

云计算业务支撑技术组 赵雨森

跨产品的组件化解决方案-VUSION



- 赵雨森
- 杭州研究院 云计算技术部
- 2014年入职,毕业于西安交通大学
- 闷骚男,爱摄影,组件控
- rainfore@github, hzzhaoyusen@corp



里想

理想 / Vusion



- 解决方案
- HTML5页面
- Web端
- 幻灯片

- 移动端
- 全栈架构

PC端

- 原型设计
- 组件体系 视觉设计

- 优化目标
 - 协作效率
 - 用户体验
 - 网页性能

理想 / Vusion



- 架构模型
- 工程模板
- 干发规范
- Proto UI + Vusion UI
- Vusion CLI
- Vusion App

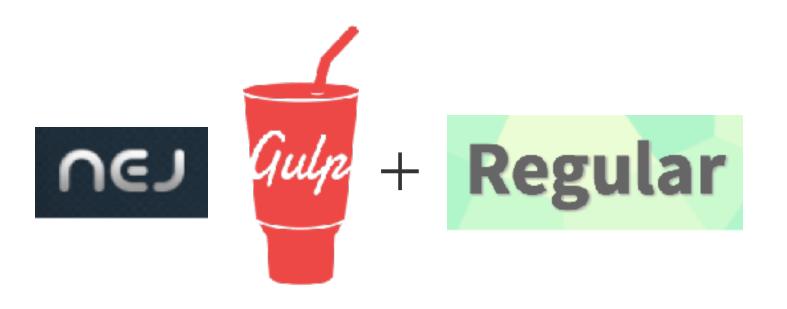


技术选型

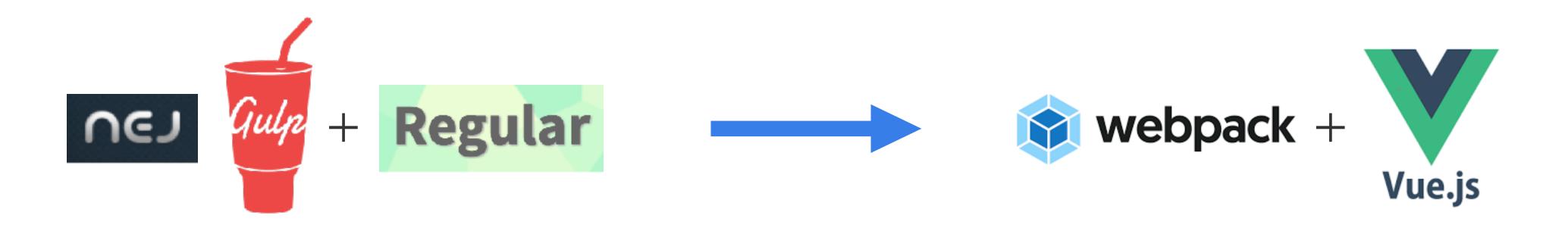
技术选型



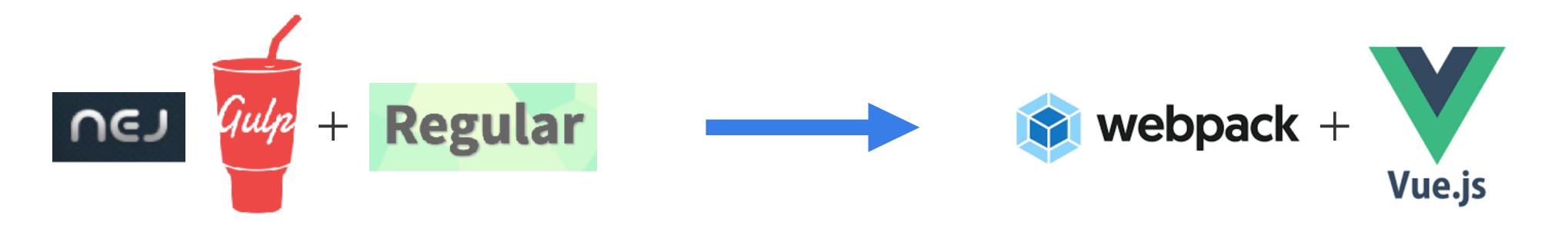






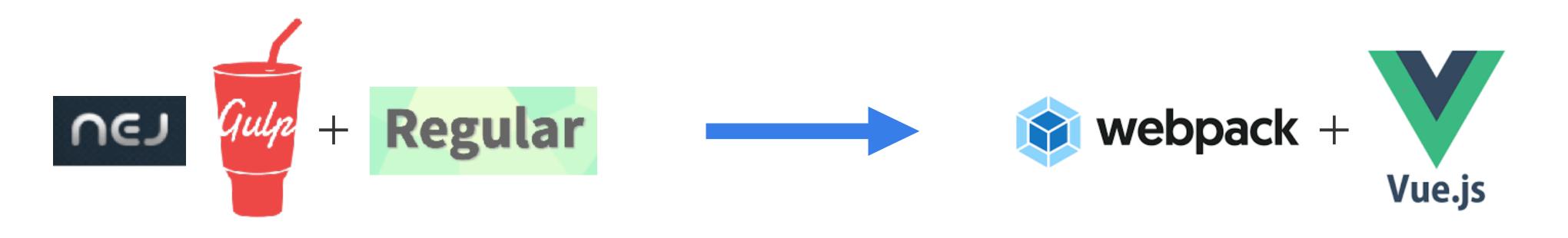






- 框架流行度
- 学习曲线
- 文档
- 社区支持
- 可扩展性
- 许可证





- 框架流行度
- 学习曲线
- 文档
- 社区支持
- 可扩展性
- 许可证

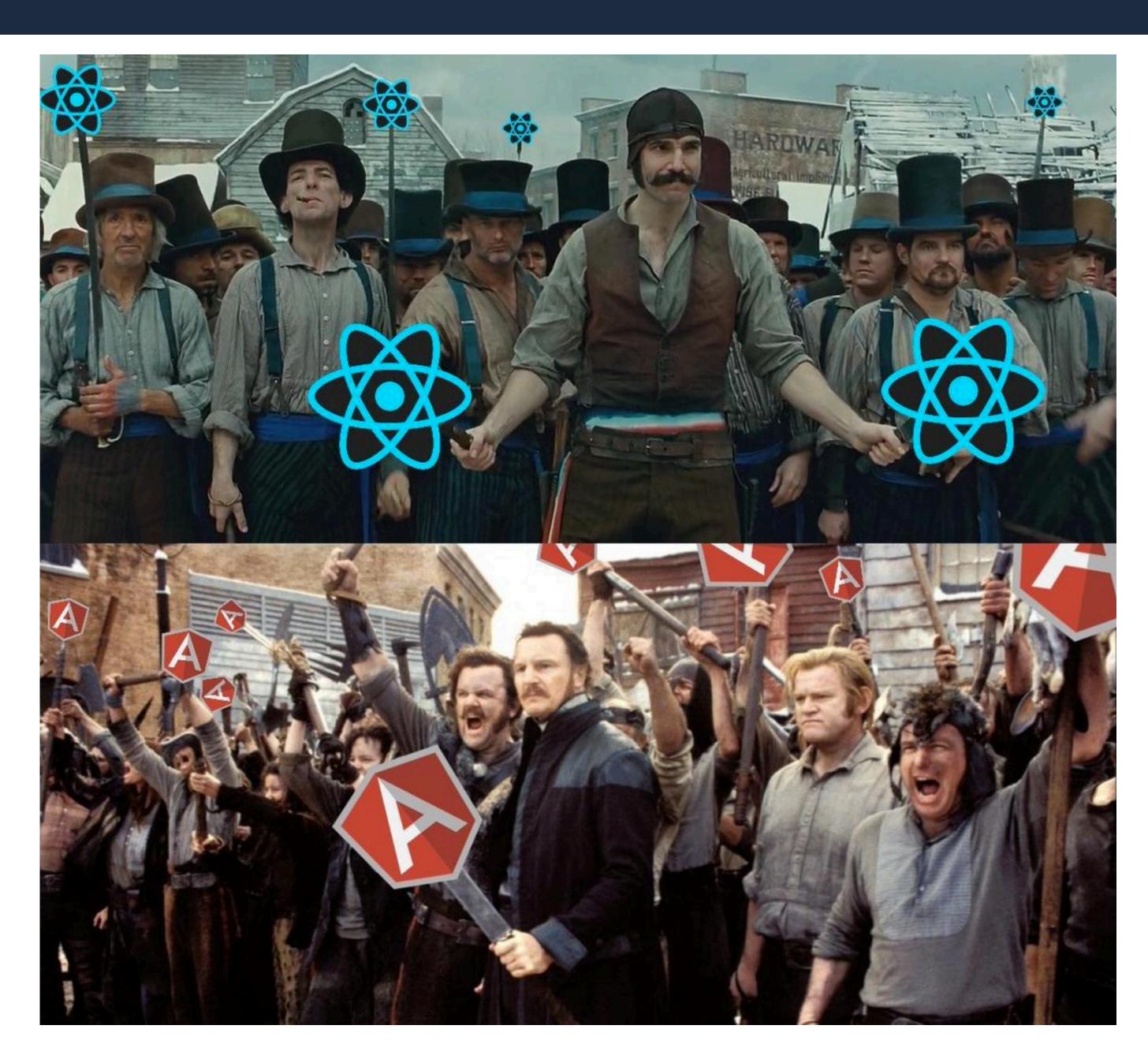


技术选型 / 为什么选Vue





- 依赖管理
- 数据绑定
- 模板、逻辑、样式的分离
- 学习曲线





模块化

模块化 / 分类



- JS的模块化
- CSS的模块化
- 一资源的模块化



完全模块化	部分模块化								
元土俣坎化	JS	CSS	资源						
Webpack	Rollup	SCSS	Icomoon						
System.js	Browserify	LESS	FontCustom						
FIS3	Require.js	Stylus	PS						
	NEJ	MCSS	Sketch						



完全模块化	部分模块化								
元土俣坎化	JS	CSS	资源						
Webpack	Rollup	SCSS	Icomoon						
System.js	Browserify	LESS	FontCustom						
FIS3	Require.js	Stylus							
	NEJ	MCSS	Sketch						



一个北古+カイレ	部分模块化								
完全模块化	JS	CSS	资源						
Webpack	Rollup	SCSS	Icomoon						
System.js	Browserify	LESS	FontCustom						
FIS3	Require.js	Stylus	PS						
	NEJ	MCSS	Sketch						

完备性

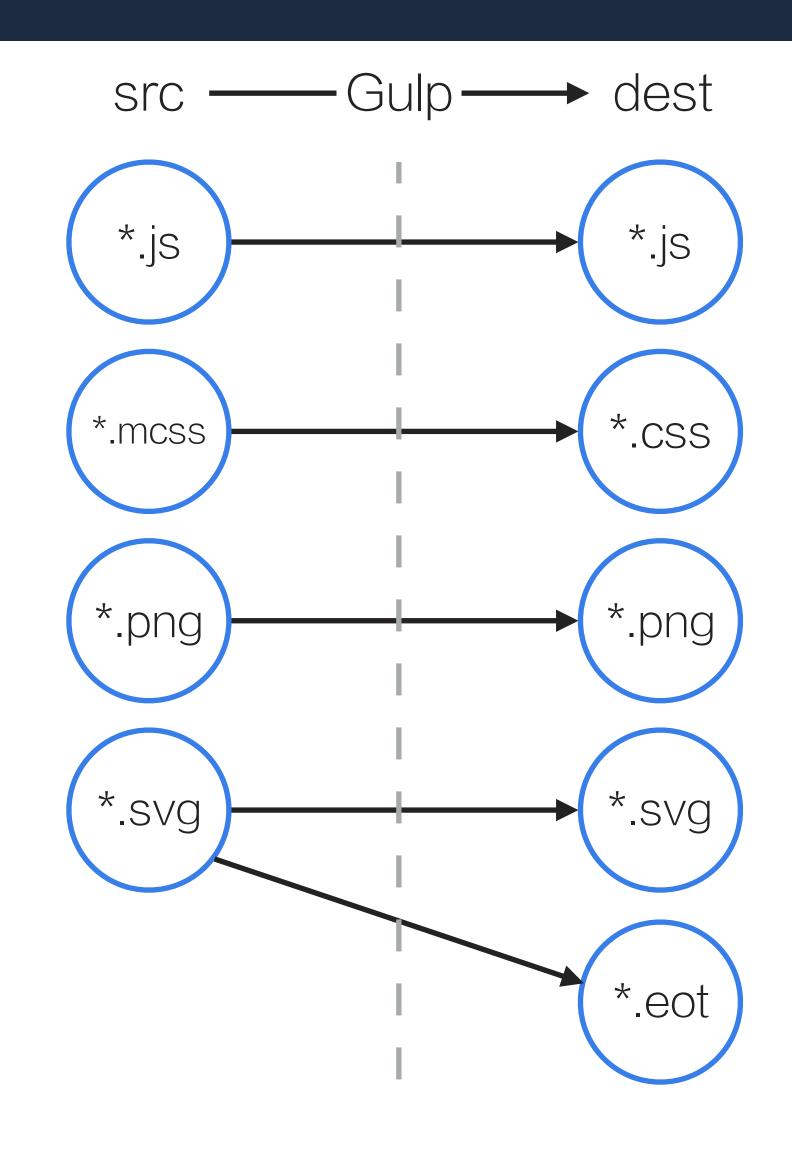
模块化 / 完备性的优点

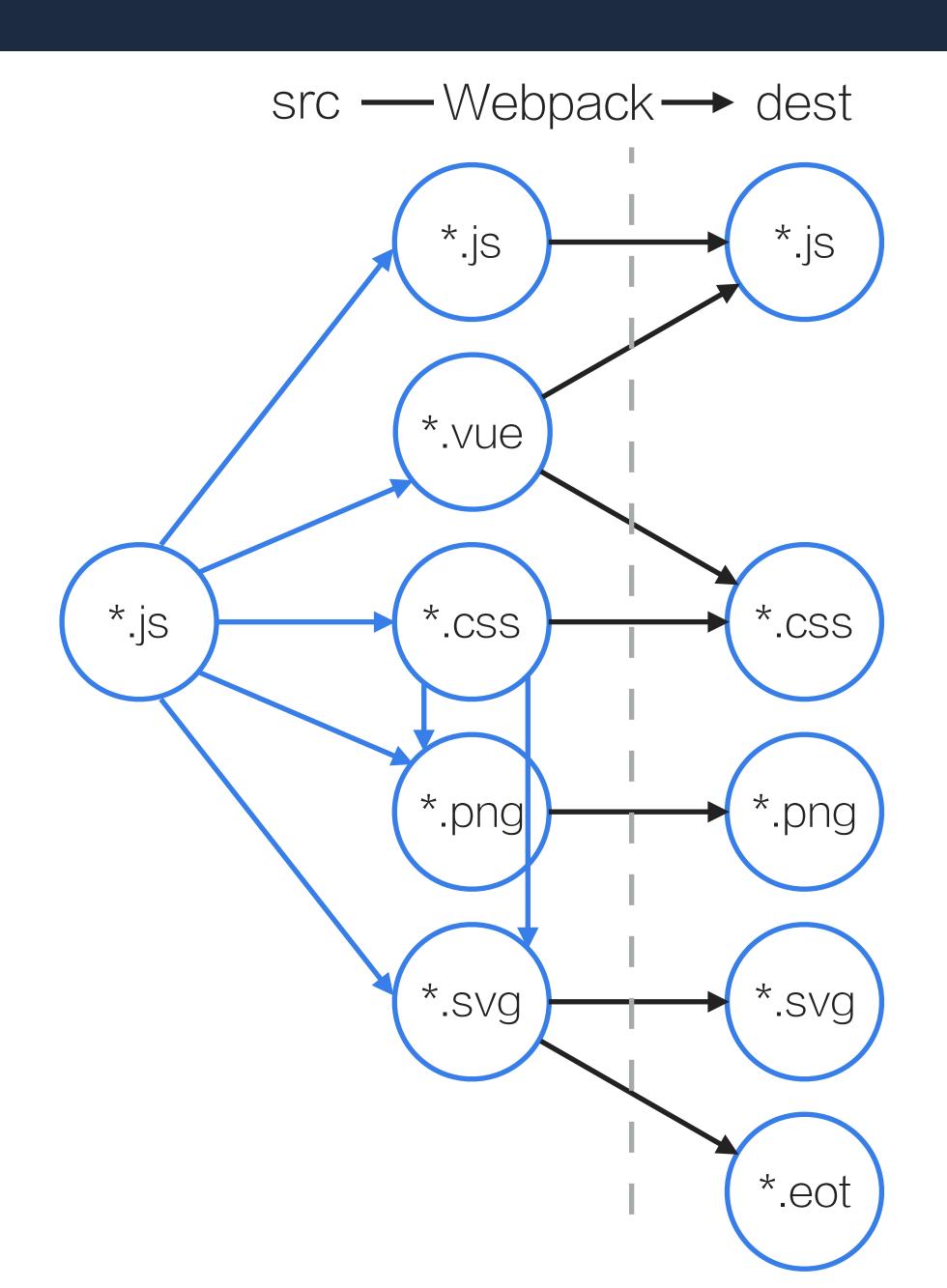


- 依赖关系单一化
- 一资源处理集成化
- 通用库的标准化

模块化 / 完备性的优点 / 依赖关系单一化

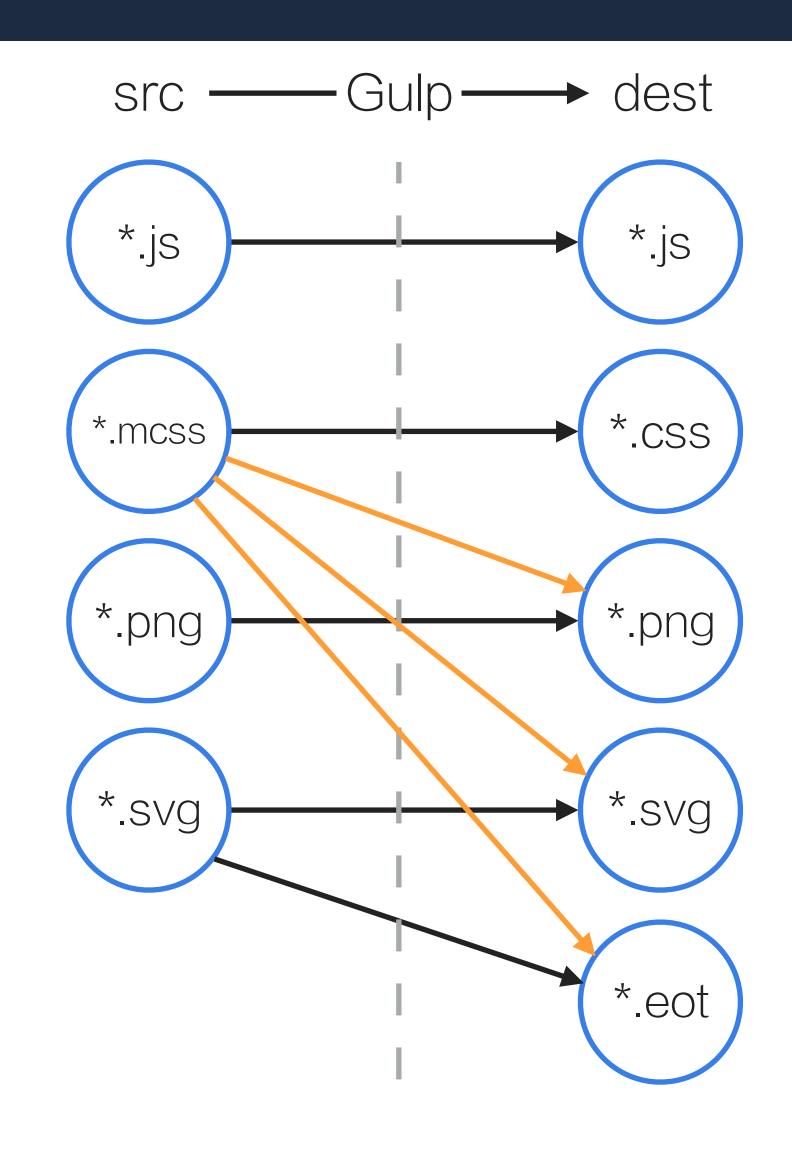


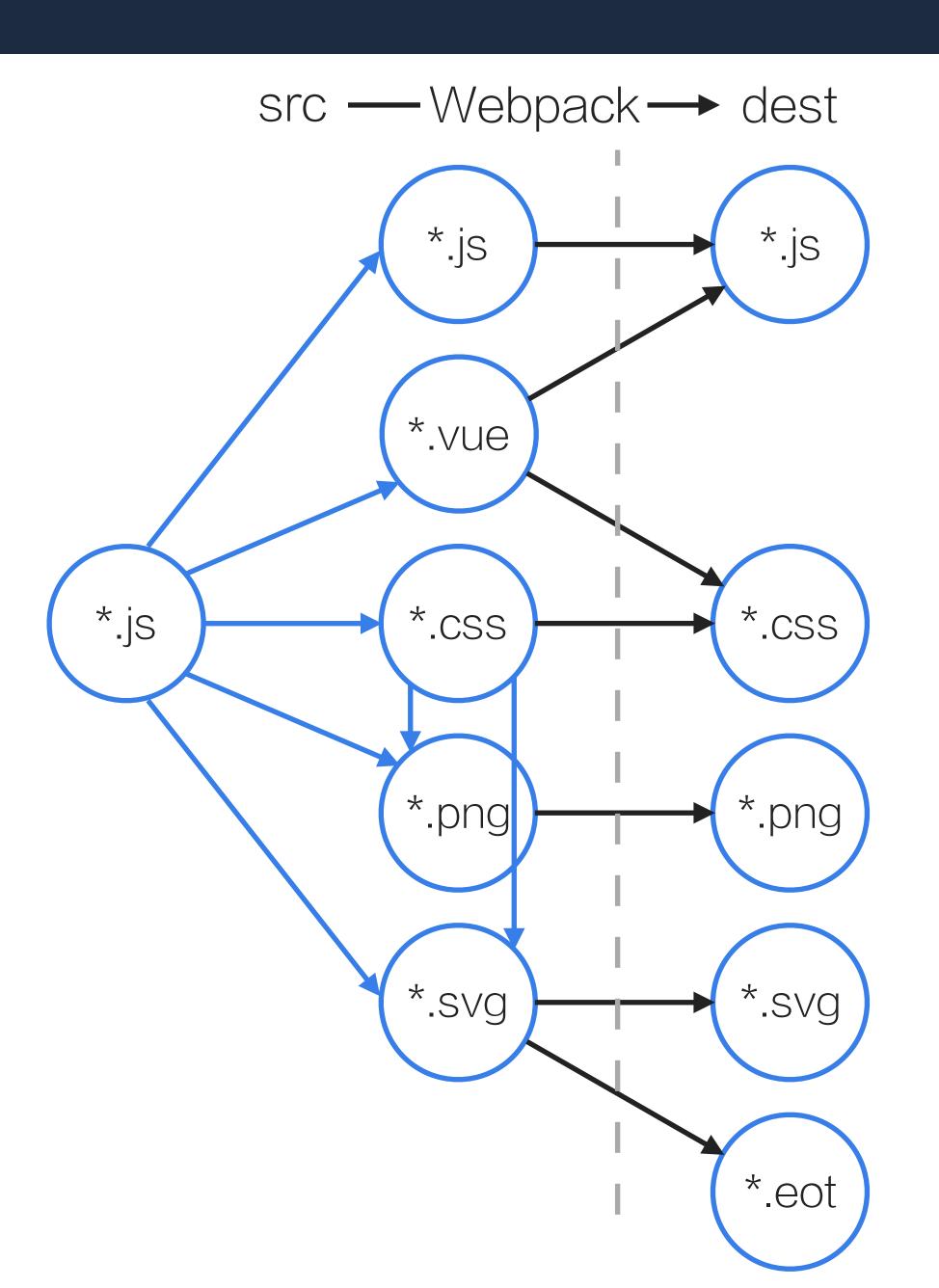




模块化 / 完备性的优点 / 依赖关系单一化

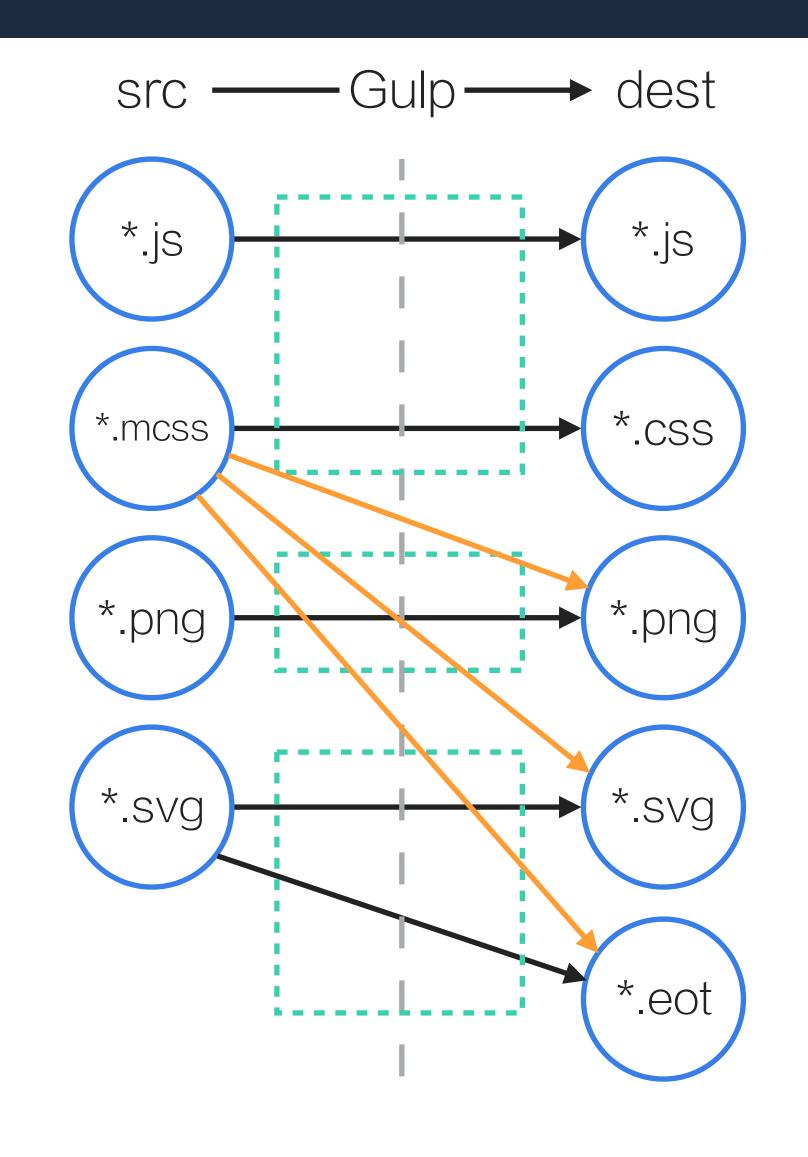


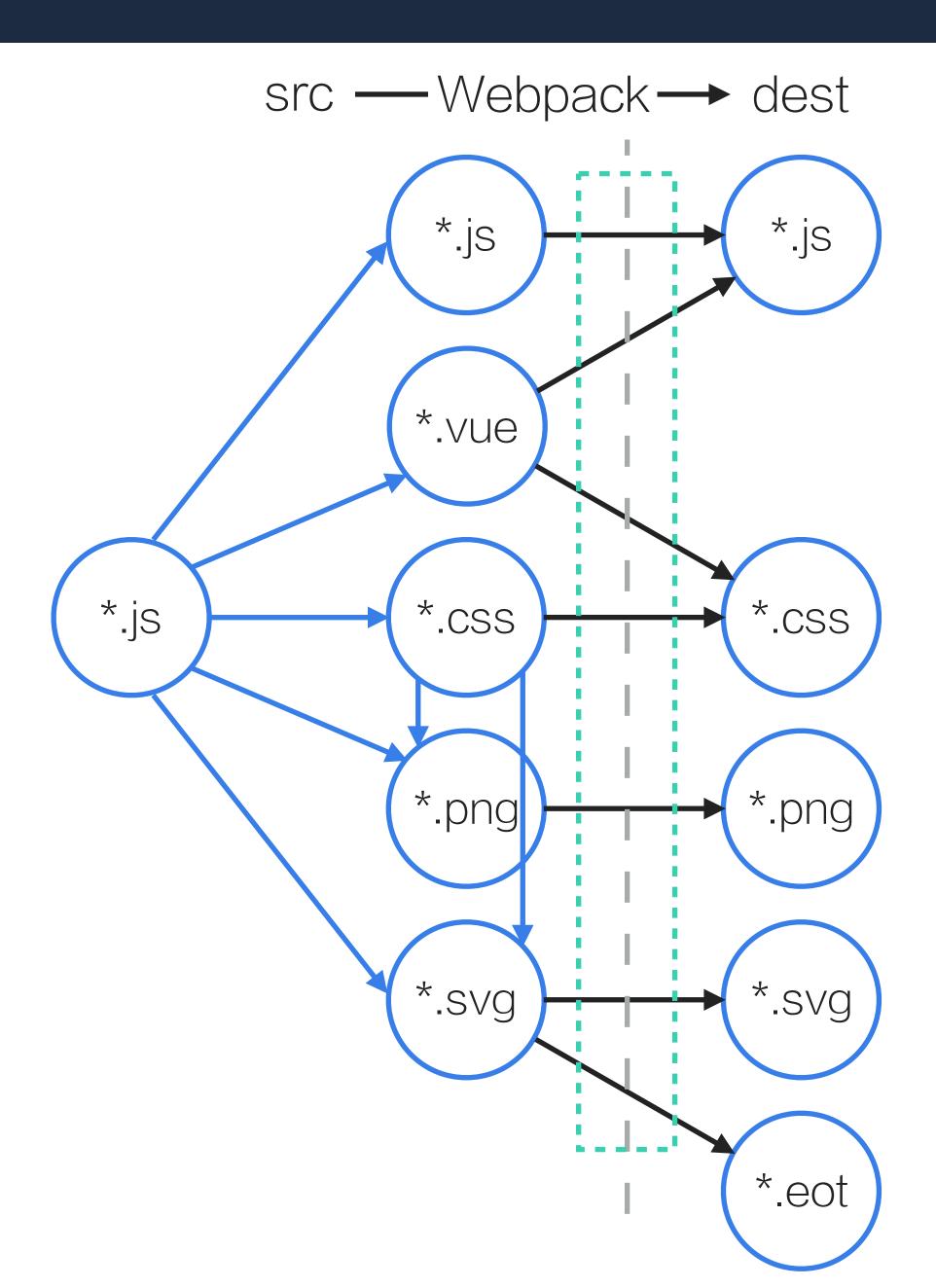




模块化 / 完备性的优点 / 资源处理集成化









Н	В	I	Ū	S	0	*	5	ŧ≡		66	(2)	A		>_	5	C^{\perp}		
欢迎使用wangEditor 富文本编辑器, 请输入内容																		





模块化/完备性的优点/通用库的标准化



```
欢迎使用wangEditor 富文本编辑器,请输入内容...
            <link href="https://cdn.xxx.com/wangEditor/dist/wangEditor.css">
            <script src="https://cdn.xxx.com/wangEditor/dist/wangEditor.js"></script>
            /public/bundle.css
            /public/bundle.js
            /public/vendor/wangEditor/dist/fonts/wangEditor.eot
            /public/vendor/wangEditor/dist/fonts/wangEditor.svg
            /public/vendor/wangEditor/dist/fonts/wangEditor.ttf
            /public/vendor/wangEditor/dist/fonts/wangEditor.woff
```



```
欢迎使用wangEditor 富文本编辑器,请输入内容...
            <link href="https://cdn.xxx.com/wangEditor/dist/wangEditor.css">
            <script src="https://cdn.xxx.com/wangEditor/dist/wangEditor.js"></script>
            /public/bundle.css
            /public/bundle.js
            /public/vendor/wangEditor/dist/fonts/wangEditor.eot
            /public/vendor/wangEditor/dist/fonts/wangEditor.svg
            /public/vendor/wangEditor/dist/fonts/wangEditor.ttf
            /public/vendor/wangEditor/dist/fonts/wangEditor.woff
            import Editor from 'wangEditor';
```

模块化 / JS的模块化



ES6 Modules



```
.m-alarmrule-2{
    .config_cell{
        .unit{}
        add{}
        •op{}
   ፟ m−configTable{
        .config_cell{
.m-instances{
   &-1{}
   label.z-dis{}
    .itmBox{}
    .ck{}
   &-sam{}
.m-alarmform{
    .formItem .ipt .txt, .head .txt{}
    .formItem .ipt .itemDesc{}
```



```
.m-alarmrule-2{
    .config_cell{
         .unit{}
         add{}
         •op{}
    ፟ m−configTable{
         .config_cell{
.m-instances{
    &-1{}
    label.z-mis{}
    .itmBox{
    .ck{}
    &-sam{}
.m-alarmform{
    formItem .ipt txt, .head .txt{}
.formItem .ipt .itemDesc{}
```



CSS Modules

```
<div :class="$style.root">
    <div :class="$style.item" v-for="page in pages">...</div>
</div>
.root {}
.item {}
item:hover {}
.item[selected] {}
.item[disabled] {}
.item[role="prev"] {}
.item[role="next"] {}
.item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```



CSS Modules

```
<div :class="$style.root">
    <div :class="$style.item" v-for="page in pages">...</div>
</div>
.root {}
                                            u-pagination_2x6Z0ugf {}
.item {}
                                            u-pagination_item_1qp9NuYW {}
.item:hover {}
                                            u-pagination_item_1qp9NuYW:hover {}
.item[selected] {}
                                            •u-pagination_item_1qp9NuYW[selected] {}
.item[disabled] {}
                                            •u-pagination_item_1qp9NuYW[disabled] {}
.item[role="prev"] {}
                                            •u-pagination_item_1qp9NuYW[role="prev"] {}
.item[role="next"] {}
                                            •u-pagination_item_1qp9NuYW[role="next"] {}
item[role="blank"] {}
                                            .u-pagination_item_1qp9NuYW[role="blank"] {}
root[disabled] .item {}
                                            •u-pagination_2x6Z0ugf[disabled] •u-pagination_item_1qp9NuYW {}
.root[disabled] .item[selected] {}
                                            .u-pagination_2x6Z0ugf[disabled] .u-pagination_item_1qp9NuYW[selected] {}
```



CSS Modules

```
<div :class="$style.root">
    <div :class="$style.item" v-for="page in pages">...</div>
</div>
.root {}
                                            __2x6Z0ugf {}
.item {}
                                            __1qp9NuYW {}
.item:hover {}
                                            ._1qp9NuYW:hover {}
.item[selected] {}
                                            ._1qp9NuYW[selected] {}
.item[disabled] {}
                                            ._1qp9NuYW[disabled] {}
.item[role="prev"] {}
                                            ._1qp9NuYW[role="prev"] {}
.item[role="next"] {}
                                            ._1qp9NuYW[role="next"] {}
.item[role="blank"] {}
                                            ._1qp9NuYW[role="blank"] {}
                                            ._2x6Z0ugf[disabled] ._1qp9NuYW {}
root[disabled] .item {}
                                            ._2x6Z0ugf[disabled] ._1qp9NuYW[selected] {}
.root[disabled] .item[selected] {}
```



```
.m-alarmrule-2{
    .config_cell{
        .unit{}
        add{}
        •op{}
   ፟ m−configTable{
        .config_cell{
.m-instances{
   &-1{}
   label.z-dis{}
    .itmBox{}
    .ck{}
   &-sam{}
.m-alarmform{
    .formItem .ipt .txt, .head .txt{}
    .formItem .ipt .itemDesc{}
```

行 5042, 列 1 空格: 4 UTF-8 LF



```
.m-alarmrule-2{
   .config_cell{
        .unit{}
        add{}
        •op{}
   ፟ m−configTable{
        .config_cell{
                                     行 5042, 列 1
.m-instances{
   &-1{}
   label.z-dis{}
   .itmBox{}
   .ck{}
   &-sam{}
.m-alarmform{
   .formItem .ipt .txt, .head .txt{}
   .formItem .ipt .itemDesc{}
```



```
.m-alarmrule-2{
   .config_cell{
       .unit{}
       add{}
       •op{}
   ፟ m−configTable{
       .config_cell{
                                 行 5042, 列 1
.m-instances{
   &-1{}
   label.z-dis{}
   .itmBox{}
   .ck{}
   &-sam{}
                             模块没有当成组件对待
.m-alarmform{
   .formItem .ipt .txt, .head .txt{}
   .formItem .ipt .itemDesc{}
```



```
.m-alarmrule-2{
   .config_cell{
      .unit{}
      add{}
      •op{}

    m−configTable{
      .config_cell{
                               行 5042, 列 1
.m-instances{
   &-1{}
   label.z-dis{}
   .itmBox{}
   .ck{}
   &-sam{}
                           模块没有当成组件对待(无处安放的青春)
.m-alarmform{
   .formItem .ipt .txt, .head .txt{}
   .formItem .ipt .itemDesc{}
```



```
.root {}
.item {}
.item:hover {}
.item[selected] {}
.item[disabled] {}
.item[role="prev"] {}
.item[role="next"] {}
.item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```



REA规范 (root, element, attribute)

```
.root {}
.item {}
.item:hover {}
.item[selected] {}
.item[disabled] {}
.item[role="prev"] {}
.item[role="next"] {}
item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```

• 根节点用root



```
.root {}
.item {}
.item:hover {}
.item[selected] {}
item[disabled] {}
.item[role="prev"] {}
.item[role="next"] {}
item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```

- 根节点用root
- 子节点用一个简单的单词表示



```
.root {}
.item {}
.item:hover {}
.item[selected] {}
item[disabled] {}
.item[role="prev"] {}
item[role="next"] {}
item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```

- 根节点用root
- 子节点用一个简单的单词表示
- 样式扩展用属性来写,命名与组件props保持一致



```
.root {}
.item {}
.item:hover {}
.item[selected] {}
item[disabled] {}
.item[role="prev"] {}
item[role="next"] {}
item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```

- 根节点用root
- 子节点用一个简单的单词表示
- 样式扩展用属性来写,命名与组件props保持一致



REA规范 (root, element, attribute)

```
.root {}
.item {}
.item:hover {}
.item[selected] {}
item[disabled] {}
.item[role="prev"] {}
item[role="next"] {}
item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```

- 根节点用root
- 子节点用一个简单的单词表示
- 样式扩展用属性来写,命名与组件props保持一致

• 禁止单词用缩写



```
.root {}
.item {}
.item:hover {}
.item[selected] {}
item[disabled] {}
.item[role="prev"] {}
item[role="next"] {}
item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```

- 根节点用root
- 子节点用一个简单的单词表示
- 样式扩展用属性来写,命名与组件props保持一致

- 禁止单词用缩写
- 如果有多个单词,统一用连字符



```
.root {}
.item {}
.item:hover {}
.item[selected] {}
item[disabled] {}
.item[role="prev"] {}
item[role="next"] {}
item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```

- 根节点用root
- 子节点用一个简单的单词表示
- 样式扩展用属性来写,命名与组件props保持一致

- 禁止单词用缩写
- 如果有多个单词,统一用连字符
- 禁止有SCSS的样式嵌套



```
.root {}
.item {}
.item:hover {}
.item[selected] {}
item[disabled] {}
.item[role="prev"] {}
item[role="next"] {}
item[role="blank"] {}
.root[disabled] .item {}
root[disabled] .item[selected] {}
```

- 根节点用root
- 子节点用一个简单的单词表示
- 样式扩展用属性来写,命名与组件props保持一致

- 禁止单词用缩写
- 如果有多个单词,统一用连字符
- 禁止有SCSS的样式嵌套
- 无需添加浏览器前缀(autoprefixer)



```
.root {}
.item {}
.item:hover {}
.item[selected] {}
.item[disabled] {}
.item[role="prev"] {}
.item[role="next"] {}
.item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```



REA规范 (root, element, attribute)

```
.root {}
.item {}
.item:hover {}
.item[selected] {}
item[disabled] {}
.item[role="prev"] {}
.item[role="next"] {}
item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```

Boolean属性: selected, disabled, active, checked



```
.root {}
.item {}
.item:hover {}
.item[selected] {}
item[disabled] {}
.item[role="prev"] {}
item[role="next"] {}
item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```

- Boolean属性: selected, disabled, active, checked
- String属性: color="primary", color="success", color="error"



```
.root {}
.item {}
.item:hover {}
.item[selected] {}
item[disabled] {}
.item[role="prev"] {}
item[role="next"] {}
item[role="blank"] {}
.root[disabled] .item {}
.root[disabled] .item[selected] {}
```

- Boolean属性: selected, disabled, active, checked
- String属性: color="primary", color="success", color="error"
- Size属性: size="normal normal", size="normal large", size="mini small"



- 优先使用svg
- 单色转icon-font、多色转svg-sprite

```
.root:before {
    icon-font: url(./icons/refresh.svg);
    color: white;
}

root {
    svg-sprite: url(./icons/refresh.svg);
    width: 200px;
    height: 120px;
```

```
.root:before {
    font-family: vusion-icon-font;
    font-style: normal;
    font-weight: 400;
    font-variant: normal;
    text-decoration: inherit;
    text-rendering: optimizeLegibility;
    text-transform: none;
    -moz-osx-font-smoothing: grayscale;
    -webkit-font-smoothing: antialiased;
    font-smoothing: antialiased;
    content: "\F101";
    color: white;
.root:before {
    background: url(/public/sprite.svg#refresh);
   width: 200px;
   height: 120px;
```



为什么不在HTML或JS中引入图片?



为什么不在HTML或JS中引入图片?

- 图片还需设置CSS属性
- 图片资源的引入本质是一种样式修改,尽可能少使用和<svg>
- 可以重写

模块化 / 异步加载



```
import(/* webpackMode: "lazy-once", webpackChunkName: "all-i18n-data" */ './moduleA.js')
```

```
component: () => import('../src/u-status-icon.vue')
```



里组件化



每个包含模板(HTML)+样式(CSS)+逻辑(JS)功能完备的结构单元,我们称之为组件。



每个功能完备的结构单元称之为组件。



每个功能完备的结构单元称之为组件。

完备性

组件化/完备性



vue-loader / vue-multifile-loader

```
/module/rds/create/index.html
/module/rds/create/index.js
/mcss/?
/mcss/?
```

组件化/完备性



vue-loader / vue-multifile-loader

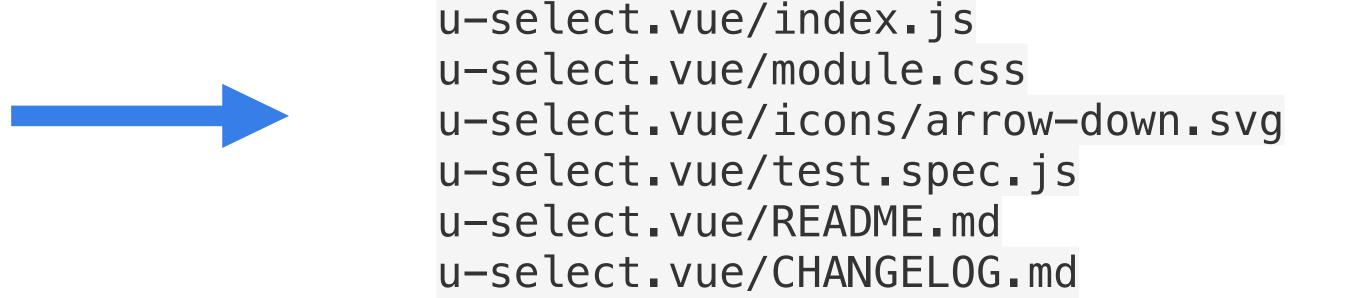
```
/module/rds/create/index.html
/module/rds/create/index.js
/mcss/?
```





vue-loader / vue-multifile-loader

```
/module/rds/create/index.html
/module/rds/create/index.js
/mcss/?
```



u-select.vue/index.html

u-select.vue/package.json



vue-loader / vue-multifile-loader

```
/module/rds/create/index.html
/module/rds/create/index.js
/mcss/?
u-select.vue/index.html
```

/components/u-select.vue
/assets/icons/arrow-down.svg
/tests/unit/components/u-select.js
/docs/u-select.md

u-select.vue/index.ntml
u-select.vue/index.ntml
u-select.vue/index.ntml
u-select.vue/module.css
u-select.vue/module.css
u-select.vue/icons/arrow-down.svg
u-select.vue/test.spec.js
u-select.vue/README.md
u-select.vue/CHANGELOG.md
u-select.vue/package.json



vusion-transform

```
u-select.vue/index.html
u-select.vue/index.js
u-select.vue/module.css
u-select.vue/test.spec.js
u-select.vue/README.md
u-select.vue/CHANGELOG.md
```



```
u-select.vue
```

<template></template>

<script></script>

<style module></style>

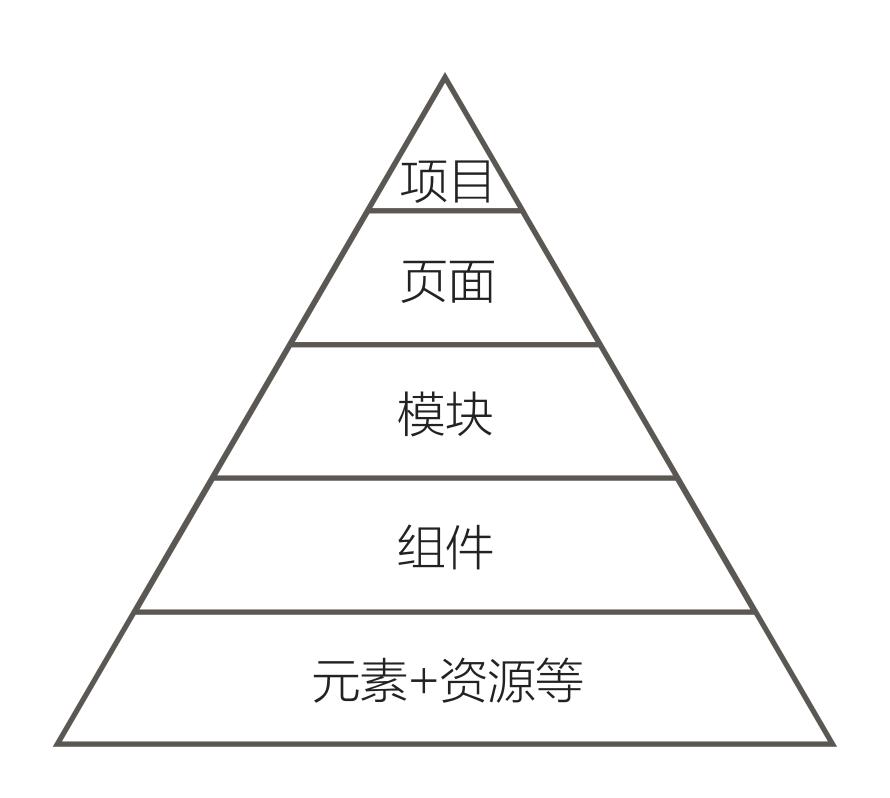
<test></test>

<docs></docs>

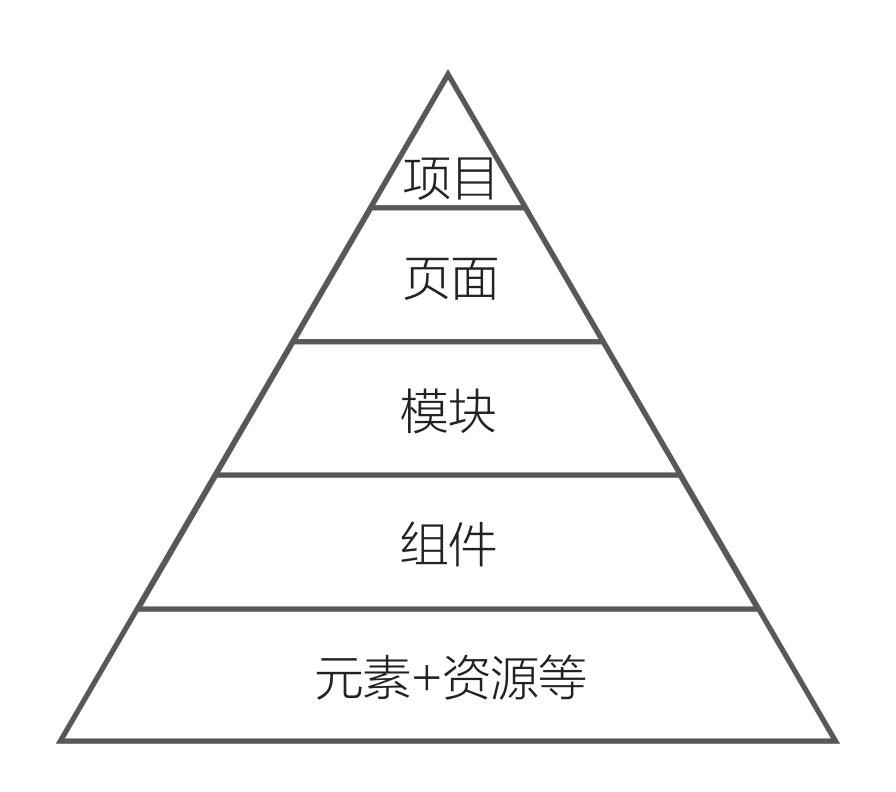
<changelog></changelog>

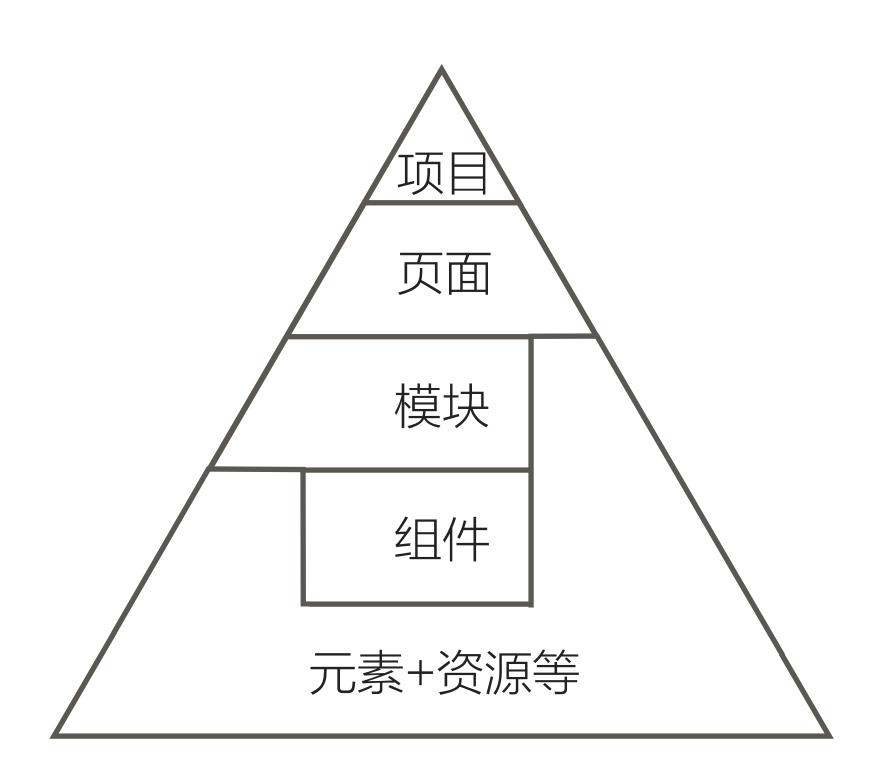
组件库/一致性



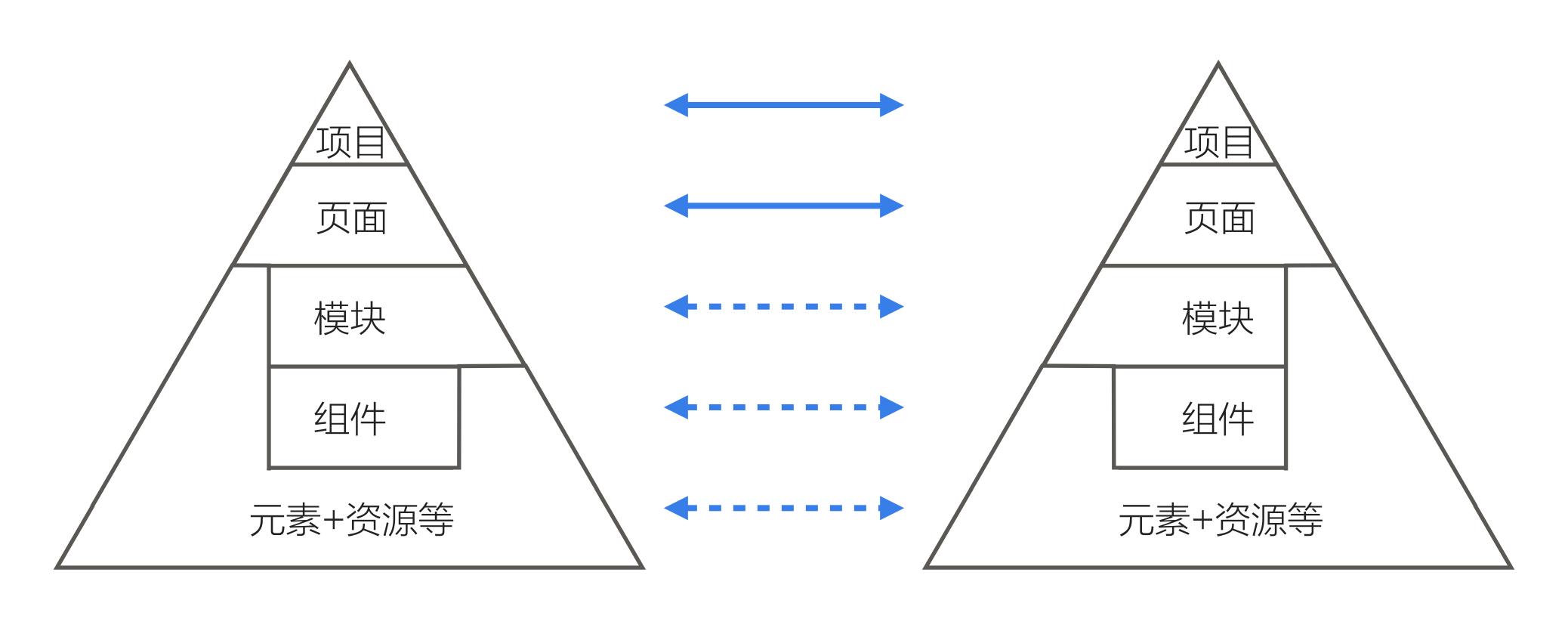








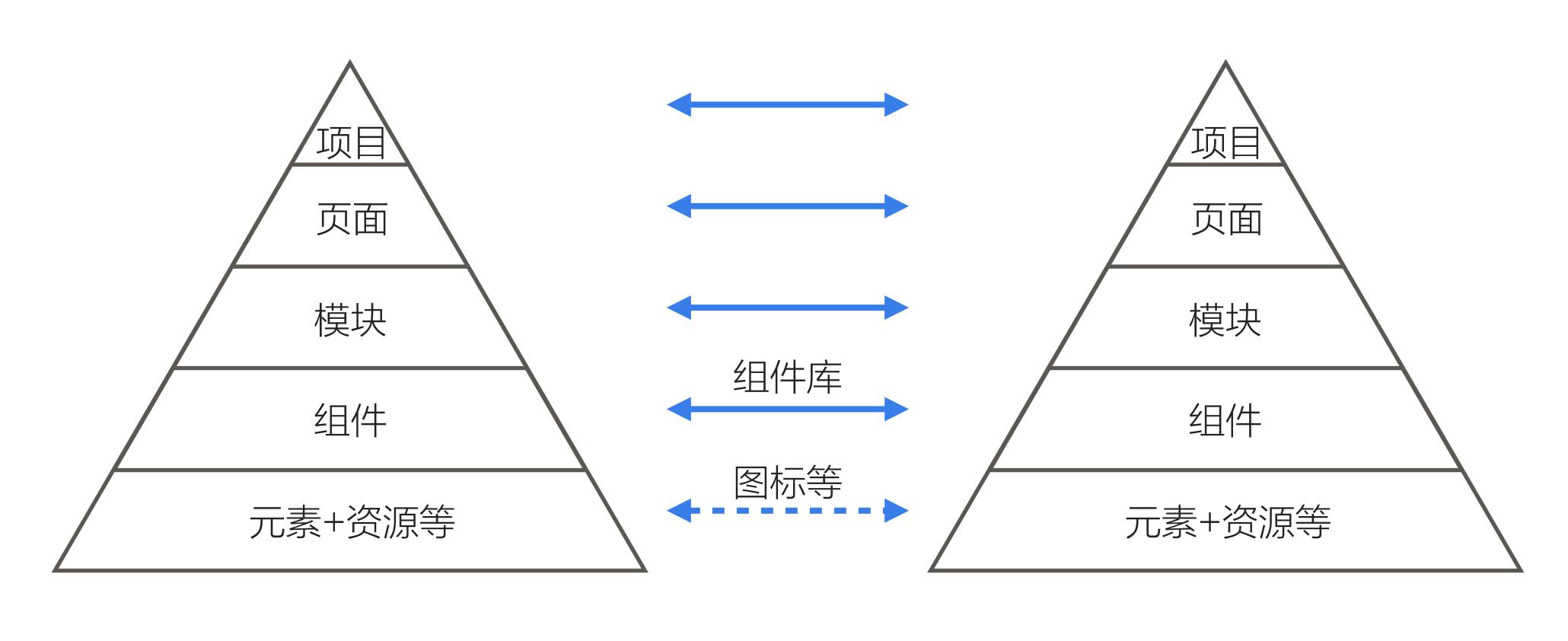




设计

组件库/一致性

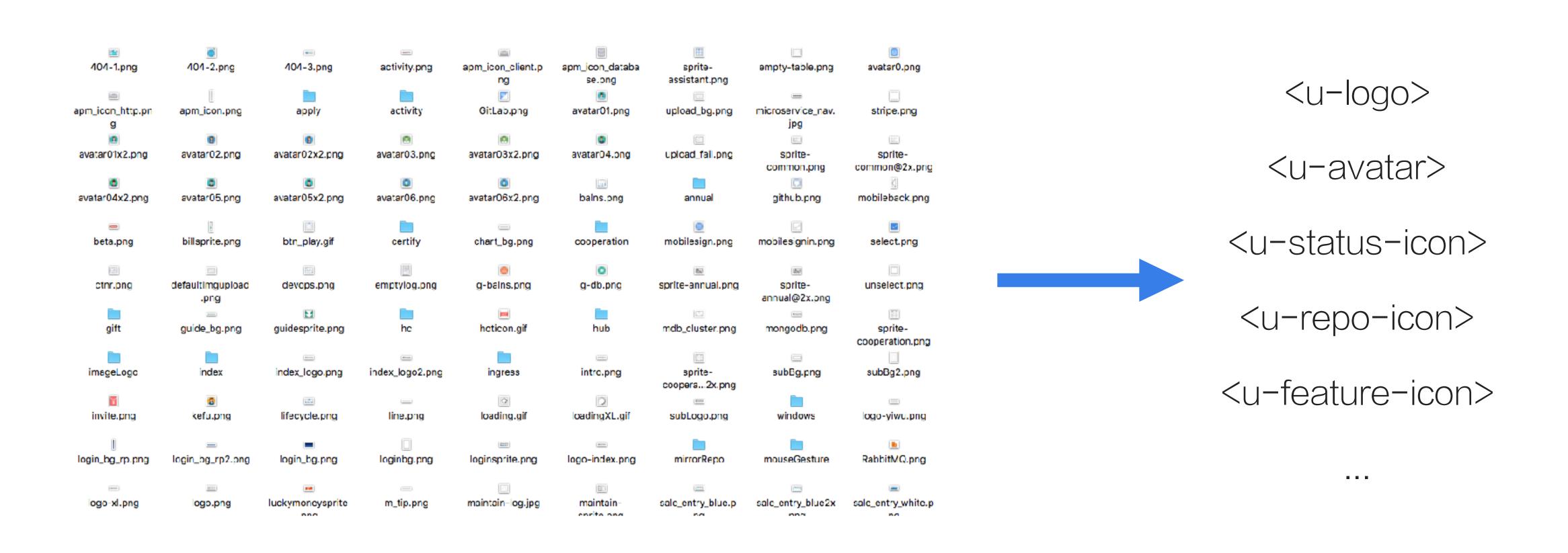




设计

组件化/化整为零







- 通用组件 (common)
- 特定组件 (specific)

```
Vue.component('u-network', Network);
import LogView from 's-log-view.vue';
export default {
    components: {
        's-log-view': LogView,
    },
}
```



■ 同名: 扩展

早名: 继承

cloud-ui.vusion/src/u-button.vue

./src/common/u-button.vue

cloud-ui.vusion/src/u-list-view.vue



cloud-ui.vusion/src/u-pills.vue

组件化/组件的扩展与继承



扩展或继承策略

	派生组件无	派生组件有
JS	报错	Mixin
HTML	继承	覆盖
CSS	继承	补充 / 合并 / 覆盖
README	继承	覆盖



组件库

组件库/扩展性



为什么我们不直接用ElementUI、iView等等组件库?





- 视觉交互不一致



- 视觉交互不一致
- 扩展性差



- 视觉交互不一致
- 扩展性差
- 实现得一般



蜂巢控制台

UI Library

域名系统

UI Library

专属云管理平台

UI Library

运营平台

UI Library



蜂巢控制台

UI Library

域名系统

UI Library

专属云管理平台

UI Library

运营平台

UI Library

Cloud UI



蜂巢控制台

Default

域名系统

Default

专属云管理平台

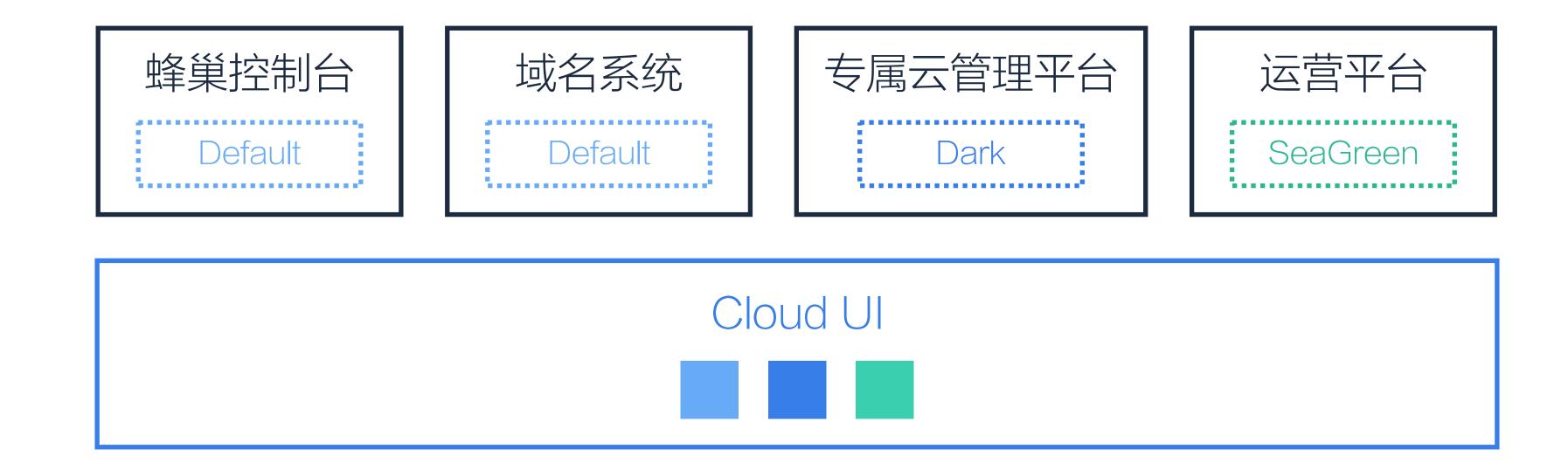
Dark

运营平台

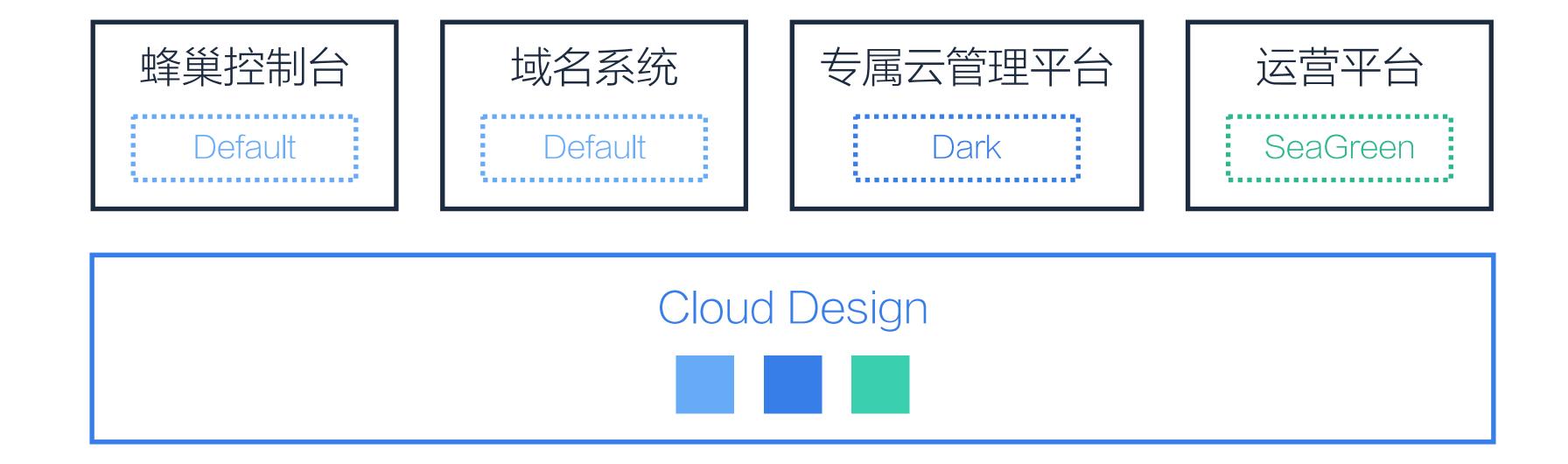
SeaGreen

Cloud UI











蜂巢控制台

域名系统

Cloud Design









蜂巢控制台

域名系统

网易云其它

Cloud Design



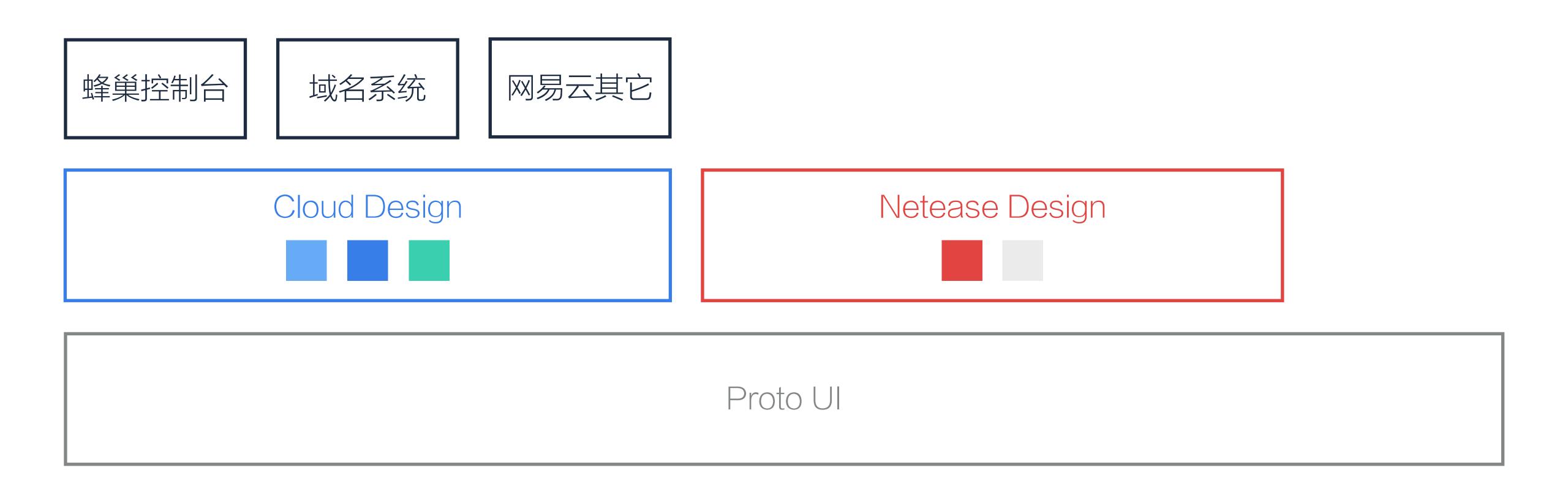




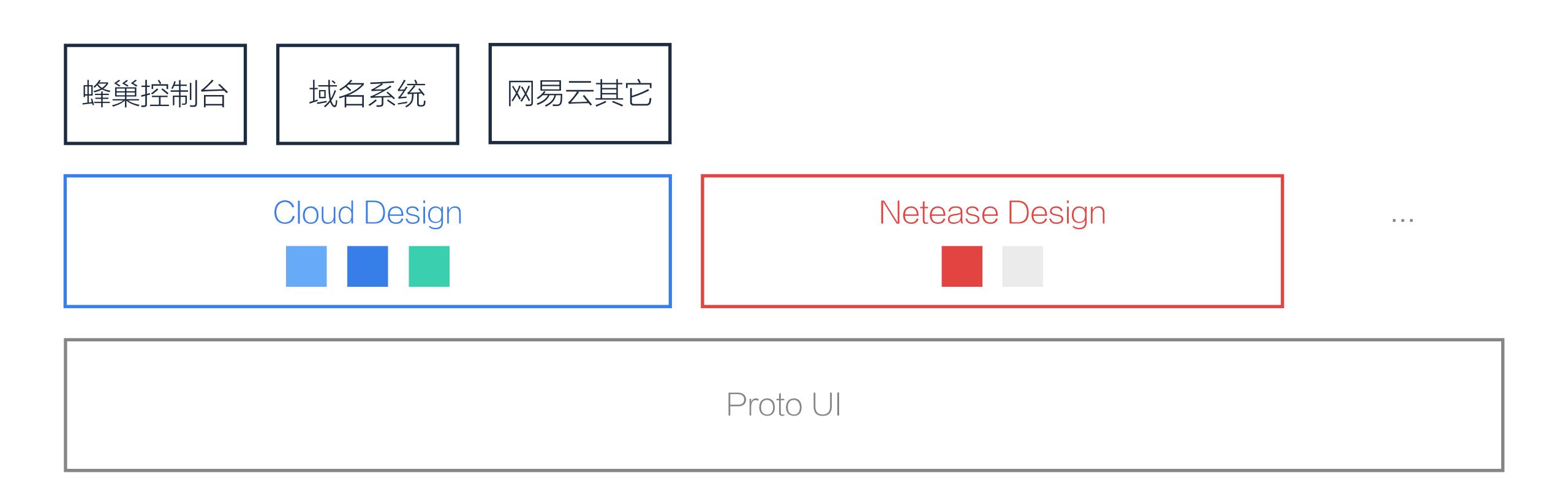


蜂巢控制台 域名系统 网易云其它 Cloud Design Proto UI

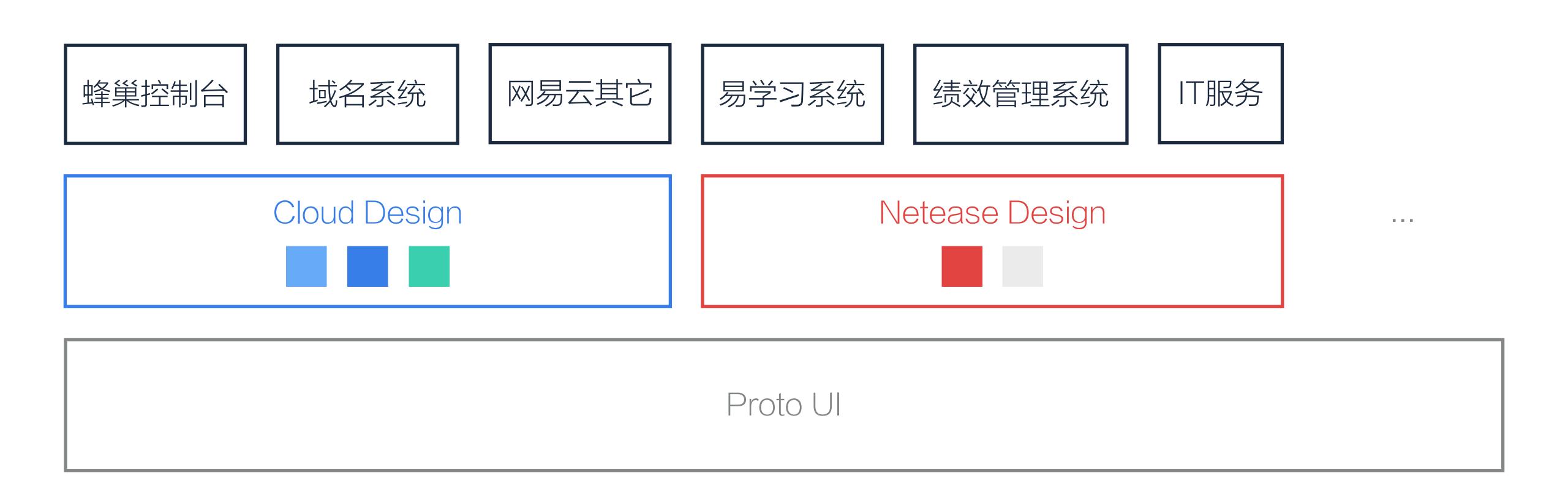












组件库/协作效率



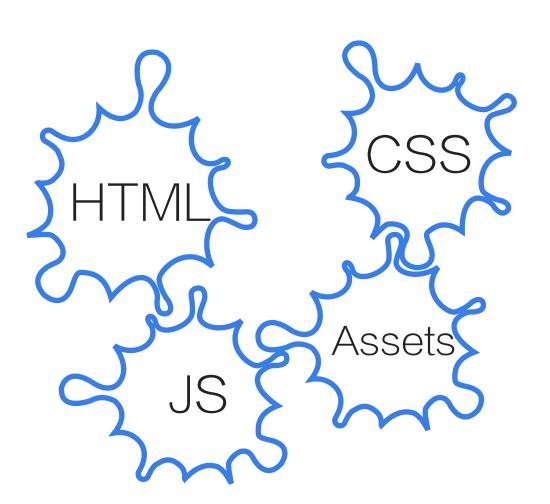
需求的灵活性 vs 版本的稳定性 修bug的急迫性 vs 发布的滞后性



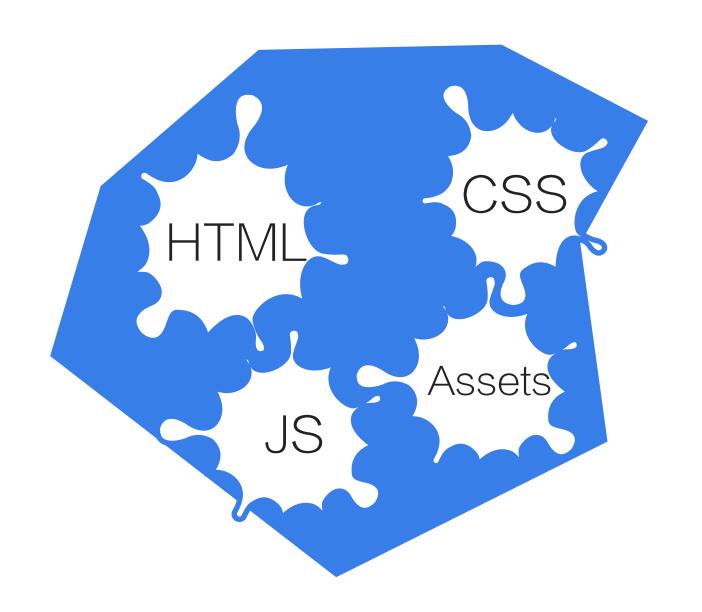
需求的灵活性 vs 版本的稳定性 修bug的急迫性 vs 发布的滞后性

快速响应 + 快速发布

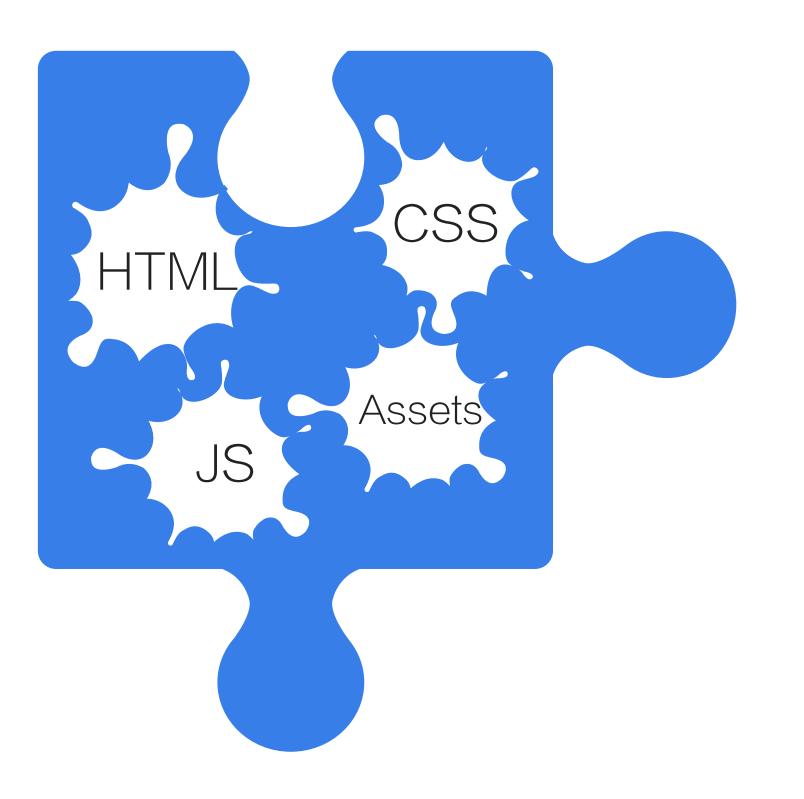




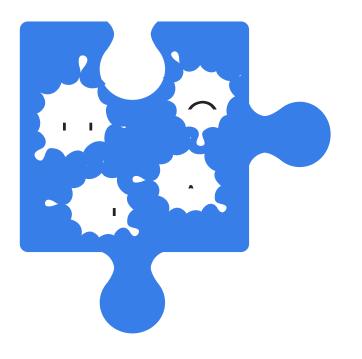




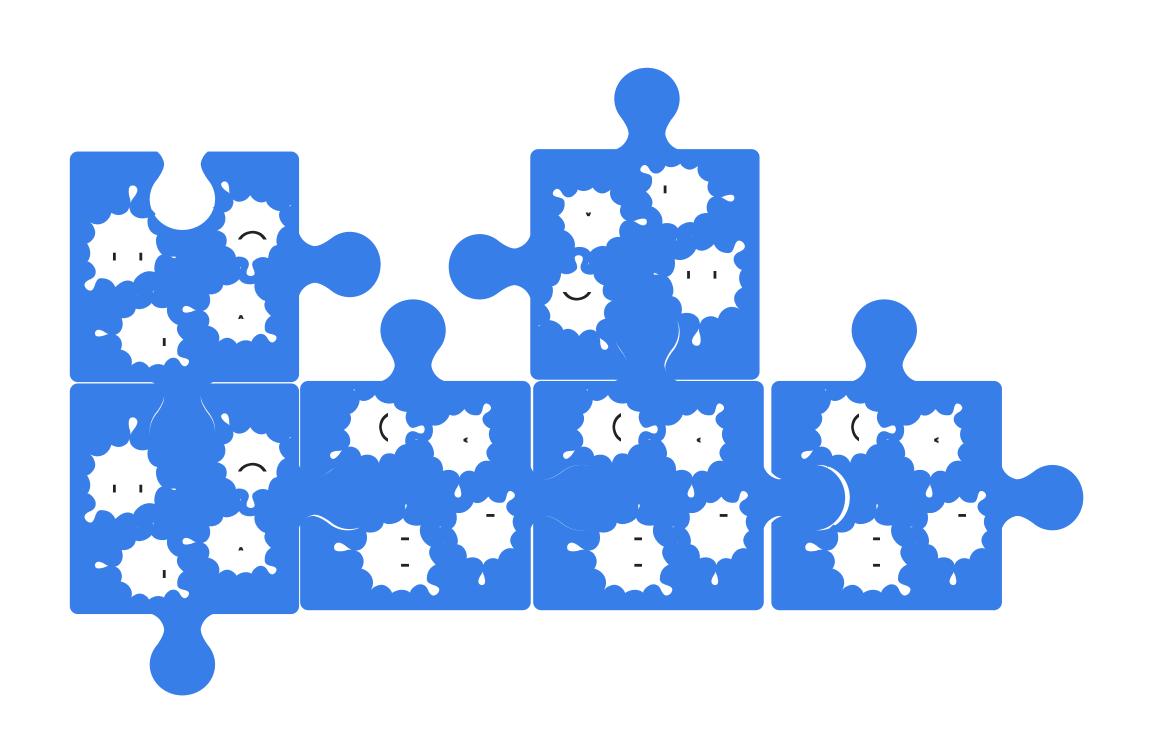




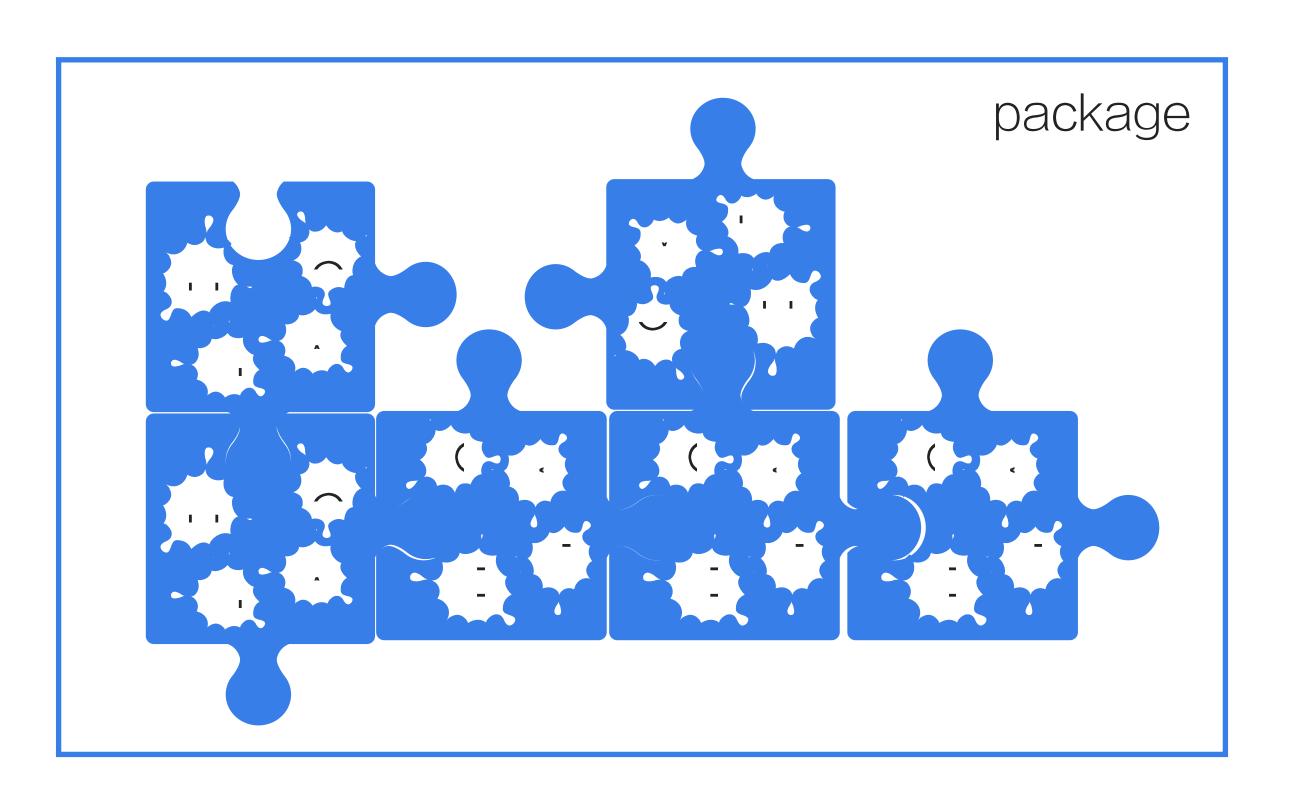




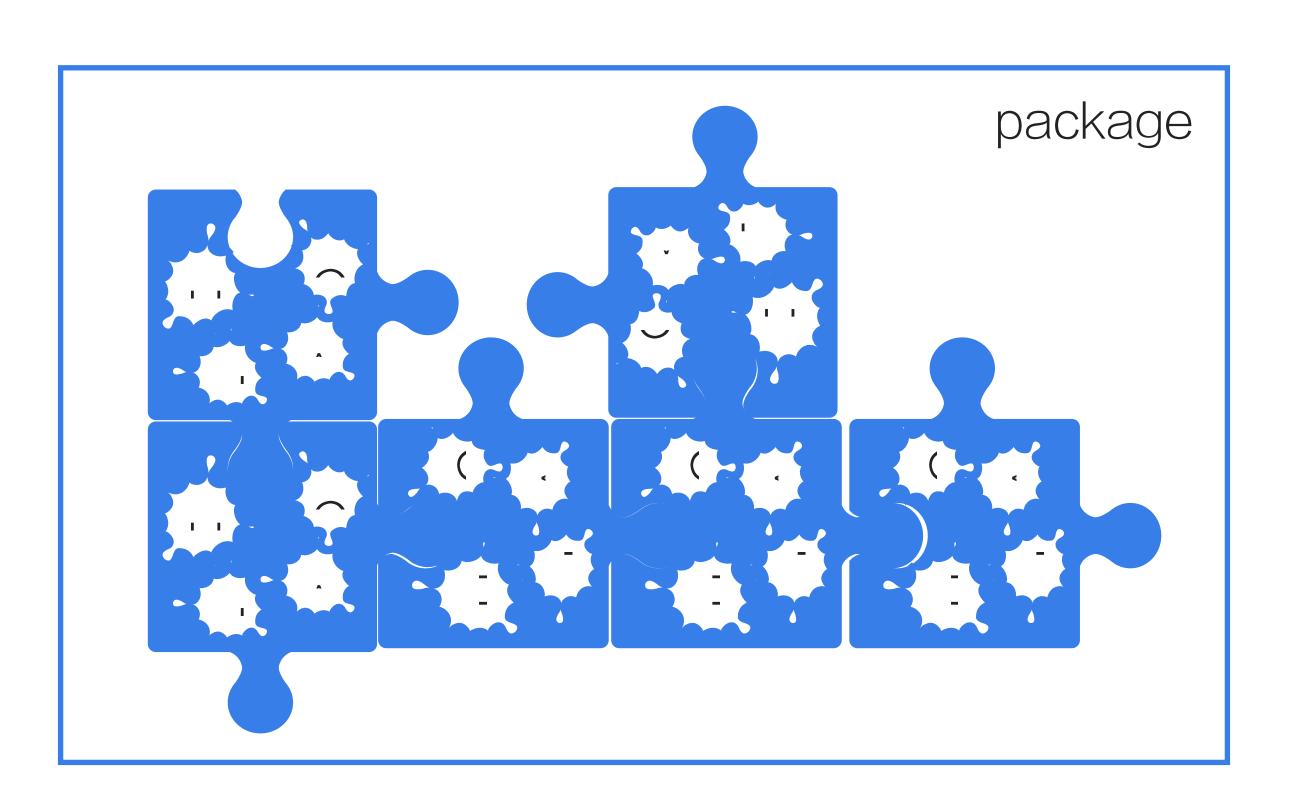












书同文、车同轨



日录结构与工程模板

目录结构与工程模板



dest = webpack(src, config);

输入和输出的完全分离

目录结构与工程模板



	静态单页	Java后端	全栈
简单应用	文档		域名系统
复杂应用		蜂巢控制台	



组件文档与接口文档



图由



测试



- 针对组件的单元测试
- 针对部分模块的集成测试
- 针对整个项目的系统测试



工具链

工具链



- vusion-cli
- vusion-app
- proto-ui



THANKS!