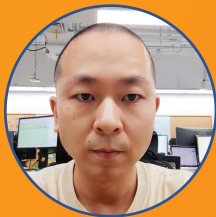




利用夜莺扩展能力打造全方位监控系统



喻波

滴滴
专家工程师



GopherChina 2021

目 录

运维监控需求来源

01

监控痛点：全面完备、跨云

02

夜莺介绍： 国产开源监控系统

03

夜莺设计实现：Agentd 数据采集

04

夜莺设计实现：Server 数据处理

05

夜莺设计实现：技术难点及细节

06

第一部分

运维监控需求来源

如果贵司的业务强依赖IT技术，IT故障会直接影响营业收入，稳定性体系一定要重视起来，而监控，就是稳定性体系中至关重要的一环



运维监控需求来源

谷歌今晨宕机5分钟：损失高达55万美元

腾讯科技  [微博] 悦潼 2013年08月17日15:18

我要分享 ▼

[导读]这点损失，对于庞大的谷歌而言，当然是无伤大雅了。

Google Error

Server Error

The service you requested is not available yet.

Please try again in 30 seconds.



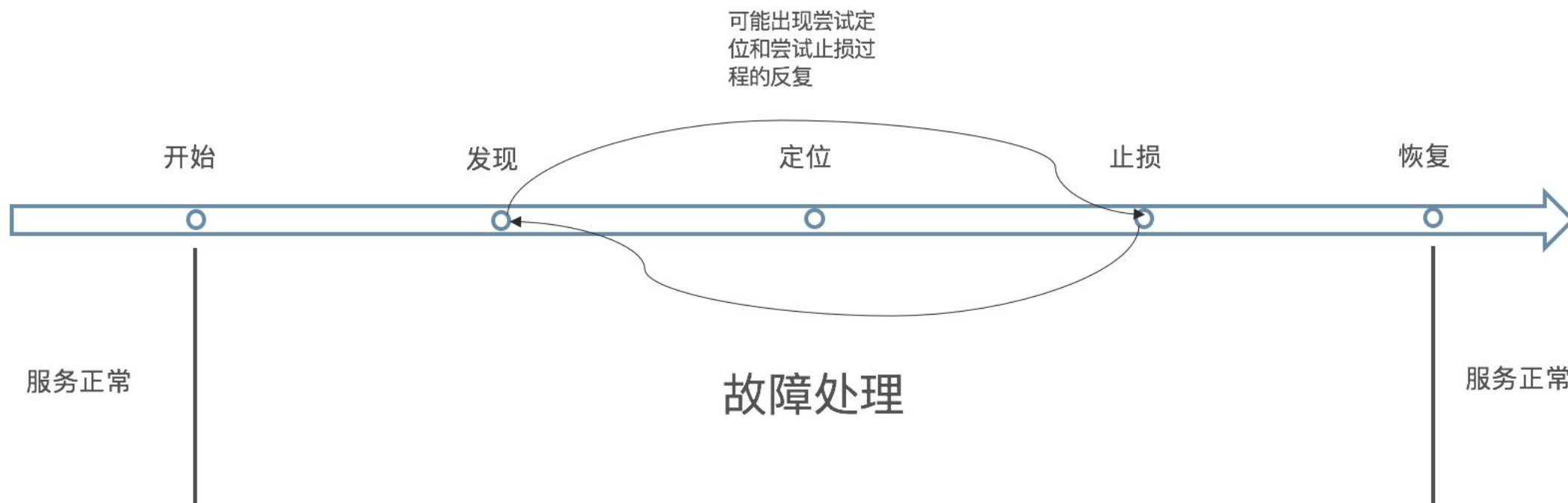
左图是2013年的一个新闻，讲Google宕机的影响。2020年也出现过aws大规模宕机的情况，影响不止是55万美元，直接影响大半个互联网！

2018年有美国调研机构指出，如果服务器宕机1分钟，银行会损失27万美元，制造业会损失42万美元

美团故障？滴滴故障？腾讯故障？

运维监控需求来源

如何减少服务停摆导致的经济损失？尽快发现故障并止损！故障处理过程中，监控是『发现』和『定位』两个环节的关键工具。故障处理过程的首要原则是『止损』，因此，过程中的『发现』和『定位』都是面向尽快『止损』来实现。



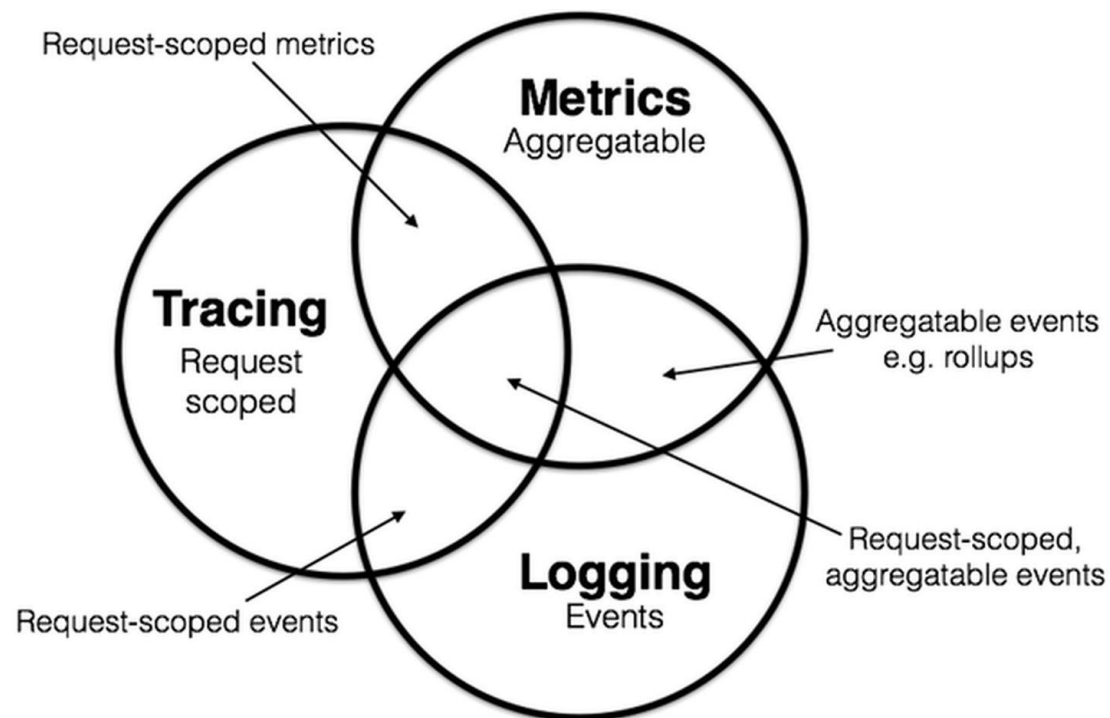
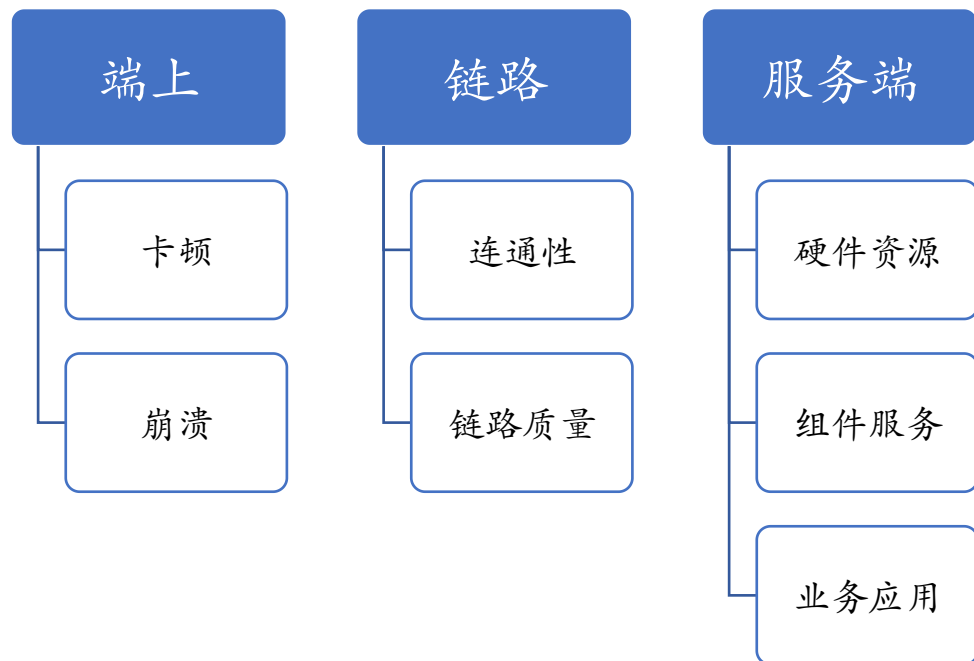
第二部分

监控痛点：全面完备、跨云

端上、链路、资源、组件、应用多维度跨云监控，不管哪个环节出问题都能及时感知



产品要求



第三部分

夜莺介绍：国产开源监控系统

国产开源监控产品相对比较匮乏，夜莺希望重新定义国产开源监控，支持云原生监控，经受了滴滴大规模生产检验



Nightingale

夜莺是**新一代国产智能监控平台**，既可以解决传统物理机虚拟机的场景，也可以解决容器的场景。衍生自Open-Falcon和滴滴Odin监控，经受了包括小米、美团、滴滴在内的数百家企业的生产环境验证，**简单可依赖，好用到爆！**

3500+

star

600+

issue

500+

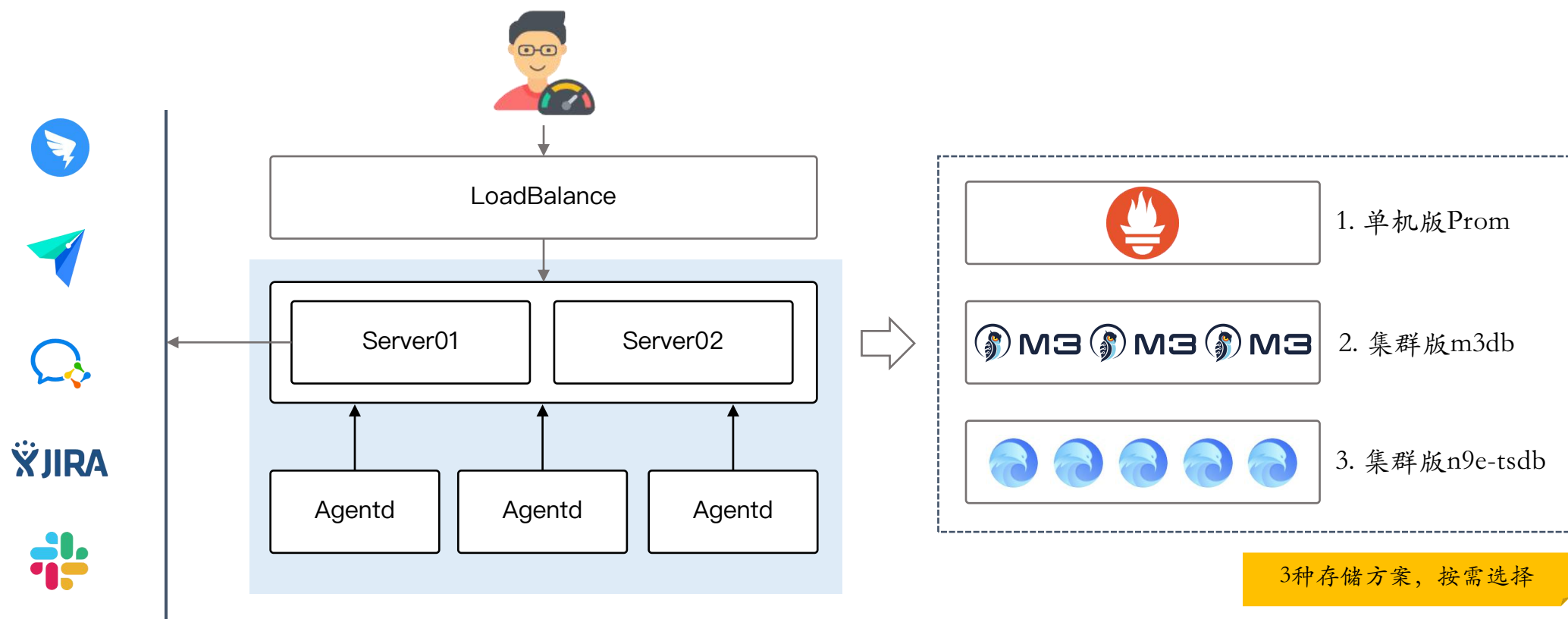
fork

项目：<https://github.com/didi/nightingale> 官网：<https://n9e.didiyun.com/>





上图展示部分社区用户，加入夜莺社群，请联系微信：UlricQin



第四部分

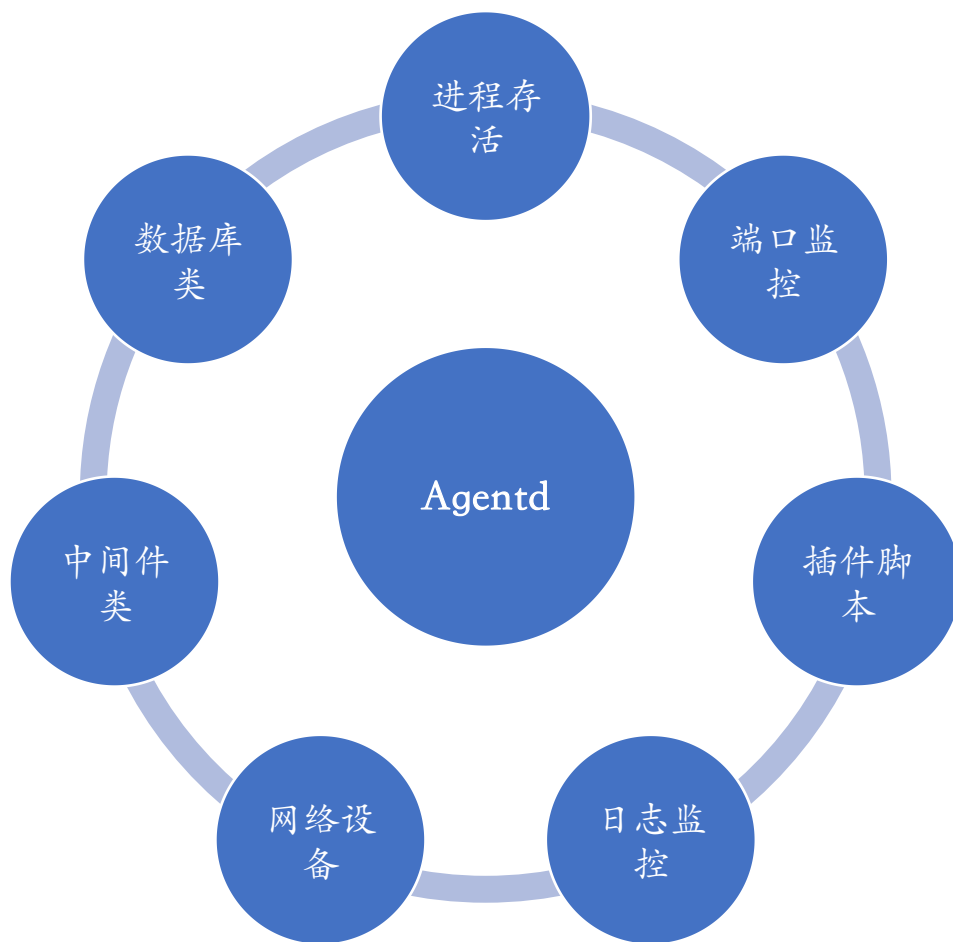
夜莺设计实现 Agentd 数据采集

监控系统的核心功能，是数据采集、存储、分析、展示，完备性看采集能力，是否能够兼容并包，纳入更多生态的能力，至关重要

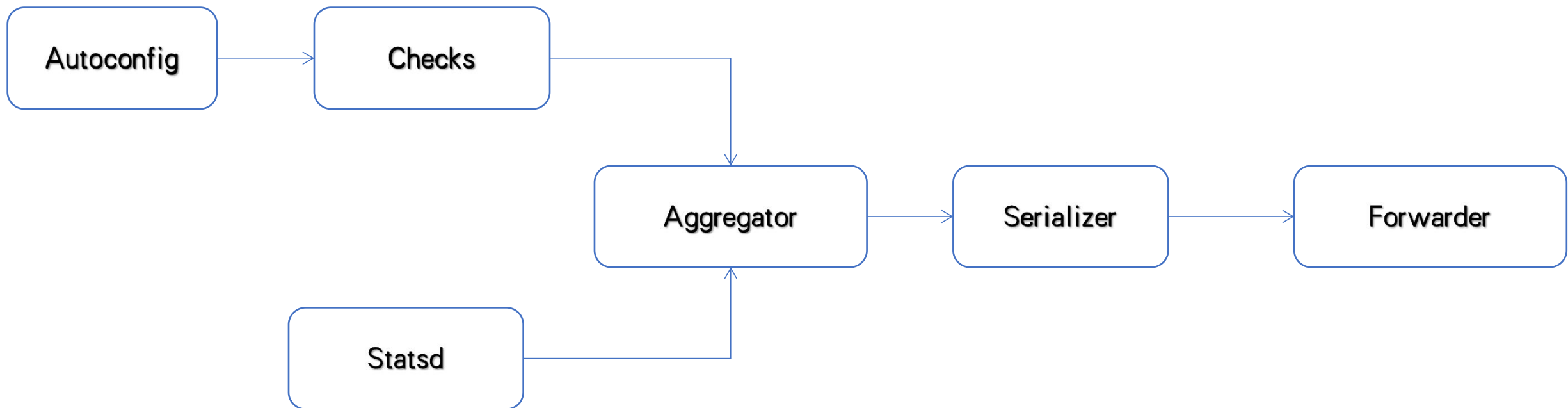


夜莺数据采集

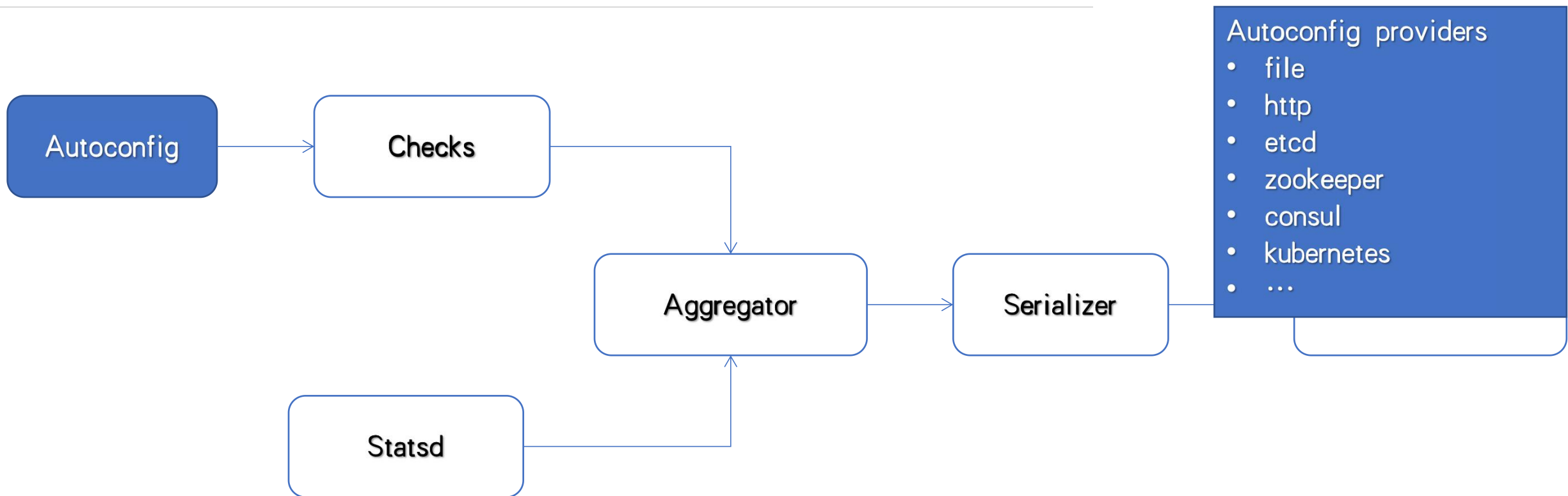
- 支持在web上配置采集策略，不同的采集可以指定不同的探针机器、目标机器，便于管理和知识传承
- 独创在端上流式读取日志，根据正则提取指标的机制，轻量易用，无业务侵入性
- 内置集成了多种数据库中间件的采集以及网络设备的采集，复用telegraf和datadog-agent的能力
- 支持statsd的udp协议，用于业务应用的apm监控分析



夜莺数据采集



夜莺数据采集



夜莺数据采集

Autoconfig

```
## n9e-agentd/etc/conf.d/log.d/conf.yaml
instances:
- name: log_count
  filePath: /var/log/nginx/access.log
  tagsPattern:
    code: HTTP\1.1"\s([0-9]{3})
    service: \s/api/(.*)\s/
  pattern: /api/
  func: count
```

Statsd

Autoconfig providers

- file
- http
- etcd
- zookeeper
- consul
- kubernetes
- ...

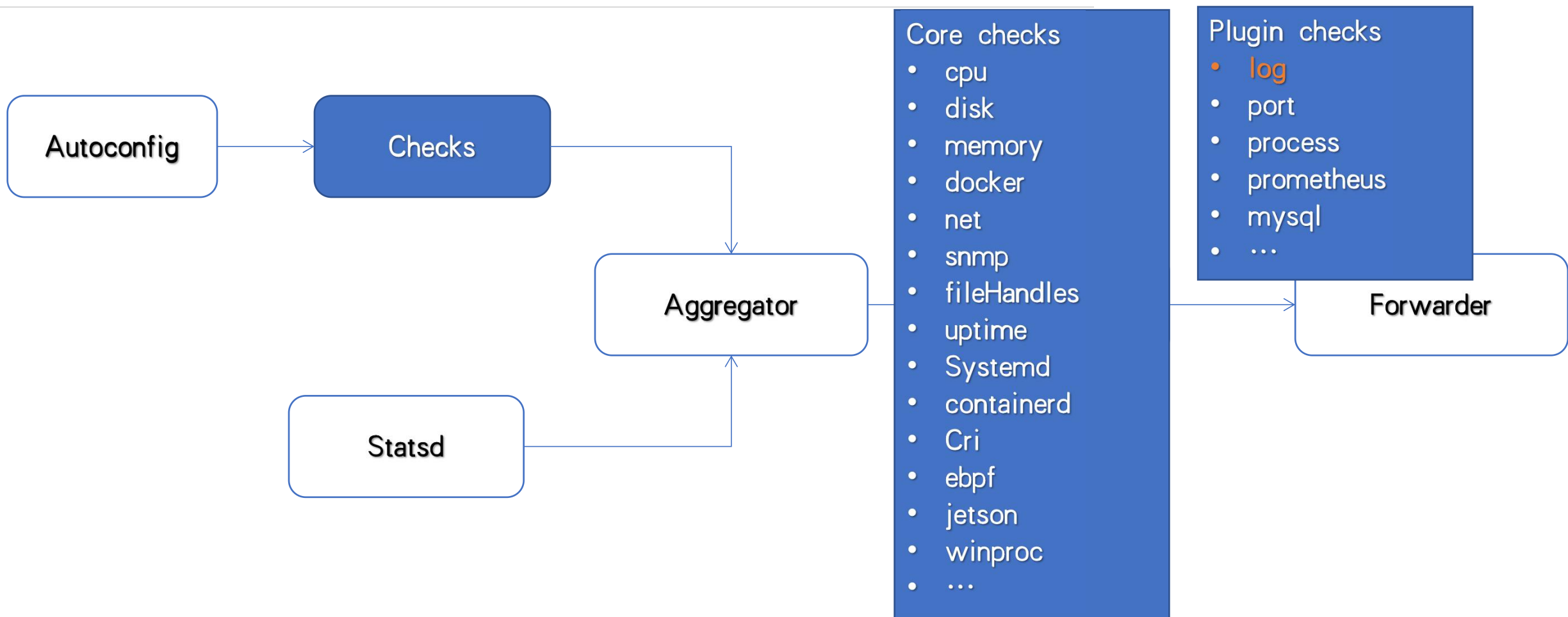
Autoconfig

```
$ curl http://172.20.70.14:8080/api/collect-rules
[
  {
    "ID": 1001,
    "name": "log-name",
    "type": "log",
    "data":
    "{\"adIdentifiers\":null,\"clusterCheck\":false,\"ignoreAutodiscoveryTags\":false,\"initC
onfig\":null,\"instances\": [{\"filePath\":\"/var/log/nginx/access.log\",\"func\":\"count
\",\"name\":\"log.count\",\"pattern\":\"/api/\",\"tagsPattern\":{\"code\":\"HTTP\\\\\\\\
/1.1\\\\\\\\\"\\\\\\\\s([0-
9]{3})\"},\"service\":\"\\\\\\\\s\\\\\\\\/api\\\\\\\\/(.*)\\\\\\\\/\"}],\"jmxMetrics\":null,\"logs\":n
ull}"
  }
]
```

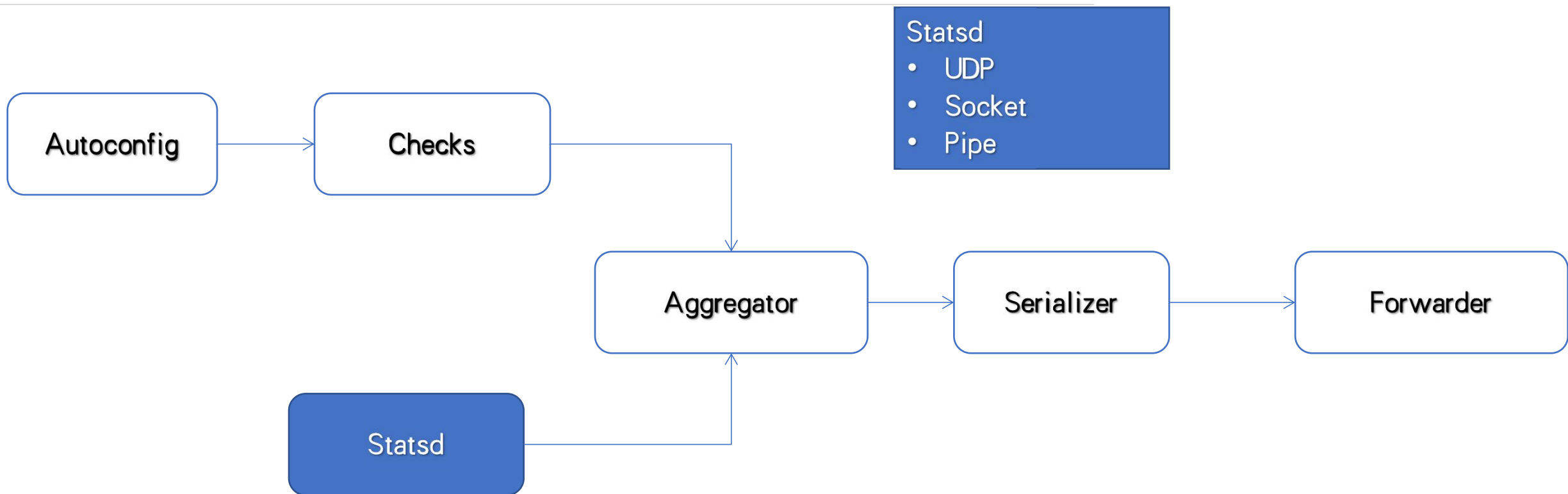
Autoconfig providers

- file
- http
- etcd
- zookeeper
- consul
- kubernetes
- ...

夜莺数据采集



夜莺数据采集



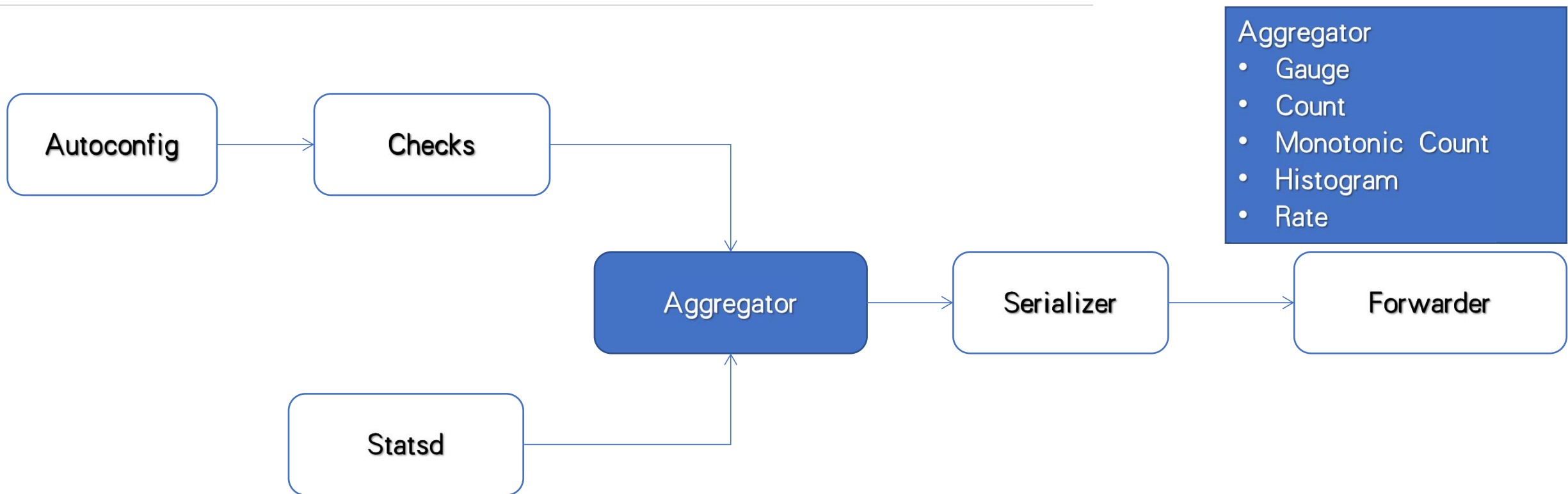
夜莺数据采集

这个报文里包含了一条 metricName: my_metric, value: 21, type=Gauge, tag: tag1=1, tag2=2 的记录
my_metric:21|g|tag1:1,tag2:2

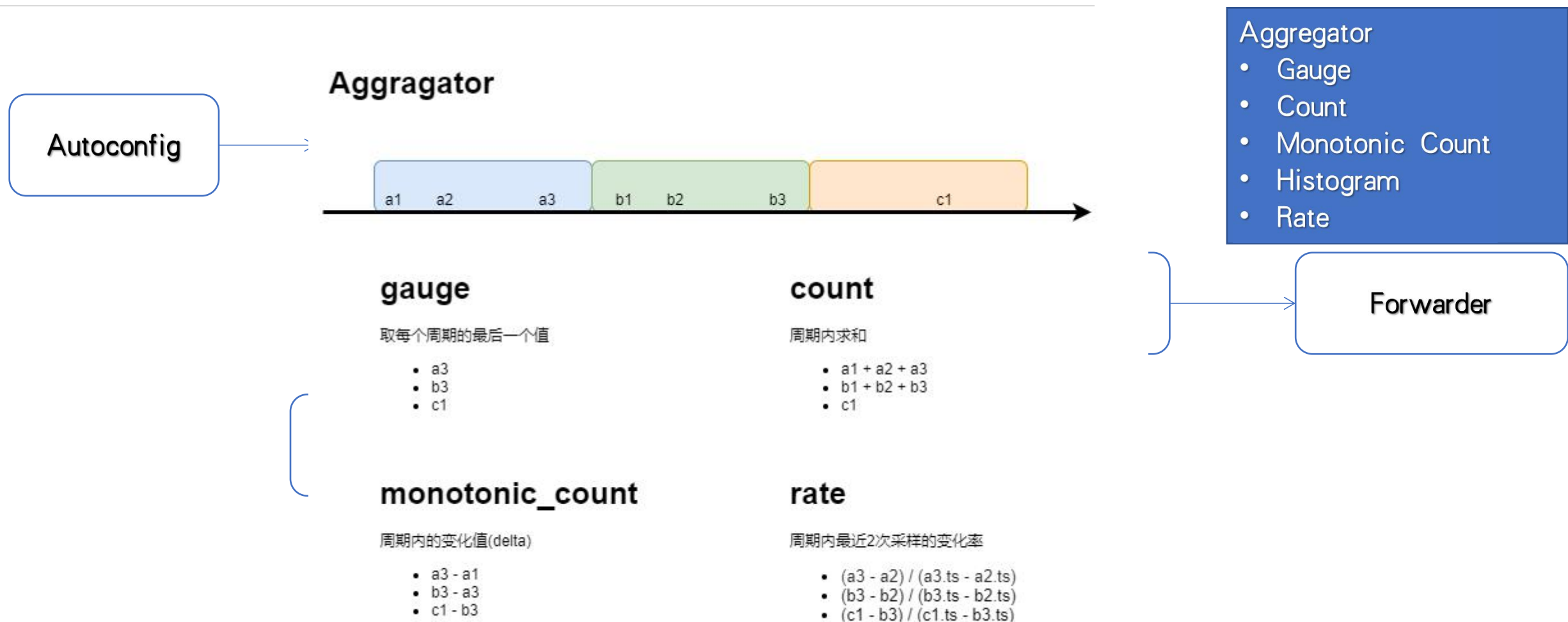
#发送到statsd的接口即可,使用sdk来实现也很容易

```
import (  
    "log "  
  
    "github.com/DataDog/datadog-go/statsd"  
)  
  
func main() {  
    client, err := statsd.New("127.0.0.1:8125",  
        statsd.WithTags([]string{"env:prod", "service:myservice"}),  
    )  
    if err != nil {  
        log.Fatal(err)  
    }  
  
    client.Gauge("my_metrics", 21, []string{"tag1:1", "tag2:2"}, 1)  
    client.Close()  
}
```

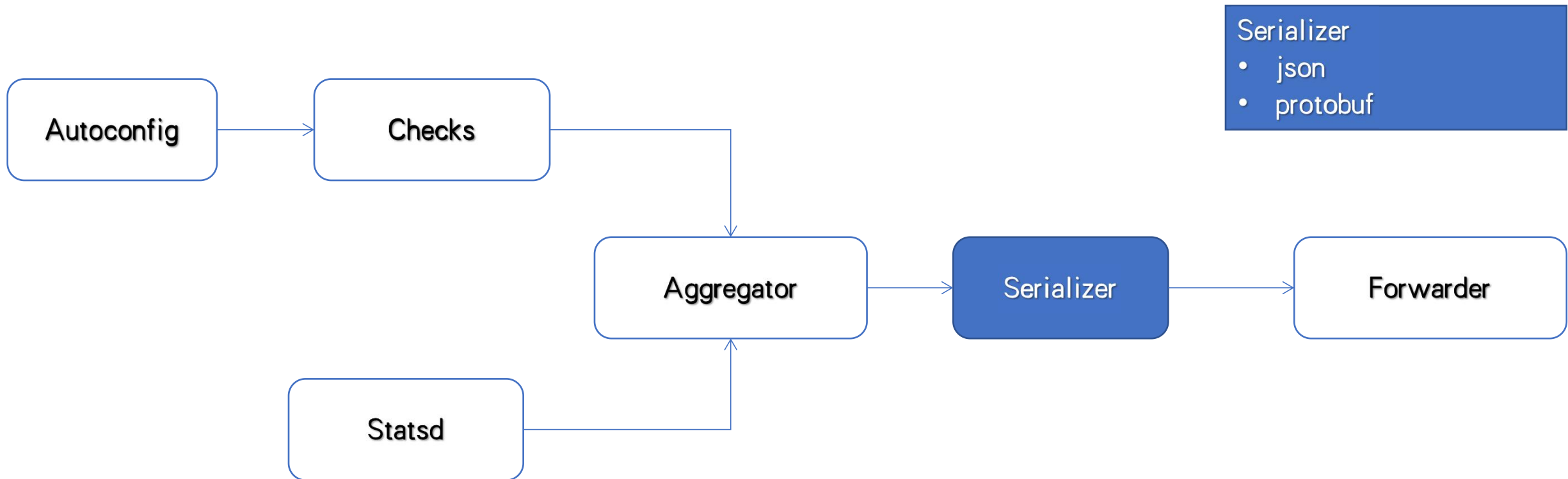
夜莺数据采集



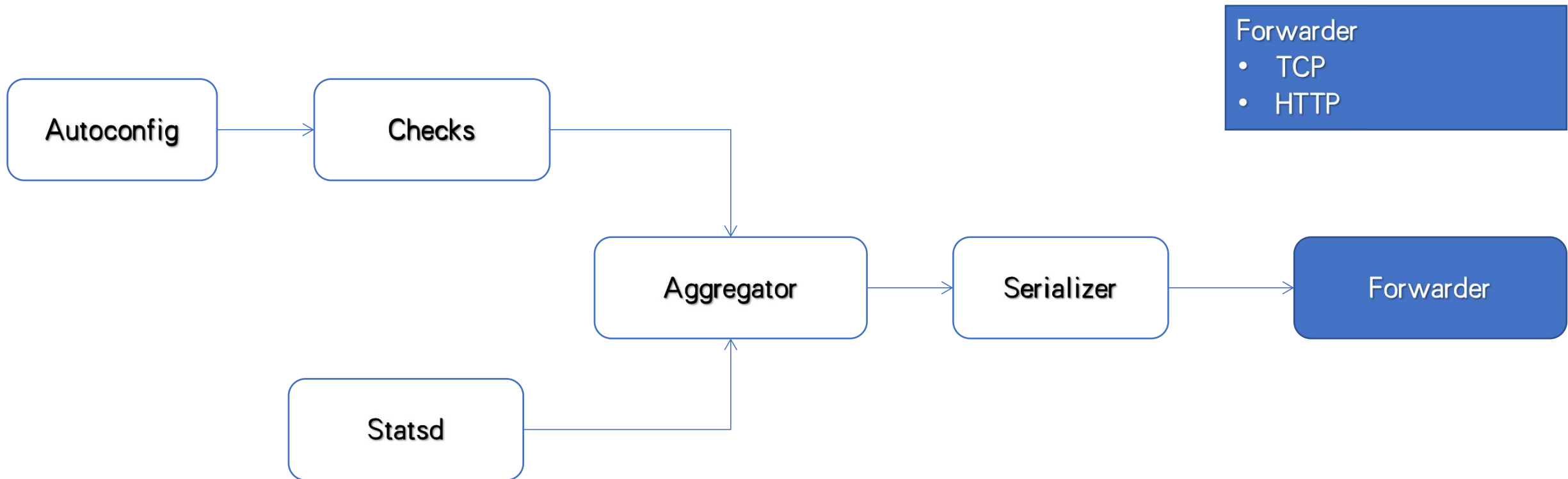
夜莺数据采集



夜莺数据采集



夜莺数据采集

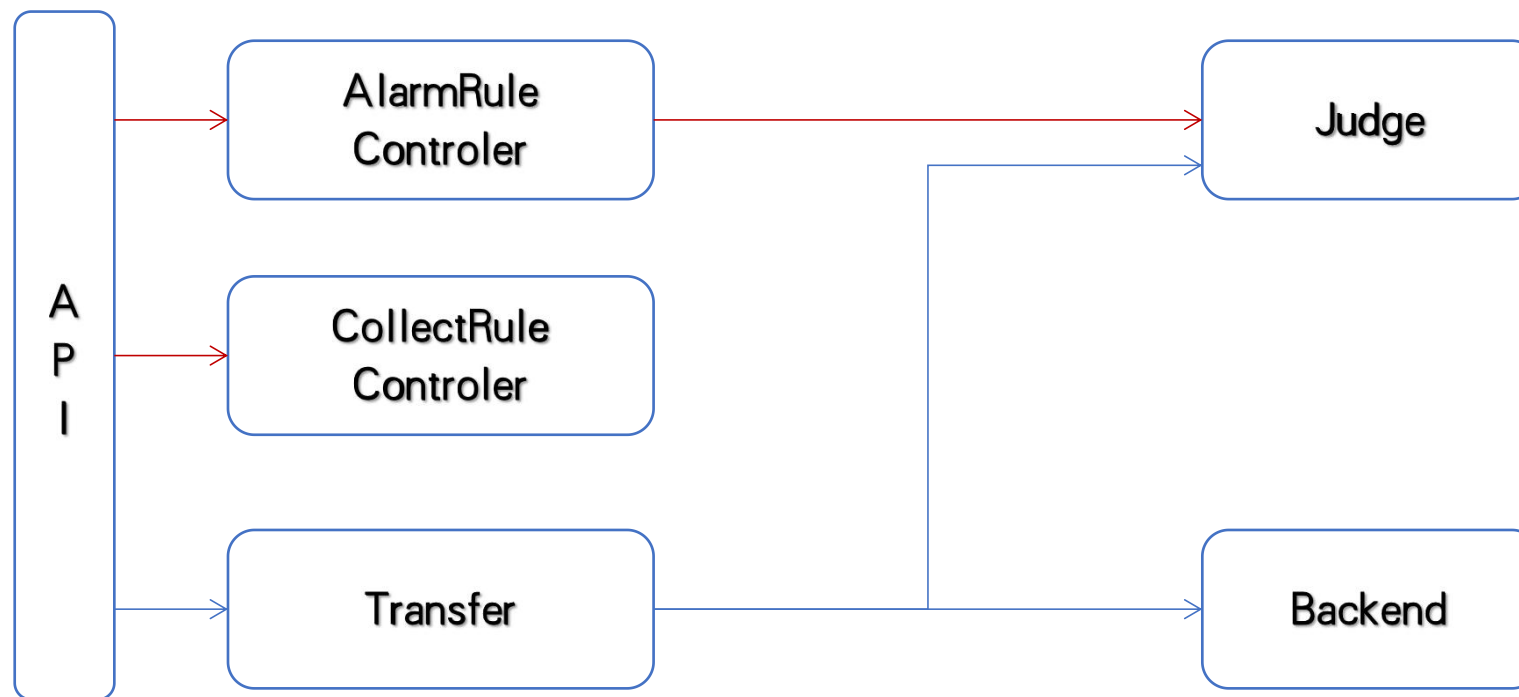


第五部分

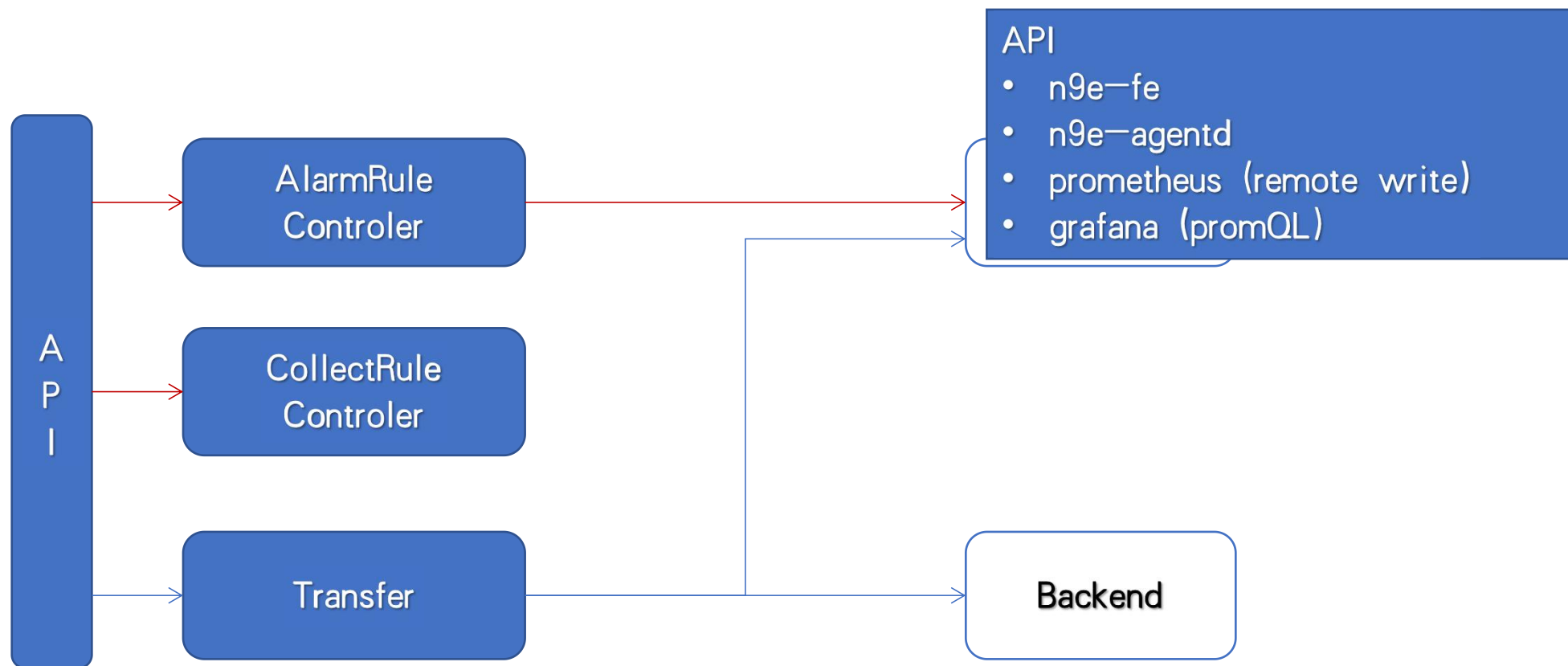
夜莺设计实现 Server 数据处理



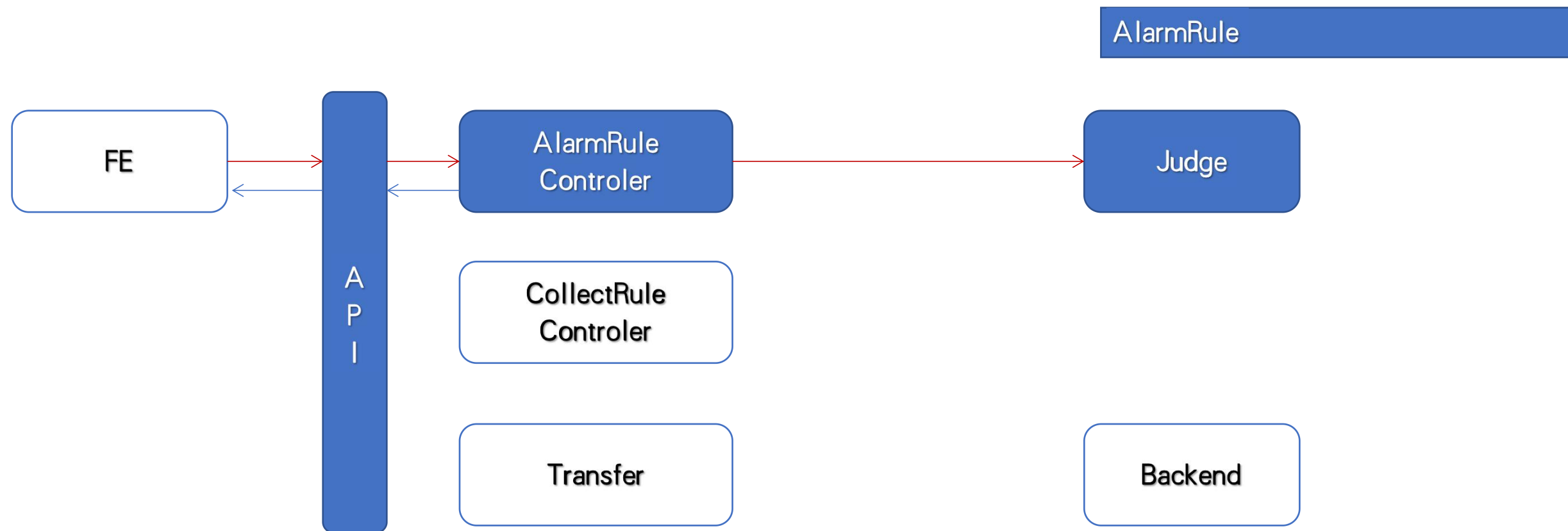
夜莺Server数据处理



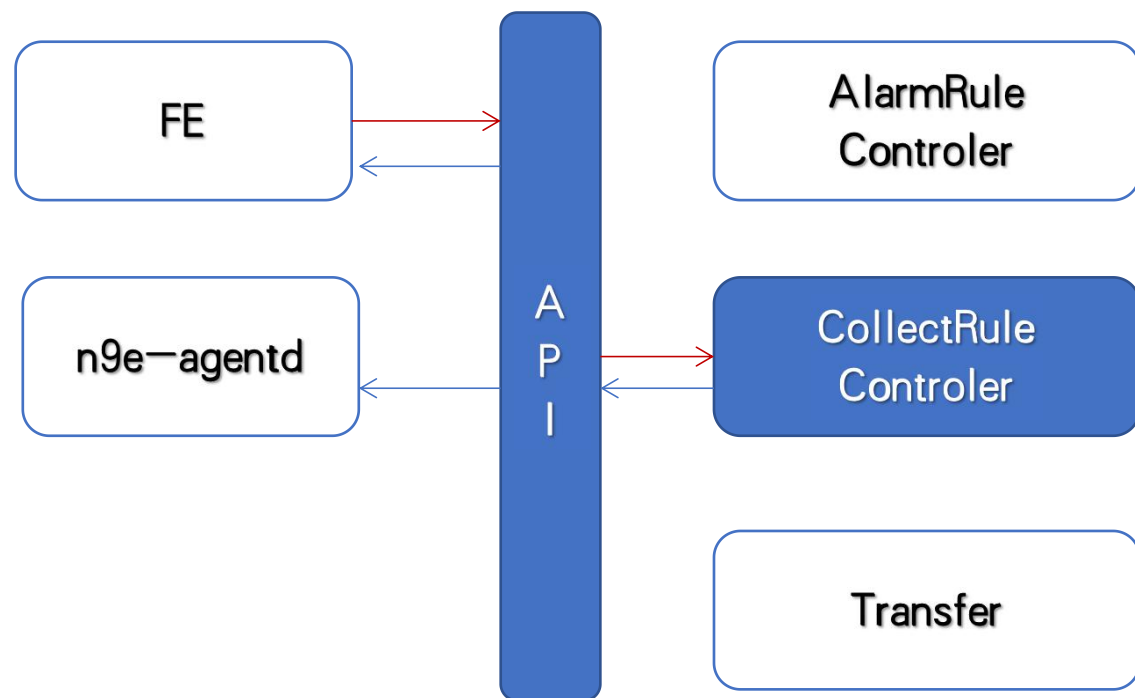
夜莺Server数据处理



夜莺Server数据处理



夜莺Server数据处理



CollectRule

- Log
- Process
- Port
- Script
- ...

Backend

夜莺Server数据处理

FE

n9e-agentd

日志采集详情

* 所属资源分组: n9e.judge

* 指标名称:

log.

count

* 计算方法:

计数: 对符合规则的日志进行计数

▼

* 日志路径:

/var/log/nginx/access.log

i

* 时间格式:

01/Jan/2006 15:04:05

▼

* 采集周期:

10

▼

* 匹配正则:

/api/

i

* 标签:

code

HTTPV1.1"\s{[0-9]

⊖

service

\s/api/(.*)/

⊕

i

验证:

请输入一行待监控的完整日志

验证

附加标签:

tagKey

=

tagValue

⊕

备注:

请填写备注信息

取消

确定

CollectRule

- Log
- Process
- Port
- Script
- ...

Backend



进程采集详情

×

进程监控指标: system_proc_num

* 所属资源分组: n9e.judge

* 采集名称: n9e_agentd

* 采集方式: 命令行 进程名

* 进程名: n9e-agentd

* 采集周期: 10 秒

附加标签: env = prd ⊖

service = n9e-agentd ⊕

备注: 请填写备注信息

CollectRule

- Log
- Process
- Port
- Script
- ...

Backend



FE

n9e-agent

端口采集详情

×

端口监控指标: system_proc_port_listen

* 所属资源分组: n9e.judge

* 采集名称:

system_proc_port

* 端口协议:

TCP

UDP

* 端口号:

8080

* 连接超时:

2

秒

* 采集周期:

10

秒

附加标签:

env

=

prd

⊖

service

=

judge

⊕

备注:

请填写备注信息

取消

确定

CollectRule

- Log
- Process
- **Port**
- Script
- ...

Backend



夜莺Server数据处理

创建插件采集

×

* 所属资源分组: n9e.judge

* 采集名称: nginx

* 文件路径: /opt/n9e-agent/script.d/10-nginx.sh

* 参数:

环境变量: 请输入变量名

请输入变量值



* 标准输入(Stdin):

* 采集周期: 10

备注: 请填写备注信息

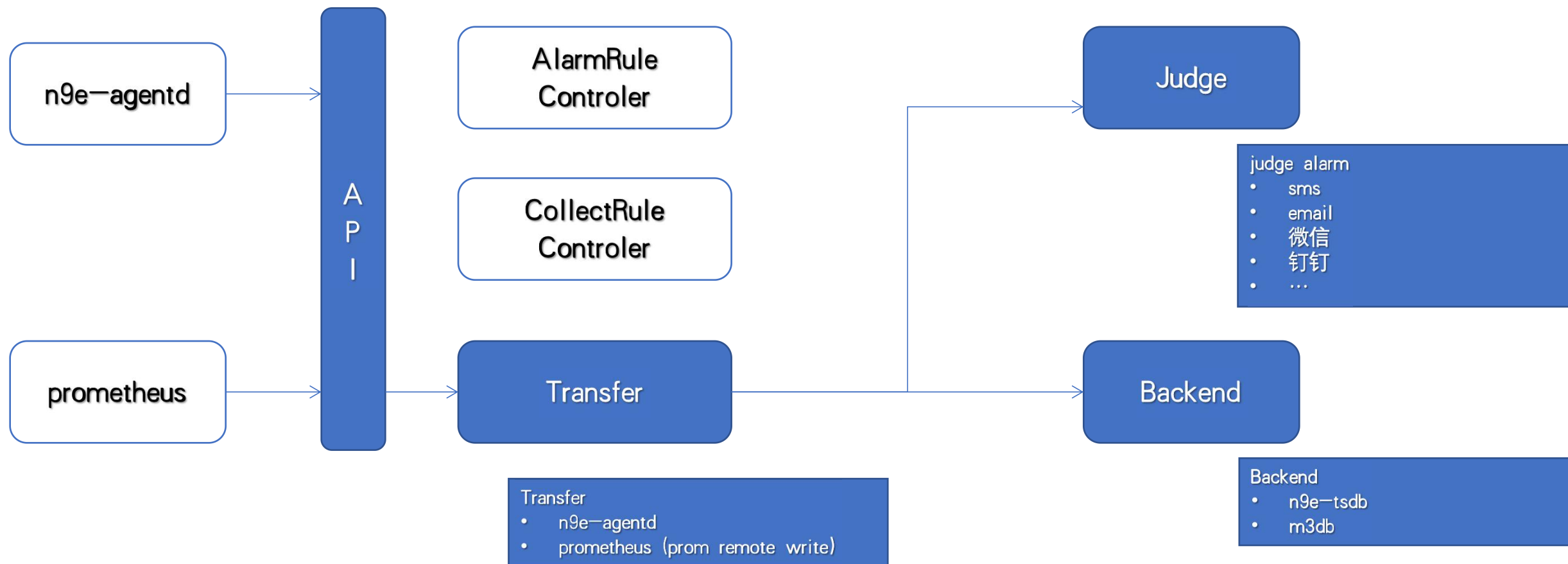
CollectRule

- Log
- Process
- Port
- Script
- ...

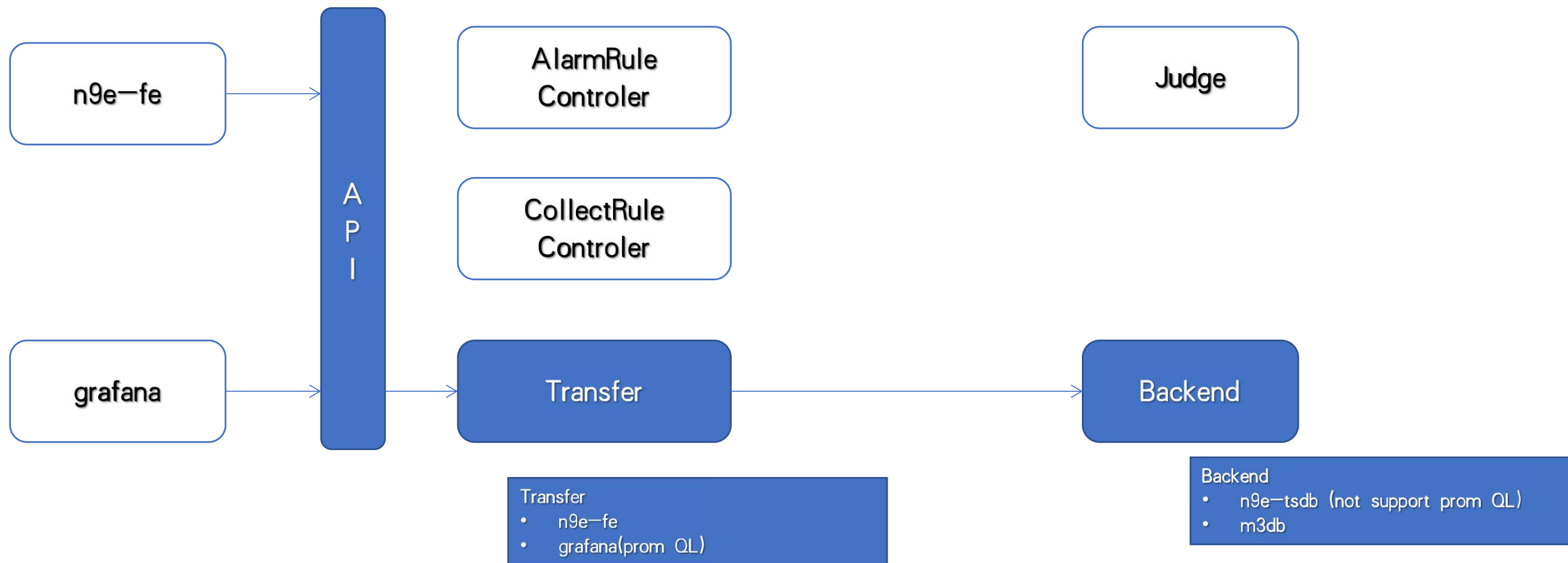
Backend



夜莺Server数据处理



夜莺Server数据处理

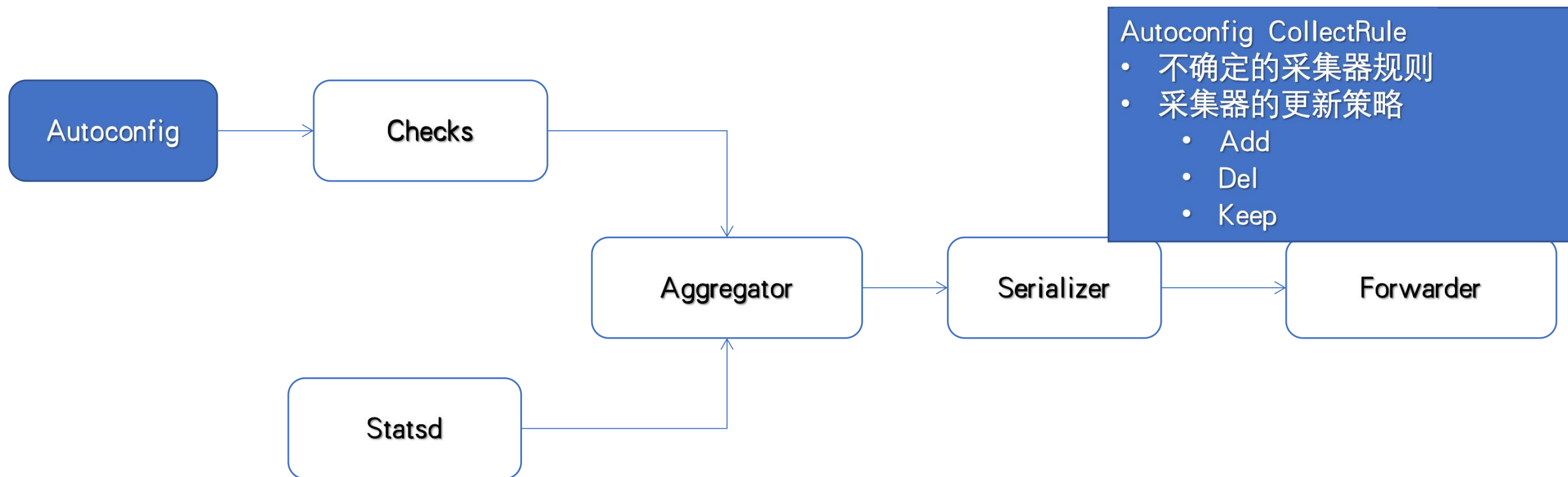


第六部分

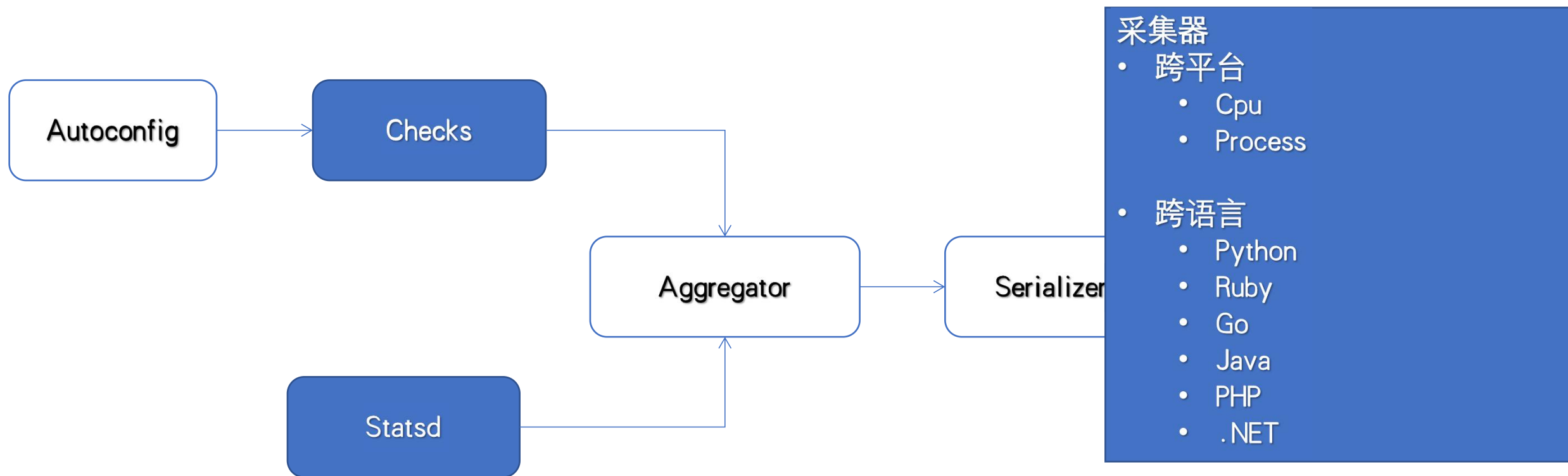
夜莺设计实现 技术难点及细节



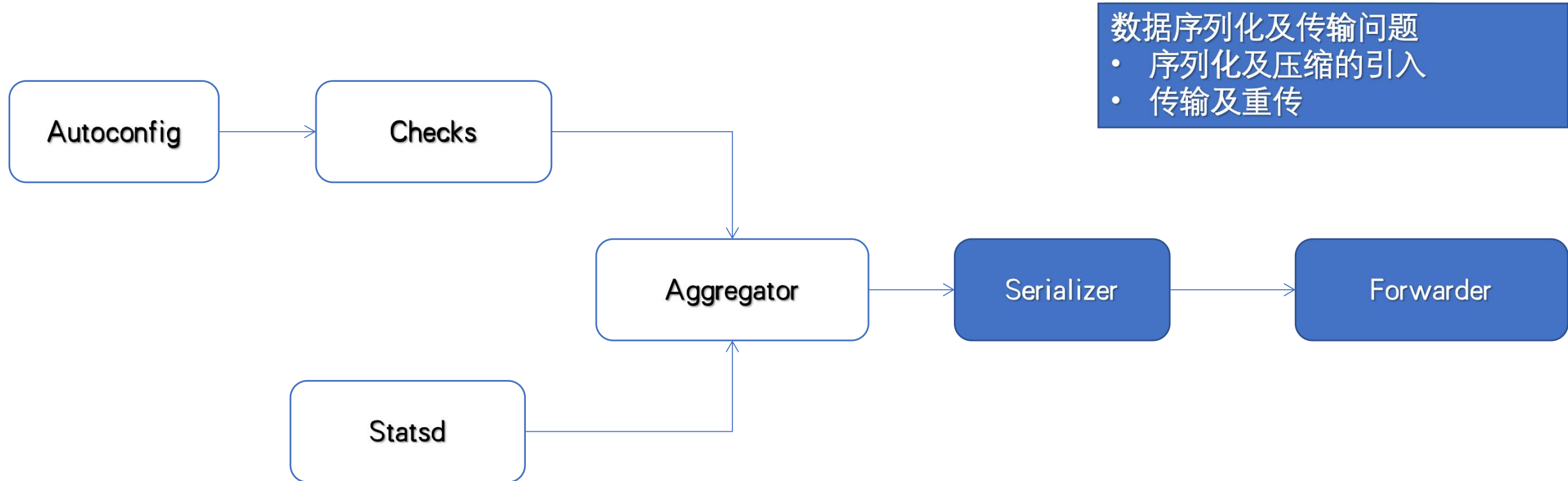
夜莺 技术难点及细节



夜莺 技术难点及细节



夜莺 技术难点及细节



Thank you