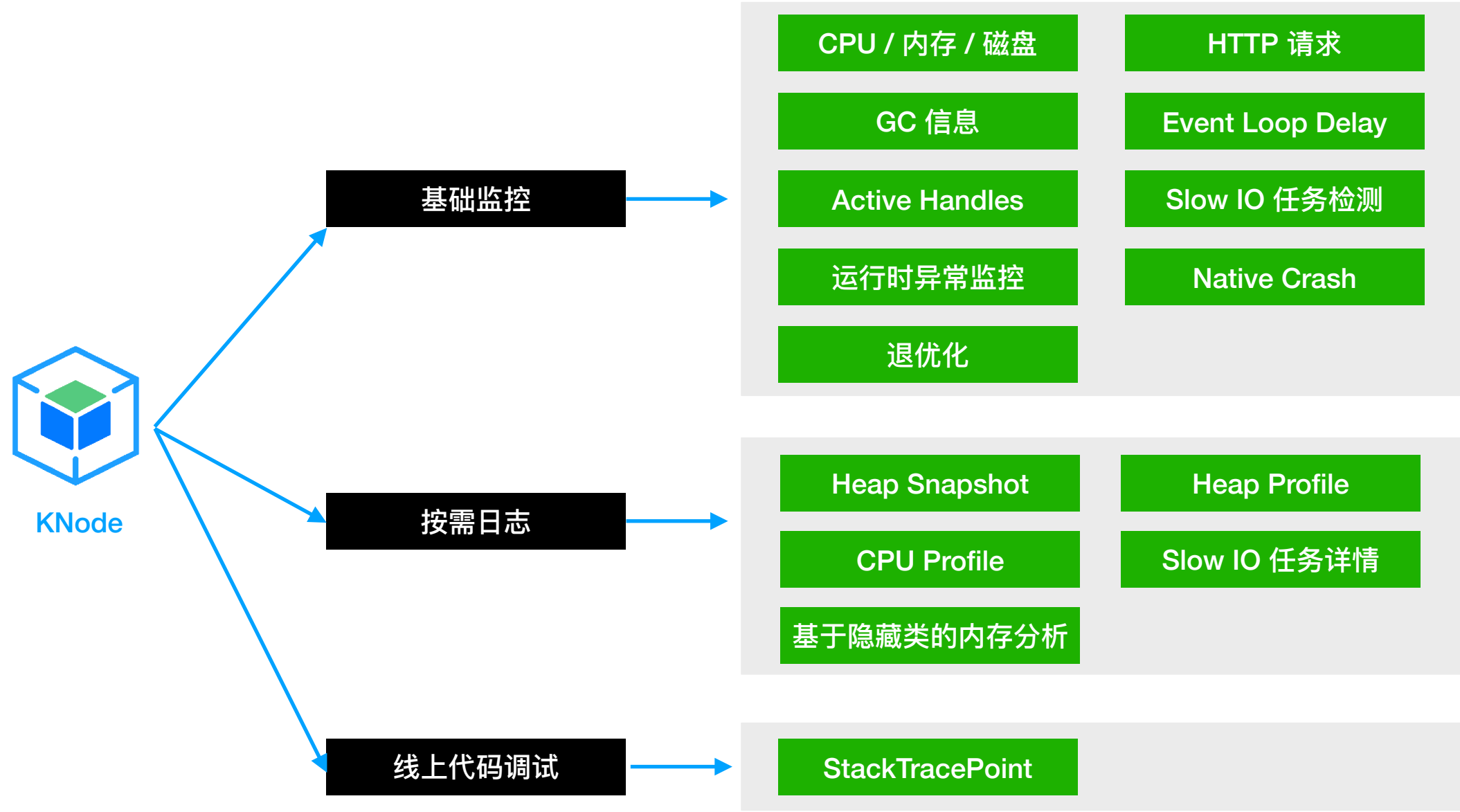
	能力受限度	业务接入成本	维护成本
JS 实现	一般	低	低
add-on 实现	低	低	一般
runtime 定制	无	无	高
操作系统能力	高	低	低~无

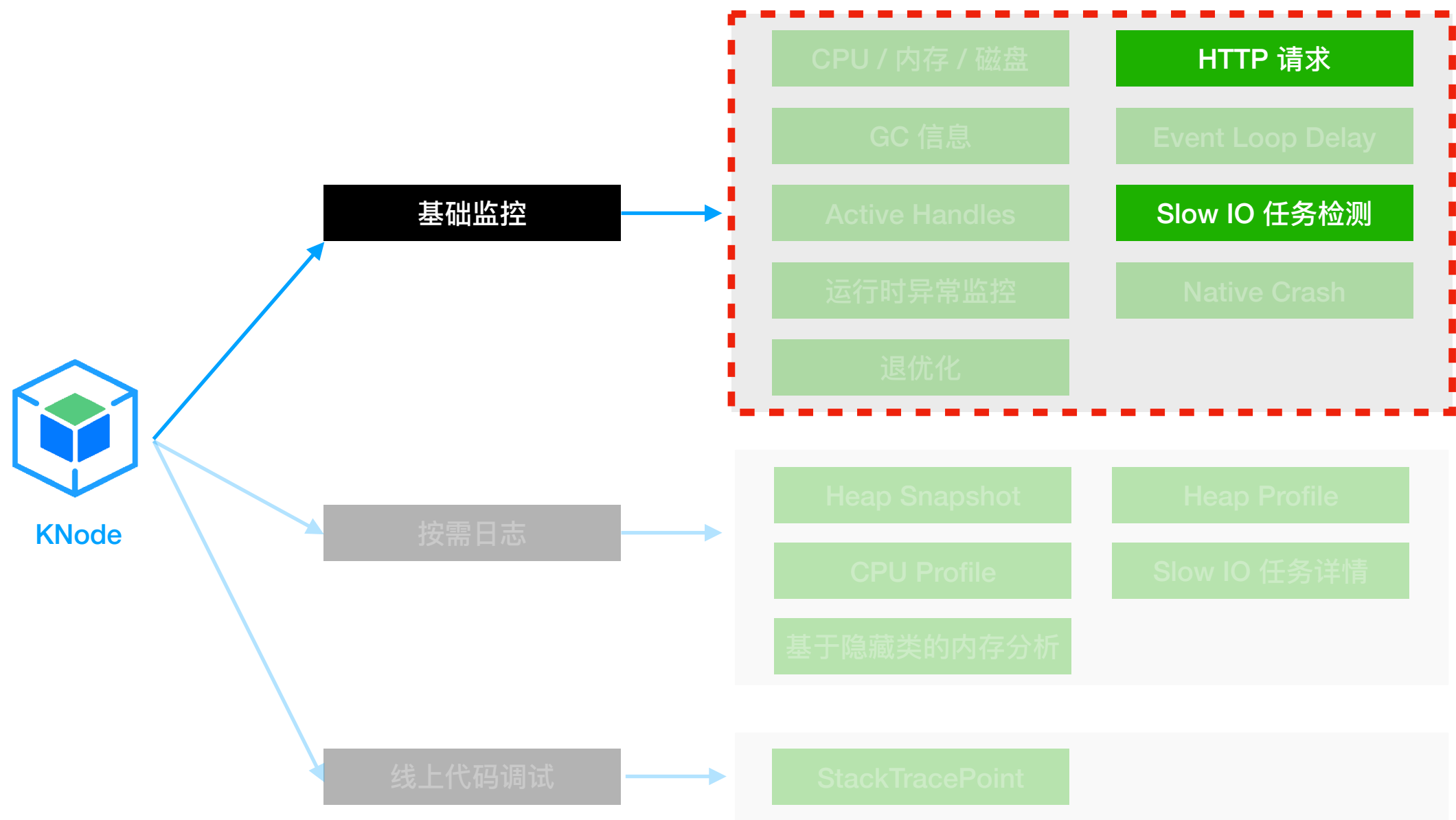
如何采集 Node.js 运行时的内部状态和信息？



小结：KNode 选择 Node.js 源码定制的方案，
来实现 **不受限的能力拓展** 与 **零成本接入**。

- ☒ 快手 KNode 的发展背景
- ☒ Node.js 运行时采集方案选型
- ☒ 运行时状态采集的技术实现
- ☐ KNode 服务数据链路 with 整体架构
- ☐ 未来展望





如何采集

HTTP 请求信息?

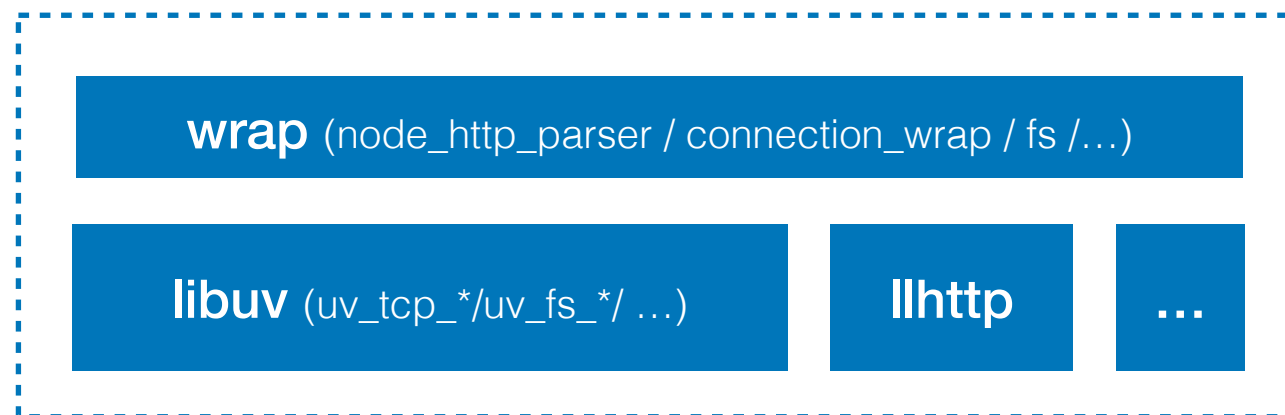
如何采集 HTTP 请求信息?

Node-core
internal js lib

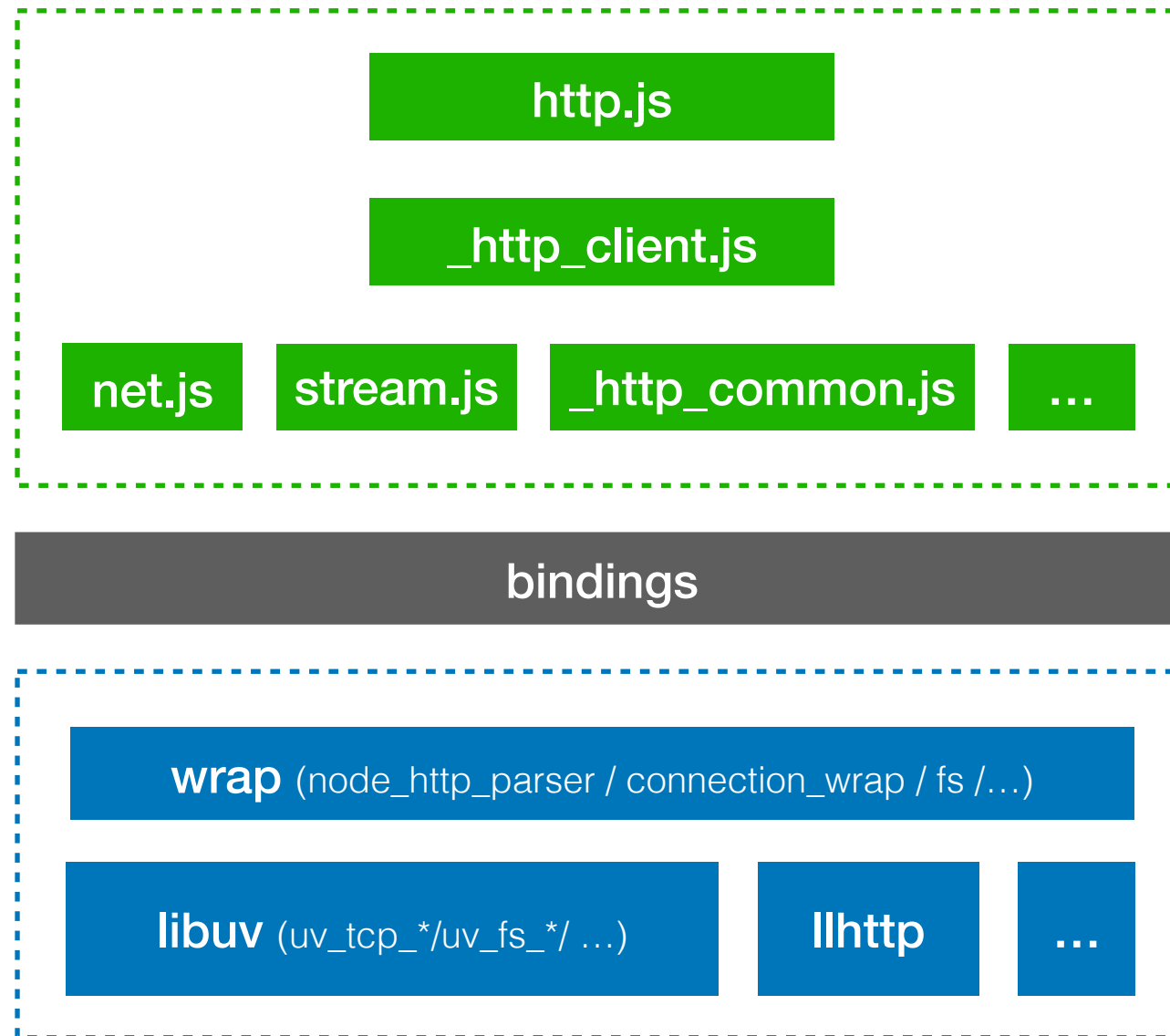


bindings

Node-core
C/C++



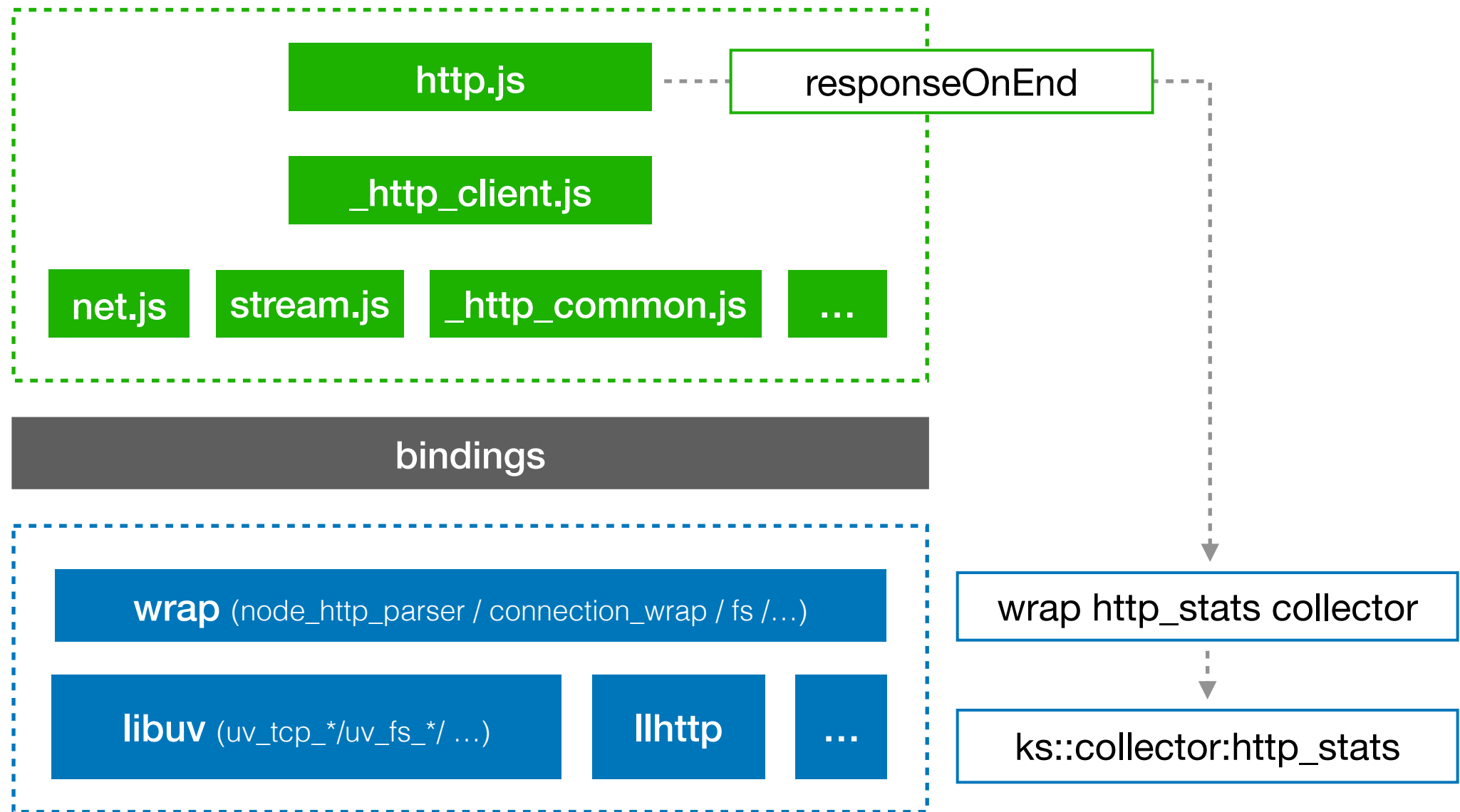
如何采集 HTTP 请求信息？



- DNS lookup
- **HTTP Parse**
- Stream

- TCP connection

如何采集 HTTP 请求信息？

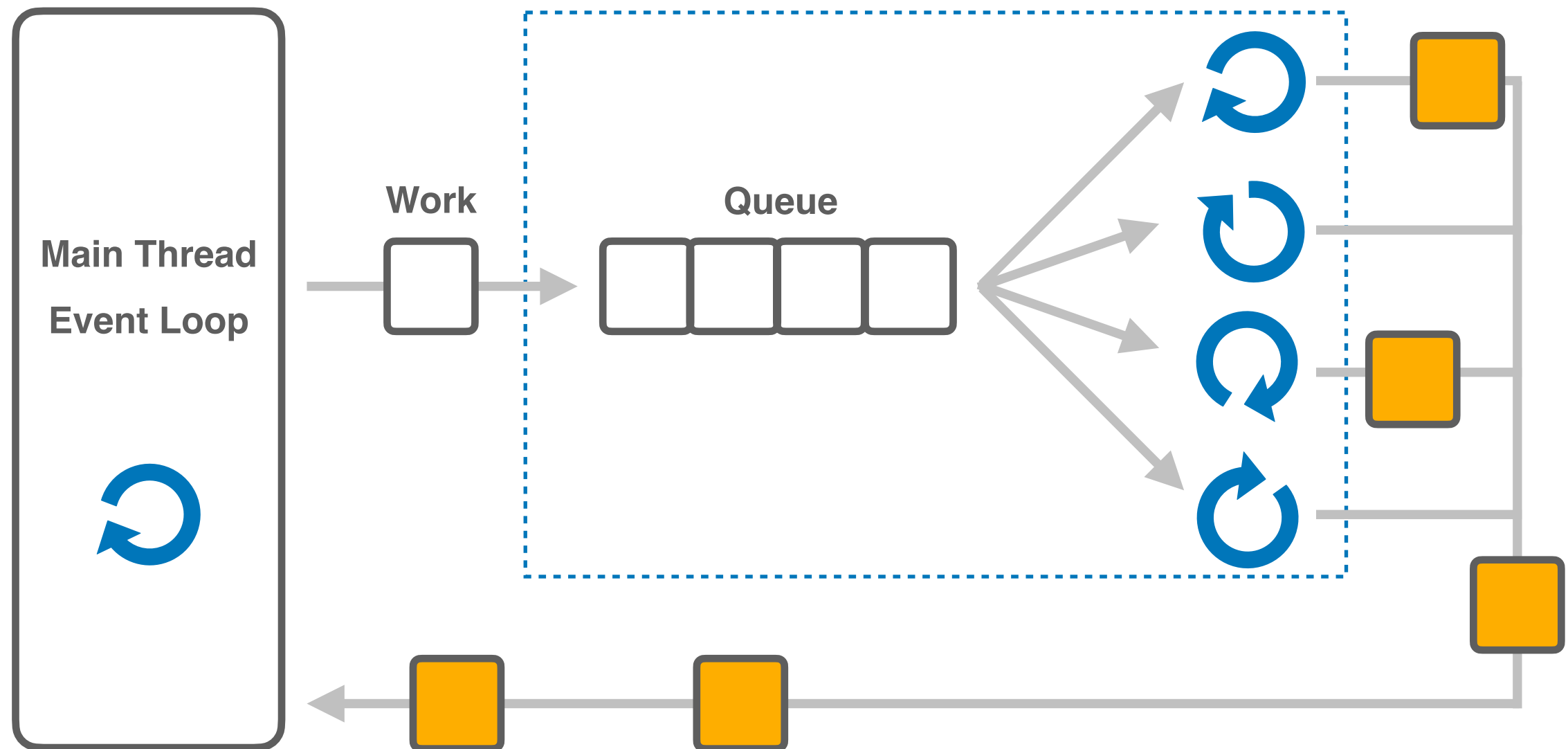


如何采集

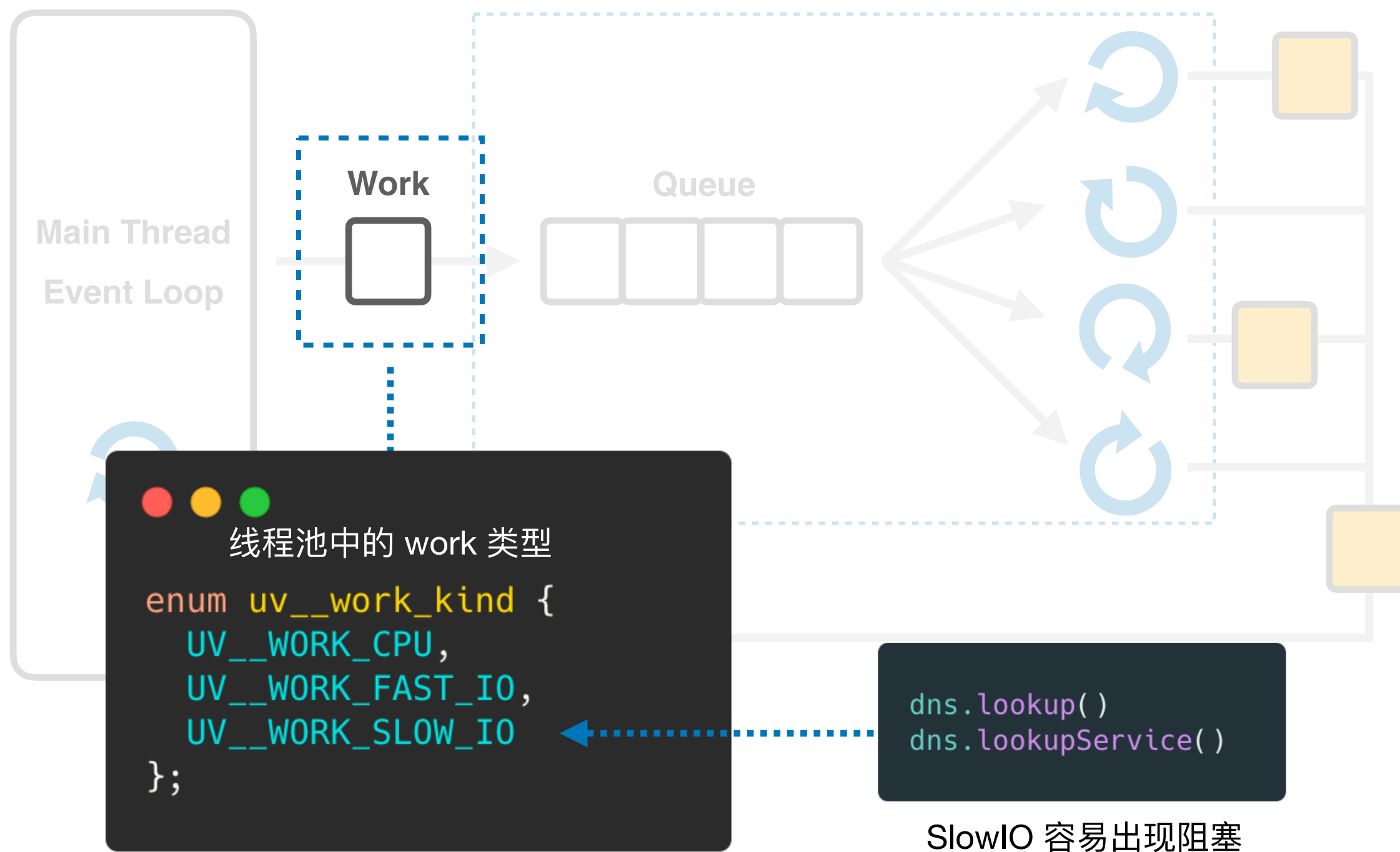
Slow IO 信息?

如何采集 Slow IO 信息？

libuv 中也会使用 **线程池** 来实现某些异步 IO



如何采集 Slow IO 信息?

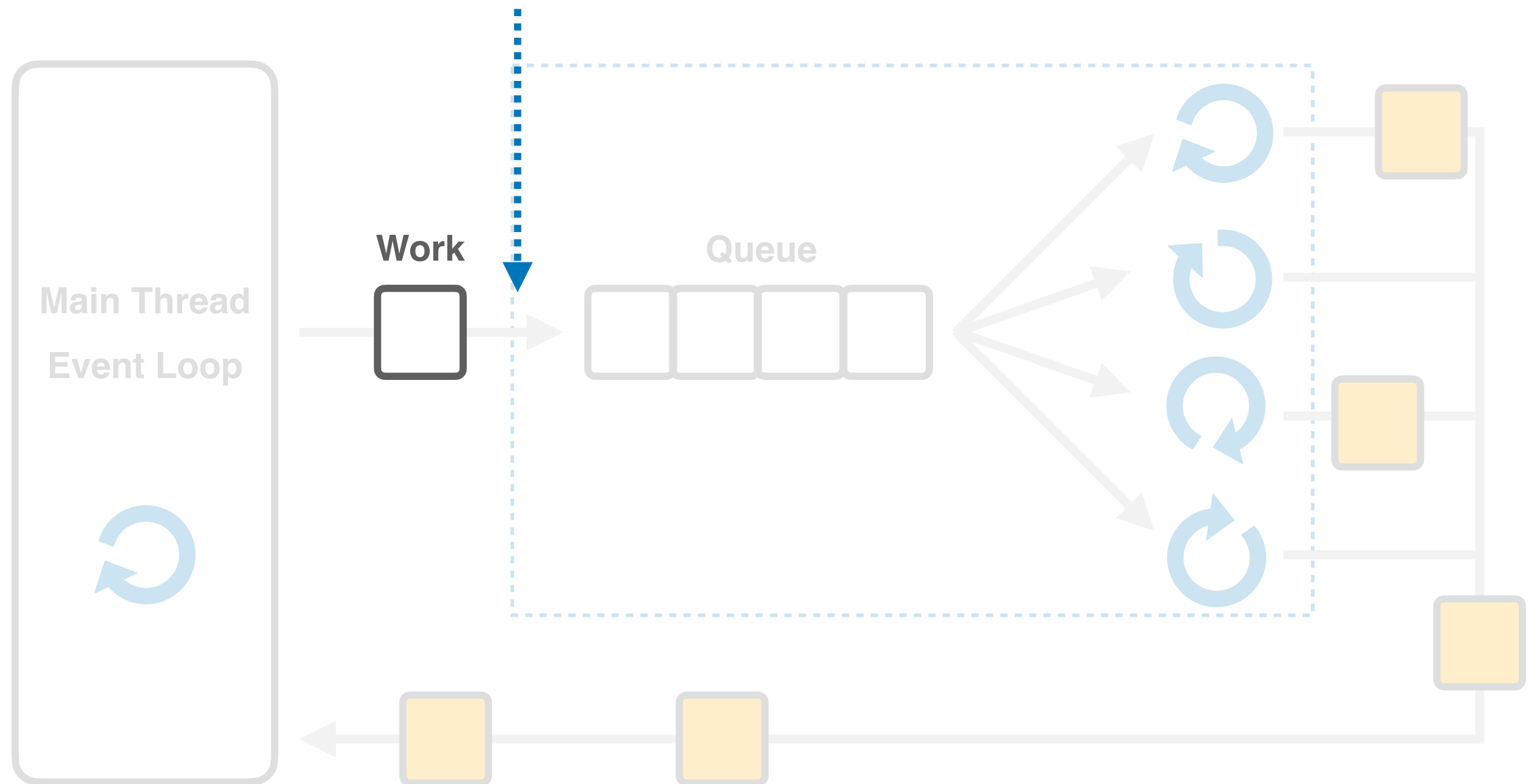


一个案例：DNS 查询导致的 Nodejs 服务疑似“内存泄漏”问题

<https://zhuanlan.zhihu.com/p/373685562>

如何采集 Slow IO 信息？

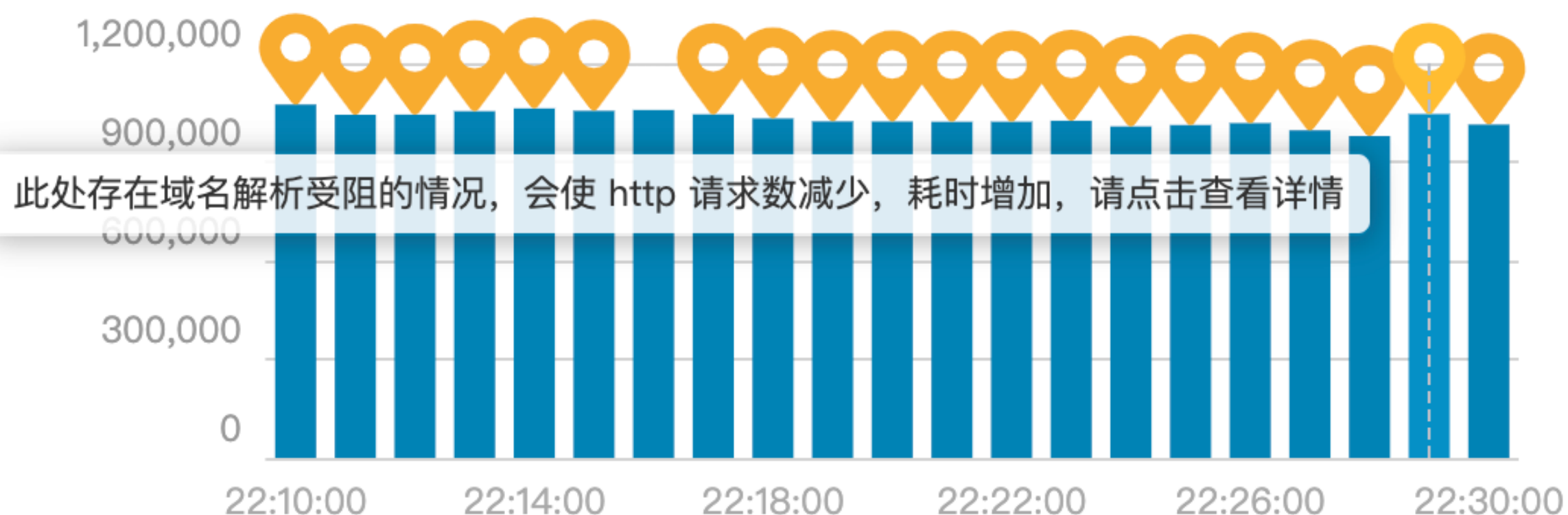
每次有新的 SlowIO work 插入后，检查当前 running 和 pending 情况，如果 **超过阈值** 则进行采集。



【一个真实案例】采集 Slow IO 信息可以排查什么问题？

Http 请求次数 ?

2xx 3xx 4xx 5xx



此处存在域名解析受阻的情况，会使 http 请求数减少，耗时增加，请点击查看详情

通过 Slow IO 监控可以直接定位到这类故障

【一个真实案例】 采集 Slow IO 信息可以排查什么问题？

21:52:01

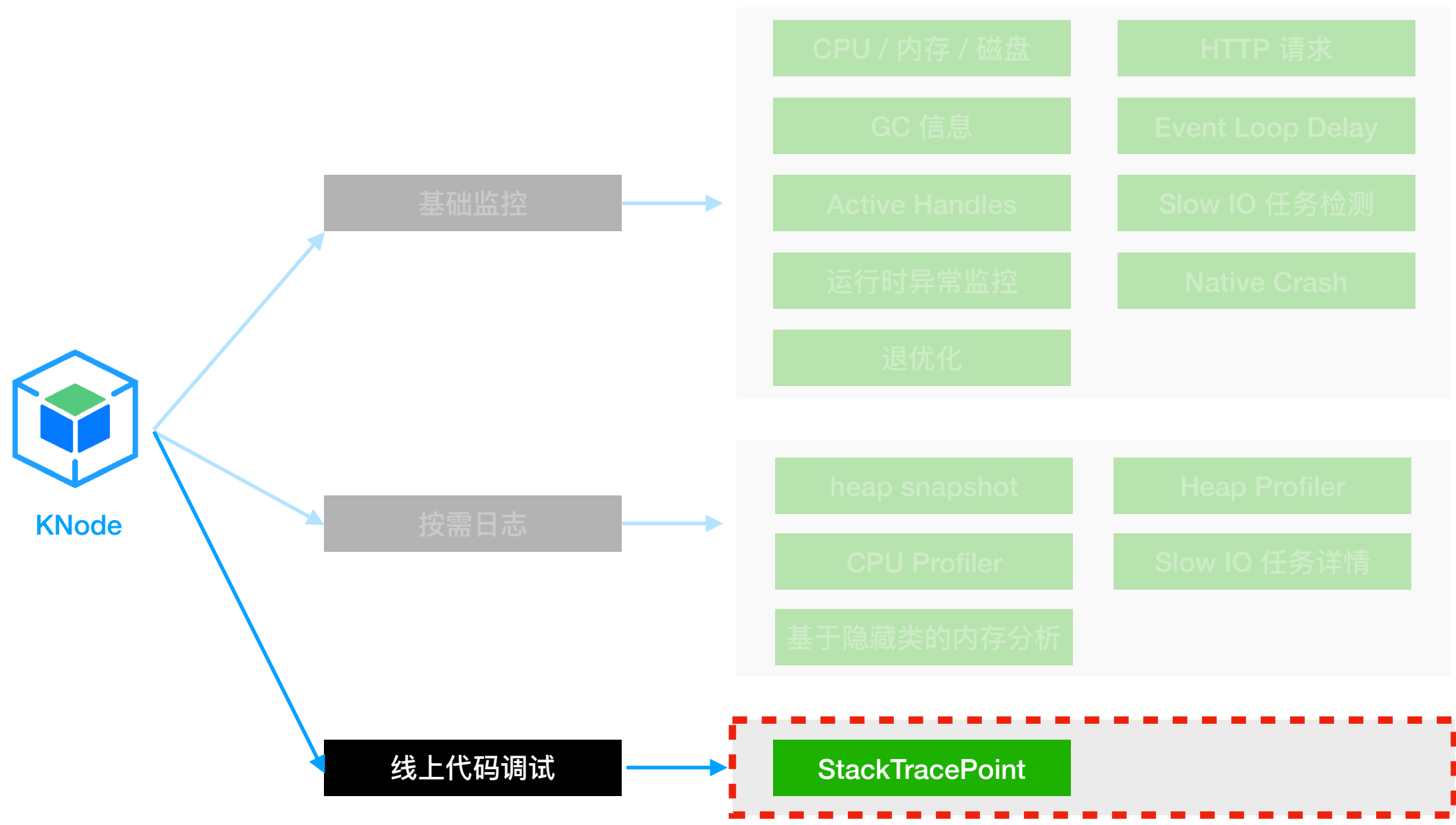
线程池大小
4

执行中及待执行任务总数
21

①仅展示线程池中前20个任务

#	任务名	内容	等待时长 (μs)
1	dns.lookup	sd.web-api.internal	4025677
2	dns.lookup	font.mobile-security.i nternal	3782256
3	dns.lookup	font.mobile-security.i	3248498

DNS 高耗情况一览无余



如何对线上服务做

在线代码调试?

如何对线上服务做在线代码调试？

实时调试信息

connect

Trace Point 信息

退出调试模式

实例名: cloud-1205247-206943-142256-11-deploy-c-b8dbd9dc9-lsxkm 进程号: 1744

latest 2021/5/30下午1:57:48 39: 311e56fd-07c5-43b2-9a1c-493547cfa918

点击查看更多 ...

eachMessage at consumer.js.map:31

2021/5/30下午1:57:46 37: 4a6a99b6-048d-4cac-8728-4997418adc5e

点击查看更多 ...

eachMessage at consumer.js.map:31

调用栈和作用域信息

2021/5/30下午1:57:43 35: aaff4165-82f1-4f8a-8520-f21996c72106

点击查看更多 ...

eachMessage at consumer.js.map:31

2021/5/30下午1:57:42 34: 8eff9fb1-c151-4e95-a490-8bc9ac26f501

点击查看更多 ...

eachMessage at consumer.js.map:31

2021/5/30下午1:57:46 37: 4a6a99b6-048d-4cac-8728-4997418adc5e

Call Stack:

eachMessage	consumer.js.map:32
processEachMessage	runner.js:151
onBatch	runner.js:287
(anonymous)	runner.js:339
invoke	concurrency.js:38
push	concurrency.js:51
(anonymous)	concurrency.js:60
(anonymous)	concurrency.js:60
(anonymous)	runner.js:329
(anonymous)	runner.js:328
processTicksAndRejections	task_queues.js:93
async function (async)	

Scope:

Block (eachMessage)
rwaMessage: {"time":"2021-05-30 13:57:05.397","logger":"v12.18.3-knode-"
Local (eachMessage)
topic: "klog_knode_data"
partition: 7
message: {...}
Closure
kafka_1: {...}
logger_1: {...}
publisher_1: {...}
util_1: {...}

2021/5/30下午1:57:39 32: 07cc98

eachMessage at consumer.js.map:31

2021/5/30下午1:57:39 31: 6d1d77

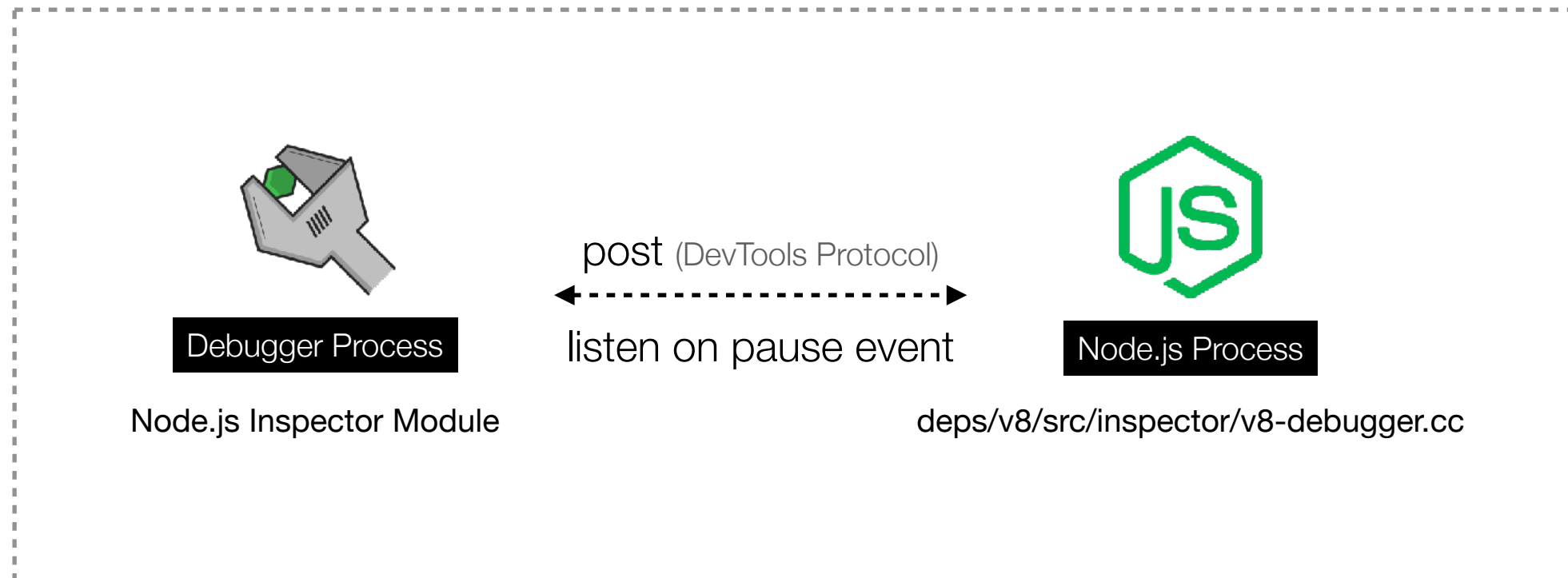
eachMessage at consumer.js.map:31

2021/5/30下午1:57:35 29: 5414d5

eachMessage at consumer.js.map:31

2021/5/30下午1:57:36 28: 20fb9a

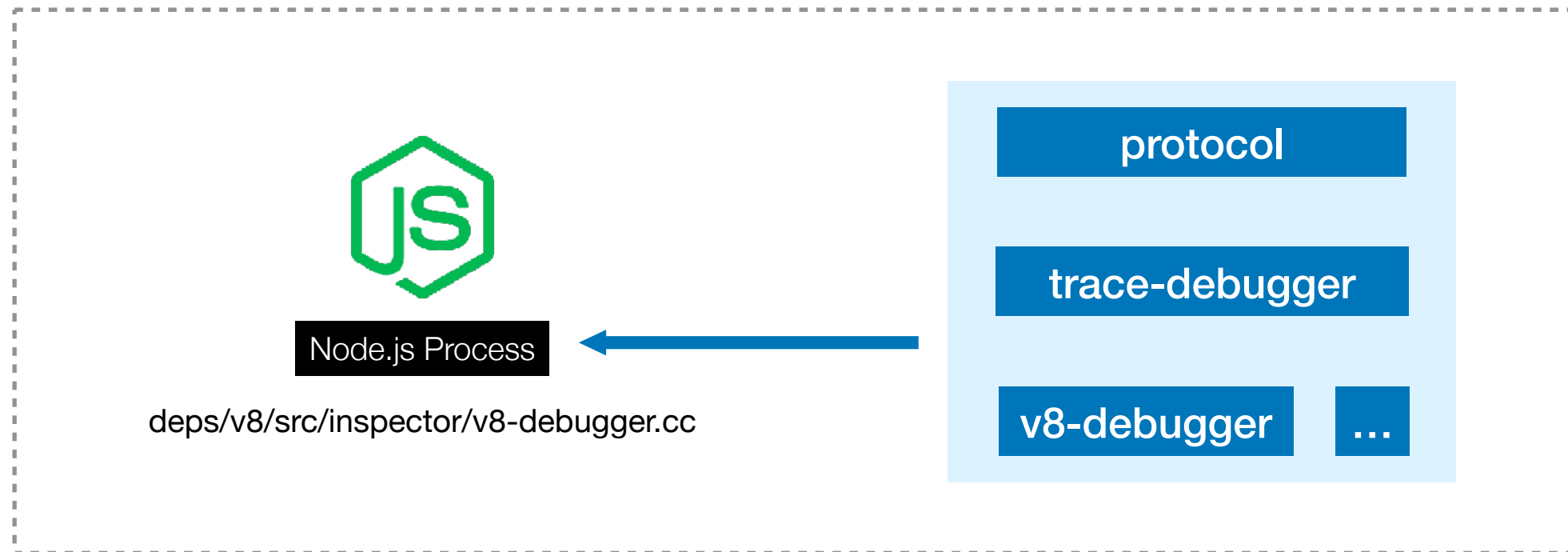
如何对线上服务做在线代码调试？



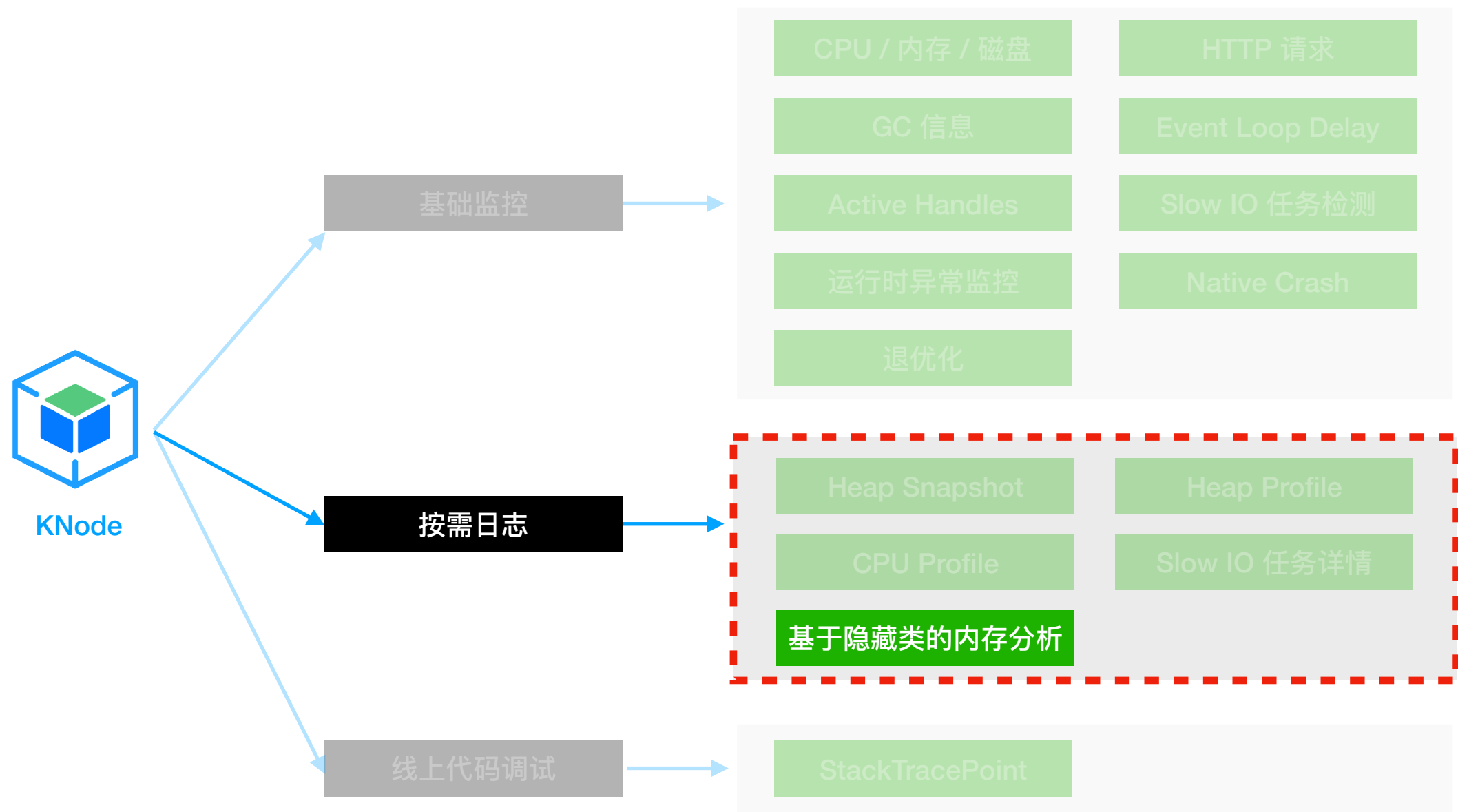
已有的 Break Point 方式

会阻断程序运行

如何对线上服务做在线代码调试？



实现无阻塞的 Trace Point



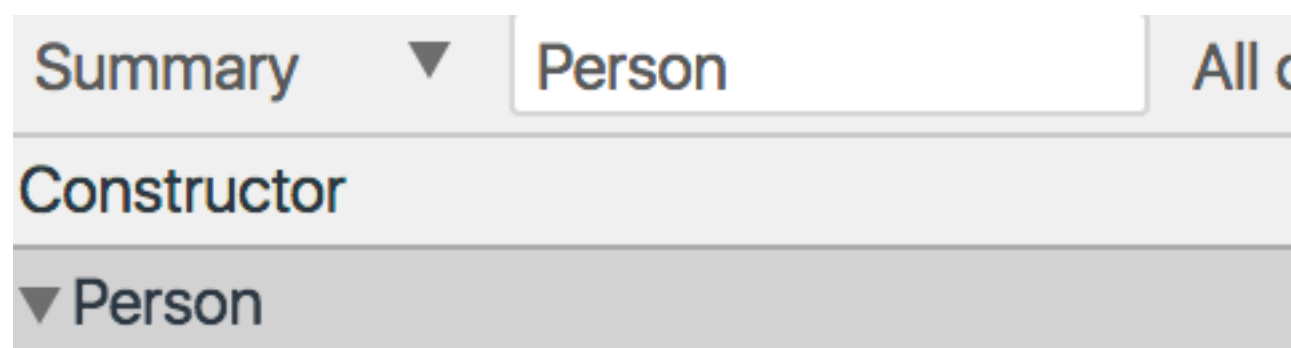
如何实现

基于隐藏类的内存分析?

如何实现基于隐藏类的内存分析？



如何实现基于隐藏类的内存分析？



```
▼ Person @84415
  ► __proto__ :: @84407
  ► map :: system / Map @148831
  ► name :: "yonehara" @85857
▼ Person @84421
  ► __proto__ :: @84407
  ► map :: system / Map @148831
  ► name :: "suzuki" @85859
```

隐藏类
hidden class

如何实现基于隐藏类的内存分析？

#	数据简介	数据大小(KB) ▾	出现次数 ▾	出现位置	操作
1	0x391aeef42031 <JSFunction isAsync (sfh = 0x67f234764a1)>	676	10809	 /node_modules/async/internal/wrapAsync.js:16:17	查看数据值
2	0x391aeef40781 <JSArray[9]>	314	10060	—	查看数据值
3	0x364612d51911 <Object map = 0x17398e827719>	552	8832	—	查看数据值
4	0x391aeef42311 <JSFunction nestLikeConsoleFormat (sfh = 0x2670d1463b09)>	426	7793	 'node_modules/nest-winston/dist/winston.utilities.js:20:31	查看数据值
5	0x391aeef410c1 <JSFunction apply (sfh = 0x288dd8e4331)>	382	6114	 'node_modules/lodash/lodash.js:471:17	查看数据值
6	0x391aeef6adc1 <Object map = 0x3a17ff54dc79>	232	5938	—	查看数据值
7	0x391aeef6ac69 <JSFunction encrypt (sfh = 0x1f1dc710b9d9)>	138	2518	 /node_modules/@infra-node/key-center/lib/core/provider.js:21:18	查看数据值

查看数据

数据类型: object

数据值:

{...}

rule: "inderoy.no"

suffix: "inderoy.no"

punySuffix: -1

wildcard: false

exception: false

① 类型为 array 和 object 的数据最多展开3层，超出的部分展示为 [...]

关闭

如何实现基于隐藏类的内存分析？

#	数据简介	数据大小(KB) ▾	出现次数 ▾	出现位置	操作
1	0x391aeef42031 <JSFunction isAsync (sfh = 0x67f234764a1)>	676	10809	/node_modules/async/internal/wrapAsync.js:16:17	查看数据值
2	0x391aeef40781 <JSArray[9]>	314	10060	-	查看数据值
3	0x364612d51911 <Object map = 0x17398e827719>	552	8832	-	查看数据值
4	0x391aeef42311 <JSFunction nestLikeConsoleFormat (sfh = 0x2670d1463b09)>	426	7793	'node_modules/nest-winston/dist/winston.utilities.js:20:31	查看数据值
5	0x391aeef410c1 <JSFunction apply (sfh = 0x288dd8e4331)>	382	6114	'node_modules/lodash/lodash.js:471:17	查看数据值
6	0x391aeef6adc1 <Object map = 0x3a17ff54dc79>	232	5938	-	查看数据值
7	0x391aeef6ac69 <JSFunction encrypt (sfh = 0x1f1dc710b9d9)>	138	2518	/node_modules/@infra-node/keycenter/lib/core/provider.js:21:18	查看数据值

- ★ 较为轻量级（耗时 ↓ 30%）
- ★ 生成的“快照”体积小（↓ 90%+）

hiiden.hiddenclassmem 19 KB 修改时间：今天 下午 4:10	h.heapsnapshot 41.1 MB 修改时间：今天 下午 4:11
--	---

如何实现基于隐藏类的内存分析？



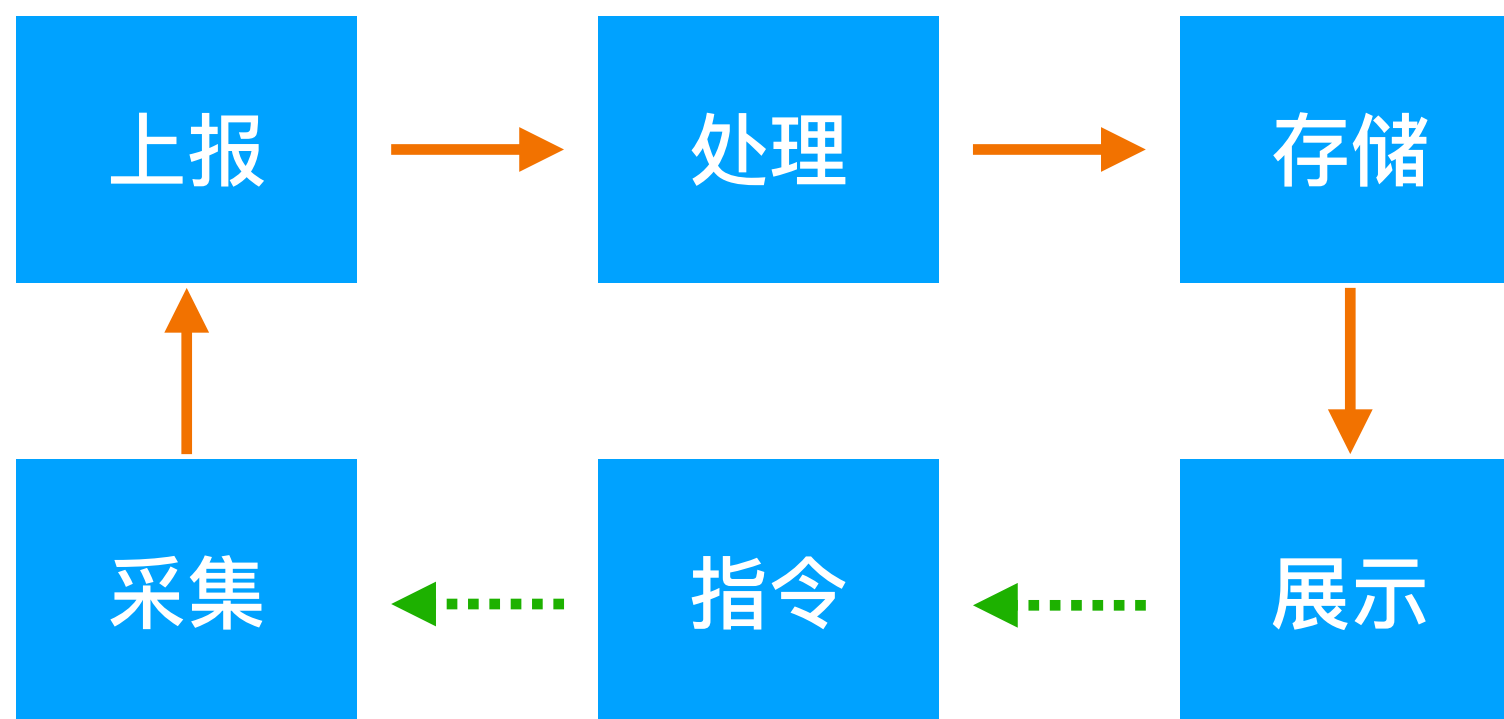
```
LargeObjectSpaceObjectIterator obj_it(heap_>old_space());  
for (HeapObject obj = obj_it.Next(); !obj.is_null(); obj = obj_it.Next()) {  
    if (obj.IsJSObject()) {  
        Map map = object.map();  
        ...  
    }  
}
```

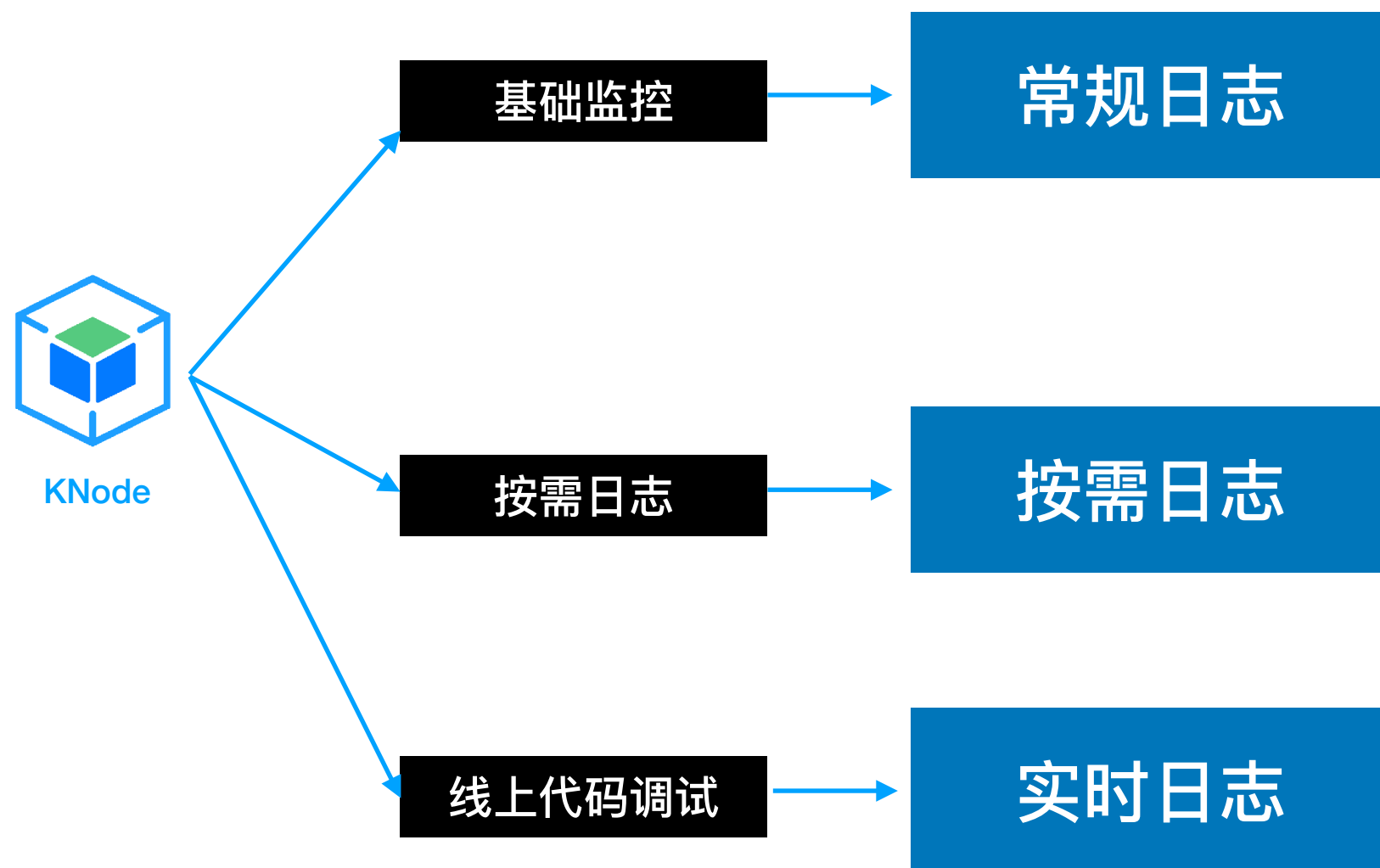
遍历 old space 中的 object, 获取对应的 map

- ☒ 快手 KNode 的发展背景
- ☒ Node.js 运行时采集方案选型
- ☒ 运行时状态采集的技术实现
- ☒ KNode 数据链路 with 整体架构
- ☐ 未来展望

服务的数据链路

监控排障系统的基本流程模型



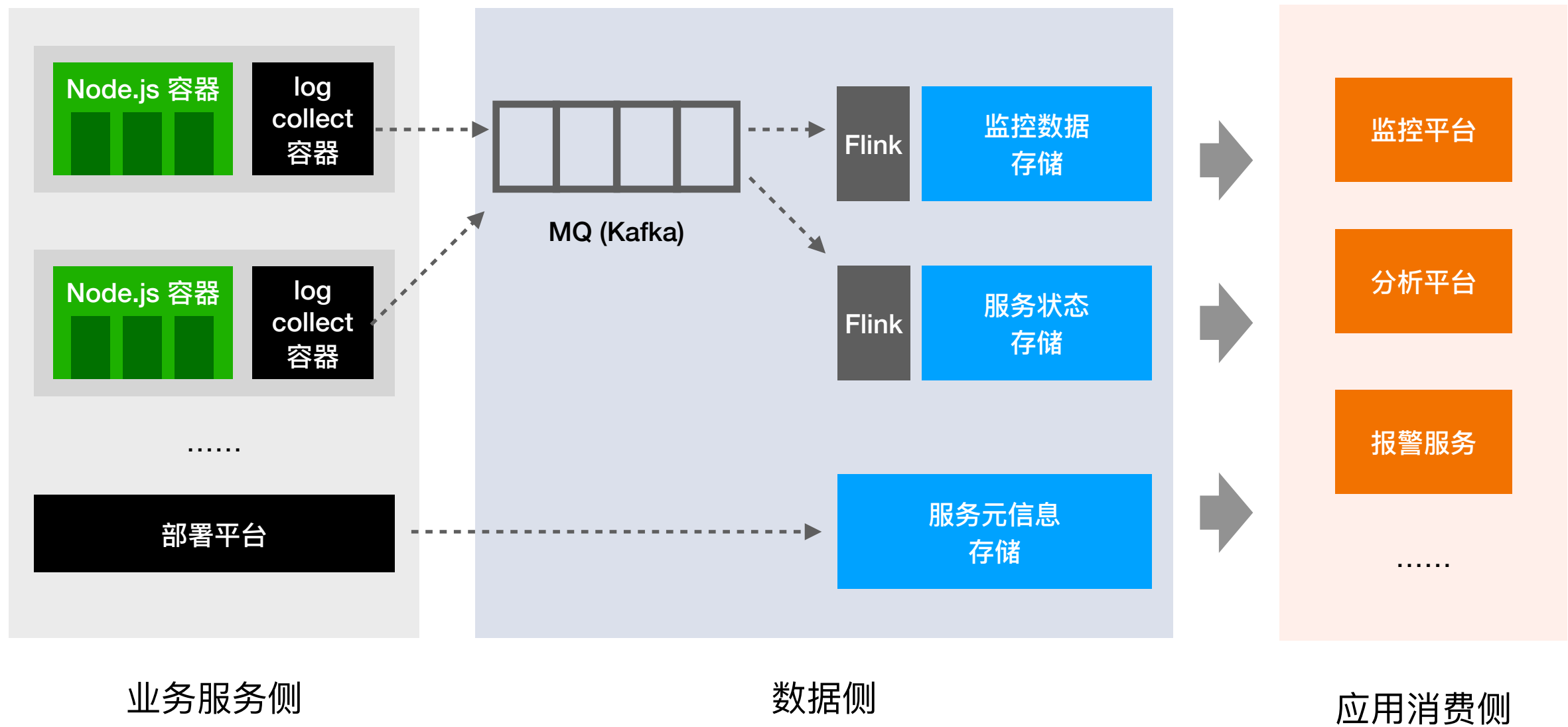


日常监控：如何实现

常规日志的监控链路？

KNode 服务数据链路与整体架构

如何实现常规日志的监控链路?

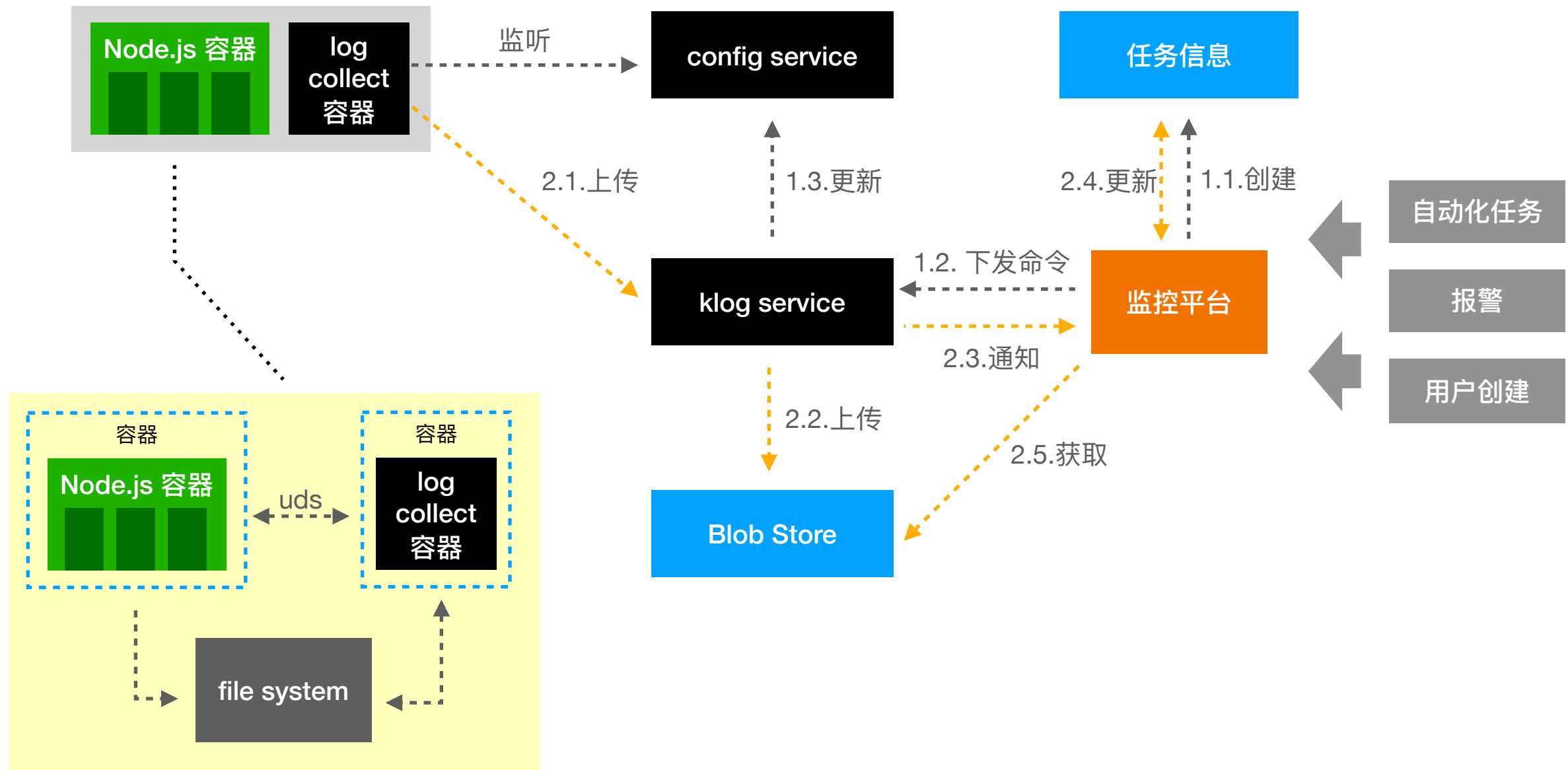


按需分析任务：如何实现

按需日志的监控链路？

KNode 服务数据链路与整体架构

如何实现按需日志的监控链路？

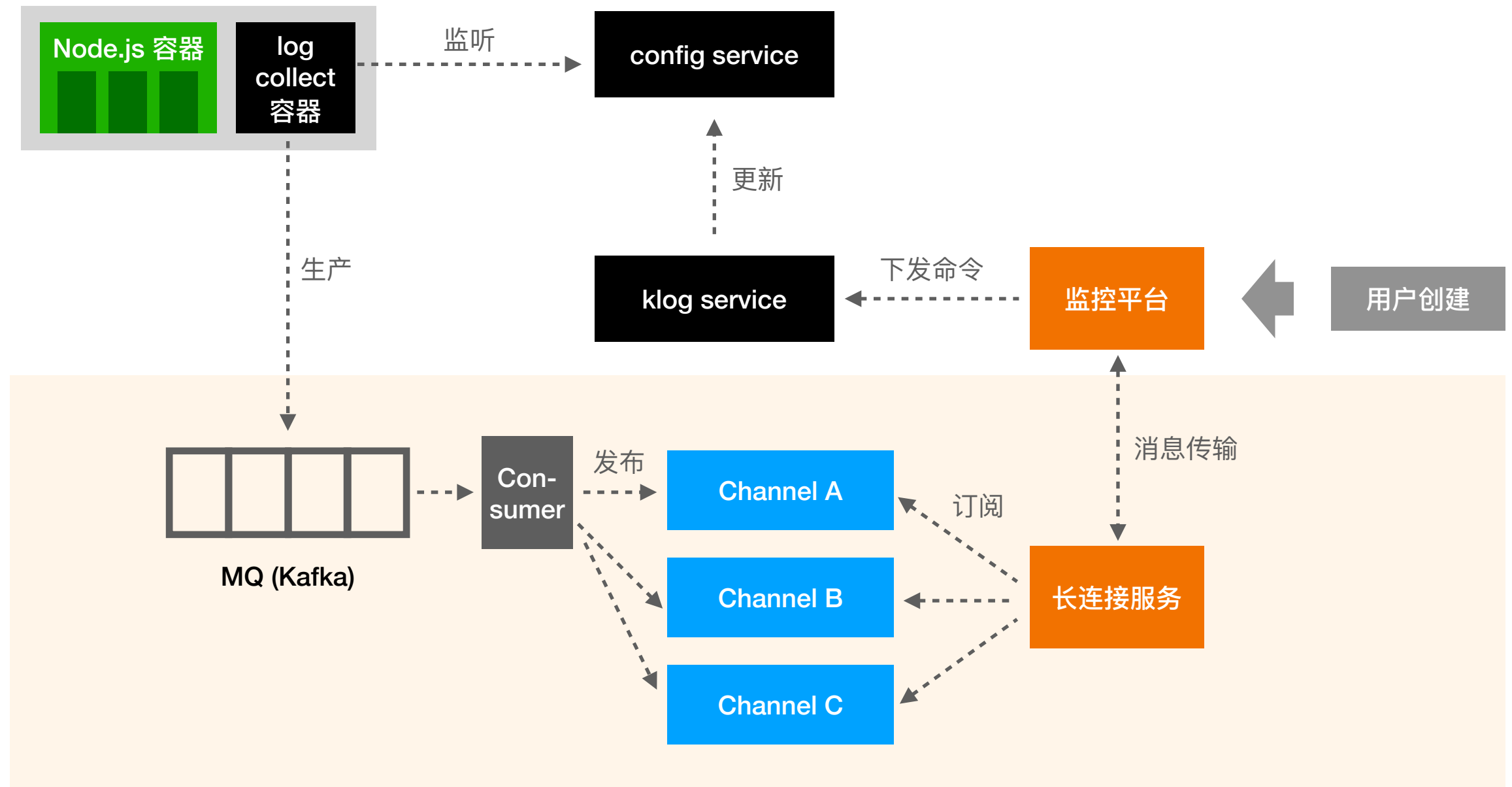


实时信息展示 —— 如何实现

实时日志的数据链路?

KNode 服务数据链路 with 整体架构

如何实现实时日志的数据链路?



服务的整体架构

应用层

监控平台、诊断报告

服务层

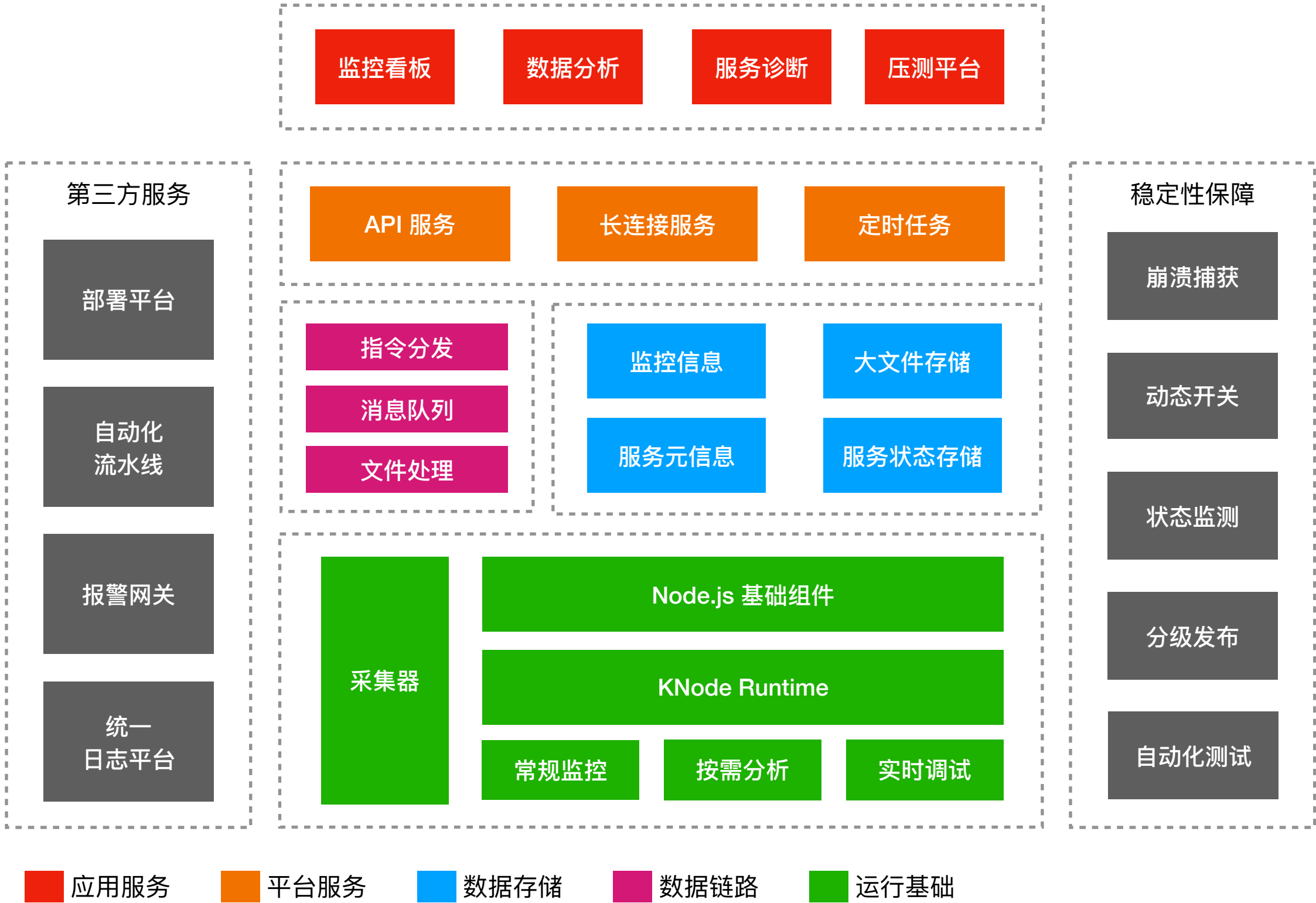
API 服务、定时任务等

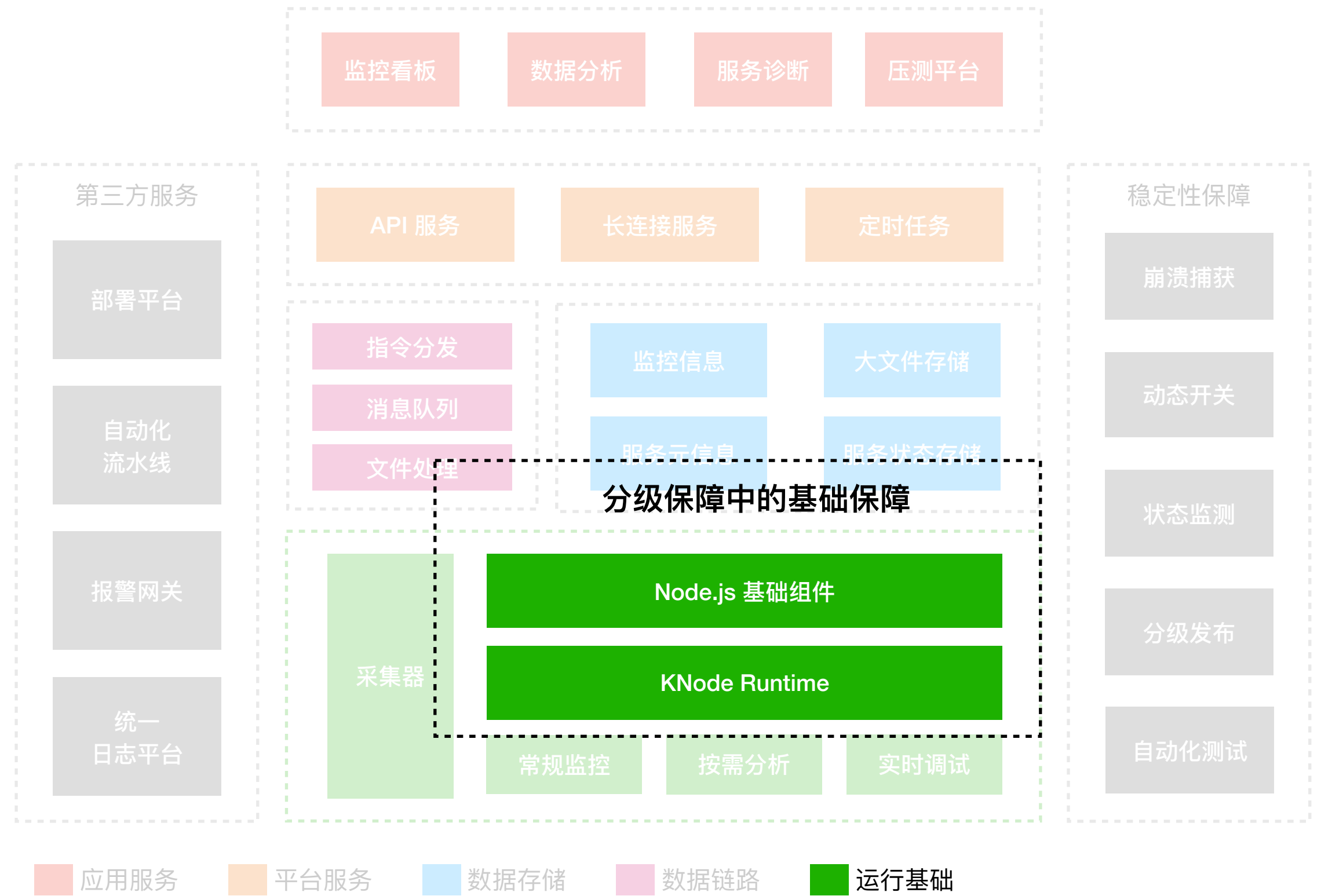
数据层

数据链路、数据存储

采集层

数据生产、数据采集





- ☑ 快手 KNode 的发展背景
- ☑ Node.js 运行时采集方案选型
- ☑ 运行时状态采集的技术实现
- ☑ KNode 整体服务架构
- ☑ 未来展望



总结

1. 介绍了在快手 Node.js 的发展背景下，Node.js **基础监控能力的缺失** 成为亟待解决的短板
2. 分析了常见的 Node.js 运行时采集方案的选型，KNode 使用了 **运行时源码定制** 方案
3. 介绍了 KNode 监控信息采集，一些典型监控排障能力的 **技术实现**
4. 介绍了 KNode 服务中各类 **数据链路的设计**，以及 **整体服务的架构**
5. 对 KNode 服务的未来展望

谢谢！



麦思博(msup)有限公司是一家面向技术型企业的培训咨询机构，携手2000余位中外客座导师，服务于技术团队的能力提升、软件工程效能和产品创新迭代，超过3000余家企业续约学习，是科技领域占有率第1的客座导师品牌，msup以整合全球领先经验实践为己任，为中国产业快速发展提供智库。



高可用架构主要关注互联网架构及高可用、可扩展及高性能领域的知识传播。订阅用户覆盖主流互联网及软件领域系统架构技术从业人员。高可用架构系列社群是一个社区组织，其精神是“分享+交流”，提倡社区的人人参与，同时从社区获得高质量的内容。