

# 第一届搜车 Node Party

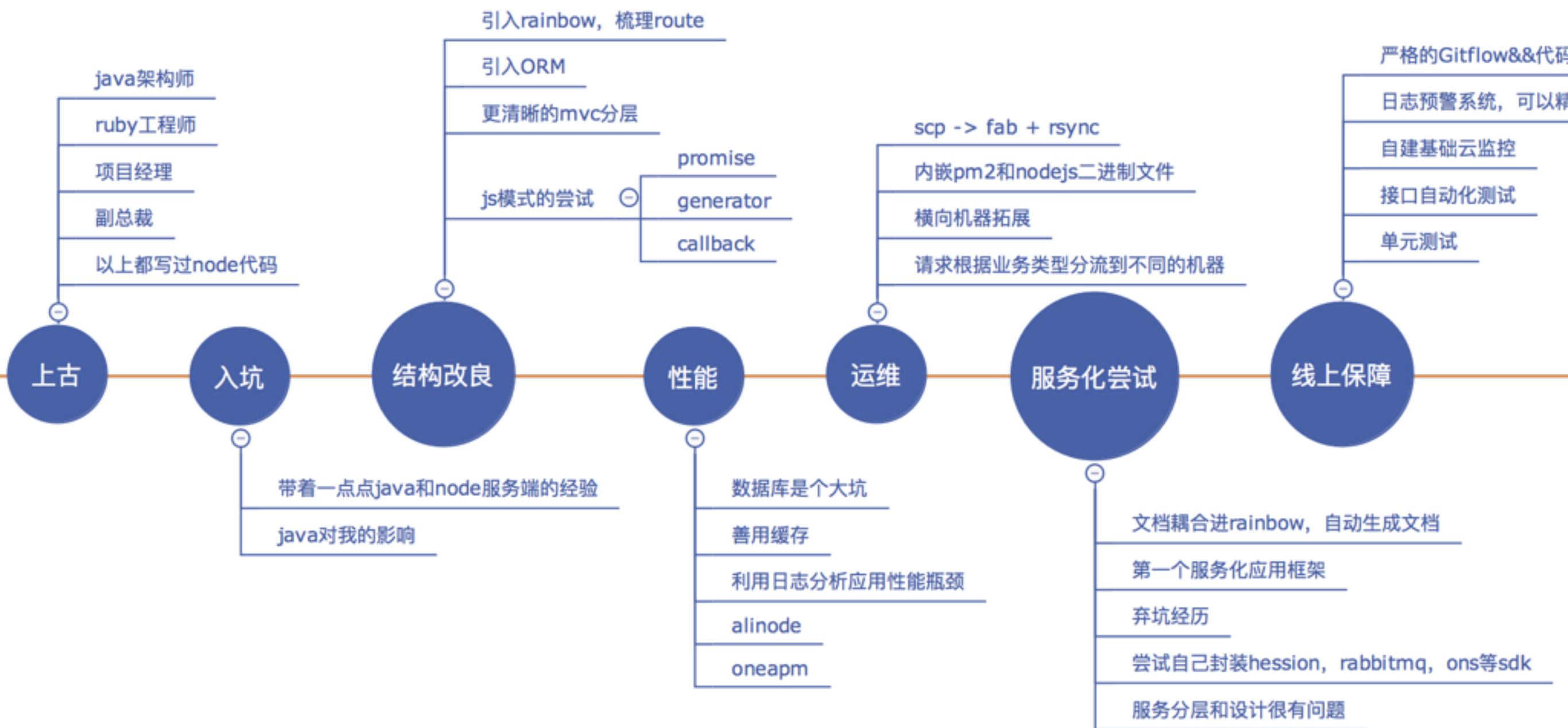
@搜车前端研发部

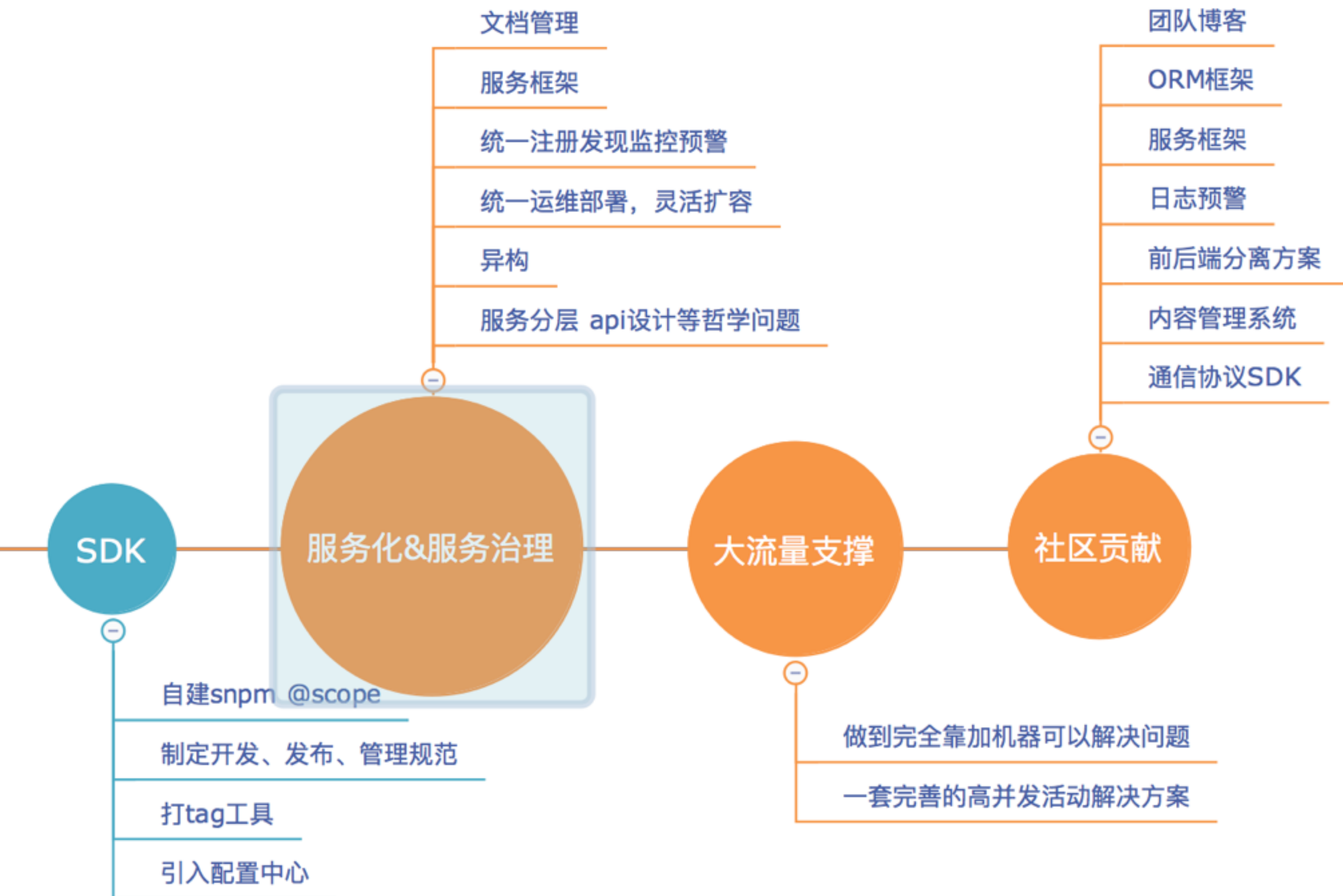
# 搜车Node进化史

@芋头

# 大事件

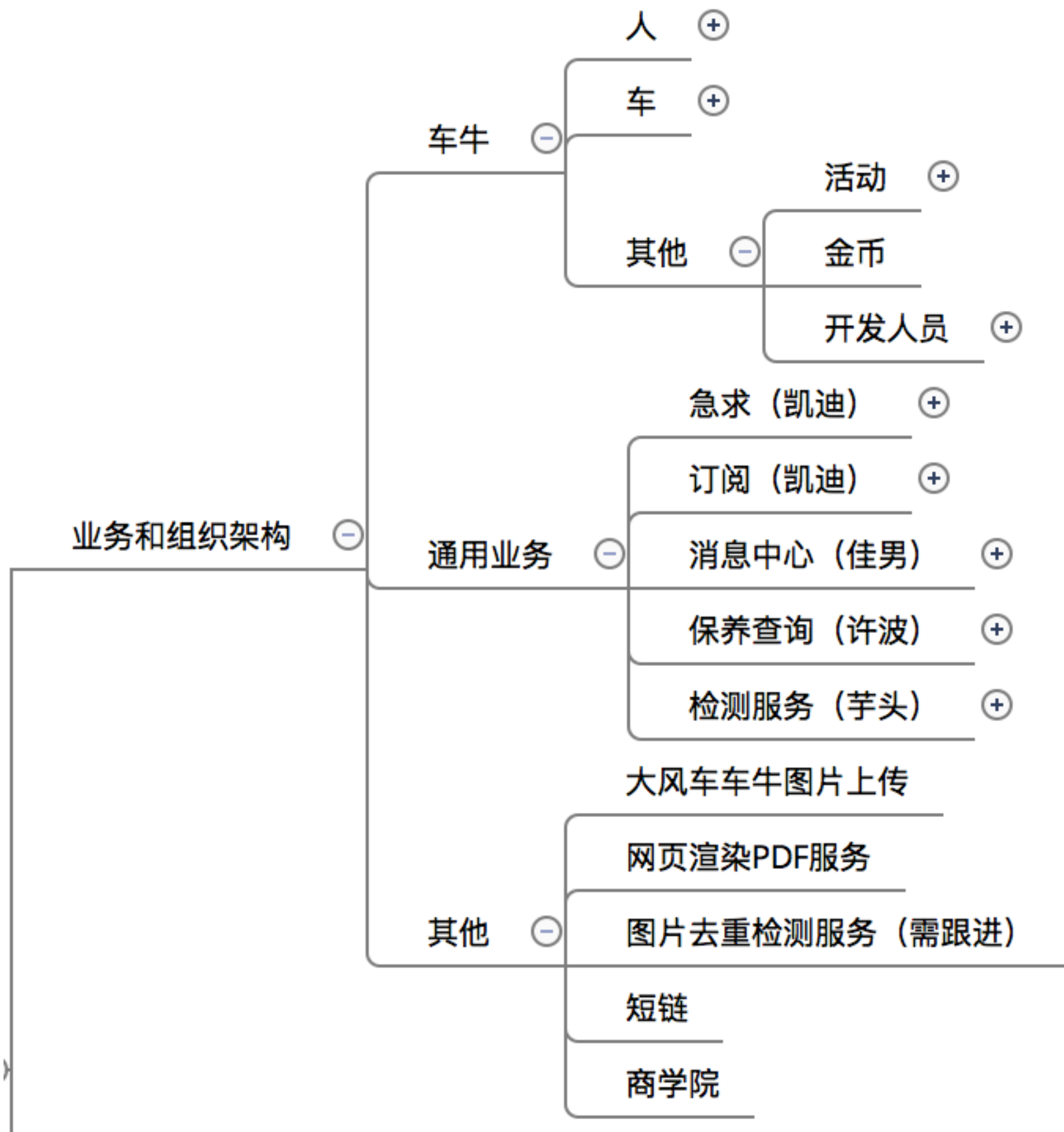
- 2年半前，选型。
- 1年半前，小白接坑。
- 半年多前，性能，运维。
- 几个月前，线上保障。
- 现在，SDK和服务。
- 未来，大流量，社区贡献。





# 支撑业务介绍

- 业务系统服务端。
- 底层通用业务
- 图片处理，爬虫，短链，PDF渲染等小型服务。



流水账开始





上古

java架构师

ruby工程师

项目经理

副总裁

以上都写过node代码



# 入坑

---

带着Java开发经验

引入Route管理

引入类似Spring分层

引入ORM

规范Callback的使用

Promise  
Generator

A red circle with the text '性能优化' (Performance Optimization) in white. A thin green horizontal line passes through the center of the circle.

性能优化

# 数据库

- 最低级的：索引。
- Mongodb的乱用。（跨表，读性能，写入性能）
- 缓存的善用。

# 其他性能点

- 长逻辑，频繁读写没有控制。
- 重逻辑，直接占用整个进程。
- 内存泄露？死循环逻辑？

# 手段

- OneAPM （模糊定位）
- AliNode （内存级别定位，贵）
- 日志发现和分析 （精确）

应用统一日志记录

ELK整体方案

分析时间和瓶颈

# 运维

feature -> develop -> master

git add tag

2.1.4+16w31b

发布机

git pull

fabirc + async

reload

内置pm2和node二进制文件，机器无需配置直接部署运行  
业务分流到机器，不能挂+可以挂+普通接口+重操作

A large red circle is centered on the page. A thin green horizontal line passes through the middle of the circle. The text '服务化尝试' is written in white inside the circle.

服务化尝试



# 服务化尝试

- 接口参数文档定义，耦合接口参数验证。
- 接口返回数据定义，耦合实体scheme。
- 服务化框架，多网络协议适配。
- 自己封装hession，mq，ons等协议适配。

# 弃坑

- 车和人被java接管，做公司级别的统一。
- 尝试在Node端再做一层服务，but意义不大。
- SDK已经能满足大部分封装了。
- 服务的分层设计不够合理（业务逻辑耦合到底层服务）

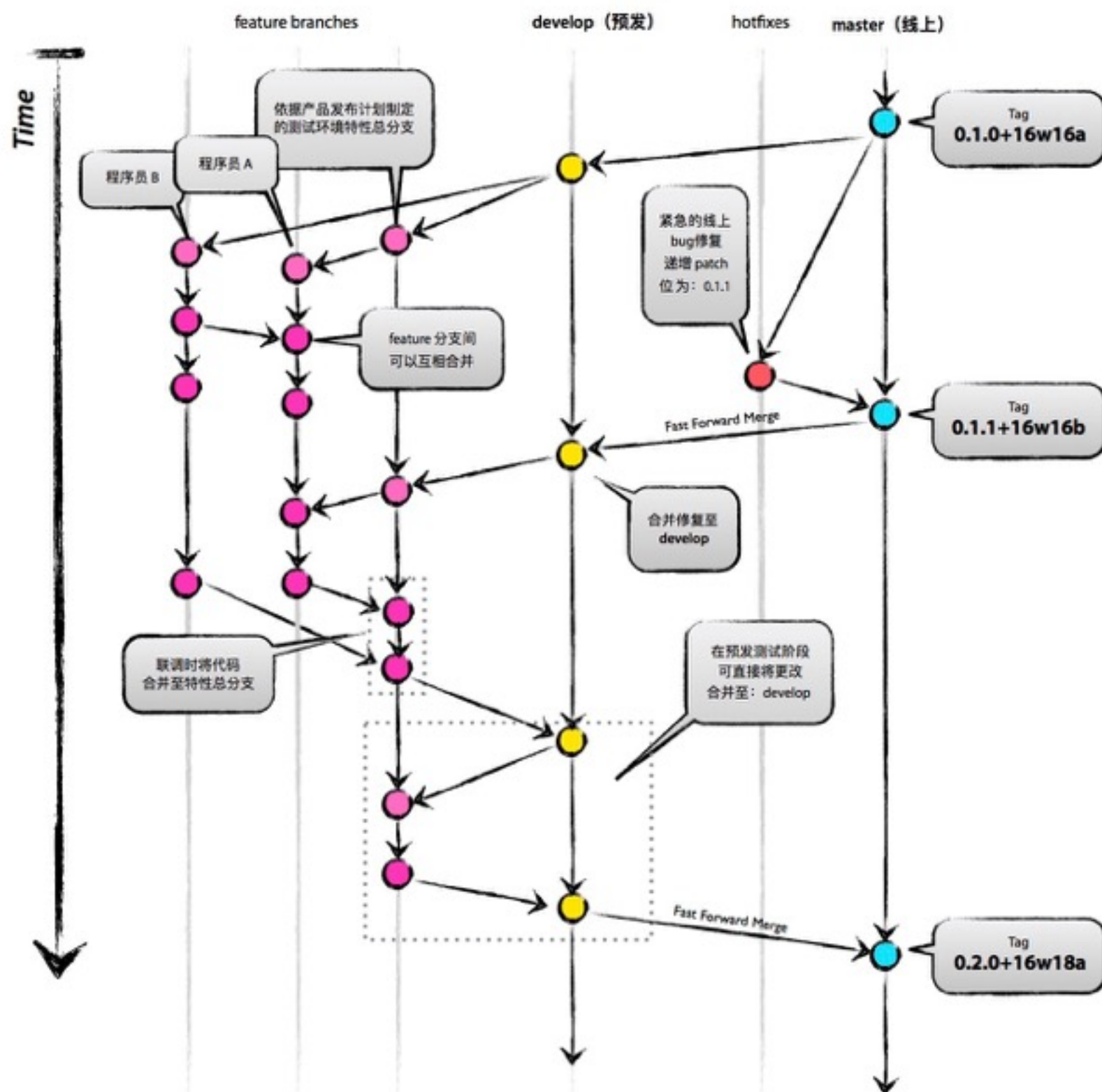
A red circle with the text '线上保障' (Online Guarantee) in white. A thin green horizontal line passes through the center of the circle.

线上保障

# 问题

- 如何及时（比老大更早）发现问题？
- 如何发掘项目中的隐患？
- 每次发布如何保障？
- 老大比你早发现问题后，如何最短时间内定位？

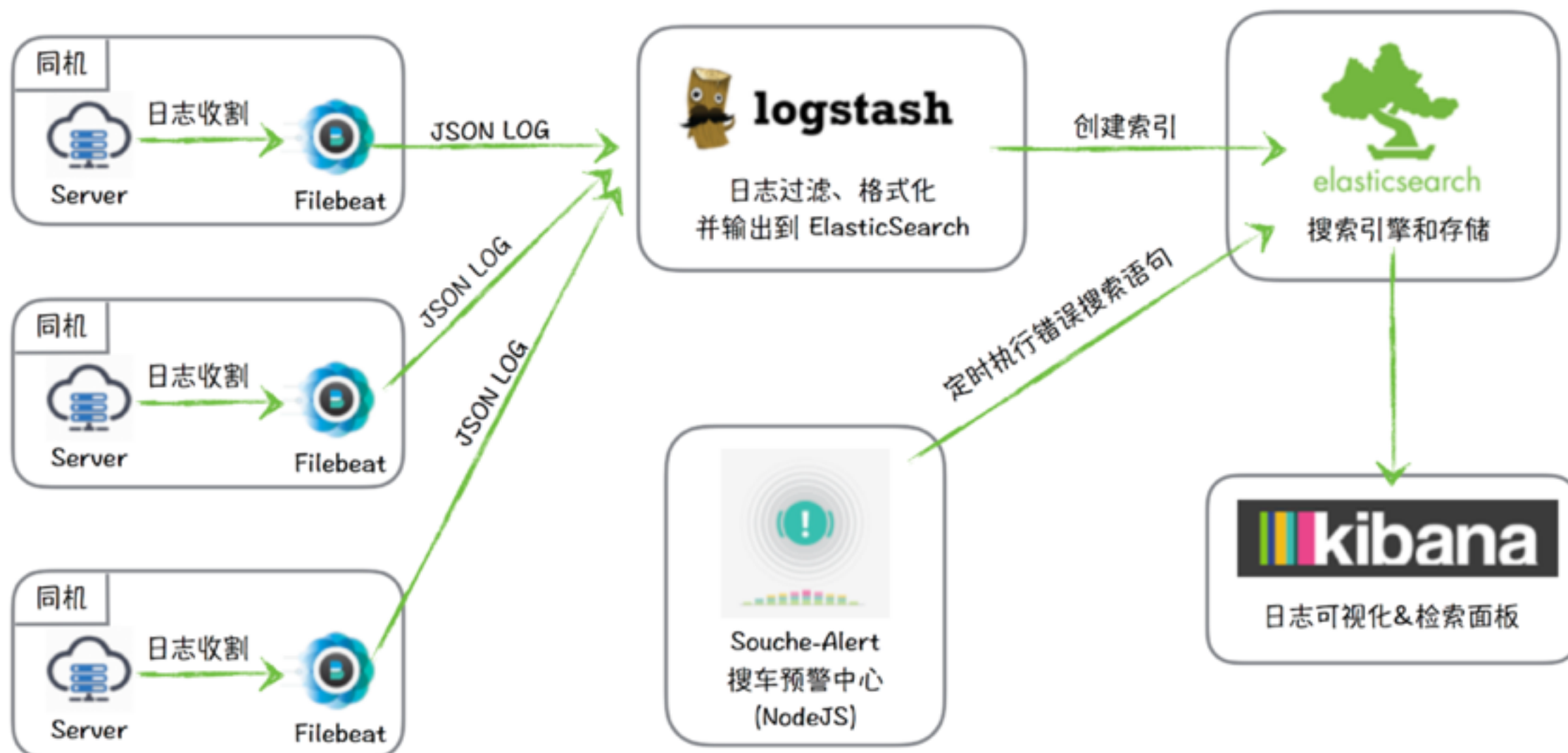
# GitFlow of souche.com II



## Important Tips:

1. Tag 命名规范为 {major}.{feature}.{patch}+{year}w{weeks}{a-z} 本周第几次发布
2. 除 develop 和 master 之间的互合外, merge 时, 请务必加上 `--no-ff` 参数
3. 尽量以 PR 形式进行代码合并, 对于重大改动, 务必进行 CodeReview。
4. 如果提交了一堆杂乱无章的记录, 需要使用 `git rebase` 重新整理这段提交。
5. Commit Message 格式  
`<type>(<scope>): <subject>`  
type 必须为以下几种之一:
  - feat: 特性提交
  - fix: Bug 修复提交
  - docs: 文档修改提交
  - style: 代码格式修改提交 (空格、格式、缺失符号等)
  - refactor: 一个既不是特性也不是修复的其它代码提交
  - perf: 性能优化
  - test: 测试用例
  - chore: 构建工具、类库等修改的提交scope 为影响范围, 可选  
subject 为针对此次提交的文字描述

# Elk + Souche-Alert 日志预警运维系统



# 日志里有什么？

- 性能数据
- 系统报错，主动报错，以及报错栈和变量。精确到行数。
- 跟踪请求路径和用户。

# 其他

- 接口自动化测试
- 接口断言报警系统 (Ruby)
- 系统监控，异常曲线预警。
- 减少依赖（请求超时，一键预案切换等）





SDK

# 目的

- 将底层service标准化
- 可服务于多个独立服务，为真正服务化做准备。
- 标准库的产生过程对开发同学自身成长的帮助。

## 业务

用户

车辆

店铺

搜索

检测

消息中心

## 中间件

用户验证

标准输出

防爬虫、黑名单等

## 底层

缓存

配置中心

日志

ORM

Mongodb

## 其他

车型库

地区管理

权限管理



未来

# 服务化&&服务治理

- 新的轻量服务框架
- 基于SDK的服务封装
- 统一运维管理，统一监控预警
- 与公司级服务治理结合
- 分层设计

# 大流量支撑

- 爆点活动，秒杀活动，抽奖活动，做到可以任意扩容
- 服务的重新设计。
- 针对性的缓存优化。

# 社区贡献

团队博客

ORM框架

服务框架

日志预警方案

通信协议SDK

内容管理系统

前后端分离方案

# The Last One



callback





Thanks  
Everyone