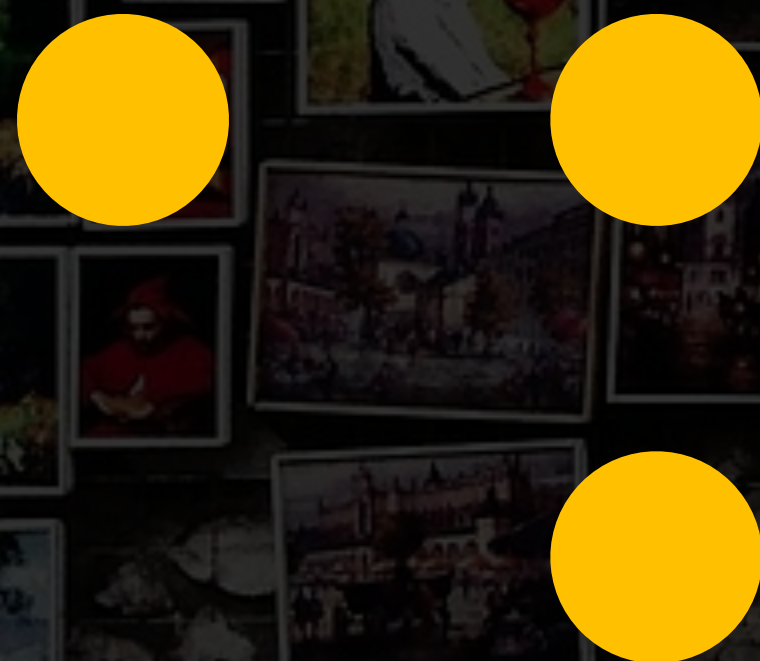


Node.js在CLI下工程化体系实践

IVWEB 程柳锋



- 2015 ~ 2017 阿里闲鱼拍卖
- 2017 ~ 至今 腾讯Now直播
- Github: <https://github.com/cpselvis>
- 喜欢研究协作规范、对前端工程化体系建设和自动化CMS系统搭建有一定积累



个人简介

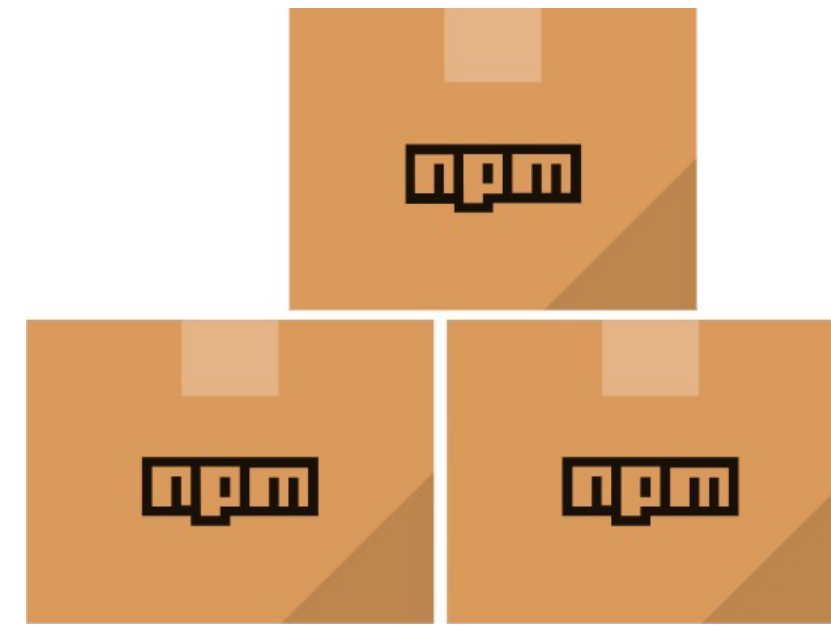
内容大纲



Why Node.js?



轻量级

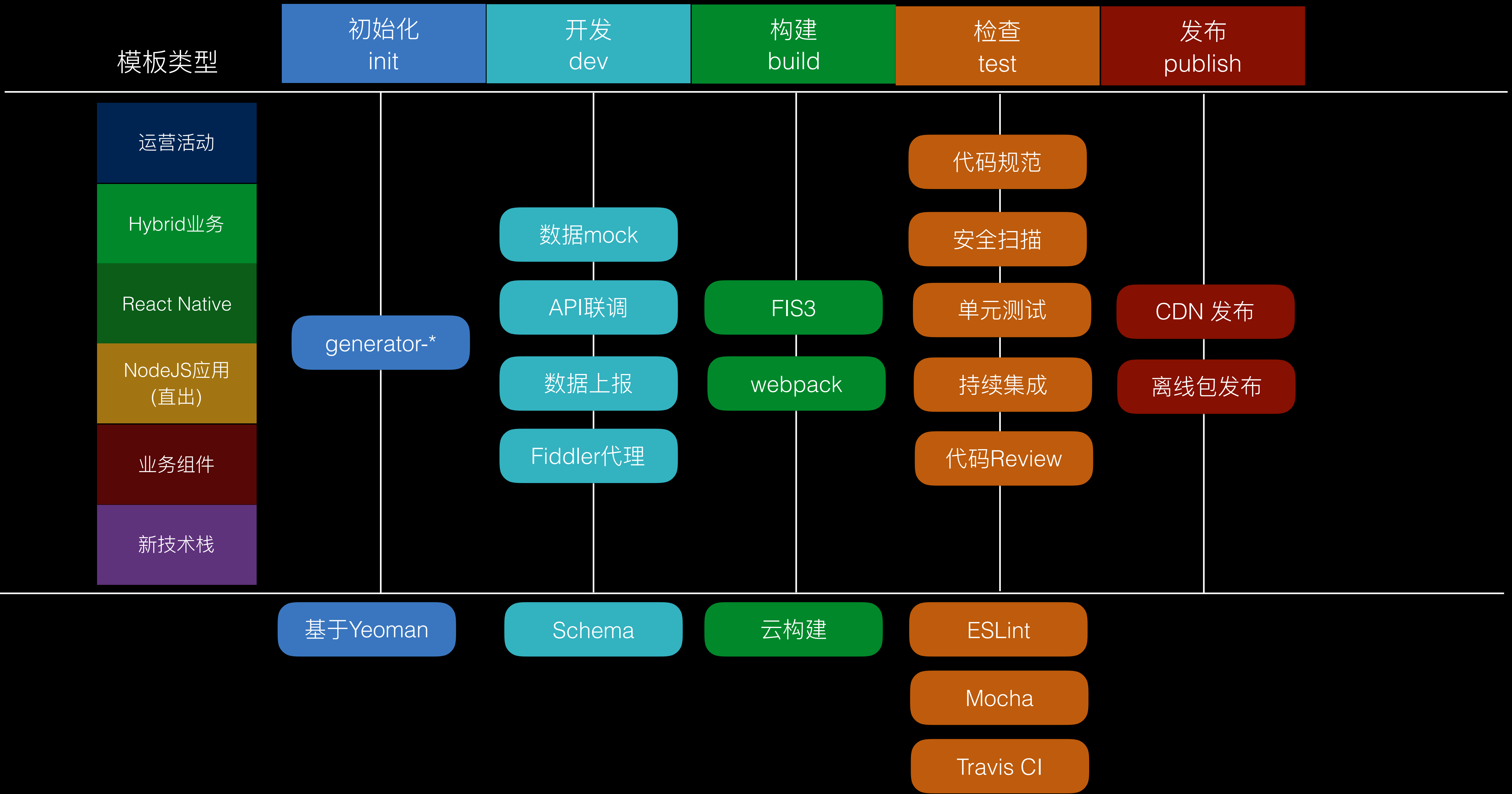


47.5万+三方包




跨平台

工程生命周期介绍



痛点1: 项目拷贝

- 容易出错
- 不同场景下对目录结构要求不同
- 新的Feature和BugFix难以同步



存在的问题？

解决方案 - Yeoman



initializing

prompting

configuring

writing

install

end

初始化

用户交互

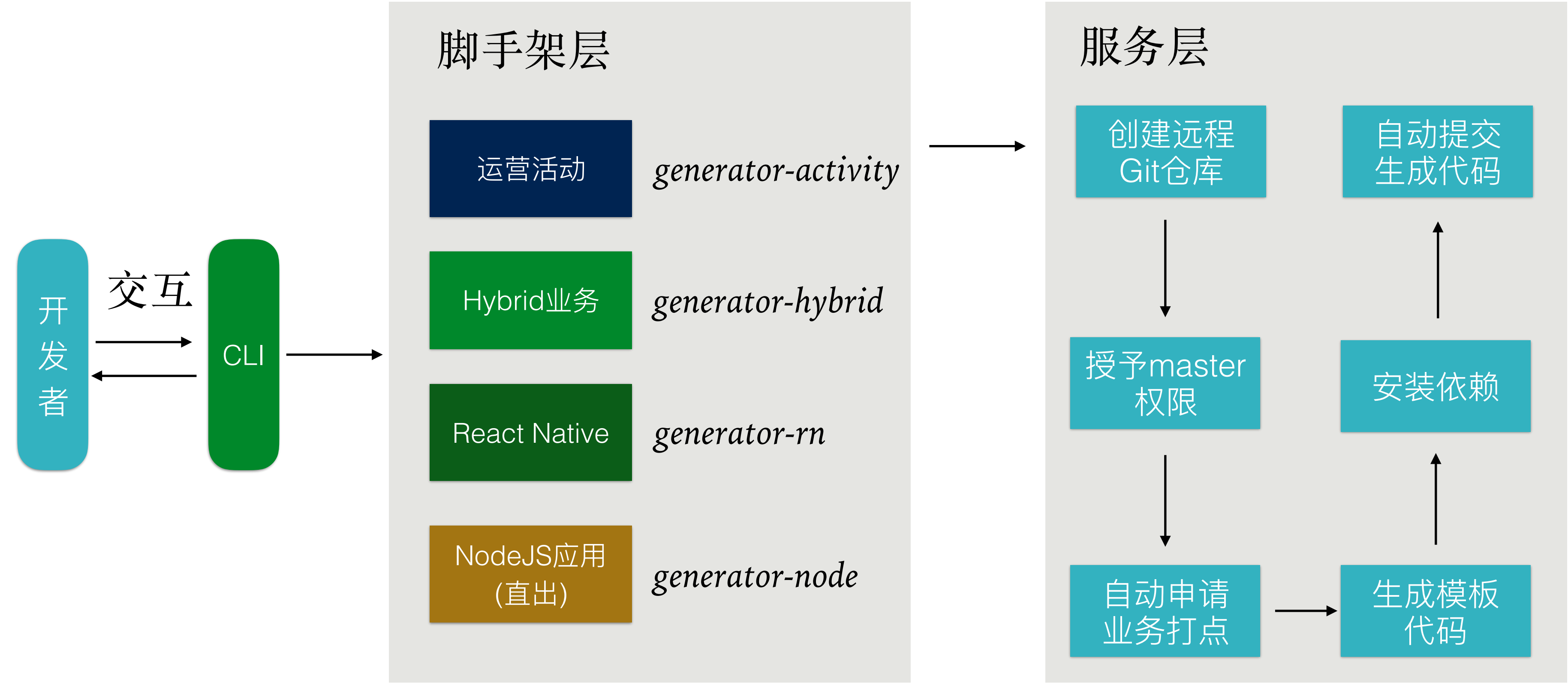
创建配置

创建项目文件

安装依赖

结束

基于Yeoman的脚手架设计思路



如何打通Git仓库

STEP 1： 申请公共邮箱，并关联git公共账号

STEP 2： 调用Git code API来进行仓库创建仓库

STEP 3： 使用CLI调用Git来执行仓库克隆、初始化功能

痛点2: 运营配置频繁修改

页面由N个组件拼接而成



Banner组件

`<Banner />`

跑马灯组件

`<Marquee />`

抽奖组件

`<Lottery />`

兑换组件

`<Exchange />`

奖品组件

`<Reward />`

获奖榜单组件

`<RankList />`

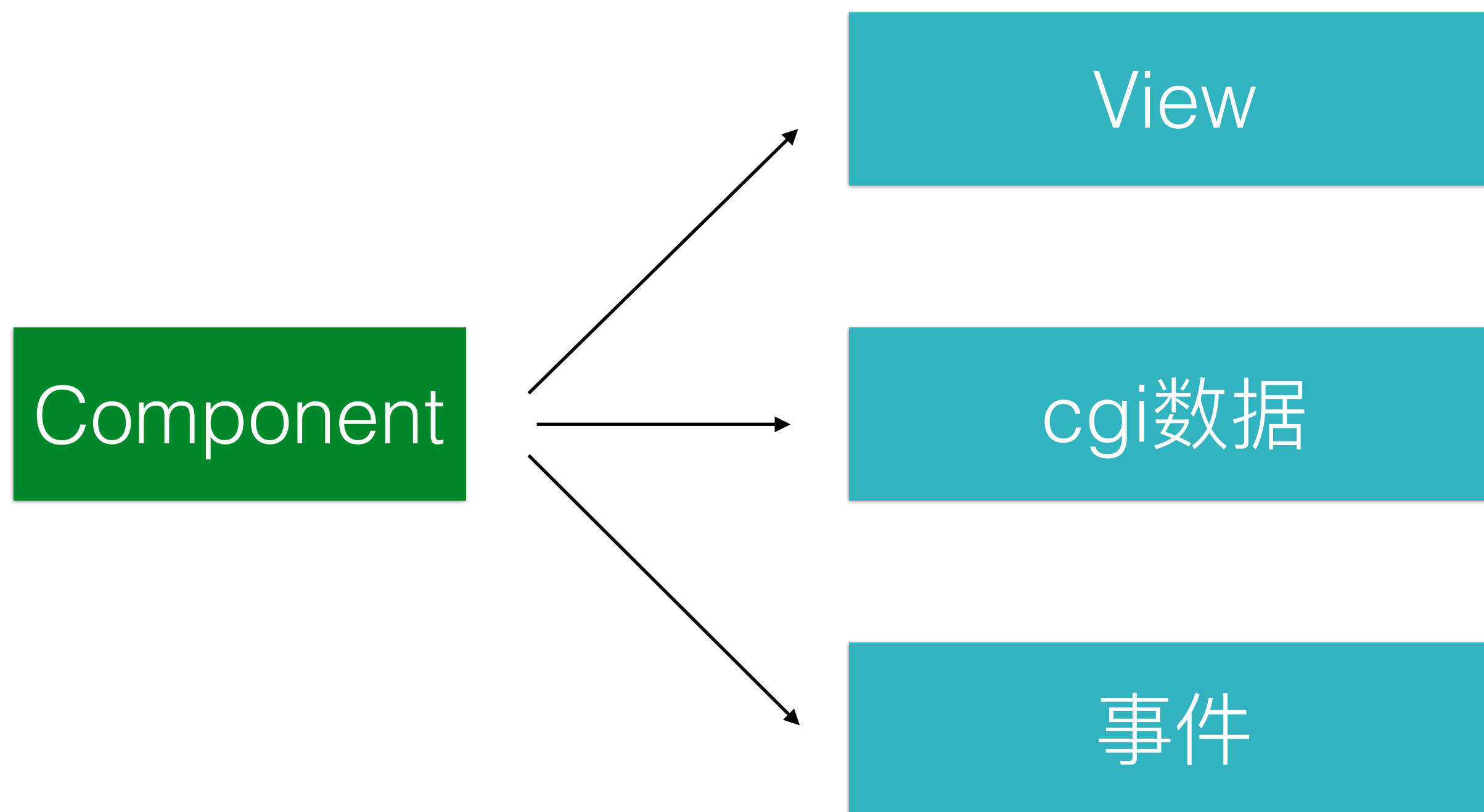
规则组件

`<Rule />`

下载bar组件

`<Downloader />`

组件的构成



静态数据
如何抽离?



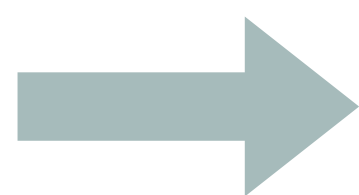
实际案例

```
render() {
  return (
    <div className="wrapper">
      <div className="section">
        <h3>活动时间: </h3>
        <p>4月10日00:00~4月17日18:00</p>
      </div>
      <div className="section">
        <h3>活动规则: </h3>
        <p>1、活动期间,在NOW录制原创小视频,上传成功即可参赛</p>
        <p>2、根据参赛小视频获得的点赞数进行排行</p>
        <p>3、按照城市评选,分别评选“明日之子”(仅限男性参加)和“闪亮女神”(仅限女性参加) </p>
        <p>4、按照城市规模每个城市分别选取Top100/Top 50/Top 10选手获得iPhone、金币、粉丝团体验特</p>
        <p>5、明日之子特别福利: 获奖选手有机会与杨幂等明日之子星推官面对面接触, 实现明星梦想</p>
      </div>
    </div>
  );
}
```


改进-Schema First

react-jsonschema

```
{
  "title": "主播任务活动数据信息配置",
  "type": "object",
  "required": [
    "actName",
    "templateType"
  ],
  "properties": {
    "ruleInfo": {
      "type": "object",
      "title": "规则模块",
      "properties": {
        "title": {
          "type": "string",
          "title": "规则标题"
        },
        "rules": {
          "type": "array",
          "title": "规则文案",
          "items": {
            "type": "object",
            "properties": {
              "subTitle": {
                "type": "string",
                "title": "子标题"
              },
              "subContent": {
                "type": "array",
                "title": "规则列表（可以只填一项）",
                "items": {
                  "type": "object",
                  "properties": {
                    "text": {
                      "type": "string",
                      "title": "规则"
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```



规则模块

规则标题

活动规则

规则文案

子标题

活动时间

规则列表（可以只填一项）

规则

2017年9月14日-9月20日

子标题

活动细则

规则列表（可以只填一项）

规则

1、活动期间，为主播赠送礼物，可帮主播击退如?

规则

2、用户给主播赠送任意礼物，入侵主播的如花将?

规则

3、用户给主播赠送1朵小红花，主播可获得1点【?

痛点3: 协作规范

Git commit规范与changelog生成

- 优势：
 - 加快Review的流程
 - 根据Commit元数据生成Changelog
 - 后续维护者可以知道feature被添加的原因
- 技术方案



提交格式要求

```
<type>(<scope>): <subject>  
<BLANK LINE>  
<body>  
<BLANK LINE>  
<footer>
```


对格式的说明如下：

- type代表某次提交的类型，比如是修复一个bug还是增加一个新的feature。所有的type类型如下：
- feat：新增feature
- fix: 修复bug
- docs: 仅仅修改了文档，比如README, CHANGELOG, CONTRIBUTE等等
- style: 仅仅修改了空格、格式缩进、都好等等，不改变代码逻辑
- refactor: 代码重构，没有加新功能或者修复bug
- perf: 优化相关，比如提升性能、体验
- test: 测试用例，包括单元测试、集成测试等
- chore: 改变构建流程、或者增加依赖库、工具等
- revert: 回滚到上一个版本


标准化的Commit信息

commit 日志:


Commits on Sep 10, 2017




docs: add plugin usage section
cpselvis committed on GitHub 2 days ago ✓




docs: update init project step.
cpselvis committed on GitHub 2 days ago ✓




style: pkg version
cpselvis committed 2 days ago ✓




fix: fix deps
cpselvis committed 2 days ago ✓



docs: changelog update
cpselvis committed 2 days ago ✓




fix: npm registry and proxy can be set when initialized.
cpselvis committed 2 days ago ✓




feat: init client before load external plugins
cpselvis committed 2 days ago ✓

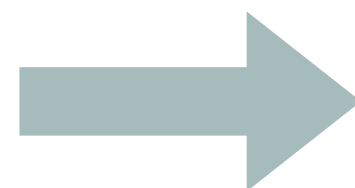
Commits on Sep 9, 2017



feat: use yaml to store local config
cpselvis committed 3 days ago ✗



feat: check node engine requirements
cpselvis committed 3 days ago ✓



changelog:

0.10.1 (2017-09-10)

Bug Fixes

- npm registry and proxy can be set when initialized. ([a9a8119](#))
- support yeoman generator app/index.js folder ([bedfaa9](#))

Features

- check node engine requirements ([499358d](#))
- init client before load external plugins ([2982f3d](#))
- print banner when no args ([3fff39c](#))
- use yaml to store local config ([c7d3241](#))

0.10.0 (2017-08-21)

Features

- add uninstall plugin command. ([84de19a](#))
- command register mechanism. ([febe1c3](#))
- Initial commit ([626def2](#))
- support -v and --version command. ([fa14d9e](#))

痛点4: 代码规范

从一次生产事故说起

- 2017年4月13日，腾讯高级工程师小圣在做充值业务时，修改了苹果iap支付配置，将JSON配置增加了重复的key。代码发布后，有小部分使用了vivo手机的用户反馈充值页面白屏，无法在Now app内进行充值。最后问题定位是：vivo手机使用了系统自带的webview而没有使用X5内核，解析JSON时遇到重复key报错，导致页面白屏。

如何避免类似
代码问题？



ESLint介绍

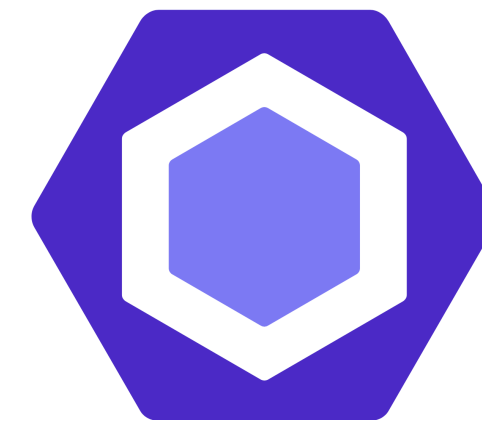
- Javascript和JSX代码规范检查工具
- 2013年6月推出，最新版本v4.6.0
- 支持插件扩展，可以自定义规则。



- 配置已经约定好，开箱即用
- 规则严格，规则无法禁用
- 无法根据错误定位到具体规则



- 支持ES6语法
- 规则可配置
- 不支持自定义规则
- 无法根据错误定位到具体规则



- 支持ES6和JSX
- 支持插件扩展
- 支持自定义配置
- 可以配置错误级别

代码规范制定原则

- 不重复造轮子，基于eslint:recommended配置并改进
- 能够帮助发现代码错误的规则，全部开启
- 配置不应该依赖于某个具体项目，而应尽可能的合理
- 帮助保持团队的代码风格统一，而不是限制开发体验
- 有对应的解释文档

创建ESLint共享配置

规则名称	错误级别	说明
for-direction	error	for 循环的方向要求必须正确
getter-return	error	getter必须有返回值，并且禁止返回值为undefined, 比如 return;
no-await-in-loop	off	允许在循环里面使用await
no-console	off	允许在代码里面使用console
no-prototype-builtins	warn	直接调用对象原型链上的方法
valid-jsdoc	off	函数注释一定要遵守jsdoc规则
no-template-curly-in-string	warn	在字符串里面出现{和}进行警告
accessor-pairs	warn	getter和setter没有成对出现时给出警告
array-callback-return	error	对于数据相关操作函数比如reduce, map, filter等， callback必须有return
block-scoped-var	error	把var关键字看成块级作用域，防止变量提升导致的bug
class-methods-use-this	error	要求在Class里面合理使用this，如果某个方法没有使用this,则应该申明为静态方法
complexity	off	关闭代码复杂度限制
default-case	error	switch case语句里面一定需要default分支

集成进push hook

1, 安装*husky*

```
npm install husky --save-dev
```

2, 增加*npm script*

```
// Edit package.json
{
  "scripts": {
    "precommit": "validate-commit-msg",
    "prepush": "eslint src ./eslinttrc.js --ext '.js,.jsx'",
    "...": "..."
  }
}
```


CLI设计

整体架构

插件生态



核心层



参数parser

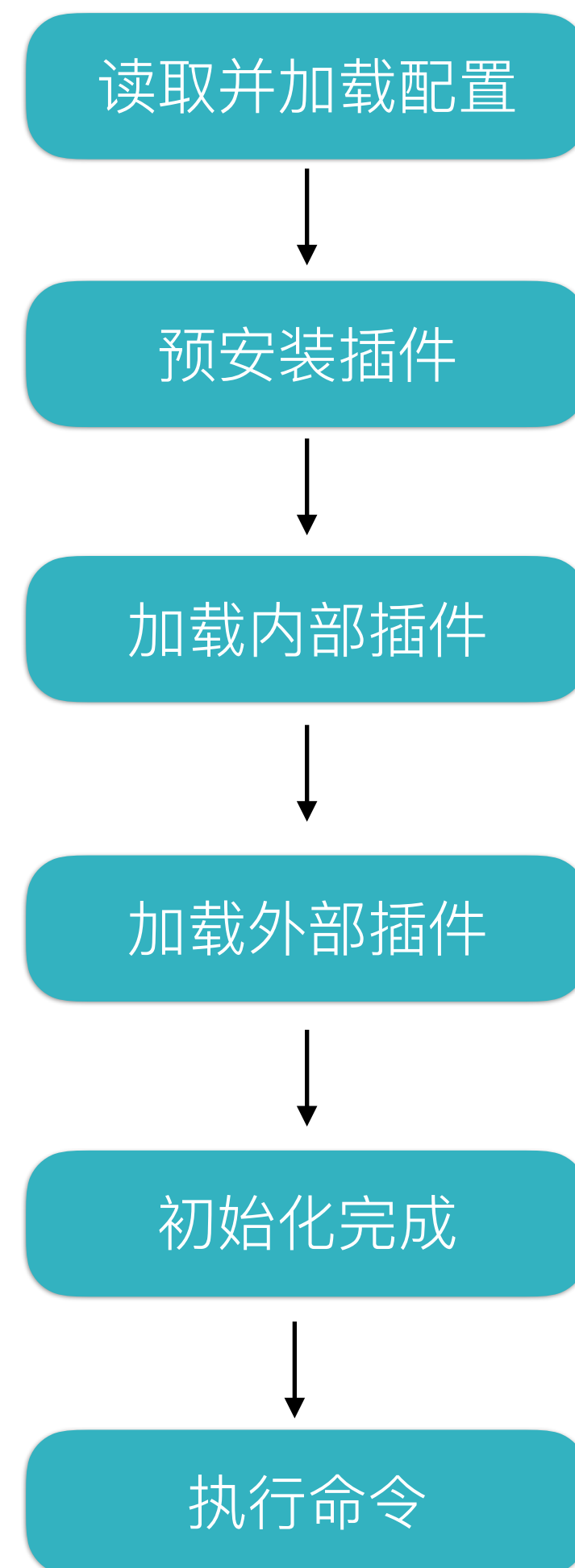


控制台



插件机制实现

CLI执行过程



通过vm注入上下文变量

```
loadPlugin(path, callback) {
  const self = this;

  return fs.readFile(path).then((script) => {

    const module = new Module(path);
    module.filename = path;
    module.paths = Module._nodeModulePaths(path);

    function require(path) {
      return module.require(path);
    }

    require.resolve = function(request) {
      return Module._resolveFilename(request, module);
    };

    require.main = process.mainModule;
    require.extensions = Module._extensions;
    require.cache = Module._cache;

    // Inject feflow variable
    script = '(function(exports, require, module, __filename, __dirname, feflow){' +
      script + '});';

    const fn = vm.runInThisContext(script, path);

    return fn(module.exports, require, module, path, pathFn.dirname(path), self);
  }).asCallback(callback);
}
```


插件上下文

➤ 命令注册

```
feflow.cmd.register('deps', 'Config ivweb dependencies', function(args) {  
  console.log(args);  
  // Plugin logic here.  
});
```

➤ 日志

```
const log = feflow.log;  
log.info()    // 提示日志, 控制台中显示绿色  
log.debug()   // 调试日志, 命令行增加--debug可以开启, 控制台中显示灰色  
log.warn()    // 警告日志, 控制台中显示黄色背景  
log.error()   // 错误日志, 控制台中显示红色  
log.fatal()   // 致命错误日志, , 控制台中显示红色
```

➤ 帮助函数

➤ spawn、request、yaml、fs等

END



更多技术信息，请加入腾讯IVWEB交流群