

当前端在讨论技术方案的时候，我们在讨论什么？

当前端在改进工作流程的时候，我们在改进什么？

当前端在考虑提升效率的时候，我们在考虑什么？

当前端和后端在扯皮的时候，我们在扯什么？

浅谈前端工程化

什么是工程化

工程一词最早是指人类有计划、有步骤地改造自然的活动

工程化即系统化、模块化、规范化的一个过程

前端工程化

工程学在前端领域的应用

将前端开发的流程、工具、技术、经验等规范化、标准化 ...

前端是一种特殊的GUI软件，它有两个特殊性：

一是前端由**三种编程语言**组成

二是前端代码在用户端运行时**增量安装**

前端开发的问题

- 如何进行高效的多人协作?
- 如何保证项目的可维护性?
- 如何提高项目的开发质量?
- 如何提升开发效率?
- 如何进行BUG追踪?
- 如何进行资源管理?
- 如何进行性能优化?
- ...



前端工程化主要内容

技术选型

搭建构建

开发连调

部署测试

性能优化

发布

技术选型

开发语言?

库?

框架?

工具?



技术选型



场景

热闹驱动开发

(Hype Driven Development, [HDD](#))

开发者爱好

团队

老板驱动开发

搭建构建



- 目录结构
- 开发环境
- 版本控制
- 编译SASS
- 检查JS
- 压缩JS
- 图片合并
- 压缩CSS
- ...

开发连调 - 开发规范

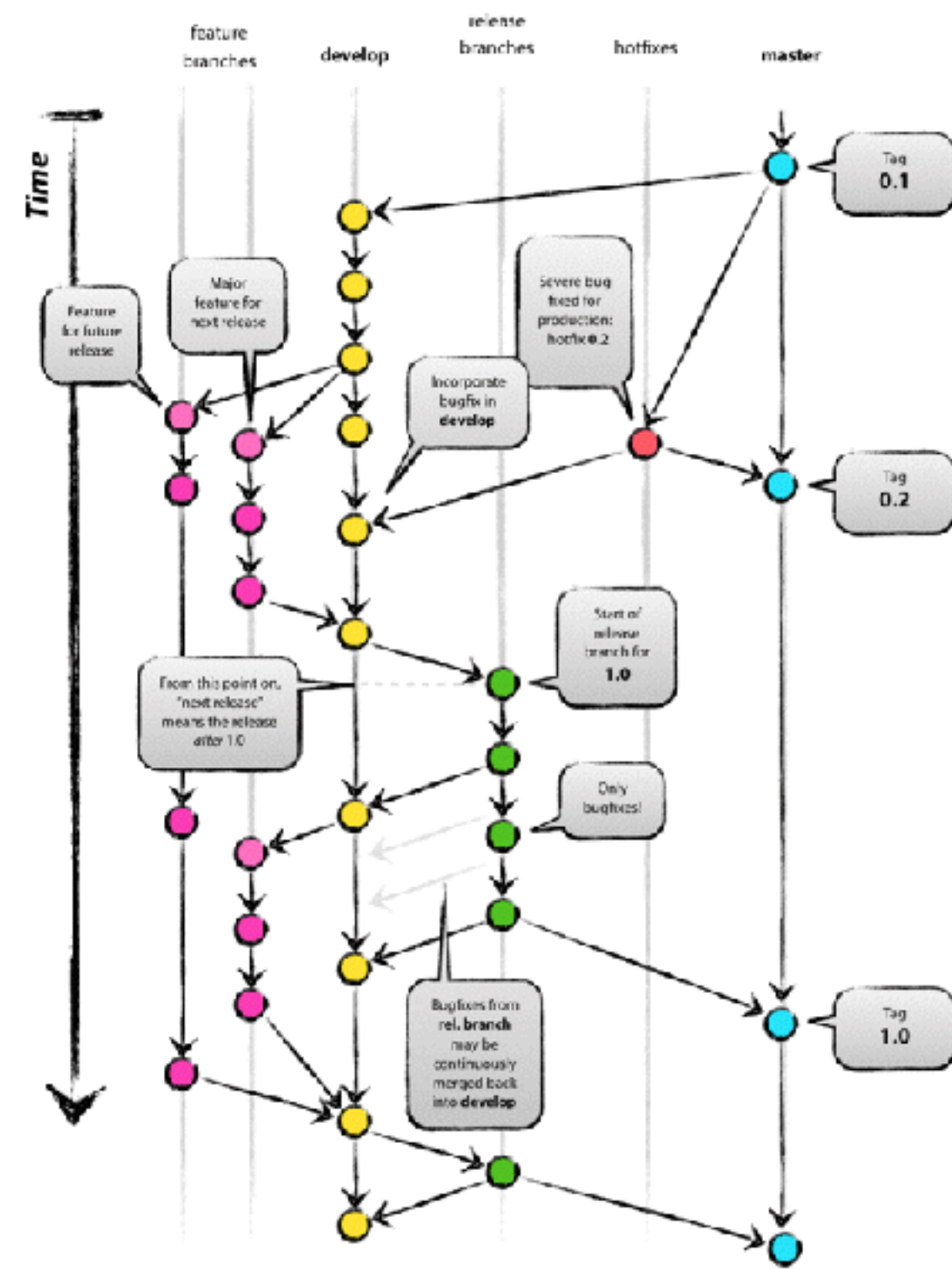
1. 编码规范 (vue、JS、css、html、CodeReview)
2. 命名规范 (文件命名、代码命名、Commit描述规范)
3. 流程规范 (测试流程、开发流程、[Git分支管理](#))
4. 前后端接口规范

Git分支管理最佳实践

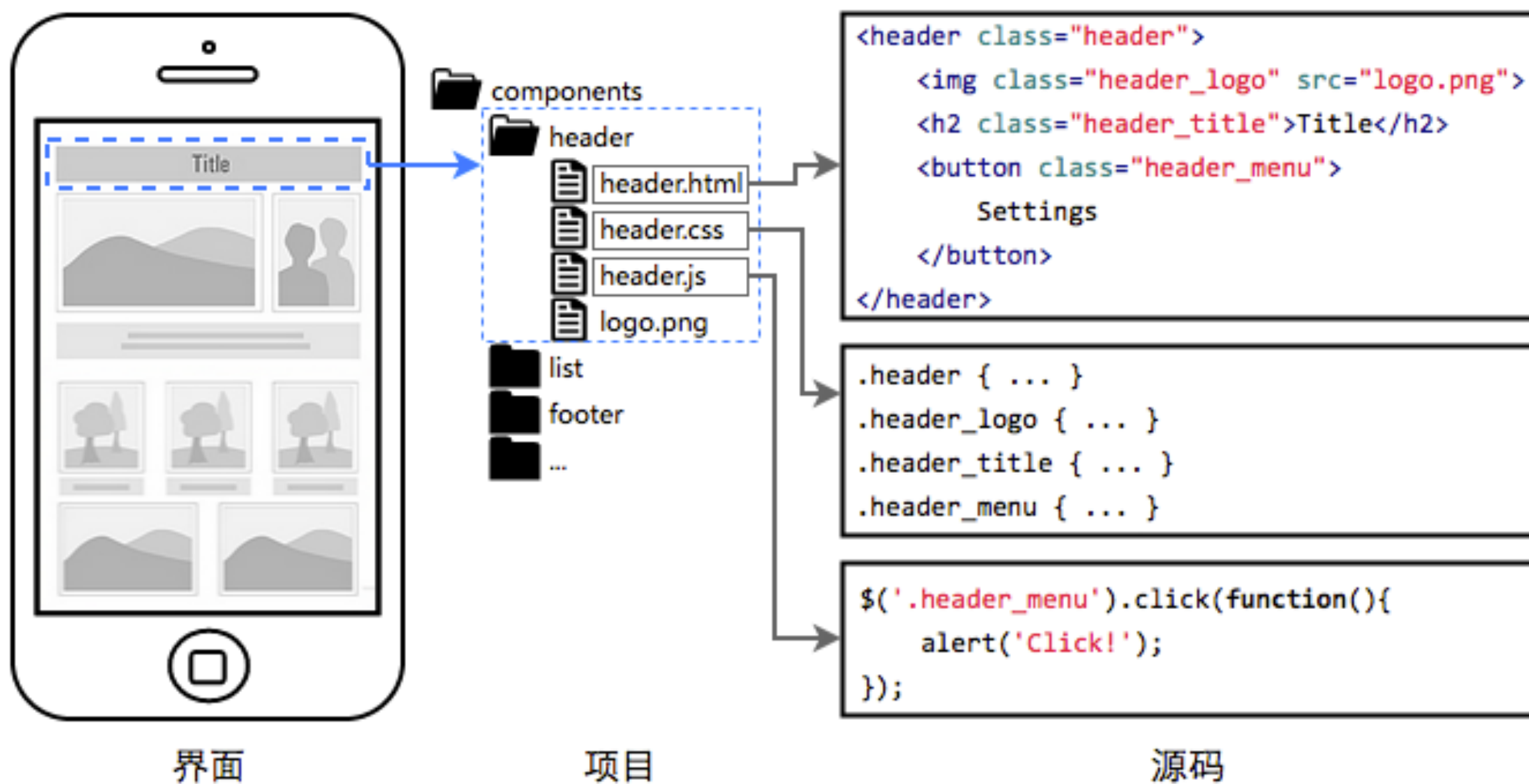
[git-flow](#)

版本发布

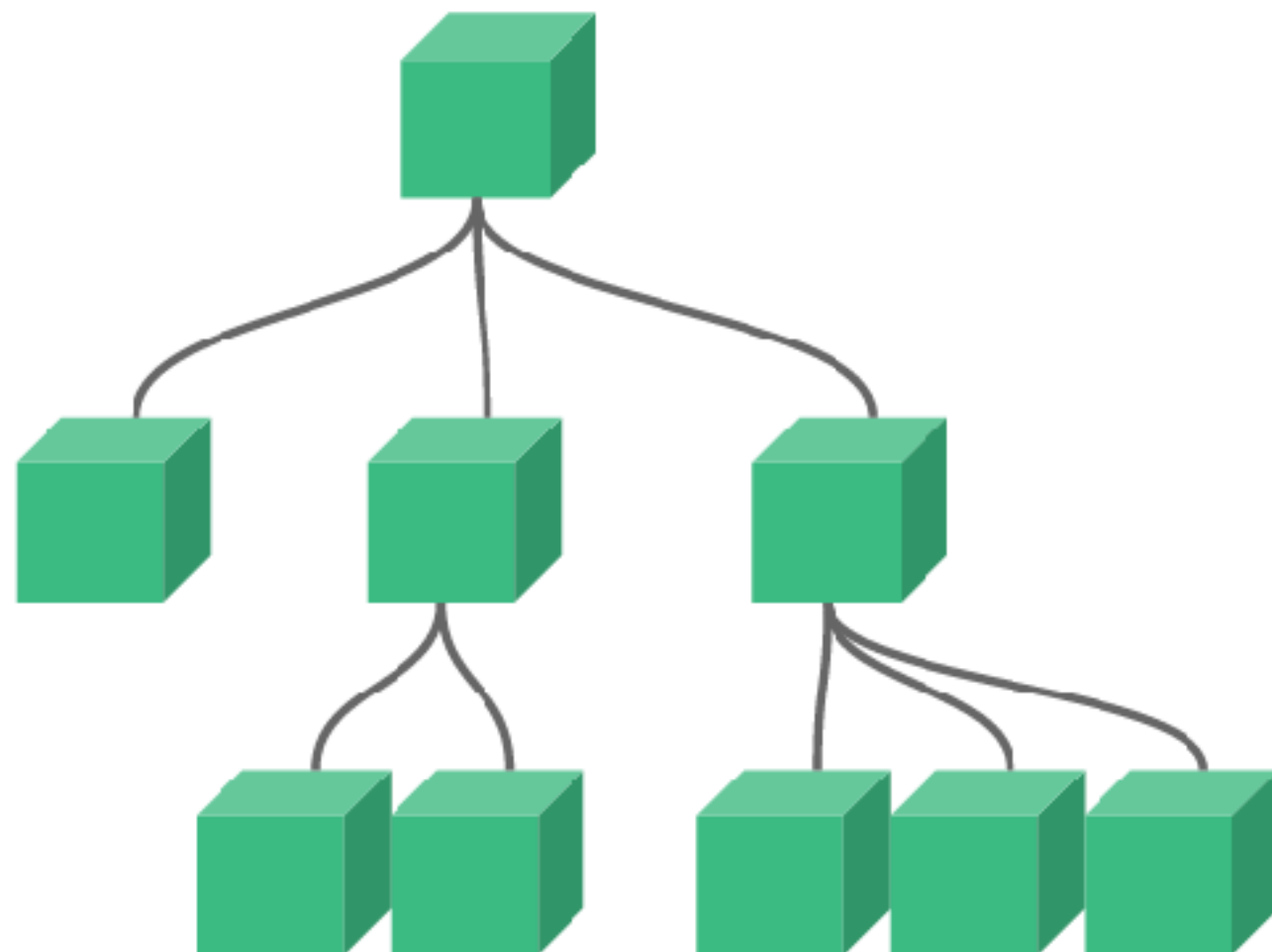
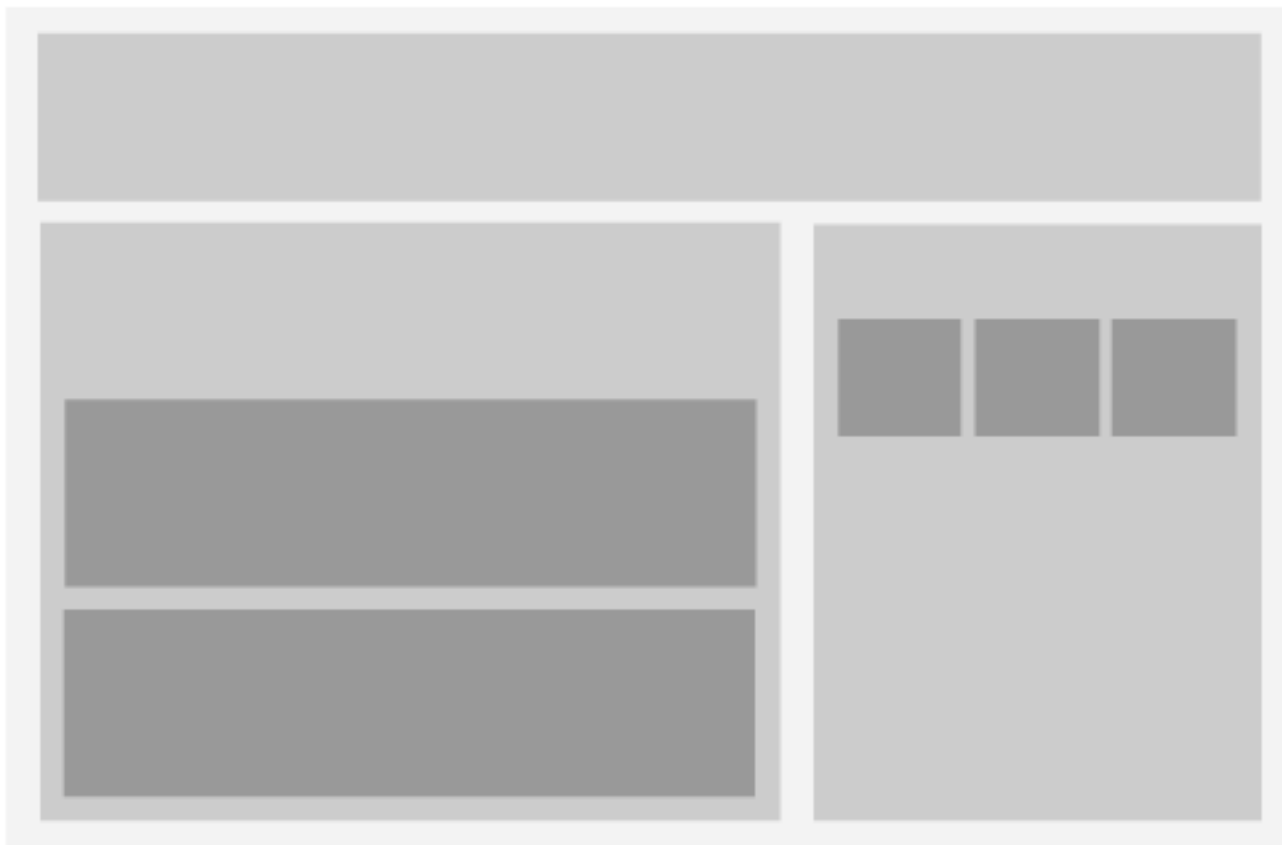
持续集成、持续交付



开发连调 - 组件化开发



组件树



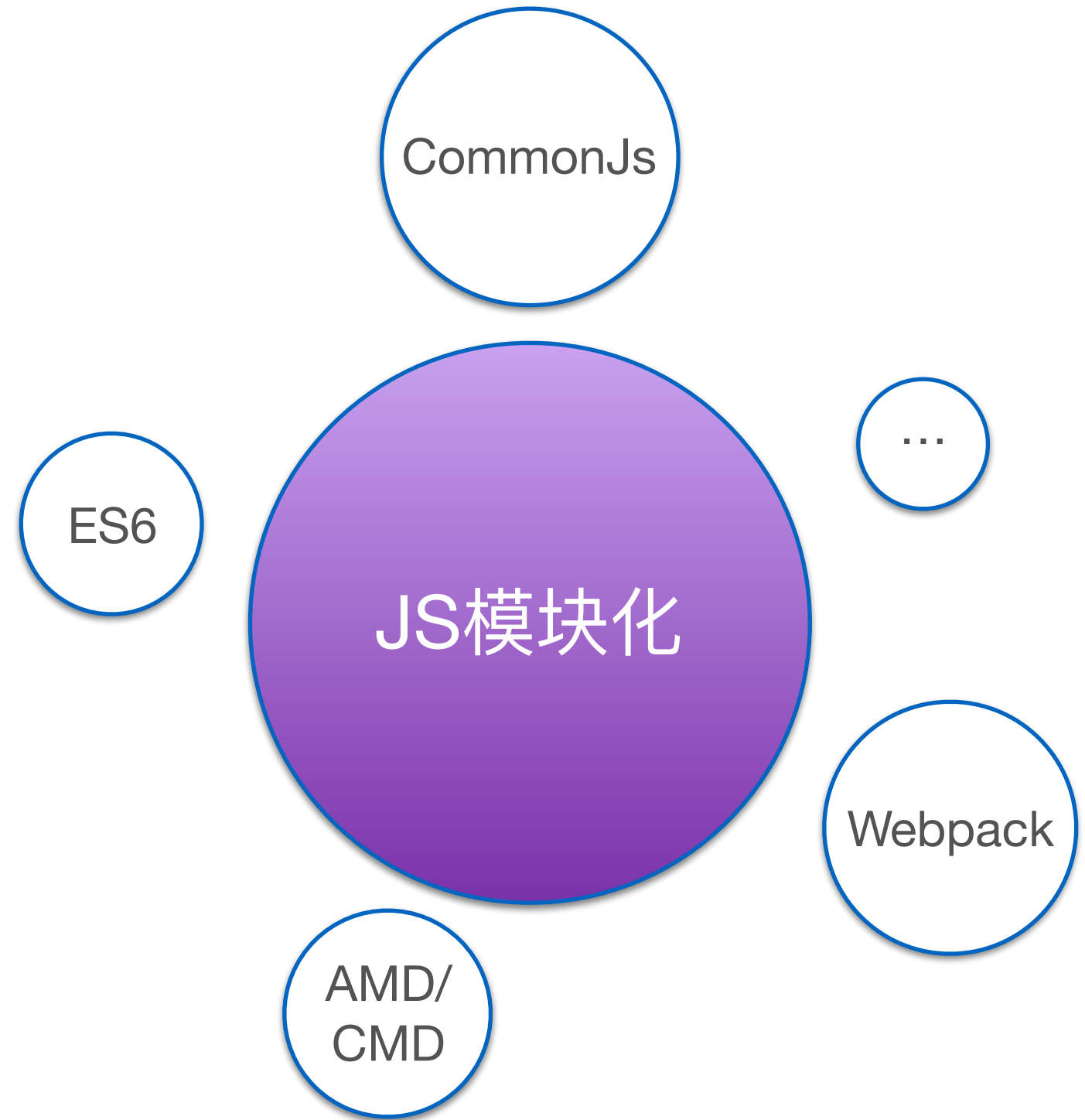
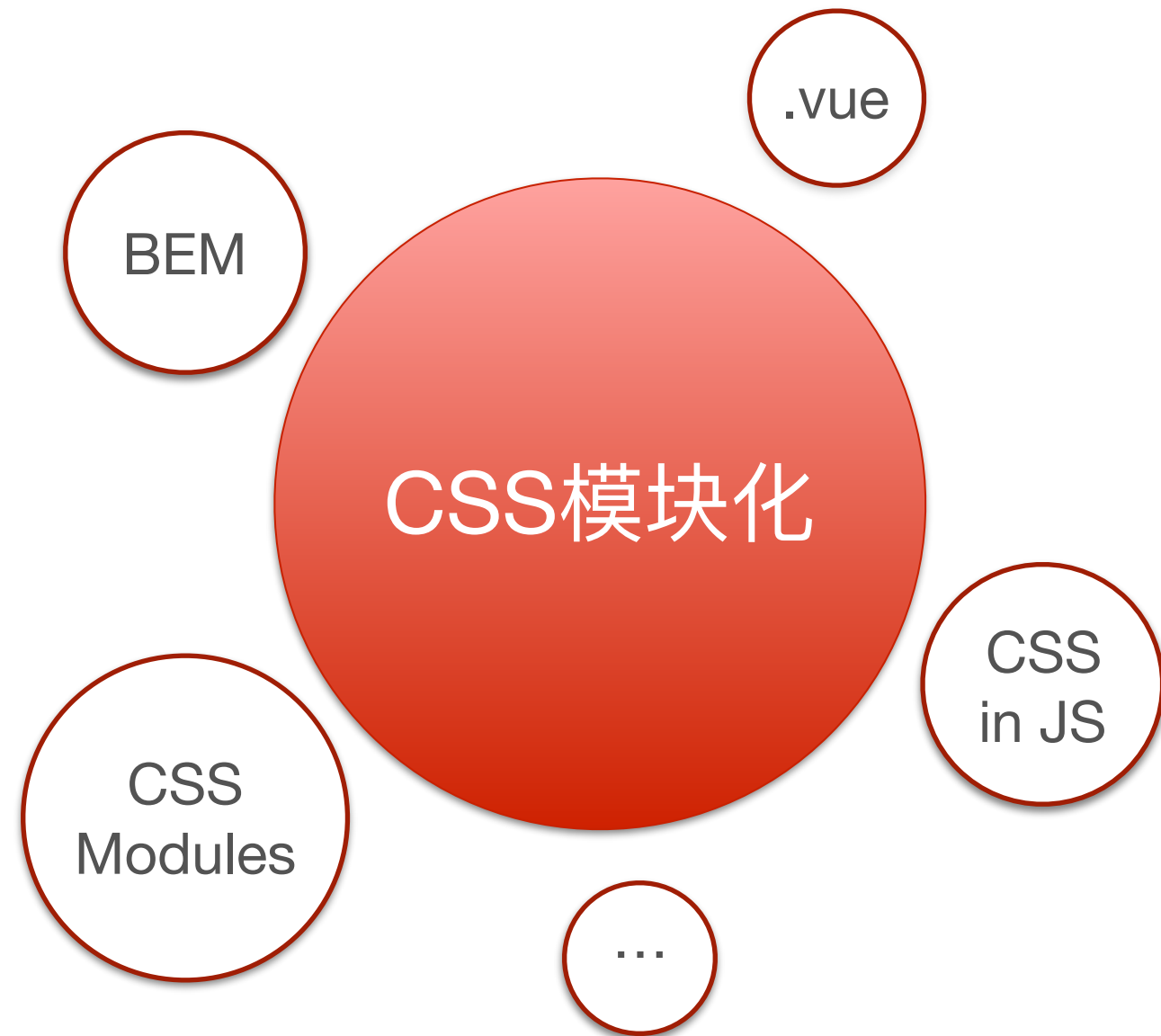
四类组件

1. 纯展示组件
2. 接入性组件container: 会和数据层打交道
3. 交互型组件: 如各类表单组件，通常强调复用
4. 功能型组件: 作为功能扩展而存在，如 `<router-view>`，`<transition>`

怎么写上面的组件？ Template or JSX

关注点分离(**Separation of Concerns**) and **Colocation**

开发连调 - 模块化



很多人觉得模块化开发的工程意义是**复用**，我不太认可这种看法，在我看来，模块化开发的**最大价值**应该是**分治**，是**分治**，**分治**！（重说三）。

性能优化

优化方向	优化手段
请求数量	合并脚本和样式表，CSS Sprites，拆分初始化负载，划分主域
请求带宽	开启GZip，精简JavaScript，移除重复脚本，图像优化
CDN	CDN缓存
页面结构	将样式表放在顶部，将脚本放在底部，尽早刷新文档的输出
代码校验	避免CSS表达式，避免重定向

测试

人工测试 or 自动化测试

浏览器兼容性测试

功能样式测试

性能测试



部署发布

容器化[部署](#) 自动化发布（一键）

当前端在讨论技术方案的时候，我们在讨论什么？ (技术选型)

当前端在改进工作流程的时候，我们在改进什么？ (规范化、标准化)

当前端在考虑提升效率的时候，我们在考虑什么？ (自动化)

当前端在和后端扯皮的时候，我们在扯什么？ (流程化、前后端分离)