Infra Design System

从v4到v5

底层能力升级，助力 Space 加速产品统一

Weiwei Duan

2024.11.06



1 V5 版本的核心优势

2 为什么 Space 需要升级 V5

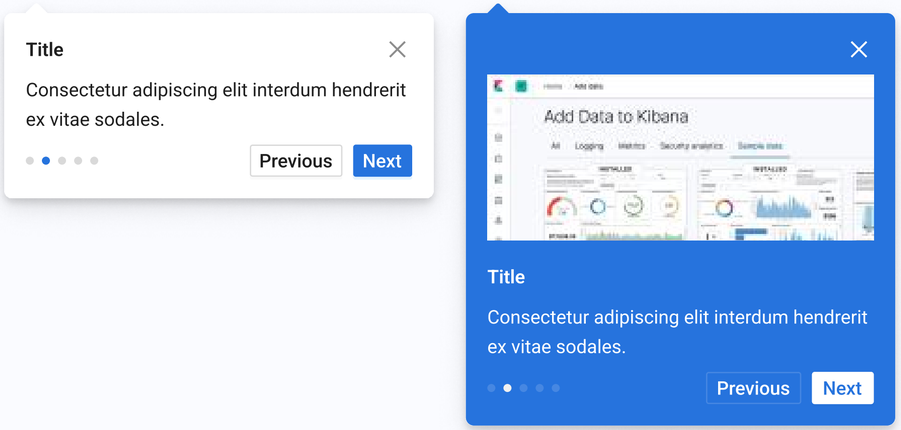
3 Space 升级 V5 的收益

V5 版本的核心优势

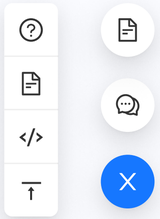
1. 设计升级　　2. 全新 Design Token 模型　　3. 支持动态主题

1 设计升级

Tour New



FloatButton New

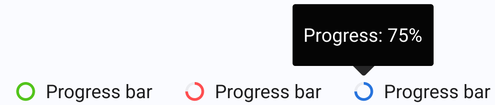


QrCode New

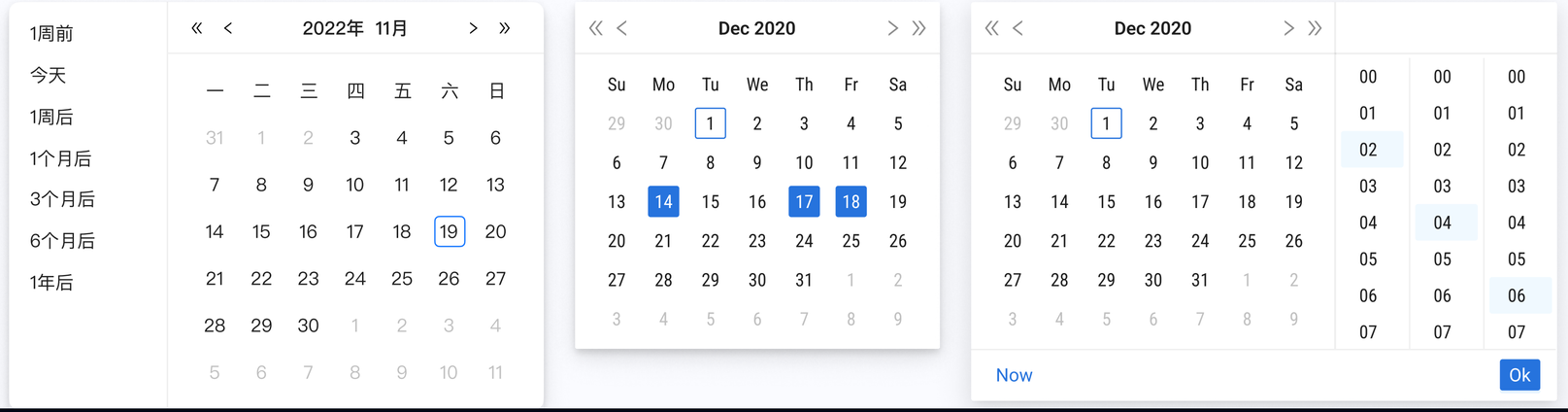


WaterMark New

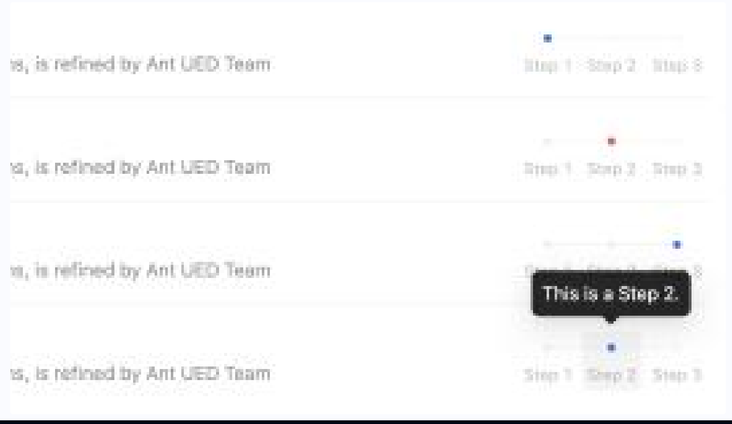
Progress Variant



DatePicker Variant



Steps Variant

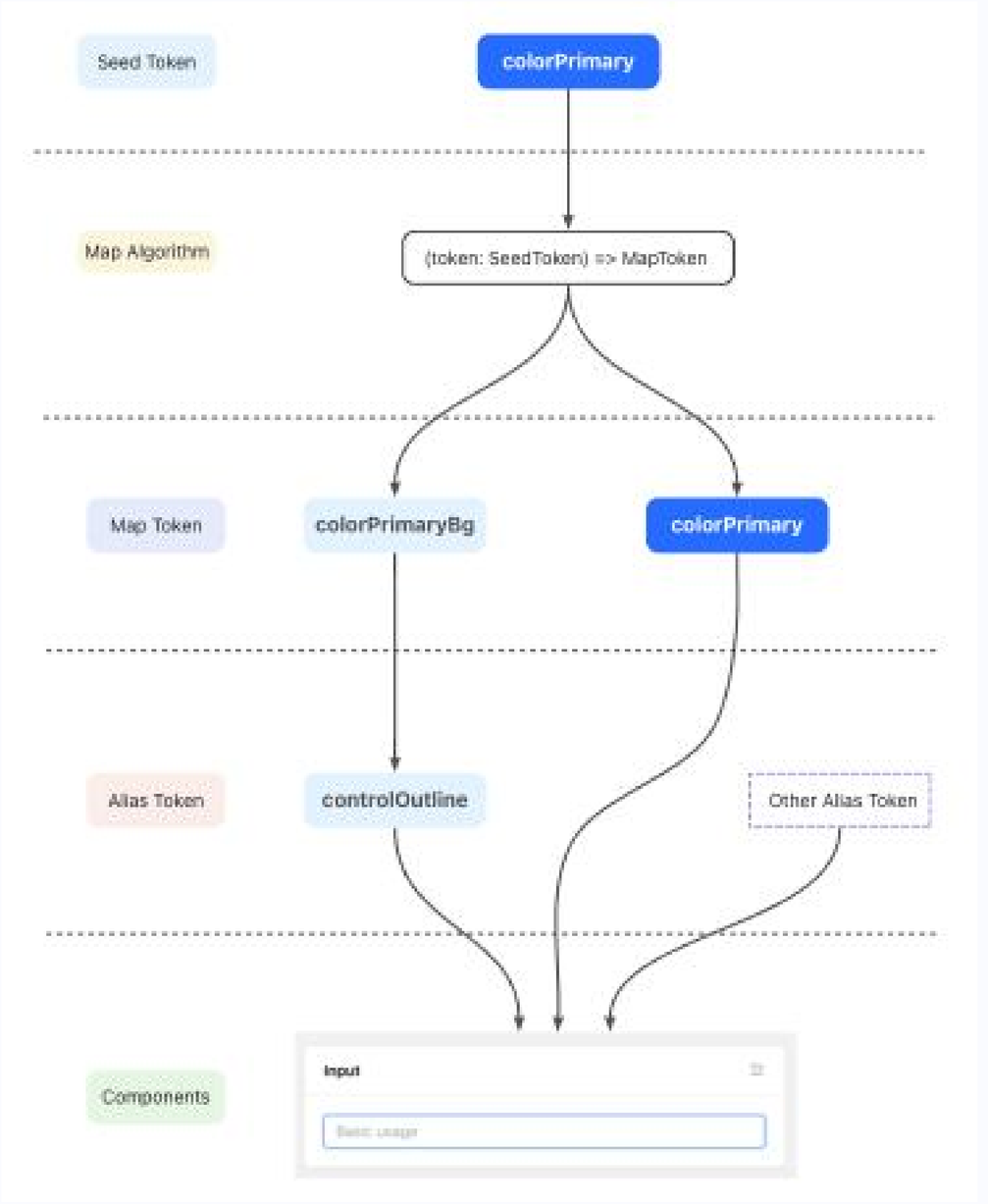


2 Design Token 模型

全新 Design Token 模型
代替 v4.x 中大量的 less 变量

在 v4 版本中，AntD 提出更多的 less 变量以支持主题定制能力。
然而除了色板支持完全的派生能力外，其他如字体、行高、间距都没有对应的算法。

在 v5 中，AntD 改造了所有的 Token，使其 基于 Seed + Algorithm 派生出所有的 Design Token。



Design Token 模型为三层结构

基于 Seed + Algorithm 派生出所有 Token

Seed Token 基础变量

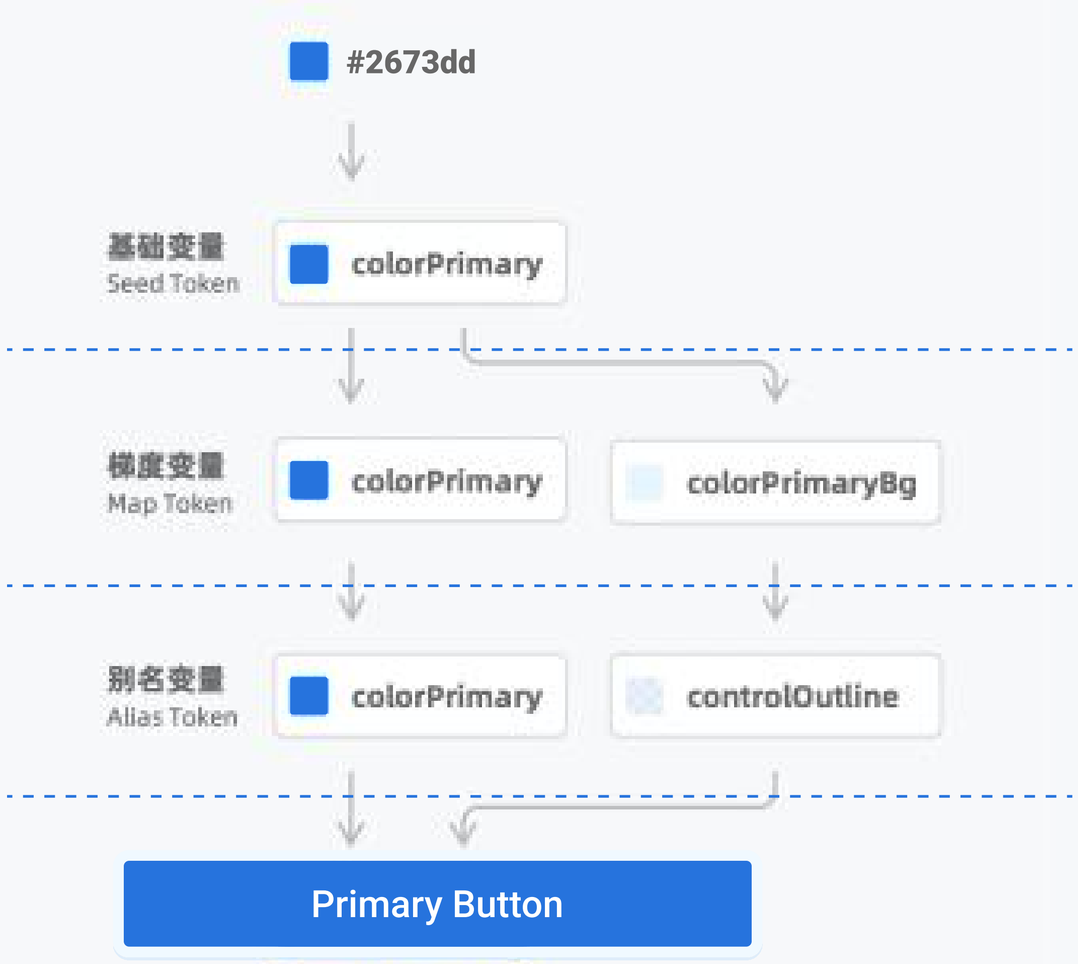
意味着所有设计意图的起源。比如 colorPrimary 表示主题色。

Map Token 梯度变量

是基于 Seed 派生的梯度变量，通过 theme.algorithm 算法计算得来，以保证 Map Token 之间的梯度关系。比如 colorPrimaryBg

Alias Token 别名变量

用于批量控制某些共性组件的样式，基本上是 Map Token 别名，或者特殊处理过的 Map Token。比如 controlOutline



设计意图的起源

Seed + Algorithm 派生出 Map Token

Algorithm 是代表派生规则的算法模型，V5 提供了如颜色/字体/行高/间距/圆角... 等的默认算法

Seed Token → Map Token → Alias Token

品牌色 colorPrimary **#2673DD**

Algorithm 基本算法

Map Token

主色浅色背景色 colorPrimaryBg	#f0f9ff
主色浅色背景悬浮态 colorPrimaryBgHover	#cfe9ff
主色描边色 colorPrimaryBorder	#a6d4ff
主色描边色悬浮态 colorPrimaryBorderHover	#79b6f7
主色悬浮态 colorPrimaryHover	#4d94eb
品牌主色 colorPrimary	#2673dd
主色激活态 colorPrimaryActive	#1654b8
主色文本悬浮态 colorPrimaryTextHover	#4d94eb
主色文本 colorPrimaryText	#2673dd
主色文本激活态 colorPrimaryTextActive	#1654b8

字号 fontSize **14**

Algorithm 基本算法

Map Token

字号		
默认字号 fontSize	14	Aa
fontSizeSM	12	Aa
fontSizeLG	16	Aa
fontSizeXL	20	Aa
一级标题字号 fontSizeHeading1	38	Aa
二级标题字号 fontSizeHeading2	30	Aa
三级标题字号 fontSizeHeading3	24	Aa
四级标题字号 fontSizeHeading4	20	Aa
五级标题字号 fontSizeHeading5	16	Aa
行高		
lineHeight	1.57142857142...	Aa
lineHeightSM	1.6666666668...	Aa
lineHeightLG	1.5	Aa
一级标题行高 lineHeightHeading1	1.21052631578...	Aa
二级标题行高 lineHeightHeading2	1.2666666666...	Aa
三级标题行高 lineHeightHeading3	1.3333333333...	Aa

尺寸步长 sizeStep **4**

尺寸变化单位 sizeUnit **4**

Algorithm 基本算法

Map Token

外间距	
外边距 XXS marginXXS	4
外边距 XS marginXS	8
外边距 SM marginSM	12
外边距 margin	16
外边距 MD marginMD	20
外边距 LG marginLG	24
外边距 XL marginXL	32
外边距 XXL marginXXL	48
内间距	
超小内间距 paddingXXS	4
特小内间距 paddingXS	8
小内间距 paddingSM	12
内间距 padding	16
中等内间距 paddingMD	20
大内间距 paddingLG	24
特大内间距 paddingXL	32

基础文本色 colorTextBase **#000000**

基础背景色 colorBgBase **#FFFFFF**

Algorithm 基本算法

Map Token

文本	
一级文本色 colorText	rgba(0, 0, 0, 0.8...
二级文本色 colorTextSecondary	rgba(0, 0, 0, 0.6...
三级文本色 colorTextTertiary	rgba(0, 0, 0, 0....
四级文本色 colorTextQuaternary	rgba(0, 0, 0, 0.2...
描边	
一级边框色 colorBorder	#d9d9d9
二级边框色 colorBorderSecondary	#f0f0f0
填充	
一级填充色 colorFill	rgba(0, 0, 0, 0.1...
二级填充色 colorFillSecondary	rgba(0, 0, 0, 0.0...
三级填充色 colorFillTertiary	rgba(0, 0, 0, 0.0...
四级填充色 colorFillQuaternary	rgba(0, 0, 0, 0.0...
背景	
组件容器背景色 colorBgContainer	#ffffff
浮层容器背景色 colorBgElevated	#ffffff
布局背景色 colorBgLayout	#f5f5f5

3 动态主题

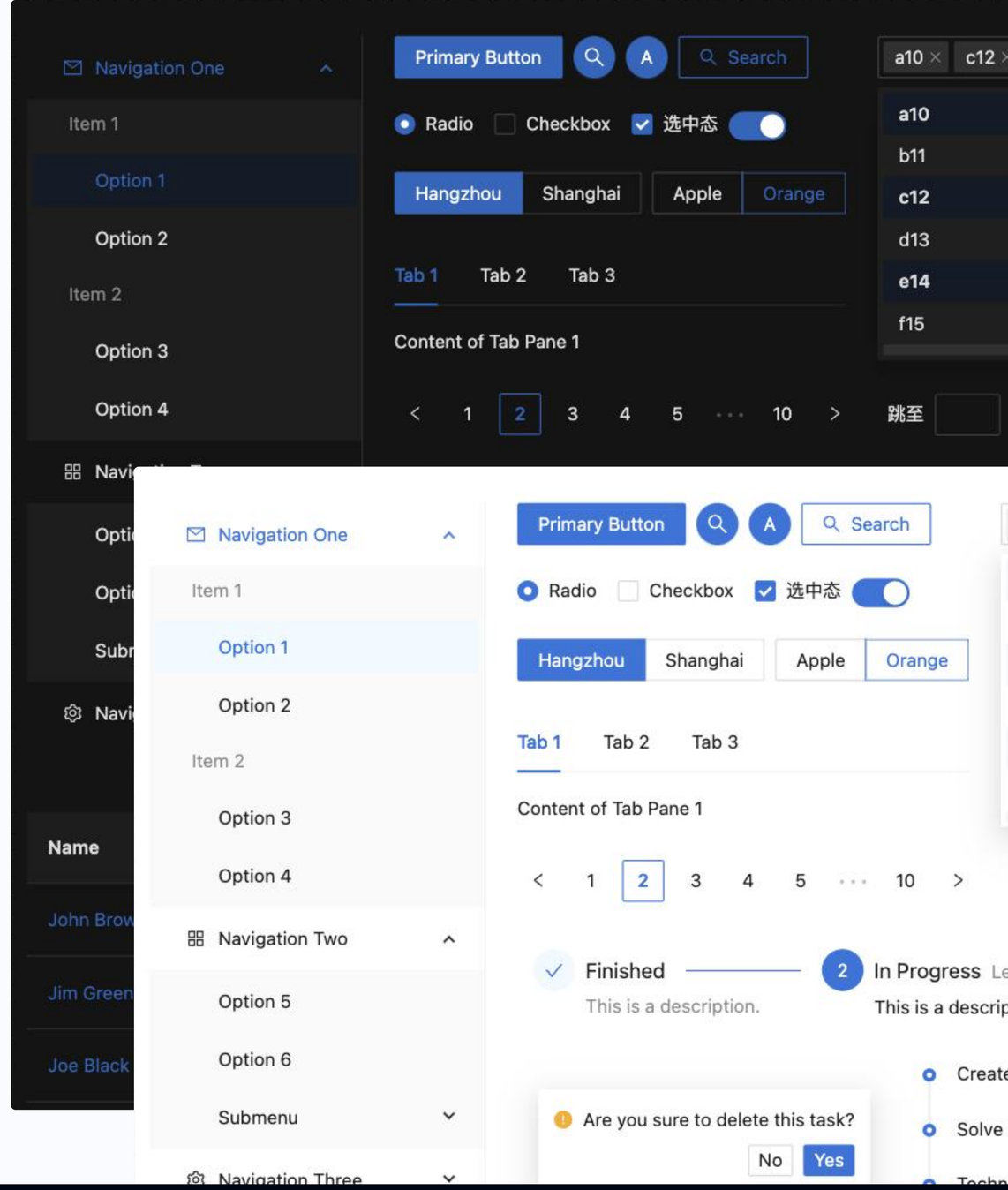
采用 CSS-in-JS 支持动态主题

支持切换、多主题并存、组件级修改主题变量等

在 v4 时期，CSS Variables 停留在了动态主题色而没有进一步提供暗色、其他 Token 的动态能力。在 v5 中改用了 CSS-in-JS 不需要维护中间变量，但是有更多的运行时消耗，在经过一系列尝试后，我们研发了一套针对组件级别的 CSS-in-JS 库 @ant-design/cssinjs

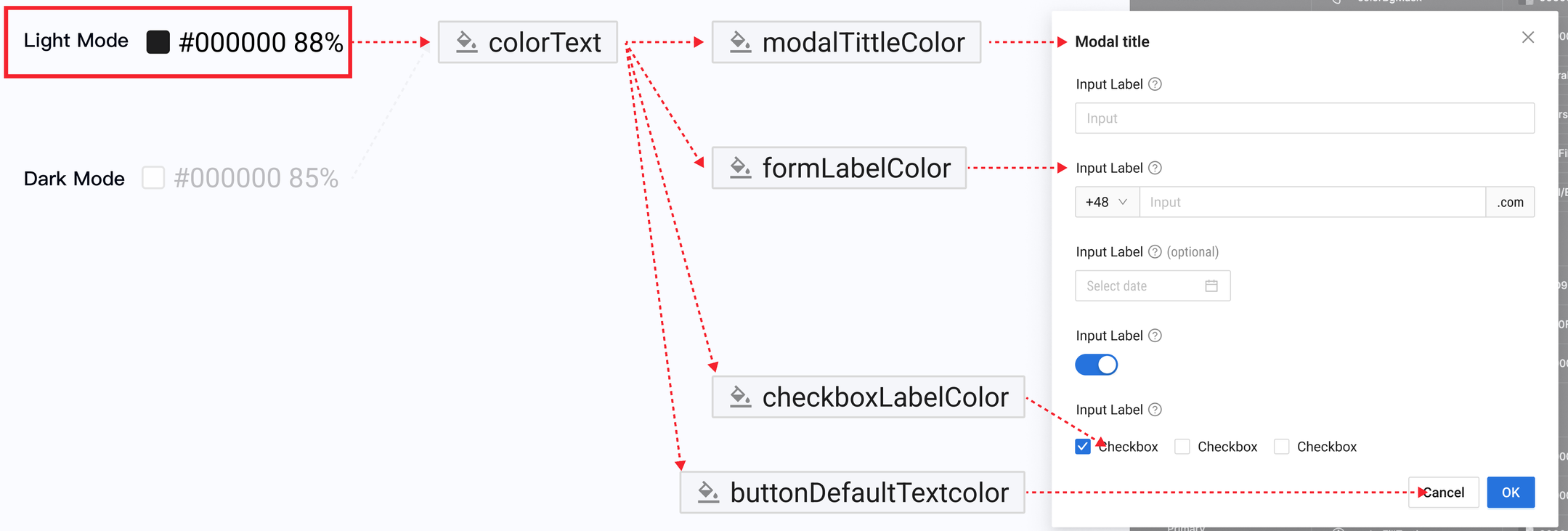
有了 CSS-in-JS 的加持后，动态主题的能力也得到了加强，包括但不限于：

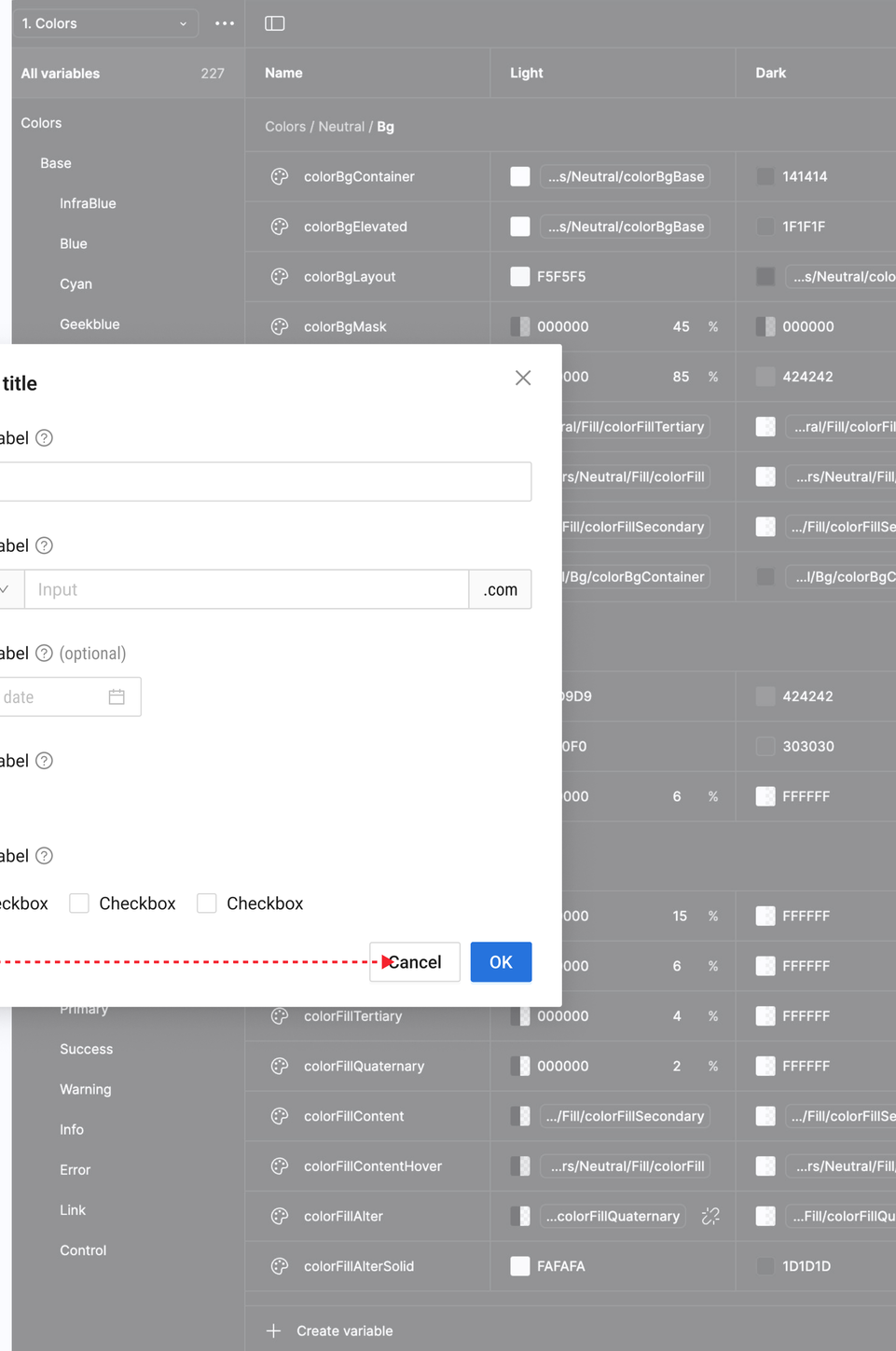
1. 支持动态切换主题；
2. 支持同时存在多个主题；
3. 支持针对某个/某些组件修改主题变量；
4. ...



Token 内置暗黑模式变量

颗粒度精确到组件层面





Token 内置暗黑模式变量

颗粒度精确到组件层面

Light Mode #000000 88%

Dark Mode #000000 85%

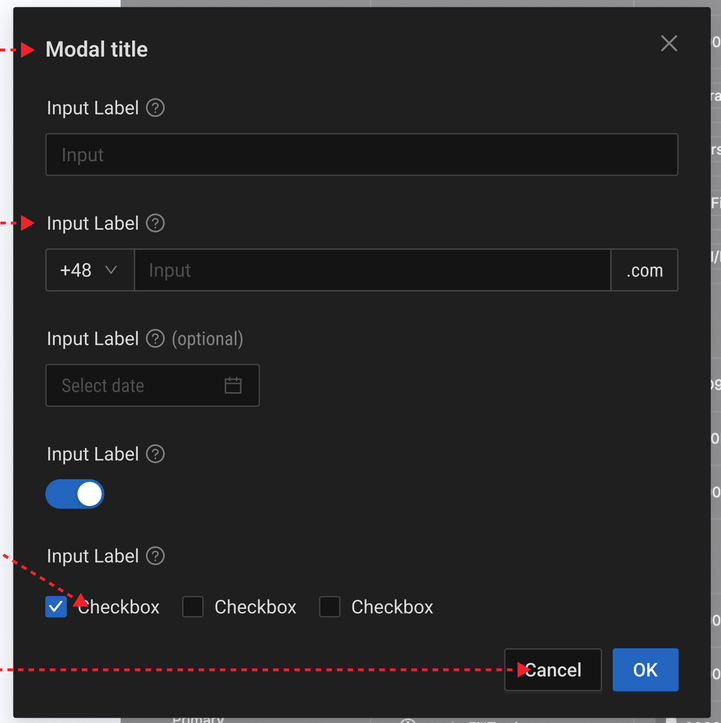
colorText

modalTittleColor

formLabelColor

checkboxLabelColor

buttonDefaultTextcolor



为什么 Space 需要升级 V5

1.Space UI/UX Standardisation　　2. 不ok的动态主题

1 Space UI/UX Standardisation

SpaceBizComponent 是设计系统的一部分，依赖设计系统的底层能力

Background

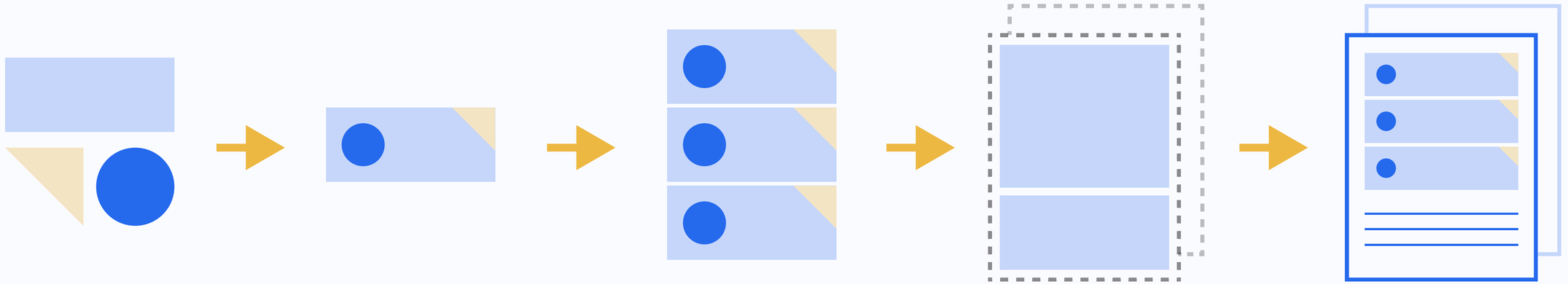
> - Mode of usage for products varies too widely across users
> - 99+ products and tools within whole Eng Infra Dept at the moment, lack of clarity on many product status - if any infra product should be made legacy or to be deprecated
> - Product category are excessive and unclear
> - Infra products metadata lack governance
> - Duplication of efforts in common abilities/tools products
> - UI/UX for products varies too widely, lack of UI/UX standardisation

- David Chow 《NEW ENGINEERING INFRA PRODUCT STANDARDS PROPOSAL》

设计系统的构建

原子化设计——科学构建设计系统的方法论

Atomic Design 是 Brad Frost 在 2016 年提出的概念，他列出了5个层级作为核心概念支架。
从抽象到具象，从元素到组件，让设计师从底层开始思考，对整个设计会有更清晰的理解。
同时也能让设计更加统一，在新增组件的时候更谨慎。



ATOMS
原子层

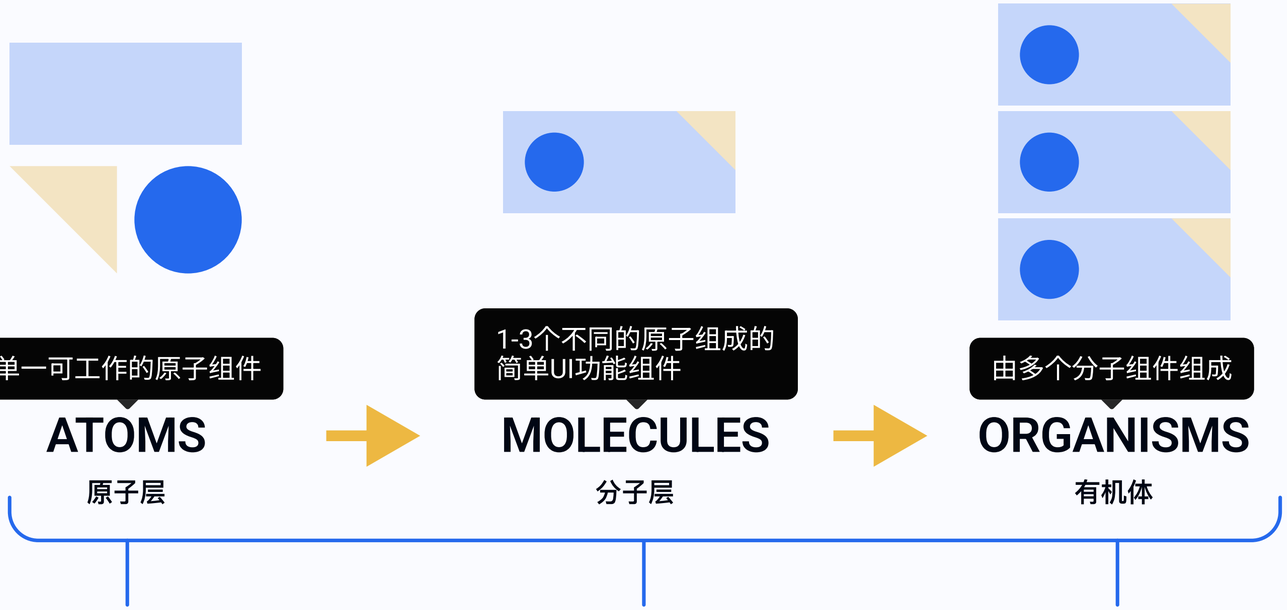
MOLECULES
分子层

ORGANISMS
有机体

TEMPLATES
模版

PAGES
页面

原子设计对应组件



单一可工作的原子组件

ATOMS

原子层

1-3个不同的原子组成的简单UI功能组件

MOLECULES

分子层

由多个分子组件组成

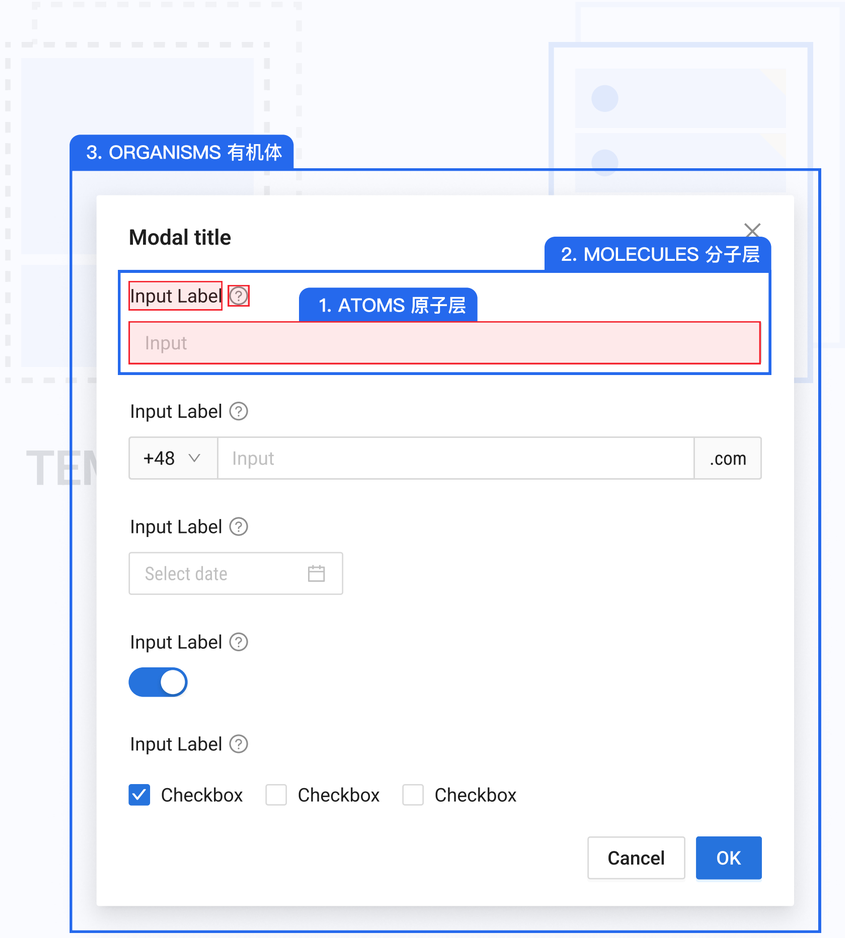
ORGANISMS

有机体

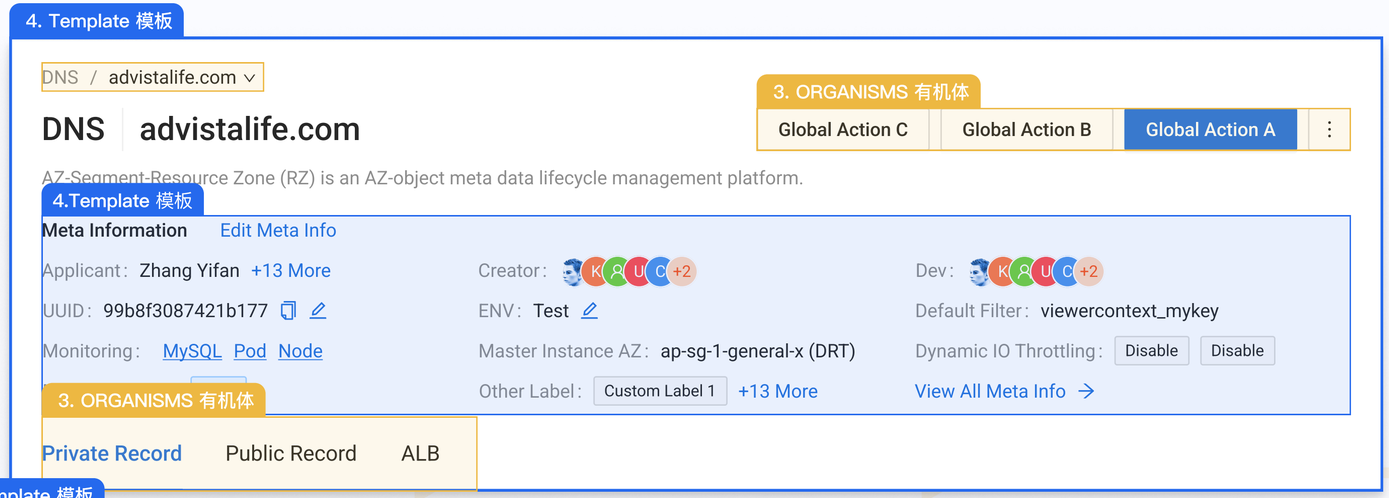
- Typographic styles 字体样式
- Color swatches 颜色色板
- Icons 图标
- Radio buttons 单选按钮
- Checkboxes 多选框
- Buttons 按钮
- Input 输入框
- Tabs 标签文字型按钮
- Tags 标签
- Tooltips 信息提示框

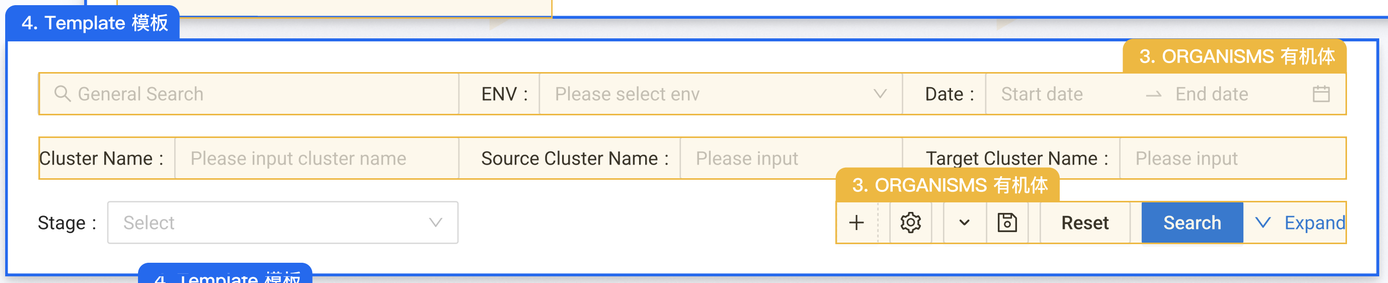
- Dropdown menus 下拉菜单
- Dropdown buttons 下拉按钮
- Date pickers 日期选择器
- Search components 搜索组件
- Breadcrumbs 面包屑导航栏
- Collapsible group items 可收缩群组
- Input fields (with labels) 带标识输入栏
- Media uploaders 上传组件（选文件按钮+标题）
- Loading components 加载模块（进度条+X+名称）
- Pagination 页面页数选择栏

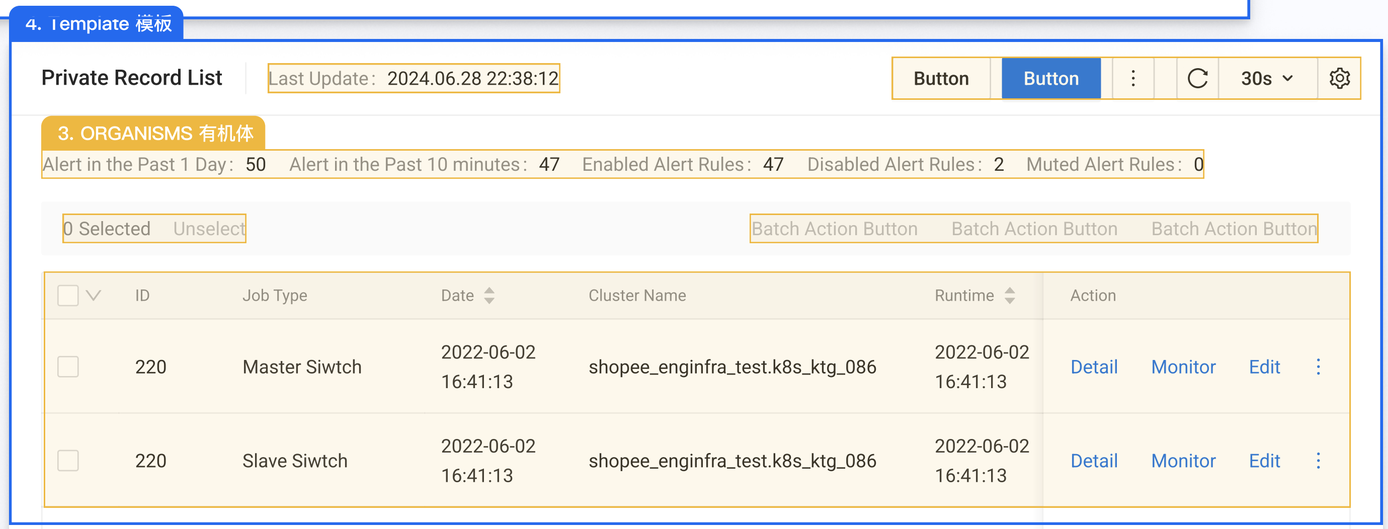
- Navigation/Tab bars 导航栏
- Tables 信息表格（常见小长条卡片组成）
- Carousels 轮播形式（常见卡片+选择箭头/...）
- Forms 信息表单



原子设计对应 SpaceBizComponent









MetaInfo 元信息

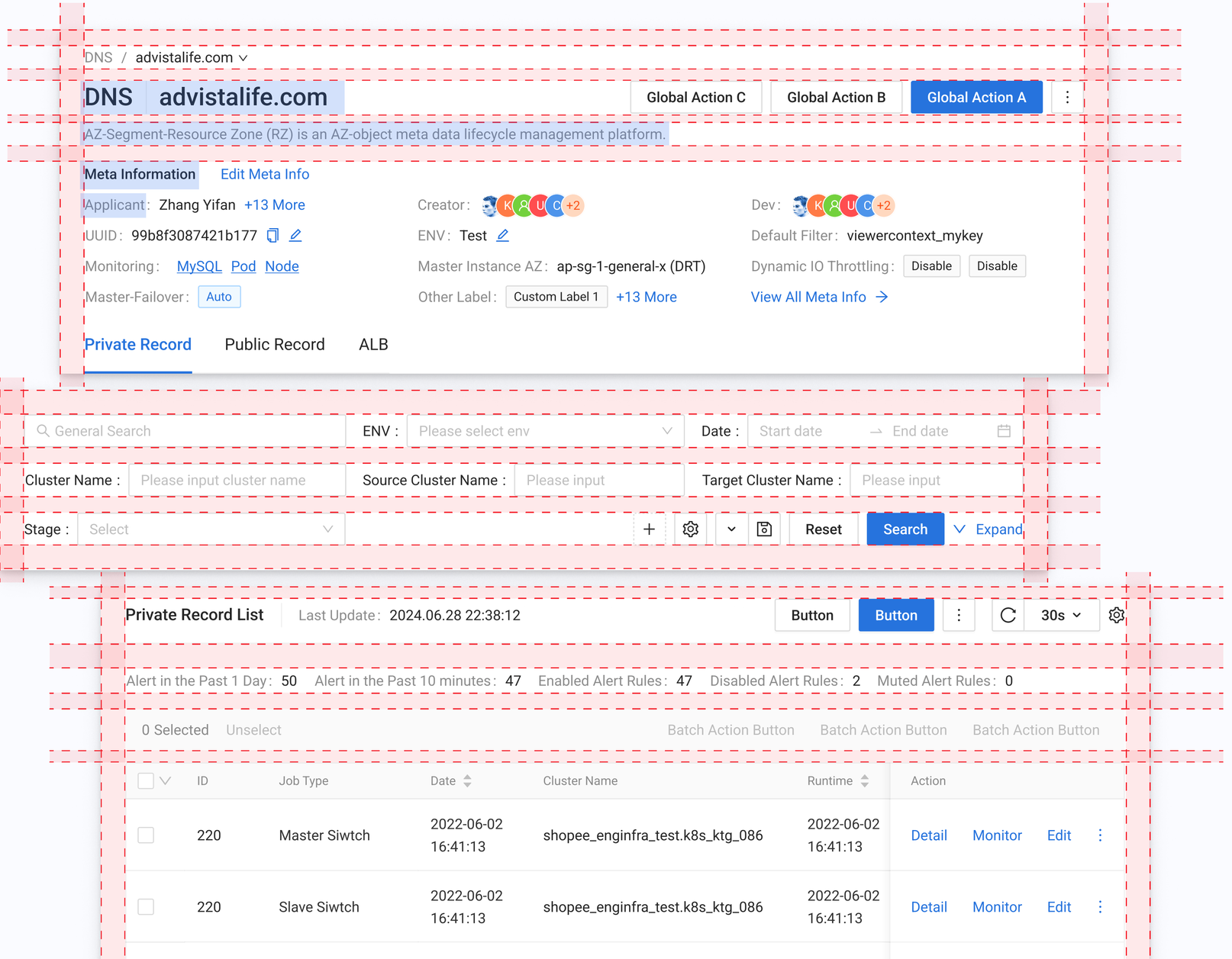
Table.SearchForm 带查询的表格

SideBar 侧导航

Service Tree 服务树

.....

TEMPLATES 级别的设计属性该如何定义呢?



- FontSize
- FontWeight
- Padding
- Margin
- TextColor
- Shadow
- ……

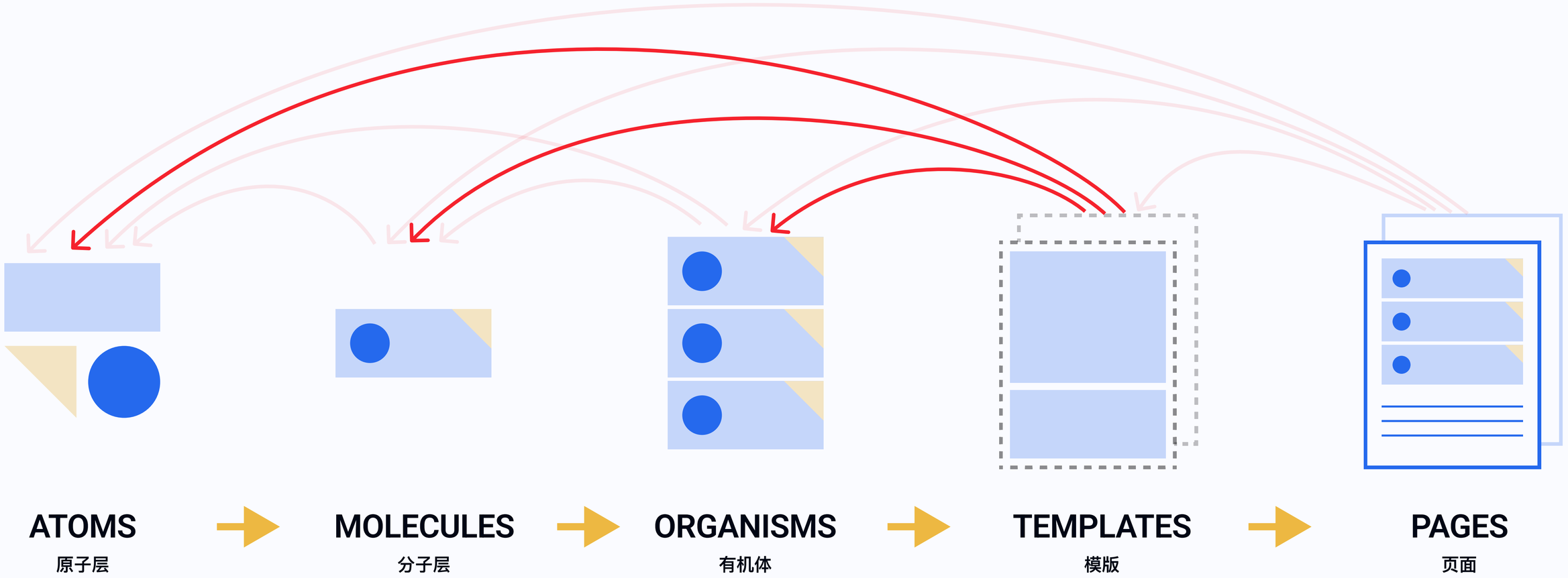
字号梯度为什么是这样?

间距用哪个? 8、16还是24? 18行不行?

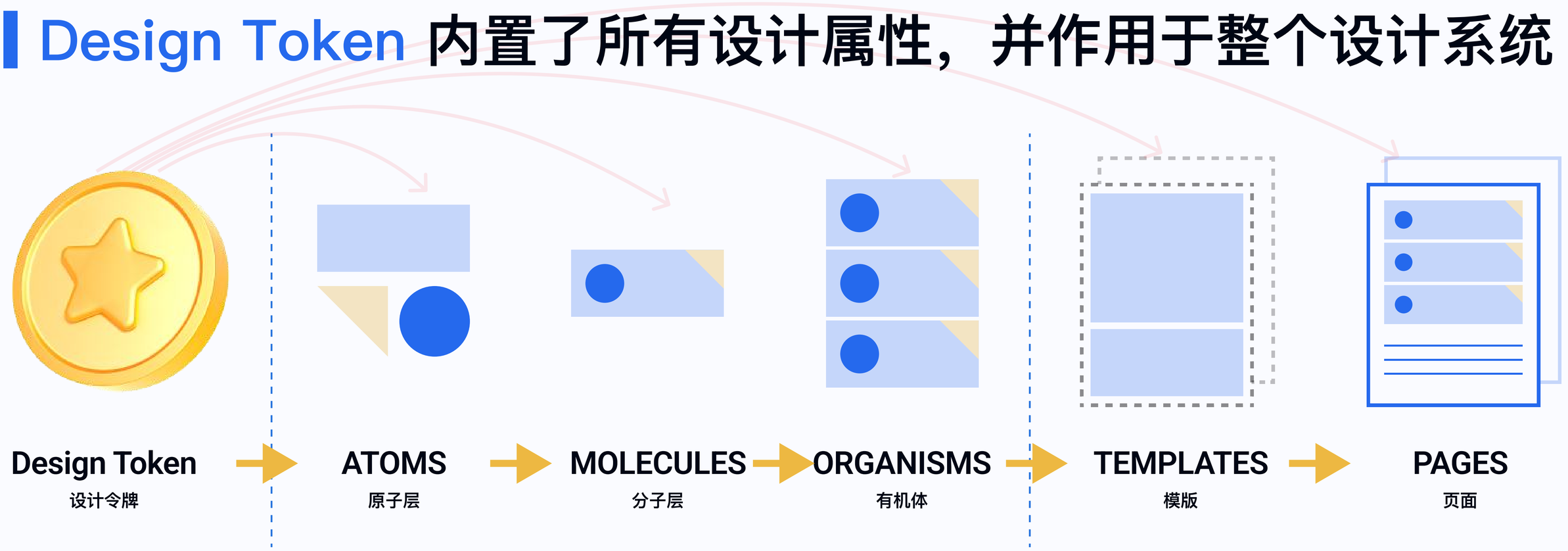
为什么要这样用?

逻辑是什么?

原子设计每一层都会继承上一层的全部属性



Design Token 内置了所有设计属性，并作用于整个设计系统



“Design Token 是正确且唯一的设计决定，助力大团队进行设计、感知和开发。”

“A Design System is the single source of truth which groups all the elements that will allow the teams to design, realize and develop a product."

—— 原子设计与设计系统

TEMPLATES 级别仍需使用 Design Token

以文字颜色为例

1. 基于 Seed + Algorithm 派生出梯度色板
2. 语意化命名的 Design Token 代表正确且唯一的设计决策

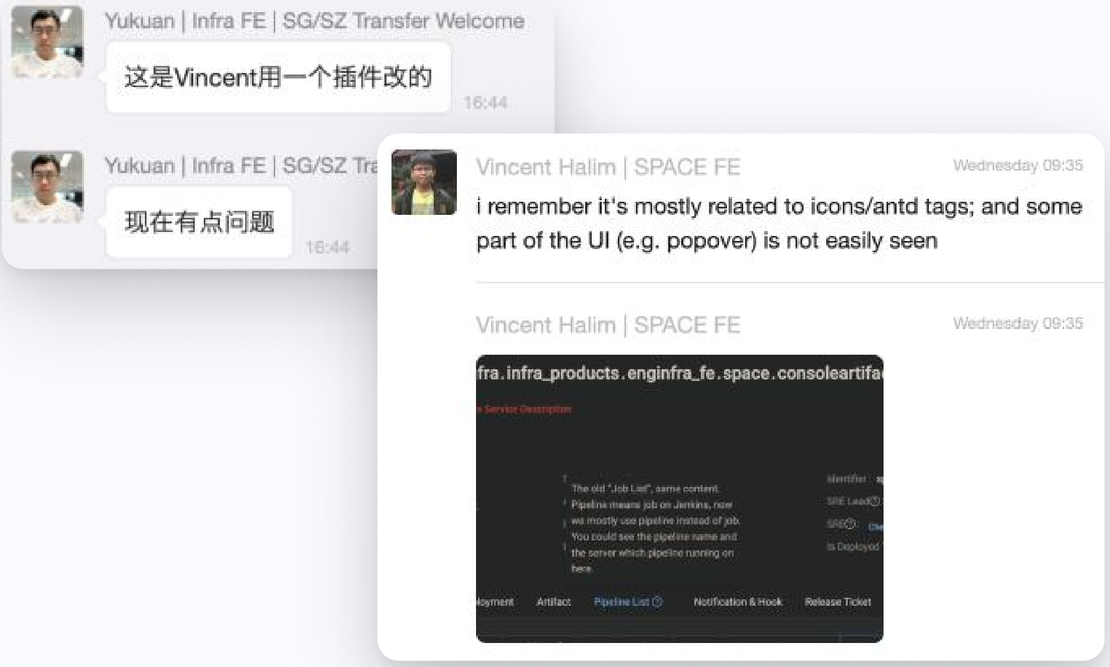


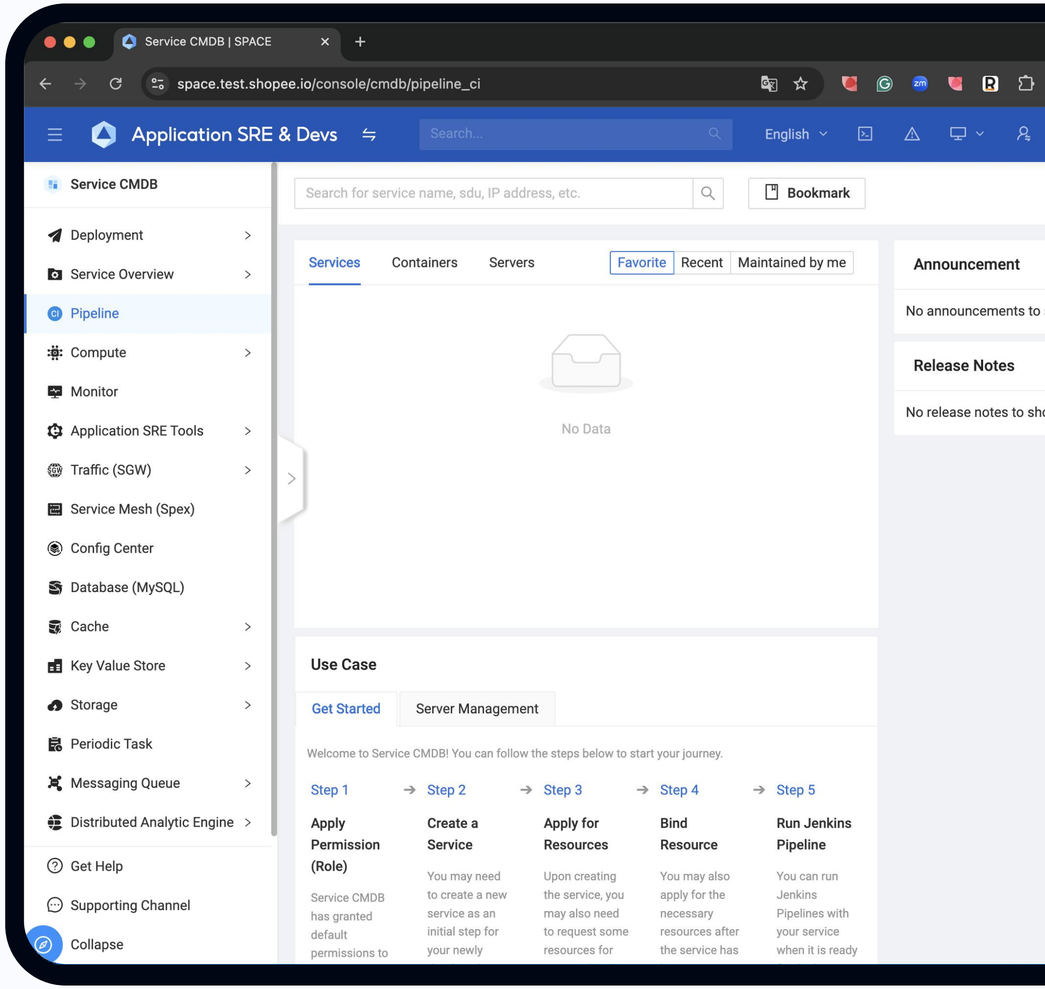
2 不👌的动态主题

Space 切换暗黑模式　　Space 主题色升级

动态主题 | 暗黑模式

前端通过插件实现了暗黑模式





动态主题 | 暗黑模式

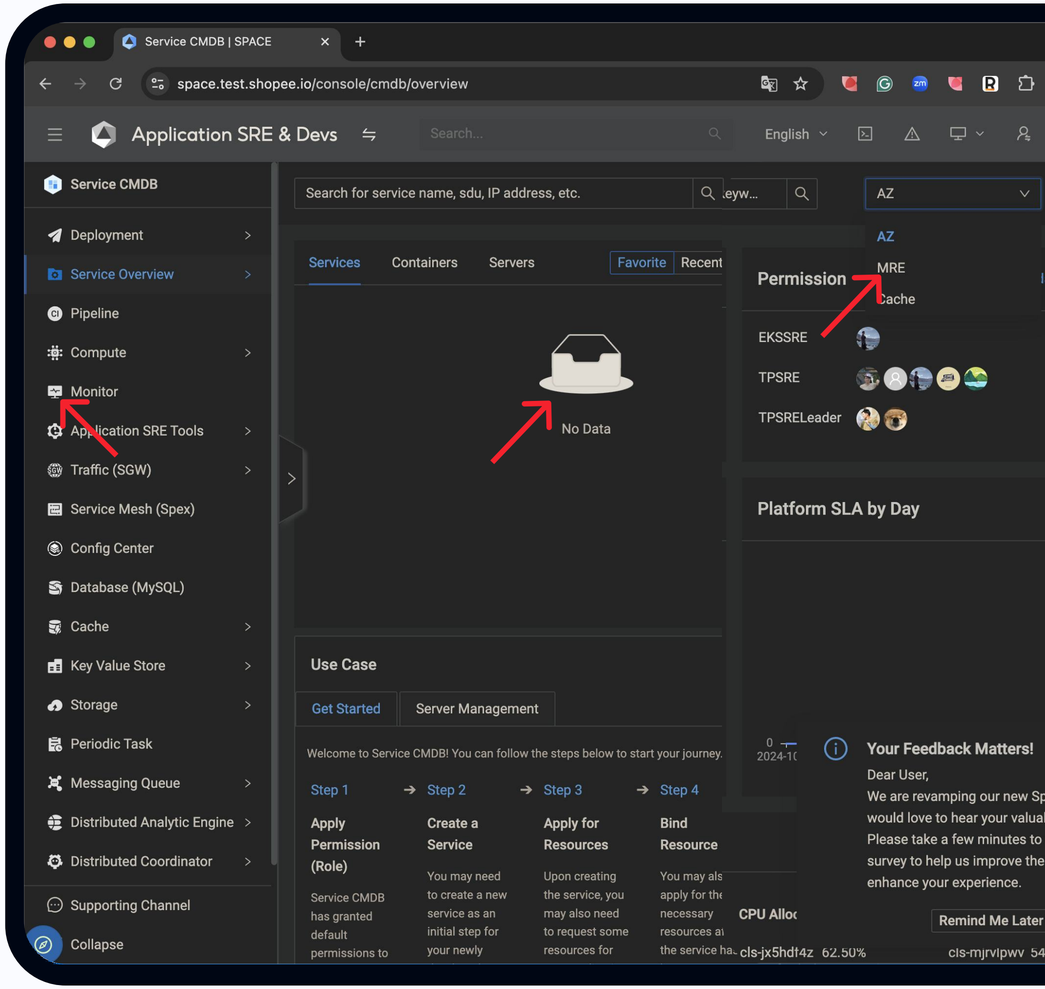
插件实现暗黑模式存在诸多问题

- 视觉障碍严重
- 难以区分信息层级
- 后期维护成本高

原因如下：

1. 颜色对比度不符合无障碍设计标准
2. 文字、可交互控件和背景的主次关系对比
3. 仅从全局层面给颜色做了一一对应的替换，未考虑同一颜色在不同组件内的使用差异，后期调整难度大

参考文档《According to the WCAG》



不OK的动态主题

动态主题 | 暗黑模式

插件实现暗黑模式存在诸多问题

- 切换主题时会卡顿
- 静态站点切换到暗色后刷新，会从亮色再变回暗色

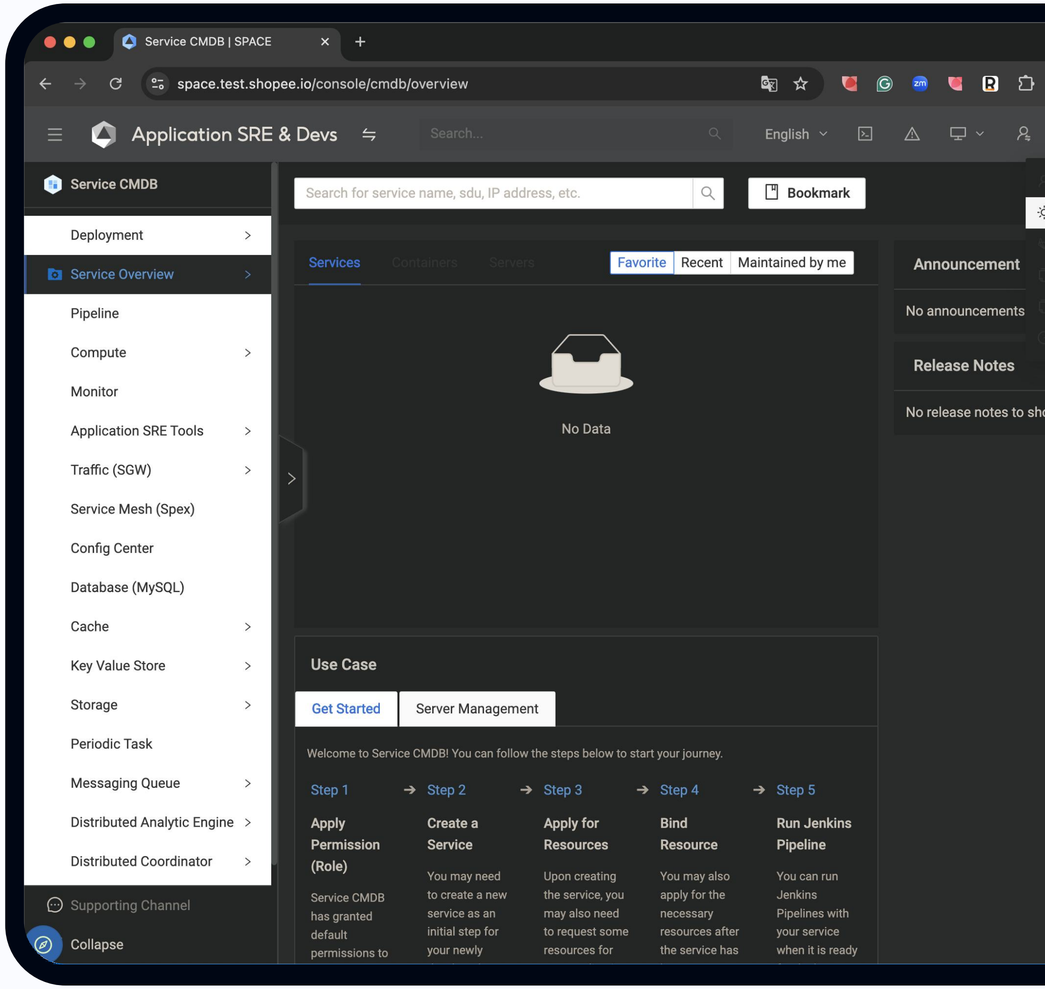
其原因是由于 cssinjs 运行时生成样式的特性：

1. 卡顿是因为需要进行一轮新的 css 序列化；
2. 不能无缝刷新是因为静态站点不能保留切换后主题的样式。

升级V5后，在 CSS 变量的主题系统下将不存在上述两个问题，原因如下：

1. 修改 CSS 变量不需要重新序列化 css，省去了这一性能消耗；
2. 在页面渲染之前 CSS 变量就可以进行注入，即通过 body 下的 script 来阻塞渲染，从而避免了渲染不必要的样式。

参考文档《当 Ant Design 遇上 CSS 变量》



动态主题 | 一键换肤

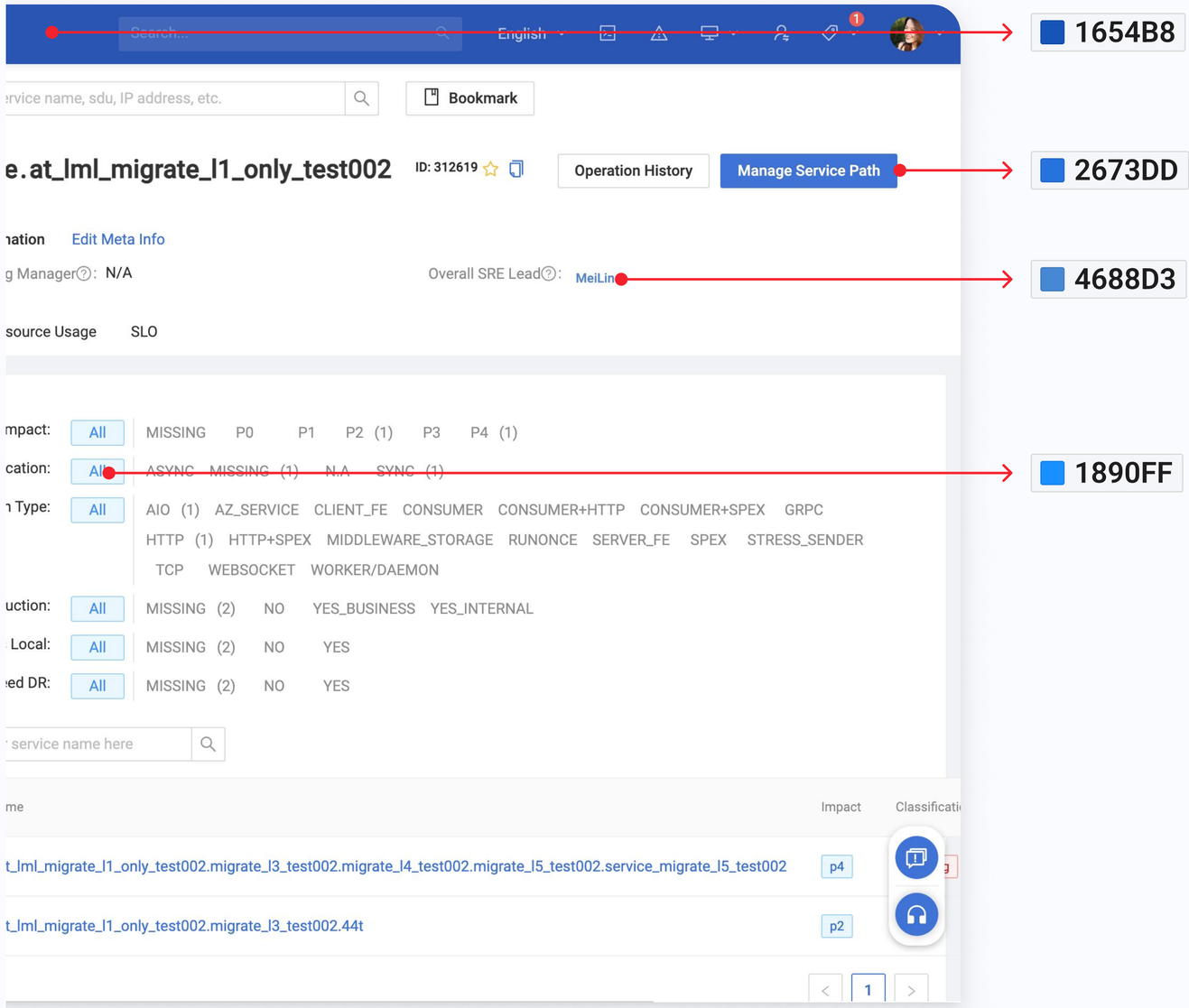
Space五彩斑斓的蓝

主题色的应用无法覆盖全局

原因：

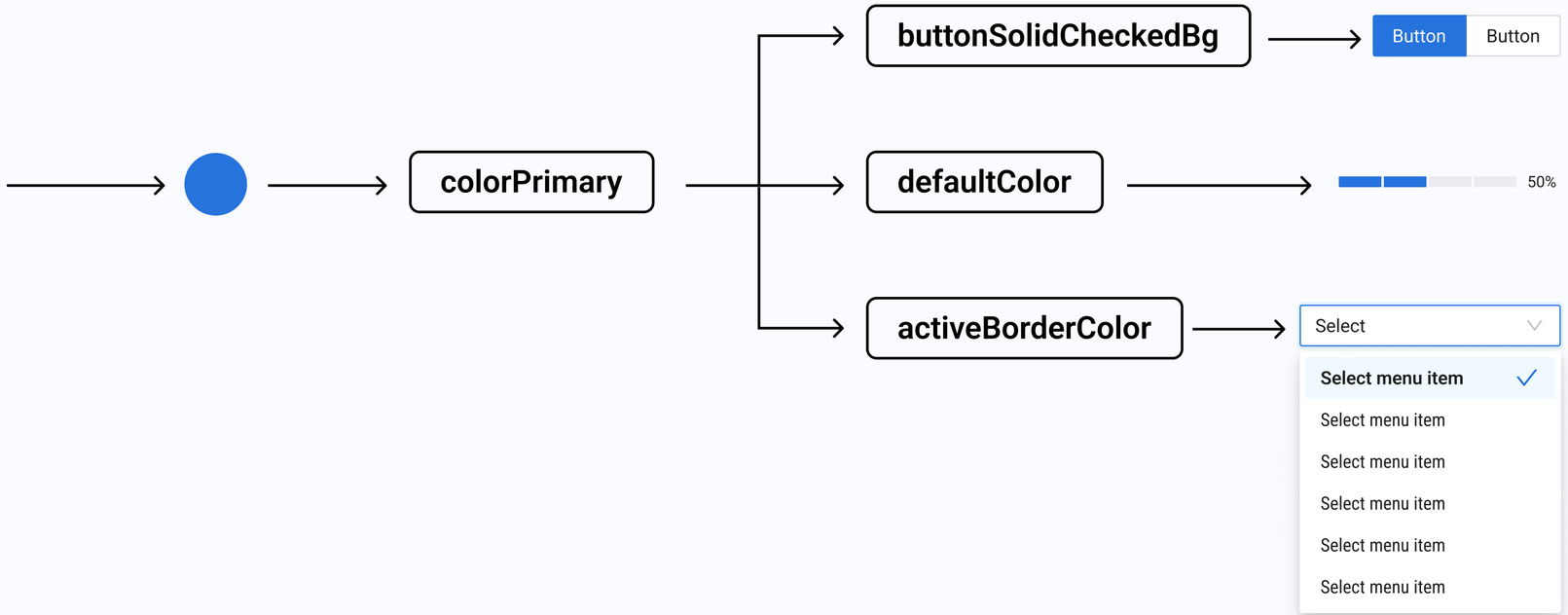
1. 没有通过 token 控制，无法一键触达 Component 层级
2. 系统中存在大量硬代码，例如 #2673dd，前端需手动查找替换





使用 Token 后，任何变更可直接触达全局

infrablue-1	#F0F9FF
infrablue-2	#CFE9FF
infrablue-3	#A6D4FF
infrablue-4	#79B6F7
infrablue-5	#4D94EB
infrablue-6	#2673DD
infrablue-7	#1654B8
infrablue-8	#0A3991
infrablue-9	#02236B
infrablue-10	#011445

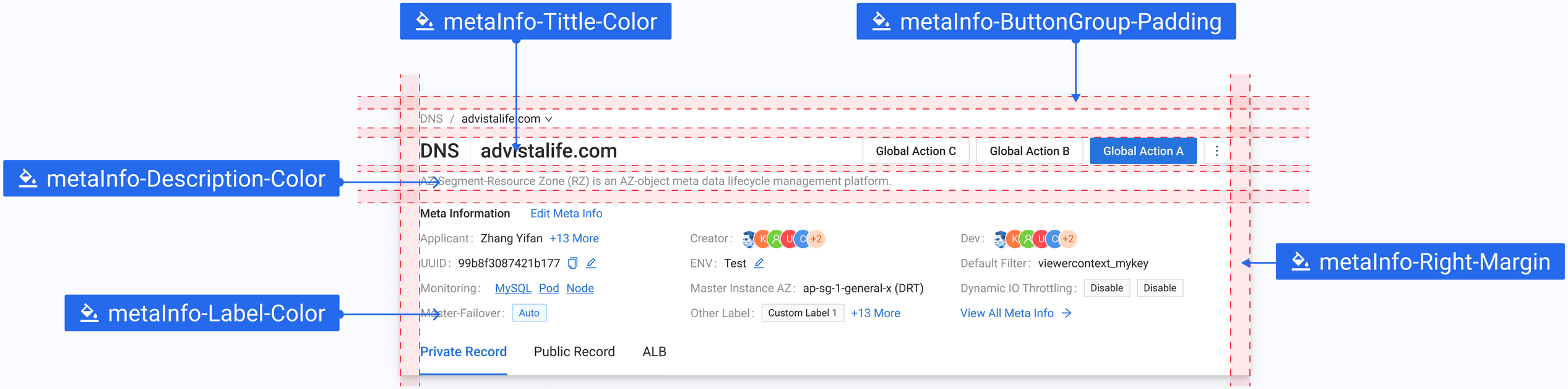


Space 升级 V5 的收益

1.设计决策效率提升　　2. 产品体验更加一致　　3. 产品迭代效率提升　　4. 研发结果更加精准

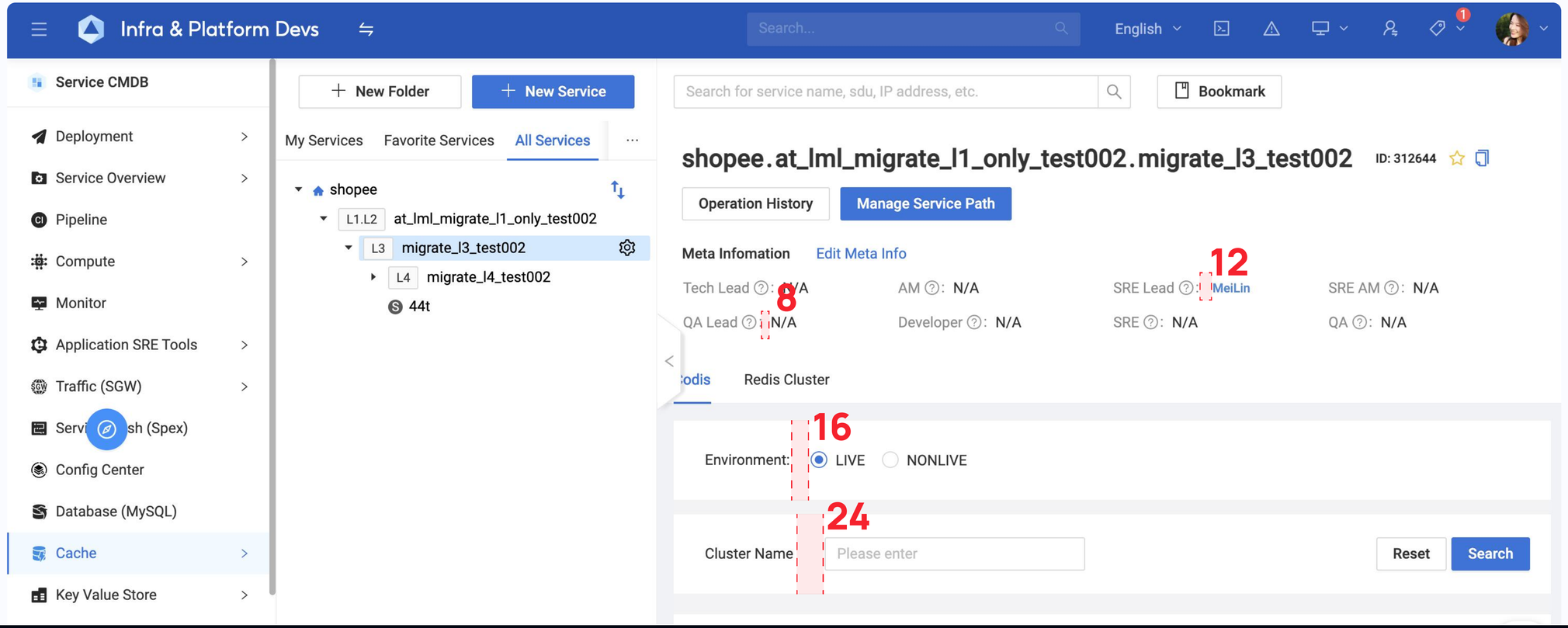
1 设计决策效率提升

将设计变量转化为语义化的 Token，帮助使用者建立界面的“画面感”



2 产品体验更加一致

在团队内使用统一的 Token 能够提升设计产出的一致性



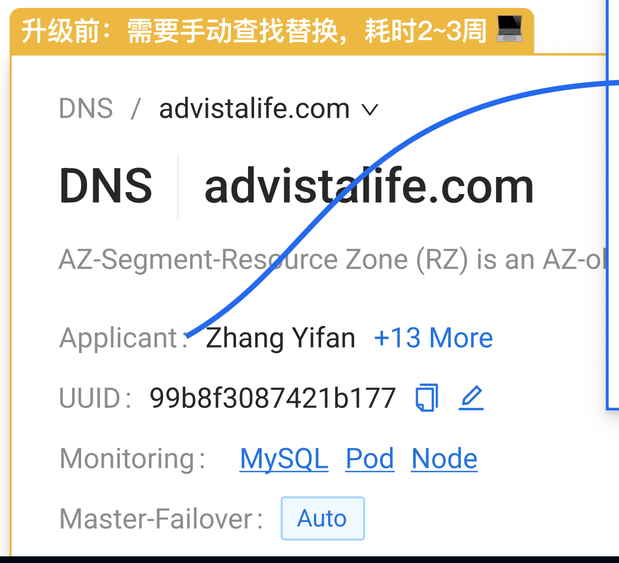
3 产品迭代效率提升

通过对 Token 的变更，任何设计改动都能够及时的应用到所有设计中

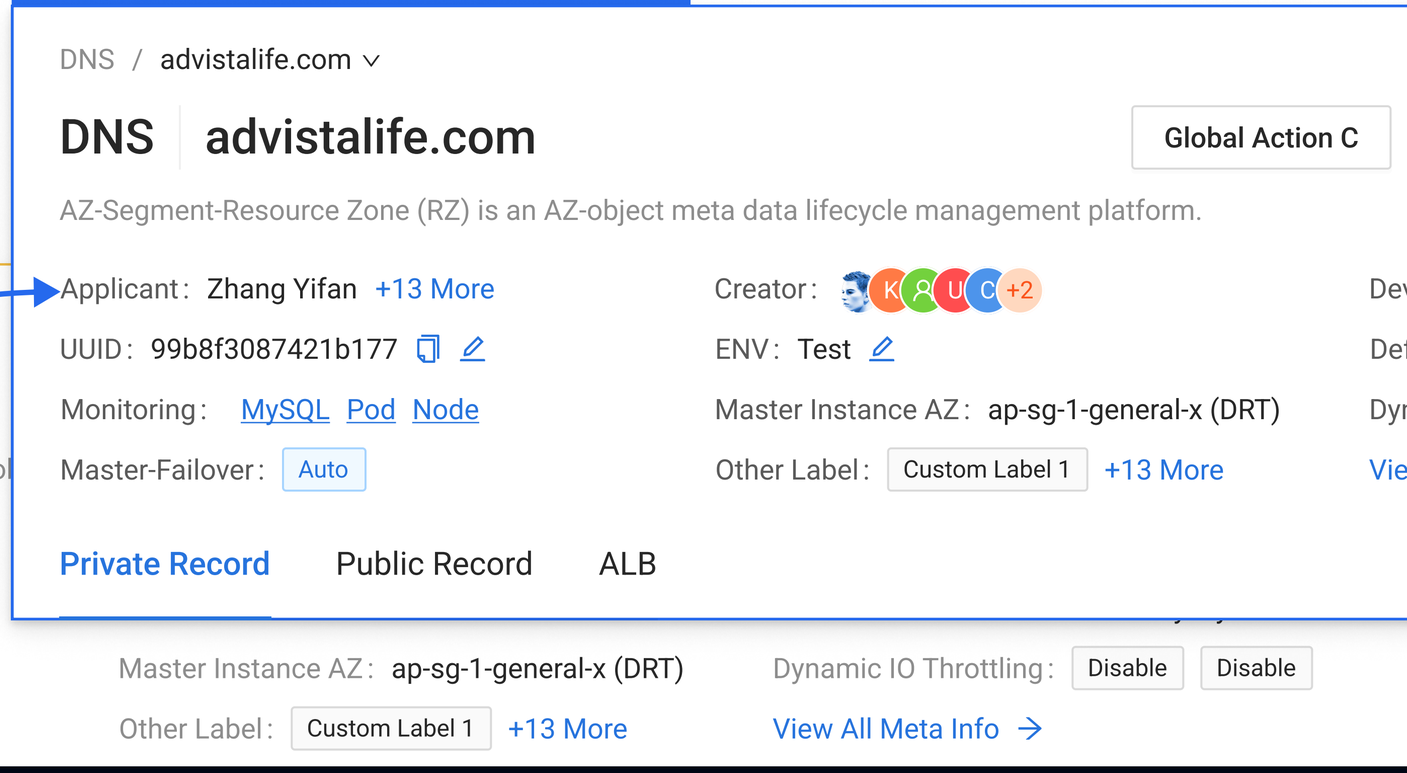
举例：新增文字颜色梯度

- colorTextHeading
- colorTextLabel
- colorTextDescription
- colorTextDisabled

升级前：需要手动查找替换，耗时2~3周



升级后：通过 Token 直接全局更新，最快半小时



4 研发结果更加精准

只要确保正确的使用 Token，设计能够高效安全的还原落地

不使用 Design Token

```
import React from 'react'  7.3k (gzipped: 3k)

const App: React.FC = () => (
  <div
    style={{
      backgroundColor: '#f0f9ff',
      padding: '16px',
      borderRadius: '2px',
      color: '#2673dd',
      fontSize: '14px',
    }}
  >
```

使用 Design Token

```
const App: React.FC = () => {
  const { token } = useToken()

  return (
    <div
      style={{
        backgroundColor: token.colorPrimaryBg,
        padding: token.padding,
        borderRadius: token.borderRadius,
        color: token.colorPrimaryText,
        fontSize: token.fontSize,
      }}
    >
```

Thank You

