

Node.js 精影

不四@Alibaba-数据平台

- Node.js 介绍
- 异步编程
- 实战体验
- 企业级 node 开发

Node.js



Node.js is a platform built on **Chrome's JavaScript runtime** for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Current Version: v0.10.26

Node.js

- 09年发布，最新稳定版本 0.10.26
- github stars 排名第三

 **twbs/bootstrap** 65,084 stars
Sleek, intuitive, and powerful front-end framework for faster and easier web development.
 Unstar

 **jquery/jquery** 29,573 stars
jQuery JavaScript Library
 Unstar

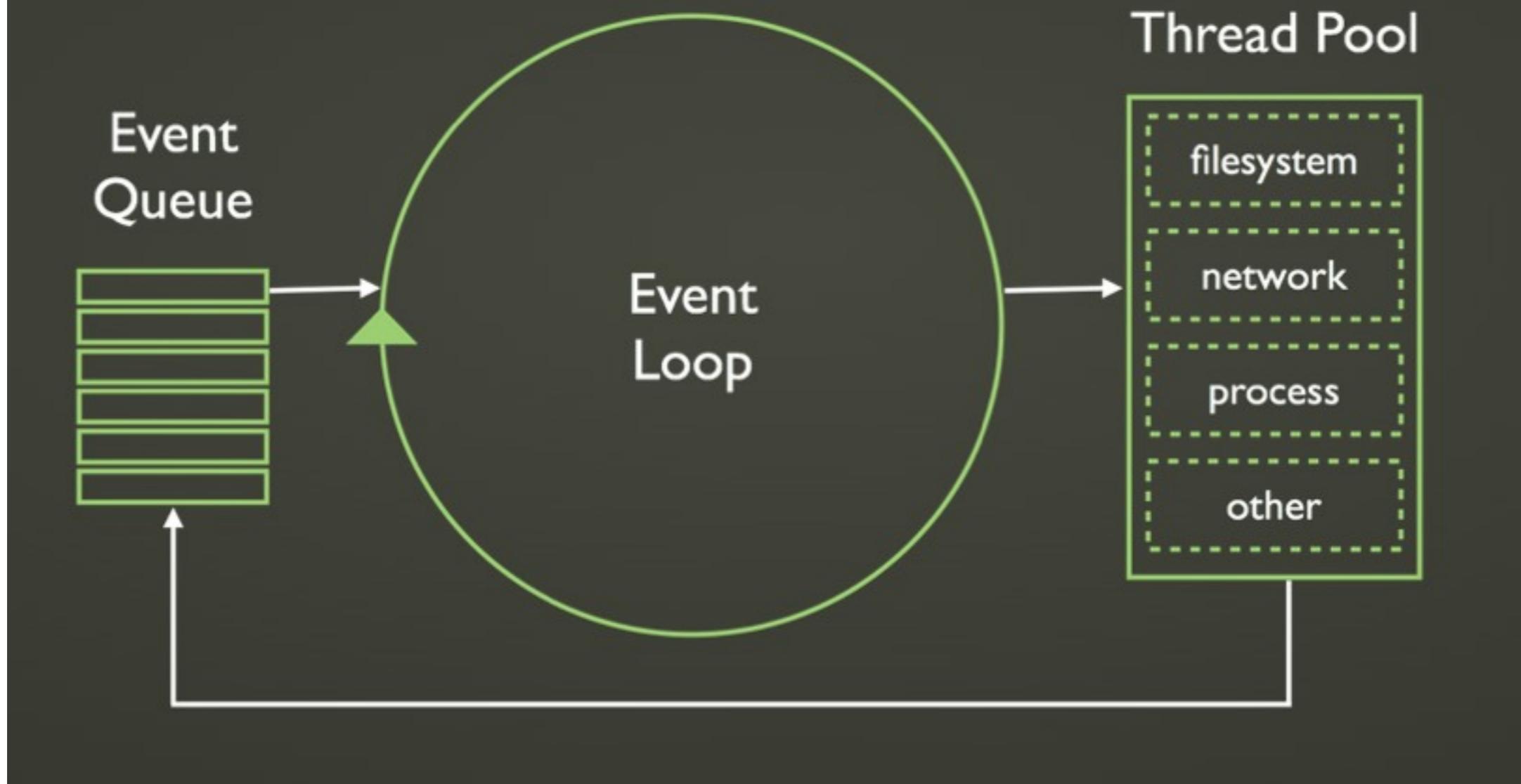
 **joyent/node** 27,921 stars
evented I/O for v8 javascript
 Unstar

Node.js 特性

- 基于 google v8 引擎（快速）
- 事件驱动（不同的编程思维）
- 非阻塞I/O（高效）
- 单线程（模型简单，无锁）

事件循环

Node.js Event Loop



单线程异步IO模型

- 无锁，逻辑简单
- 无多线程、多进程的开销，高性能
- 不用等待耗时的I/O
- 异步编程很痛苦
- 单线程未捕获异常导致进程挂掉
- CPU密集型操作堵塞整个线程

NPM

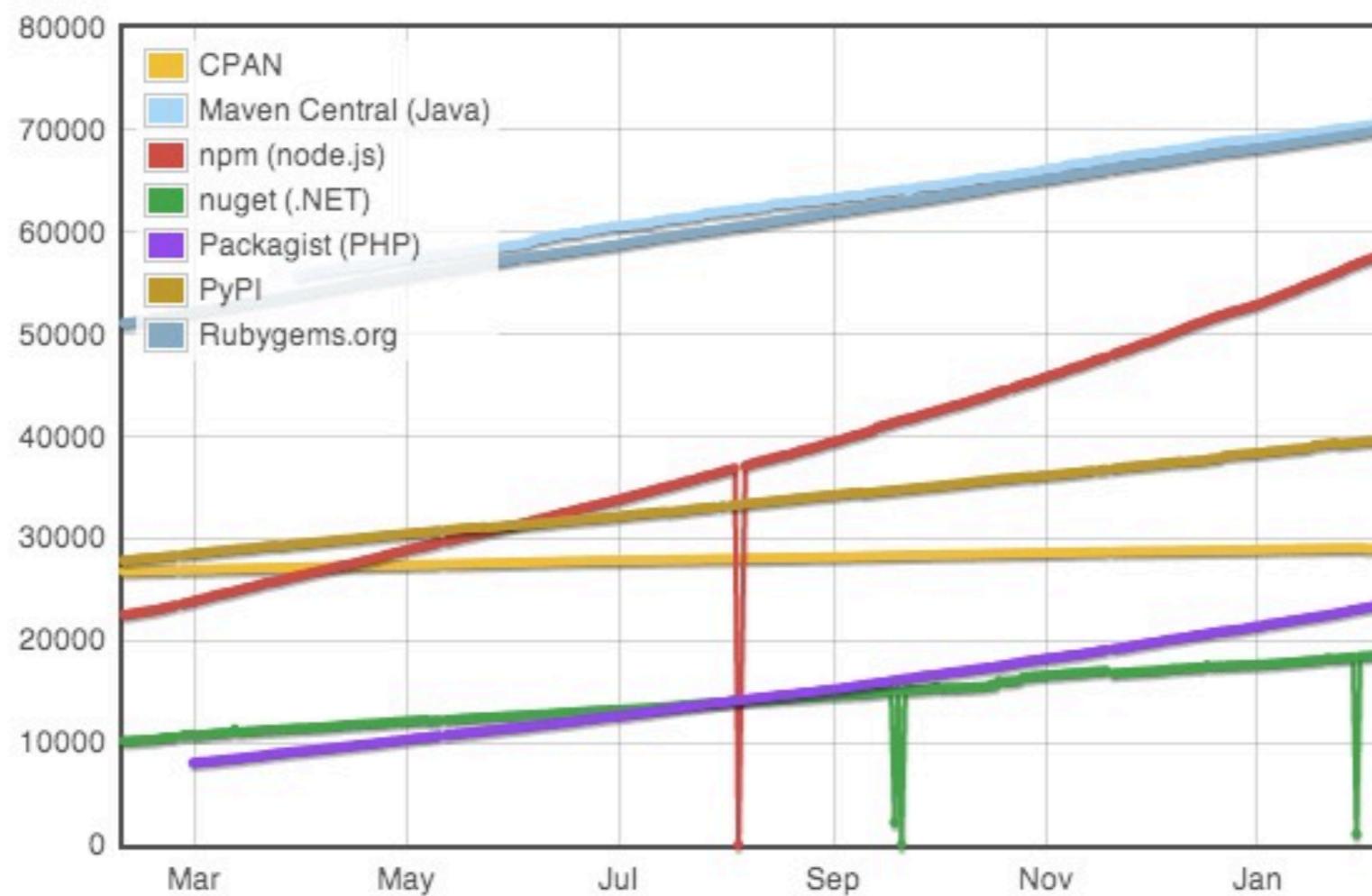
- 跟随 node 源码分发的包管理工具
- 精简的 node 内核的丰富扩展



NPM

- 包数量增速惊人

Module Counts



Company



 CBS Interactive

 LinkedIn

 Zappos[®]

 General Motors

 BARNES&NOBLE
BOOKSELLERS

 DocuSign[®]

 YAHOO!

 ebay

 Time Warner
Cable[®]

 travelocity[®]

 CONDÉ NAST

 Adobe[®]

 SimpleGeo[®]

 GoodData

 DOW JONES

 淘宝网
Taobao.com

 JETSETTER

 Yammer[®]

 UBER

 PEARSON

 Airbnb

 Chatham
Financial

 mozilla

 DELL

Server Side

- web 服务
 - restful API 服务
 - 前后端彻底分离的web前端服务
 - 全栈 web 应用
- 高性能中间层
- 高实时性应用(游戏服务器/在线聊天室)

Server Side

Connect

koa

express

3.0.0
...

restify



POMELO

sails

Meteor

Krakenjs

Front-end



Sea.js

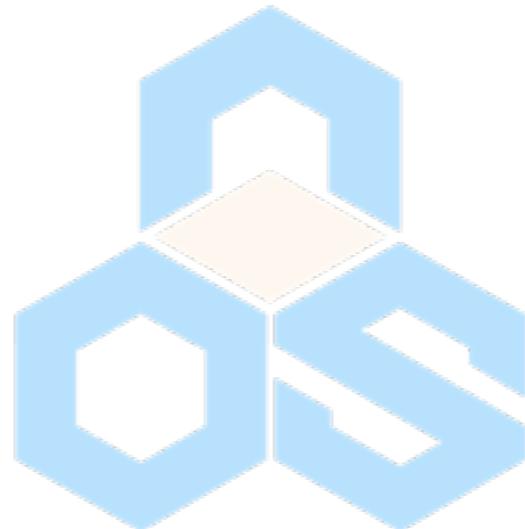
Browserify



browserling

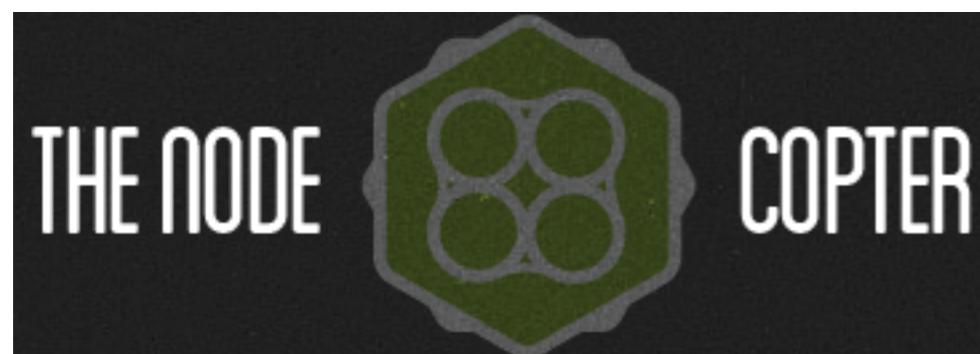


Beyond the Web



noduino

node-webkit



A T  M

小结

- node是基于v8，事件驱动的非阻塞I/O框架
- 有飞速发展的NPM和高活跃度的社区
- 深入到了 web 开发的每一个角落
- 同时不局限于 web 领域
- 虽然还不完美，但是慢慢成熟

- Node.js 介绍
- 异步编程
- 实战体验
- 企业级 node 开发

Callback Hell

```
function get(callback) {
  proxy.getUser(function (err, user) {
    if (err) {
      return callback(err);
    }
    proxy.getPosts(user, function (err, posts) {
      if (err) {
        return callback(err);
      }
      proxy.getComments(user, function (err, comments) {
        if (err) {
          return callback(err);
        }
        callback(null, {
          user: user,
          posts: posts,
          comments: comments
        });
      });
    });
  });
}
```

eventproxy

- 基于事件解耦
- 逻辑按照事件划分

```
function get(callback) {
  var ep = Eventproxy.create();
  ep.fail(callback);

  proxy.getUser(ep.done('user'));
  ep.once('user', function (user) {
    proxy.getPosts(user, ep.done('posts'));
    proxy.getComments(user, ep.done('comments'));
  });

  ep.all('user', 'posts', 'comments',
    function (user, posts, comments) {
      callback(null, {
        user: user,
        posts: posts,
        comments: comments
      });
    });
}
```

async

- 提供控制流辅助函数
- series
- waterfall
- parallel

```
function get(callback) {
  async.waterfall([
    proxy.getUser,
    getPostsAndComments
  ], callback);

  function getPostsAndComments(user, cb) {
    async.parallel([
      proxy.getPosts.bind(proxy, user),
      proxy.getComments.bind(proxy, user)
    ], function (err, res) {
      if (err) {
        return cb(err);
      }
      cb(null, {
        user: user,
        posts: res[0],
        comments: res[1]
      });
    });
  }
}
```

Q

- promise
- getUser().then
(success, fail);
- jQuery

```
var getUser = Q.denodeify(proxy.getUser);
var getPosts = Q.denodeify(proxy.getPosts);
var getComments = Q.denodeify(proxy.getComments);

function get(callback) {
  return getUser()
    .then(function (user) {
      return Q.all([user, getPosts(user), getComments(user)]);
    })
    .spread(function (user, posts, comments) {
      return {
        user: user,
        posts: posts,
        comments: comments
      };
    })
    .nodeify(callback);
}
```

generator

- es 6 新标准中的特性
- 尚未落地到 node 稳定版
- 同步方式编写异步代码
- try catch 异常

```
function *get() {
  var user = yield proxy.getUser;
  return yield {
    user: user,
    posts: proxy.getPosts(user),
    comments: proxy.getComments(user)
  };
}

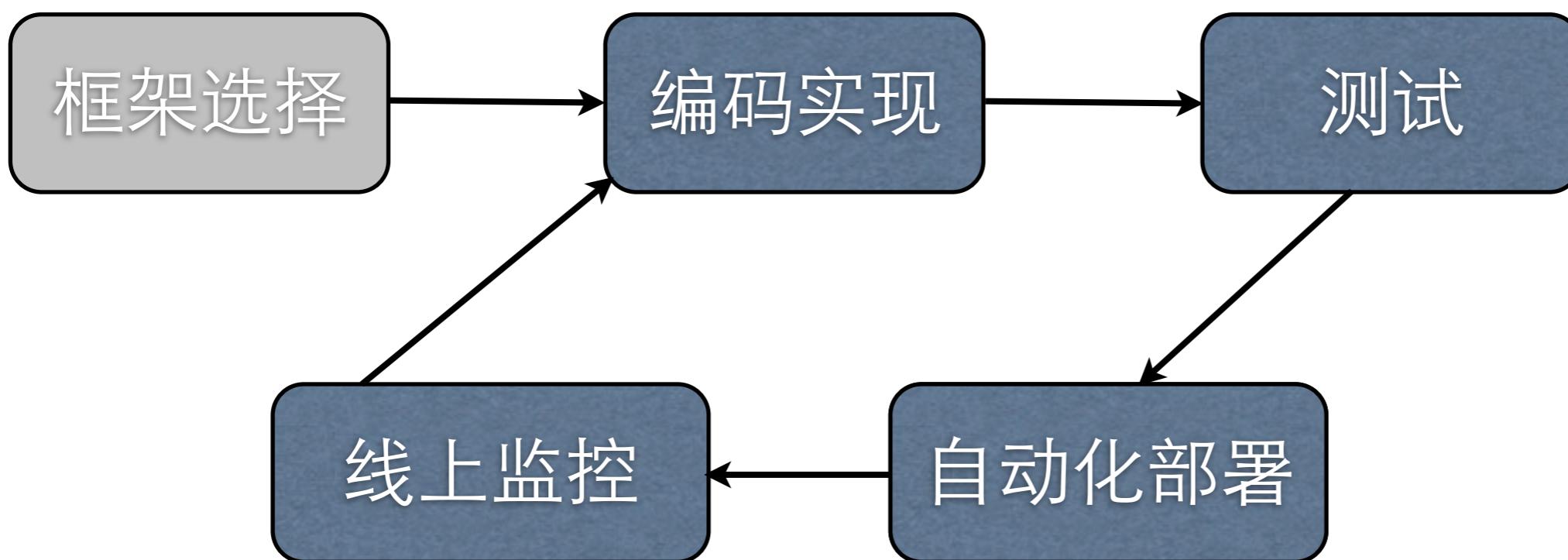
co(function *() {
  try {
    console.log(yield get);
  } catch (err) {
    console.error(err.stack);
  }
})();
```

更多解决方案？

- https://github.com/dead-horse/callback_hell

- Node.js 介绍
- 异步编程
- 实战体验
- 企业级 node 开发

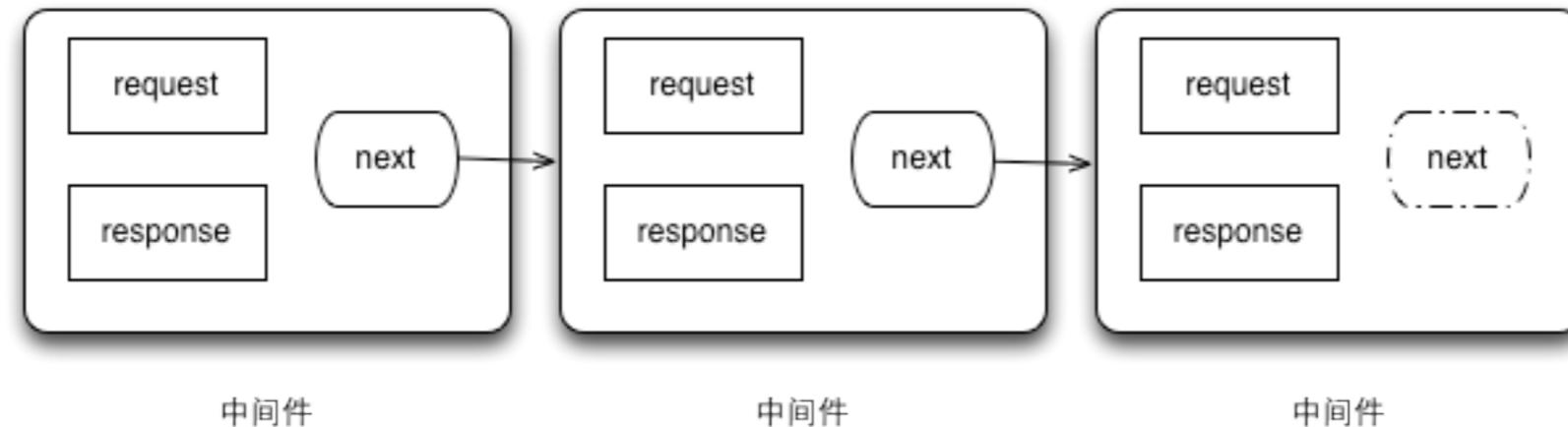
开发流程



框架选择

- express ?
 - node 社区最热门的 web 框架
 - 易于组合和插拔的中间件
 - 路由、模板引擎、等基础中间件模块很成熟

框架选择



```
module.exports = function responseTime(){
  return function(req, res, next){
    var start = new Date;

    if (res._responseTime) return next();
    res._responseTime = true;

    res.on('header', function(){
      var duration = new Date - start;
      res.setHeader('X-Response-Time', duration + 'ms');
    });

    next();
  };
};
```

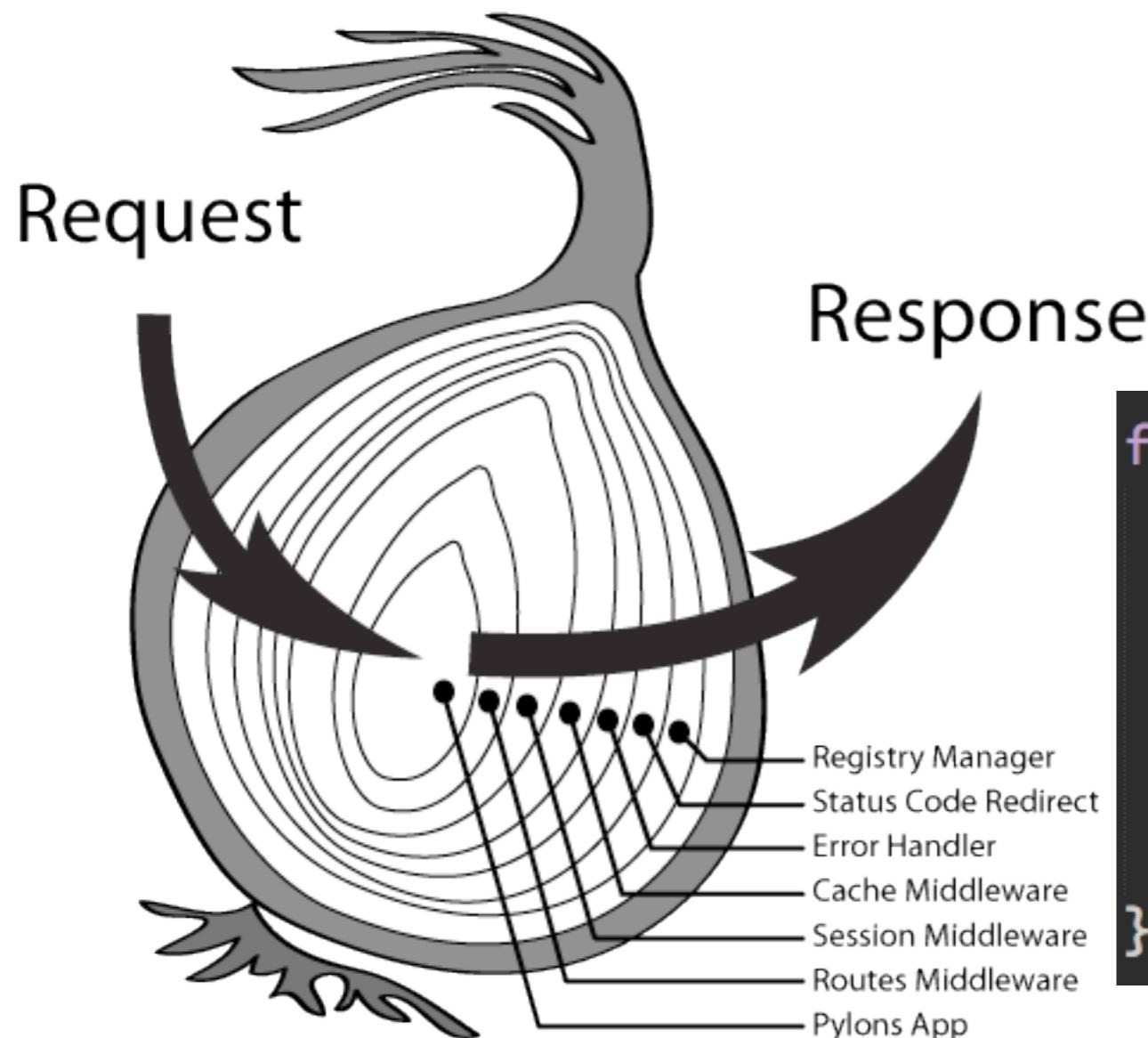
框架选择

- connect !
 - 比express更精简的内核
 - 兼容 express 中间件
 - 基础组件由自己实现，更强的掌控力

框架选择

- koa !!
 - es6 特性，完全基于 generator
 - 更轻松的异步解决方案和错误处理
 - 更灵活的中间件模型
 - getter / setter 带来的延迟初始化和更灵活的使用形式

框架选择



```
function responseTime() {  
    return function *responseTime(next){  
        var start = Date.now();  
        yield* next;  
        var delta = Date.now() - start;  
        this.set('X-Response-Time', delta);  
    };  
}
```

编码实现



编码实现

- 代码规范很有必要
- jshint
- code review
- git flow

模块化

- Node.js 社区推崇模块化
- 神器 NPM
- 活跃的社区， 6W+个模块

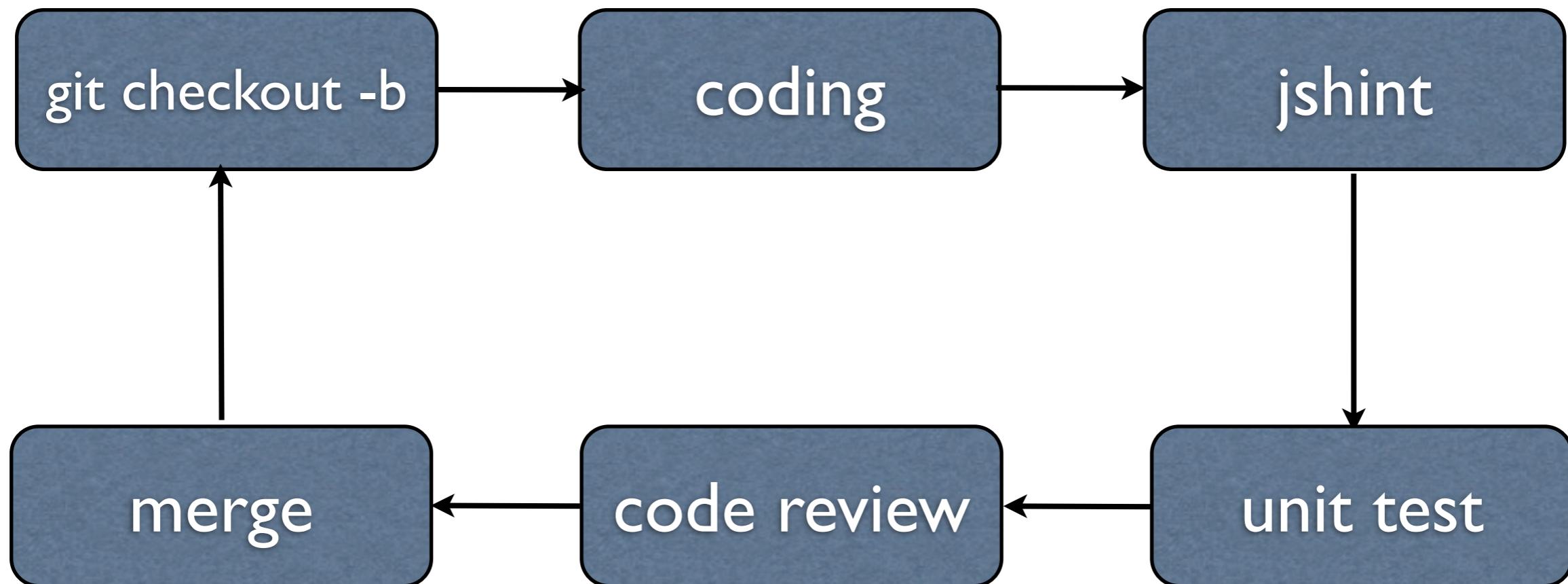
模块化

- 通过package.json文件管理依赖
- 从 NPM 安装依赖
- 抽离公用代码，封装成模块
- 如何选择模块？
 - 高活跃度、测试完备、作者靠谱
 - 尽量理解源码
 - 及时反馈bug

模块化

```
{  
  "name": "tnpm-server",  
  "version": "0.0.1",  
  "description": "alibaba Private npm registry and web",  
  "main": "index.js",  
  "dependencies": {  
    "cnpmjs.org": "0.3.1",  
    "connect": "2.13.0",  
    "connect-tair": "0.0.1",  
    "mkdirp": "0.3.5",  
    "oss-cnpm": "0.0.2",  
    "tair": "0.7.1",  
    "taobaoestatus": "0.1.1"  
  },  
  "devDependencies": {  
    "ali-contributors": "*",  
    "autod": "*"  
  },  
  "author": [  
    "不四 <busi.hyy@taobao.com>"  
  ]  
}
```

工作流



测试

- 单元测试
 - mocha + should
- 集成测试
 - github hook + travis-ci
 - gitlab hook + toast

自动化部署

- 打包机打包
 - git pull
 - npm install
 - set env
 - configure
 - tar && scp

自动化部署

- 生产环境服务启动
 - 启动 node 进程
 - 检查服务进程和端口监听
 - 启动 nginx(反向代理)
 - vip 检查
 - 服务上线

线上监控

- master + workers 多进程模式
 - worker 异常退出自动重启
 - error log (node) 报警监控
 - access log (nginx) 统计分析
 - 系统 cpu / memory / load 监控

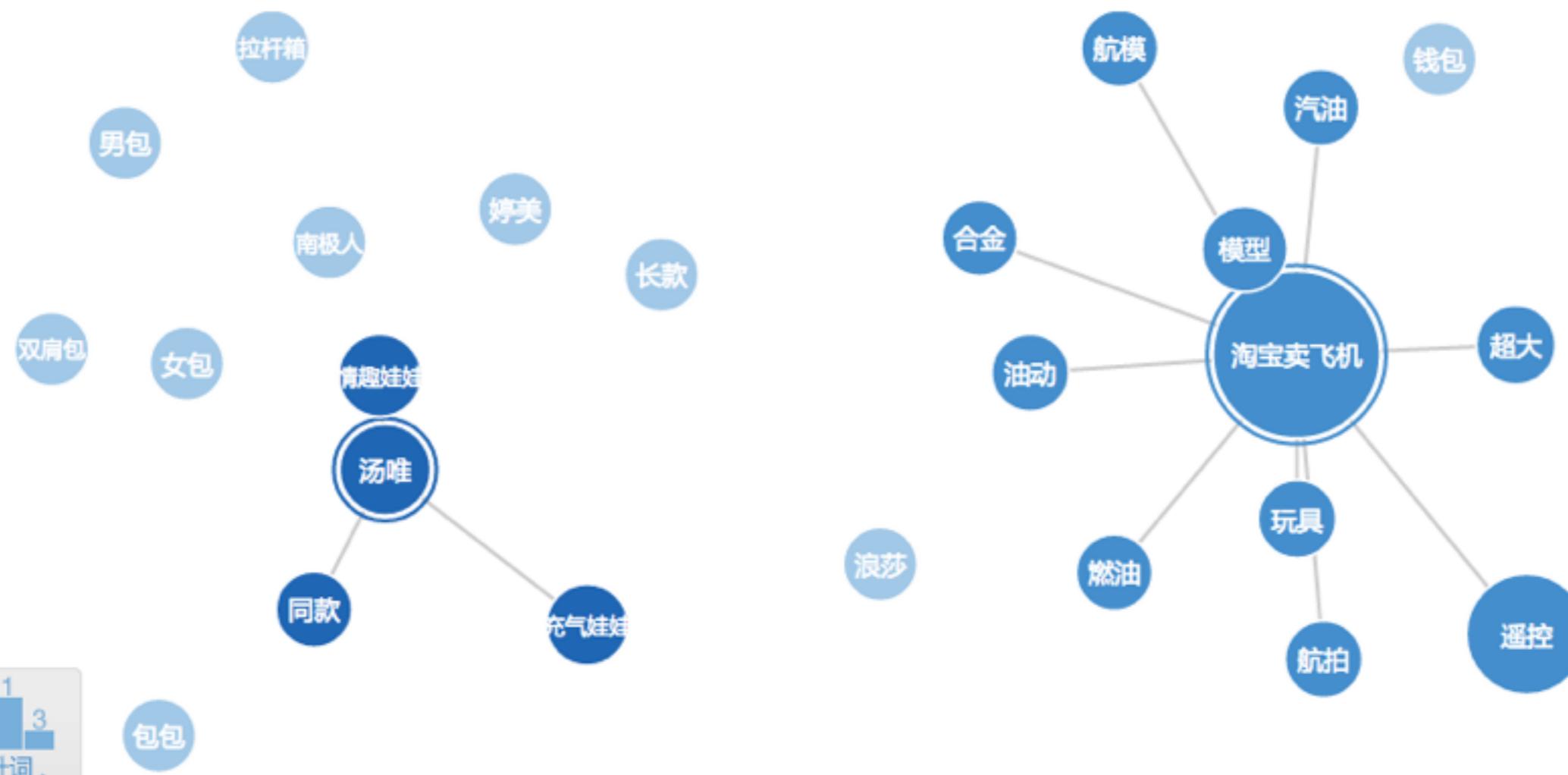
- Node.js 介绍
- 异步编程
- 实战体验
- 企业级 node 开发

node in Ali

淘宝指数

请输入您想查询的关键字

搜索



node in Ali

— 淘宝时光机 – 开启我的回忆之旅 —



立即启动

node in Ali


[首页](#) [专业版](#) [标准版](#) [版本对比](#) [帮助中心](#)


数据魔方 > 千牛移动端

数据魔方

淘宝官方数据产品
分享海量行业数据
致力帮助商家实现数据化运营

371,458 人/截止当前
累计使用人数

专业版

用数据做行业定位、点亮品牌路。[免费体验 >](#)
订购条件：集市一钻（含）以上或者天猫用户
适用人群：中大卖家,品牌商

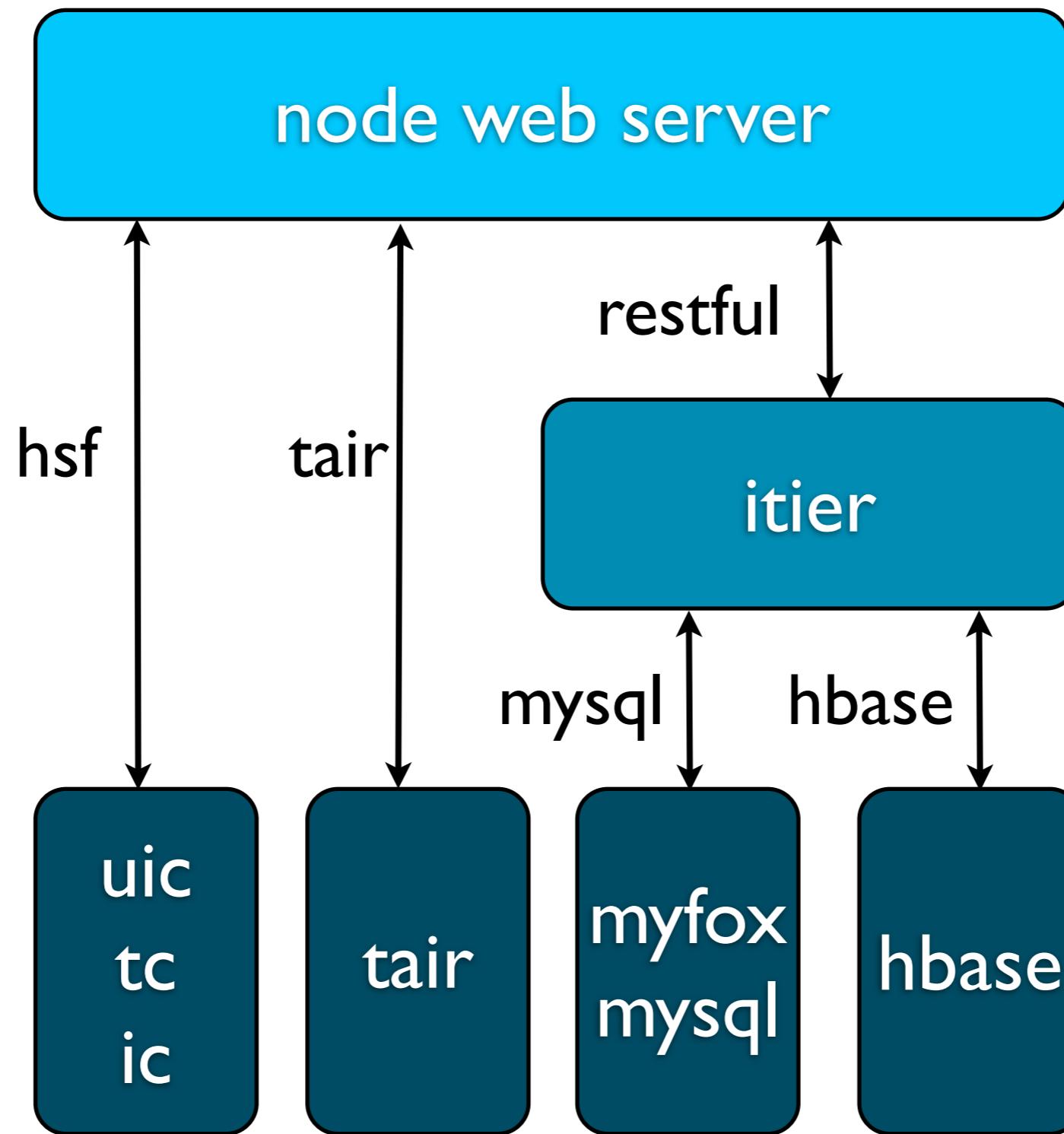
3600 元/年
按年起定

标准版

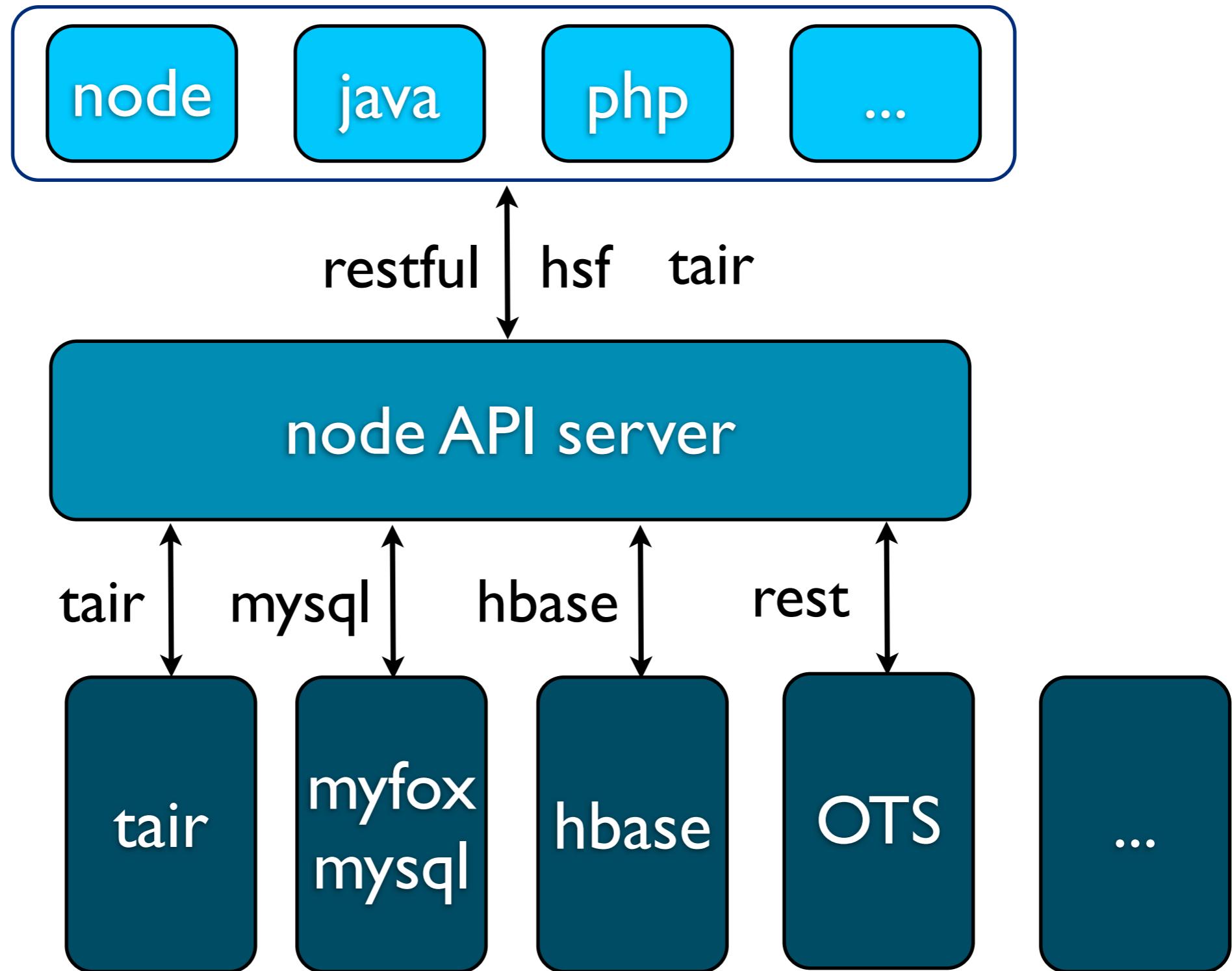
分析竞争对手，探究消费买行为 [免费体验 >](#)
订购条件：集市一钻（含）以上或者天猫用户
适用人群：中小卖家

90 元/季
按季起定

Web Server



API Server



背后的工作

- node-hsf
- node-tair
- tfs
- hbase
- ots
- session proxy
- ...

企业级NPM

- 为什么需要？
 - 私有模块无法发布，代码复用困难
 - 社区模块质量良莠不齐，无法控制
 - 官方源安装速度太慢

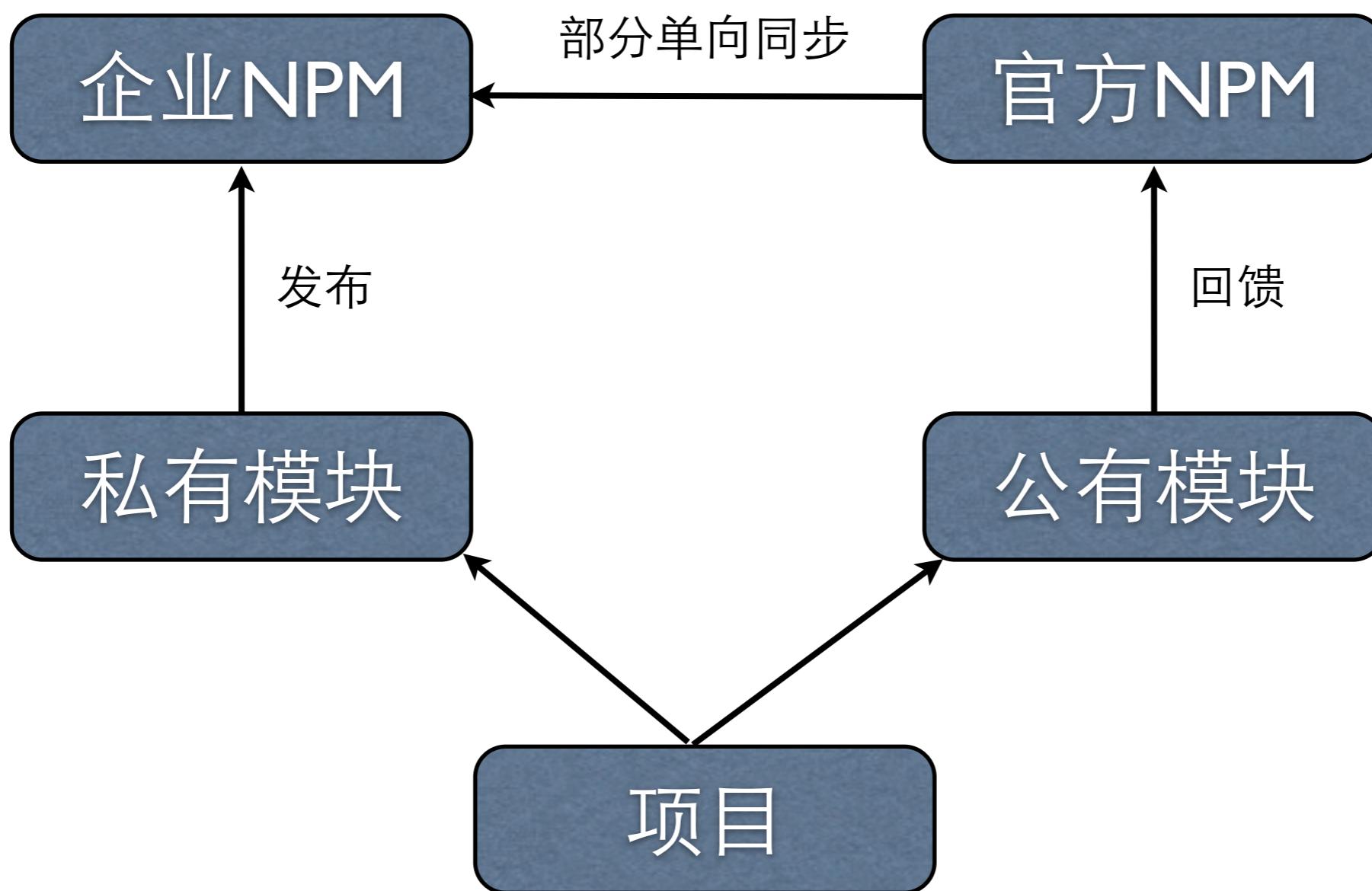
企业级NPM

- 官方提供的全量同步方案
 - 可以解决安装速度过慢的问题
 - 国内网络镜像官方源太慢
 - 并不是所有的库都会用到

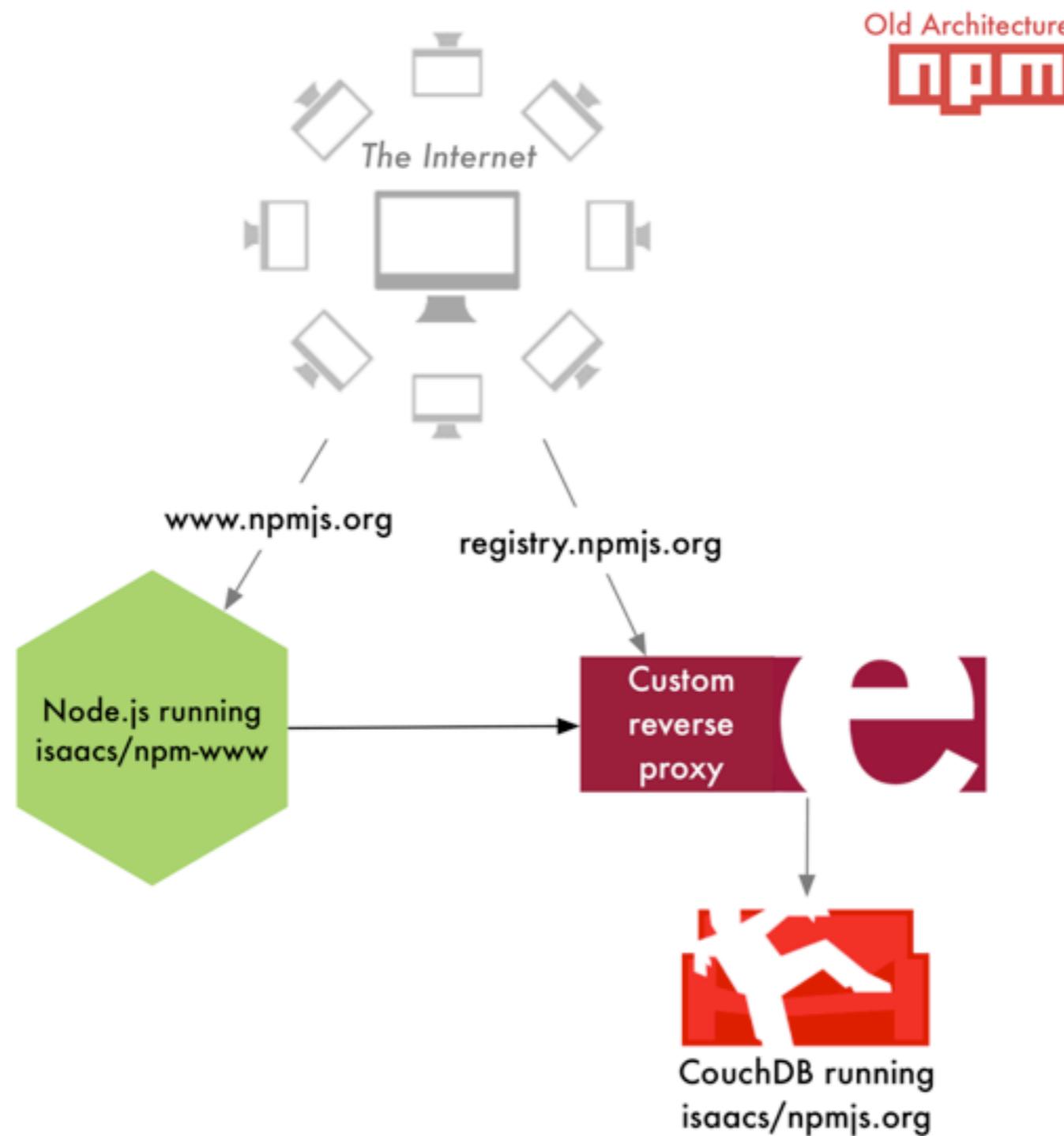
企业级NPM

- 同步需要用到的三方包
- 发布私有的二方包
- 审核机制

企业级NPM

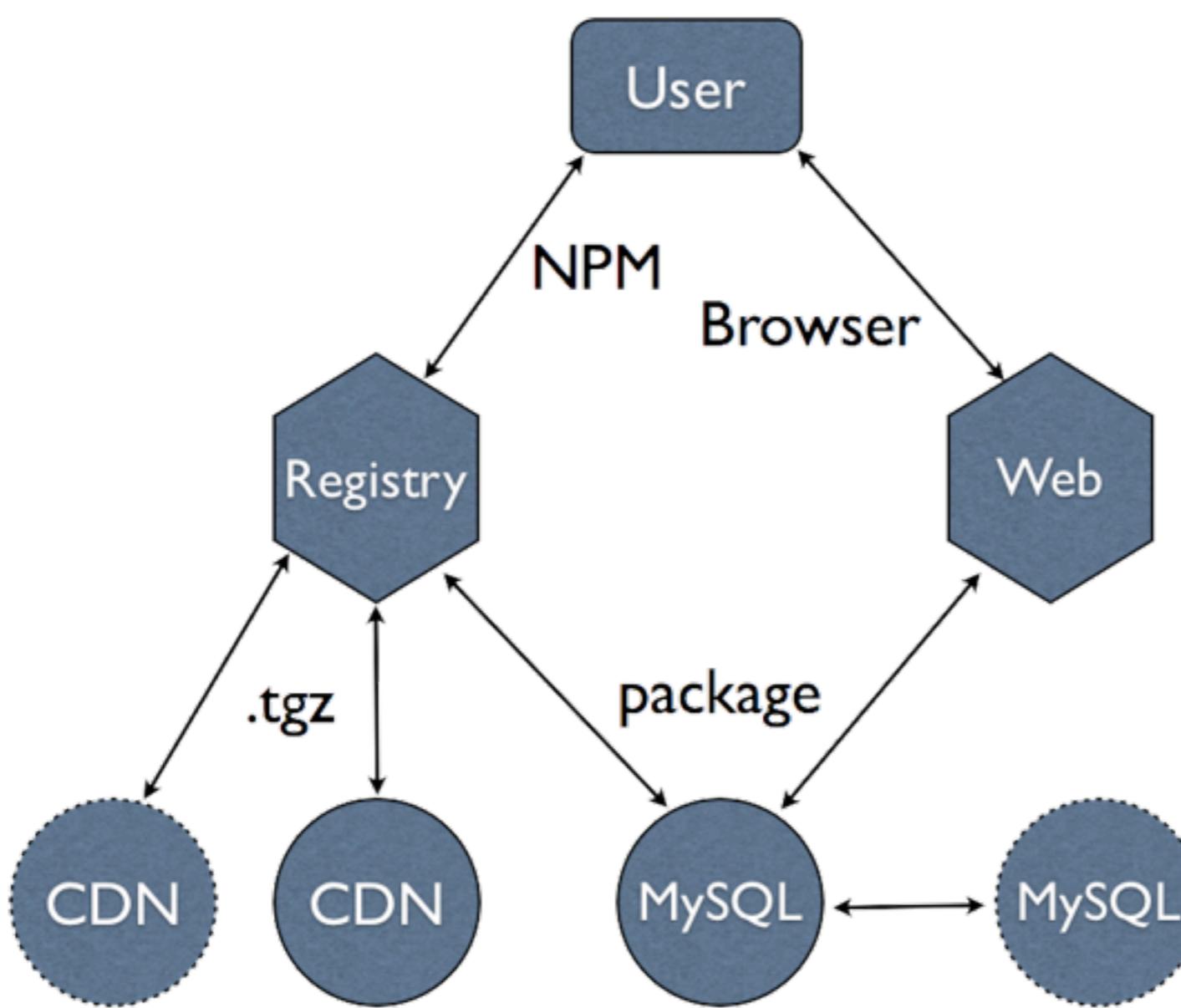


企业级NPM



企业级NPM

- 可维护性更强的企业级 NPM 架构



企业级NPM

- 基于企业NPM的管理系统
 - 命令行工具
 - 三方包直接同步 / 二方包审核
 - 发布权限分发给维护者
- wiki
- 文档中心

企业级NPM

- 阿里 NPM

2,418	total packages	28,457	total package versions	6	total delete packages
2,882	downloads today	3,705	downloads in this week	4,901	downloads in this month
823	downloads in the last day	17,324	downloads in the last week	45,700	downloads in the last month

- 国内 NPM 镜像: <http://cnpmjs.org>

62,282	total packages	318,456	total package versions	9	total delete packages
2,448	downloads today	6,090	downloads in this week	9,072	downloads in this month
3,642	downloads in the last day	62,021	downloads in the last week	154,747	downloads in the last month

Q & A
thanks