

UX In Motion

by lefayzhang

1.motion带来什么？

The image displays three screenshots of UI design guidelines pages, each featuring a large, stylized letter (G, A, M) composed of geometric shapes.

- Google Material Design Motion:** Shows the "Understanding motion" section. It includes a sidebar with "MATERIAL DESIGN" and "Motion" sections like Sound, Iconography, Shape, and Speed. The main content discusses motion's role in UI expressiveness and provides a "Principles" section with three cards: a red one with a black triangle and circle, a pink one with concentric circles, and a pink one with a plus sign.
- Alibaba Ant Motion:** Shows the "Motion Design" section. It features a sidebar with "CONTENTS" and "Principles" sections. The main content discusses animation specification and components of Ant Design, with a blue circular motion graphic and two call-to-action buttons.
- Medium UX in Motion:** Shows a post titled "Creating Usability with Motion: The UX in Motion Manifesto". It includes a sidebar with "CREATING USABILITY WITH MOTION" and "THE 12 PRINCIPLES OF UX IN MOTION". The main content lists 10 principles: Easing, Overlay, Cloning, Offset & Delay, Parenting, Transformation, Value change, Masking, Dimensionality, and Dolly & Zoom. It also includes a bio for Issara Willenskomer and a link to uxinmotion.net.

“ motion让UI拥有更强的表现力
并易于使用 ”

Google Material design-Motion

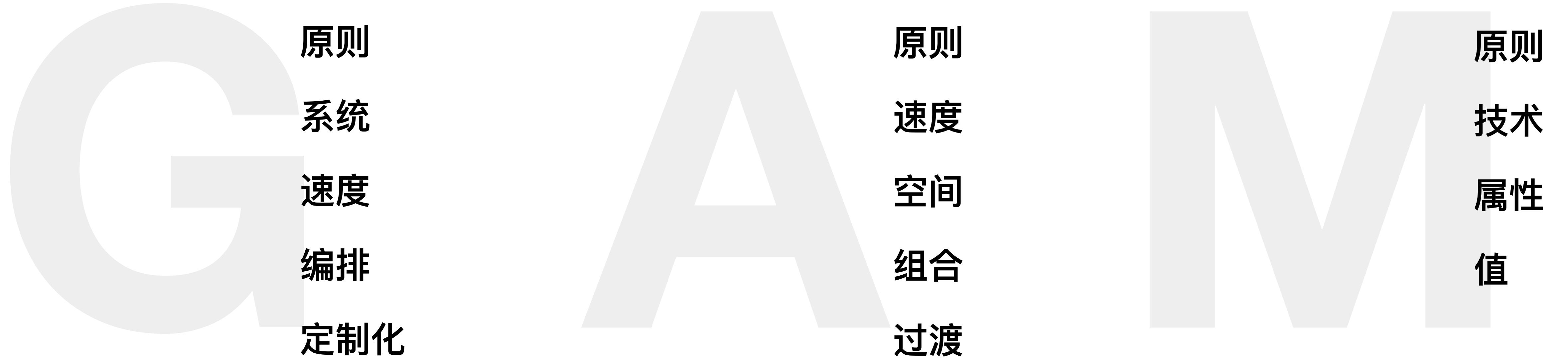
“ 界面动效能加强用户认知
且增加活力 ”

Alibaba Ant Motion

“ 通过motion创造易用性 ”

Medium UX in Motion

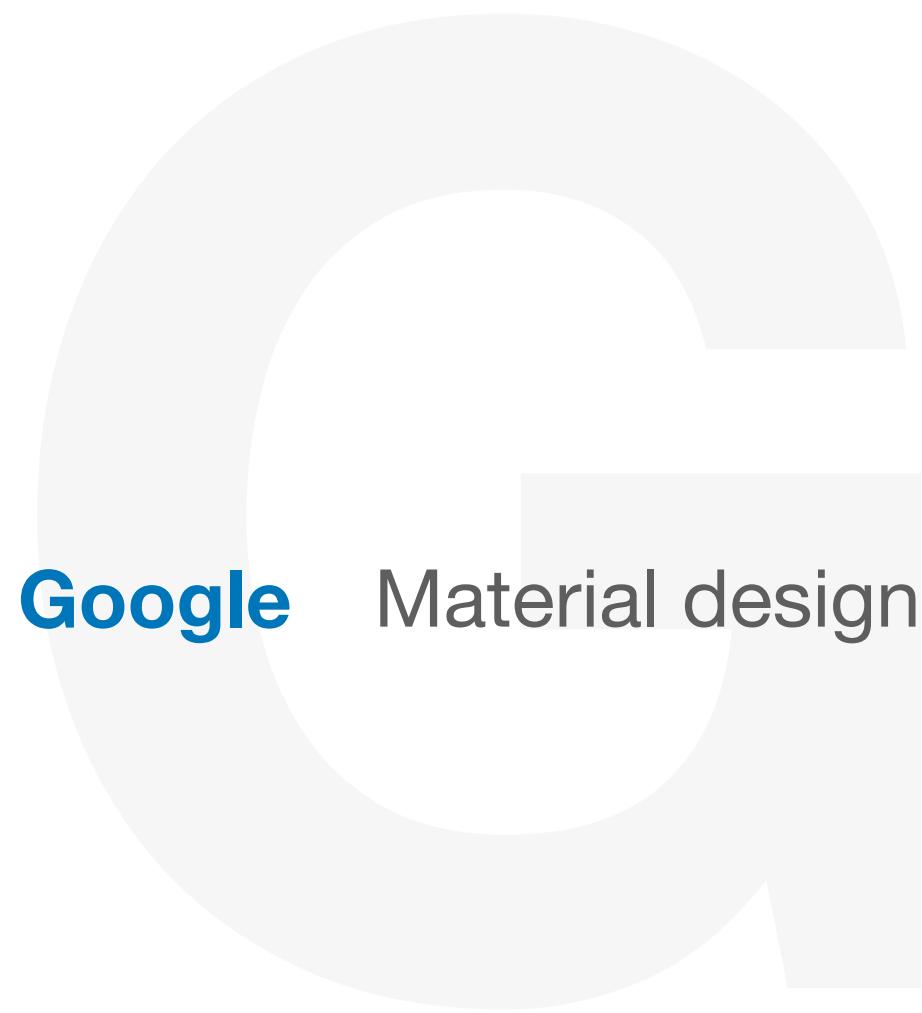
2. 对motion的理解与侧重点...



Google Material design-Motion

Alibaba Ant Motion

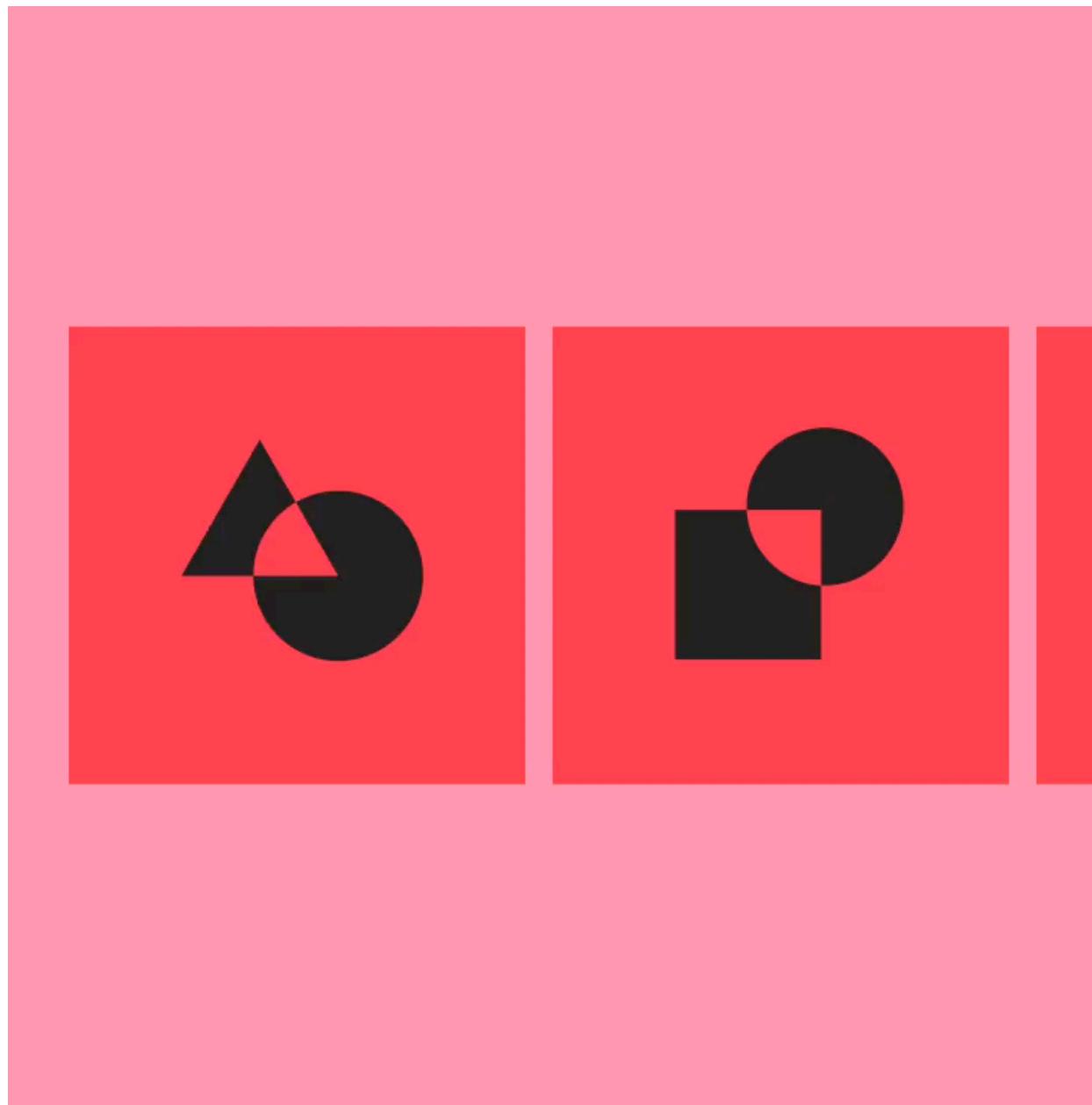
Medium UX in Motion



原则
系统
速度
编排
定制化

体系简单
细节合理

原则



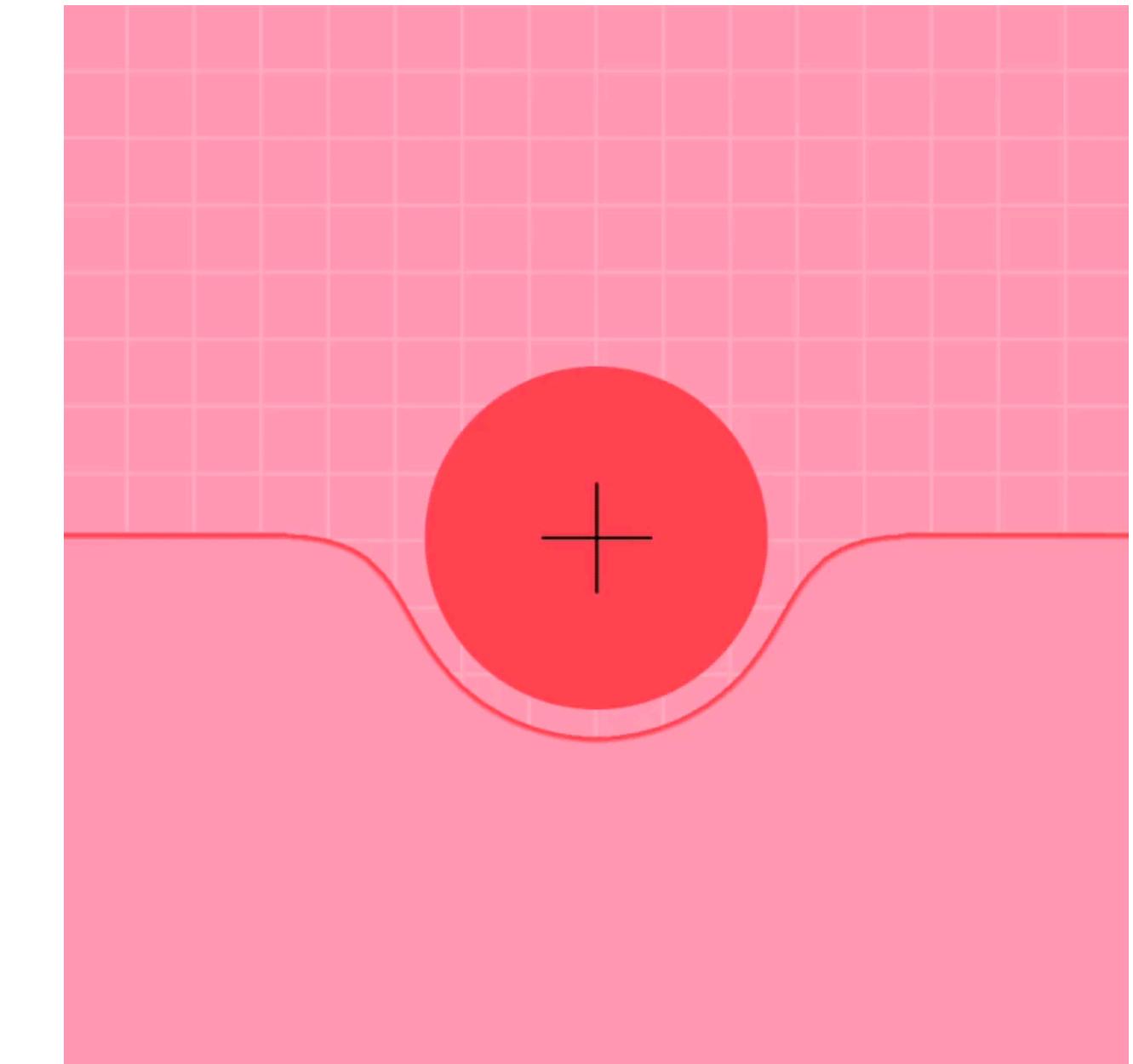
传递丰富的信息

-通过强调、展示【元素】【操作】
【操作结果】之间的关系让用户更好的
理解功能传递的信息



聚焦在重点上

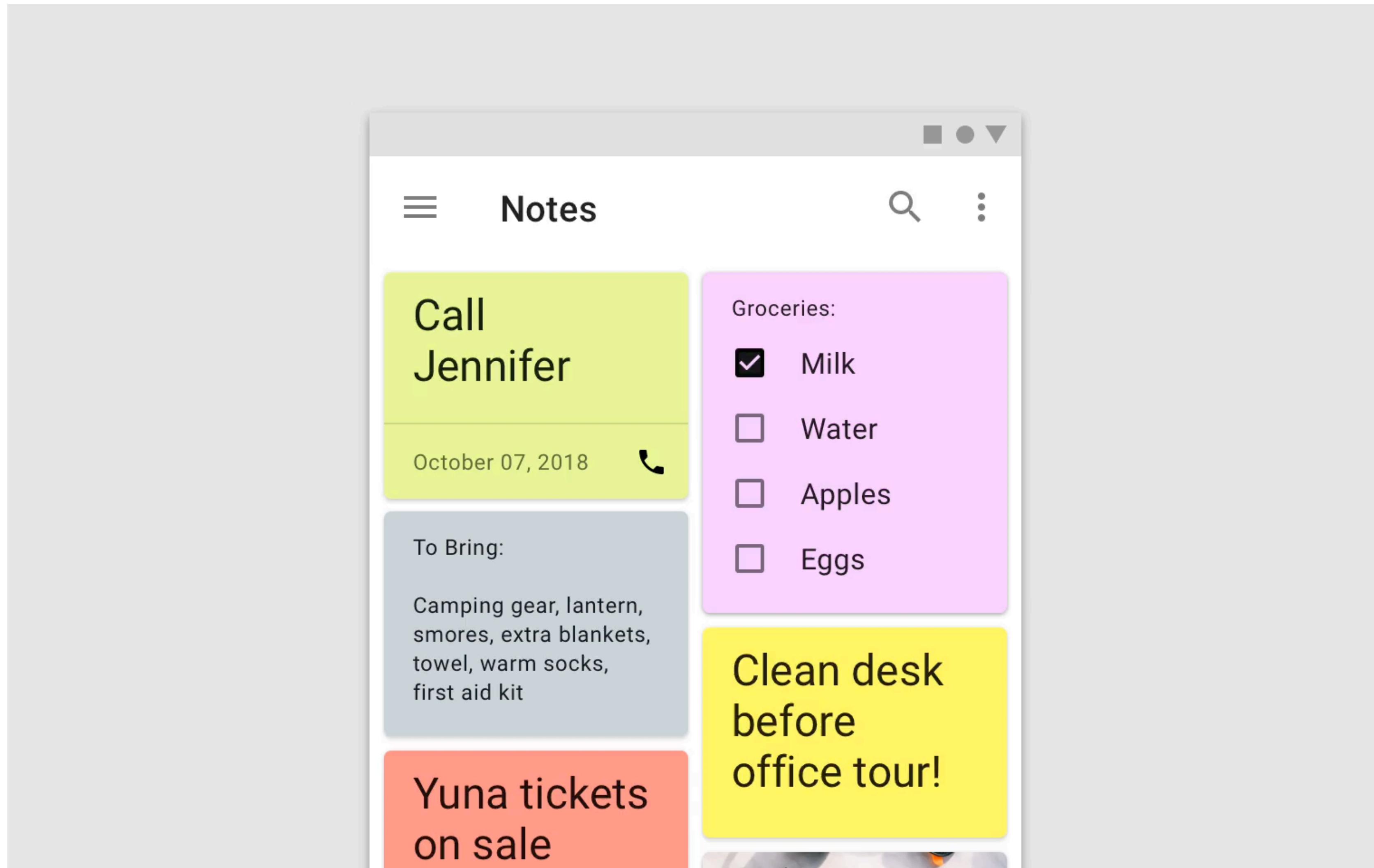
-将注意力聚焦在真正重要的功能点上
-且不会造成不必要的干扰



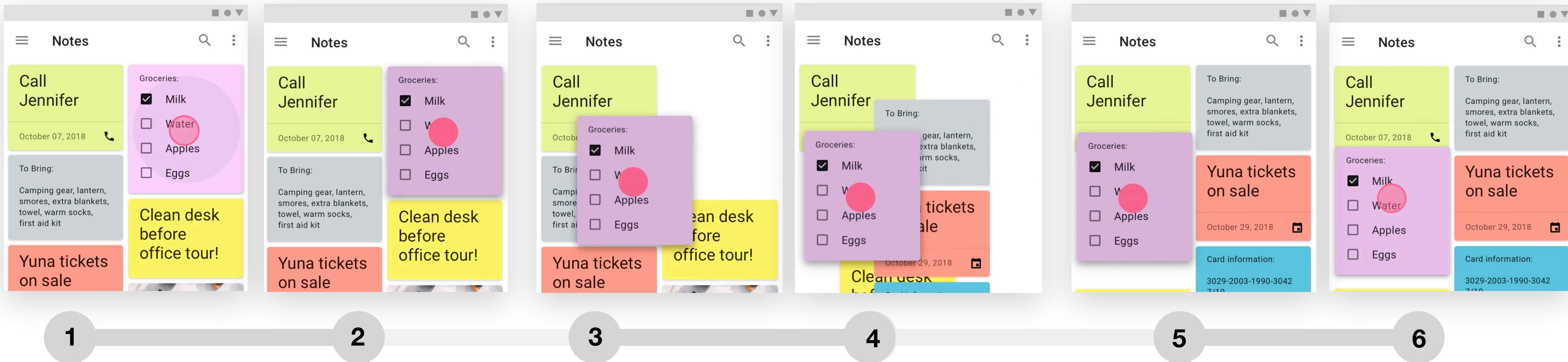
极具表现力

-凸显操作流程中的关键时刻
-强化互动亮点
-表达品牌风格

eg: 操作反馈中的motion (选中并拖拽调整一张卡片的位置)



通过motion使【选中】 【拖动定位】 【释放】 3阶段的操作清晰易懂



1

点击：

点击处扩散的圆

2

长按选中：

扩散后形成遮罩层

3

拖拽中：

如没有触发区域改变
的范围时无反馈

4

定位中：

到达可放置范围，
其他模块自动发生位移

5

确认定位：

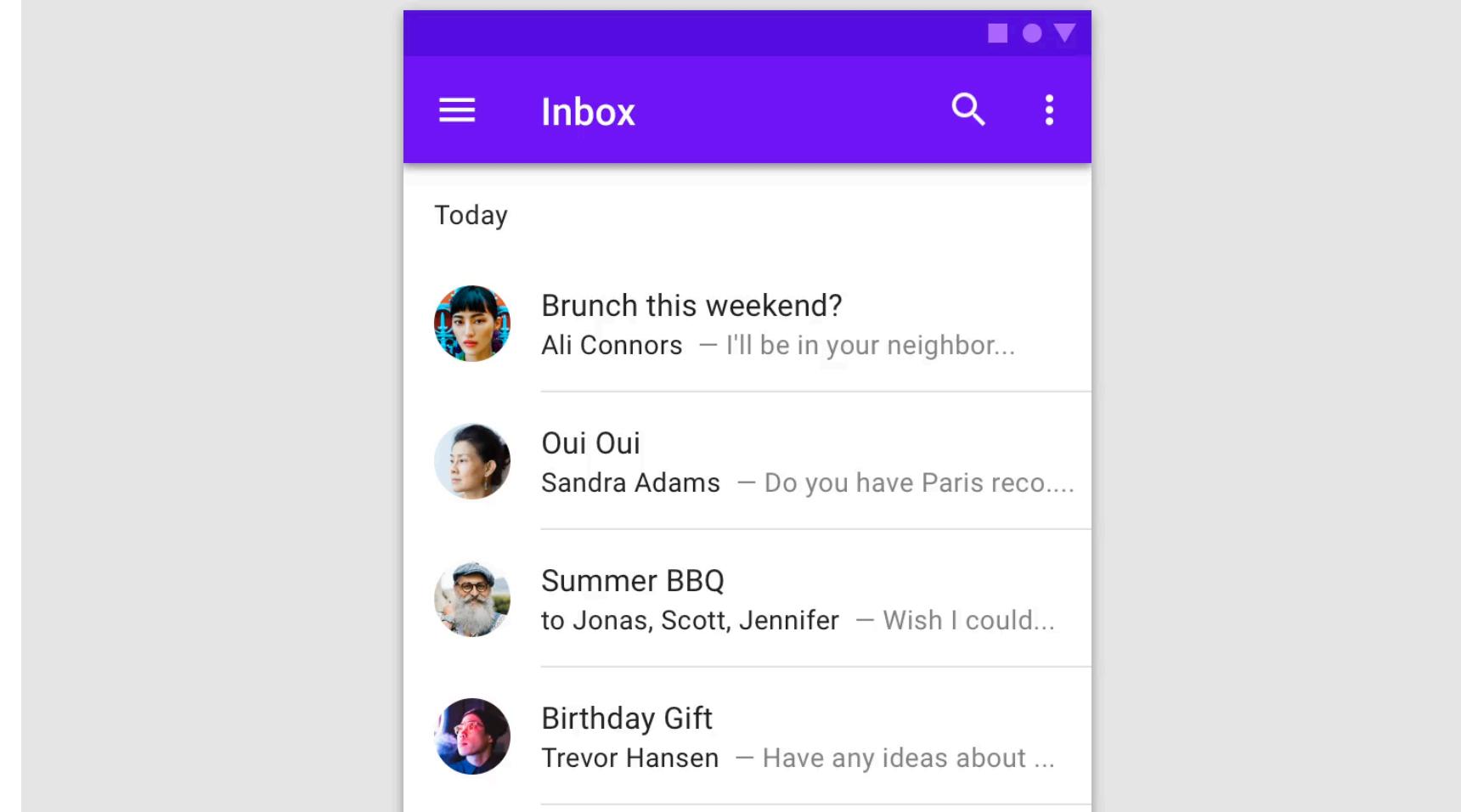
悬停确认位置，
其他模块完成位移

6

释放点击：

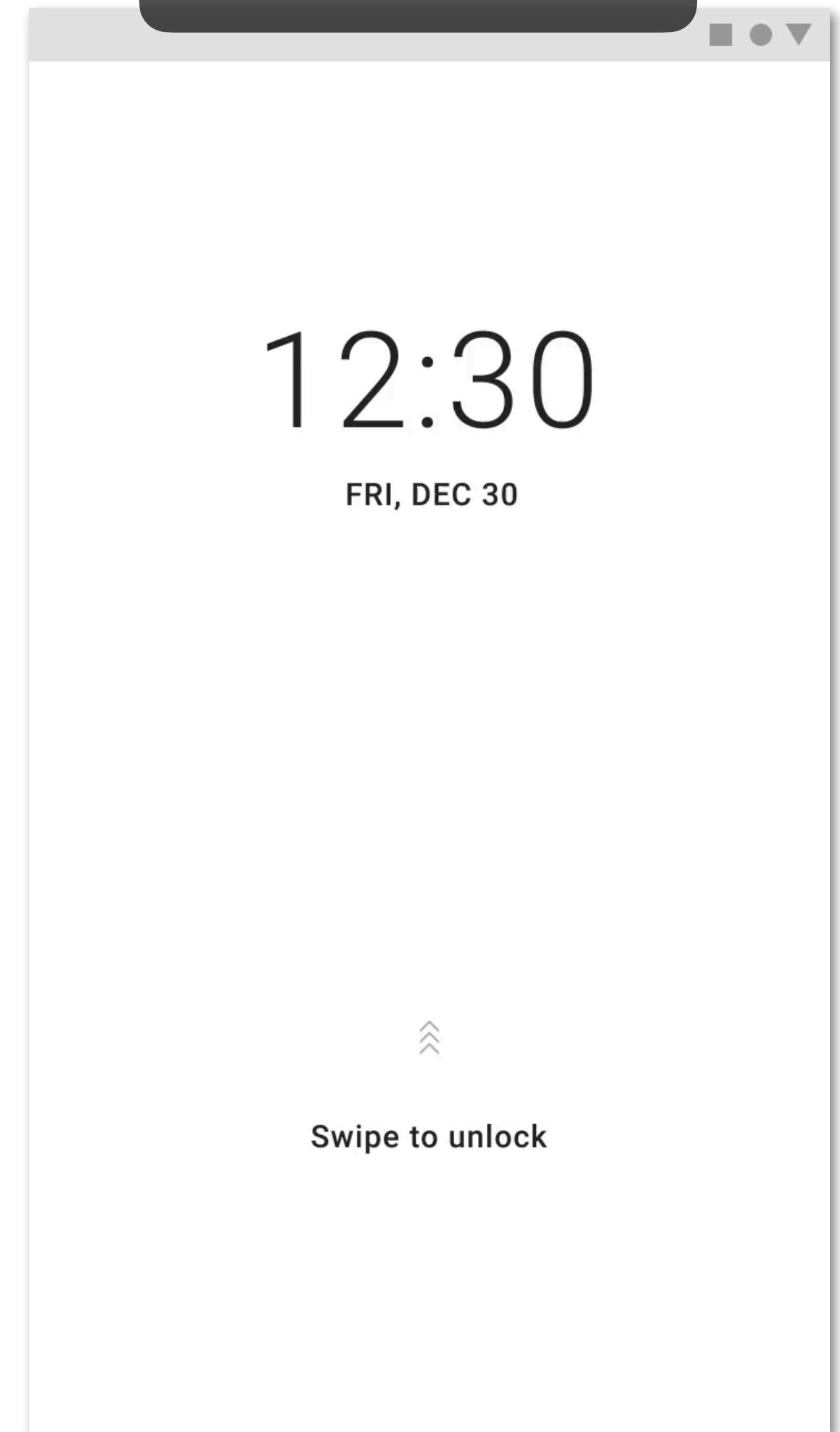
被拖拽模块
自动适配位置
遮罩层逐渐消失

展示列表（操作）与内容页（结果）之间的关系

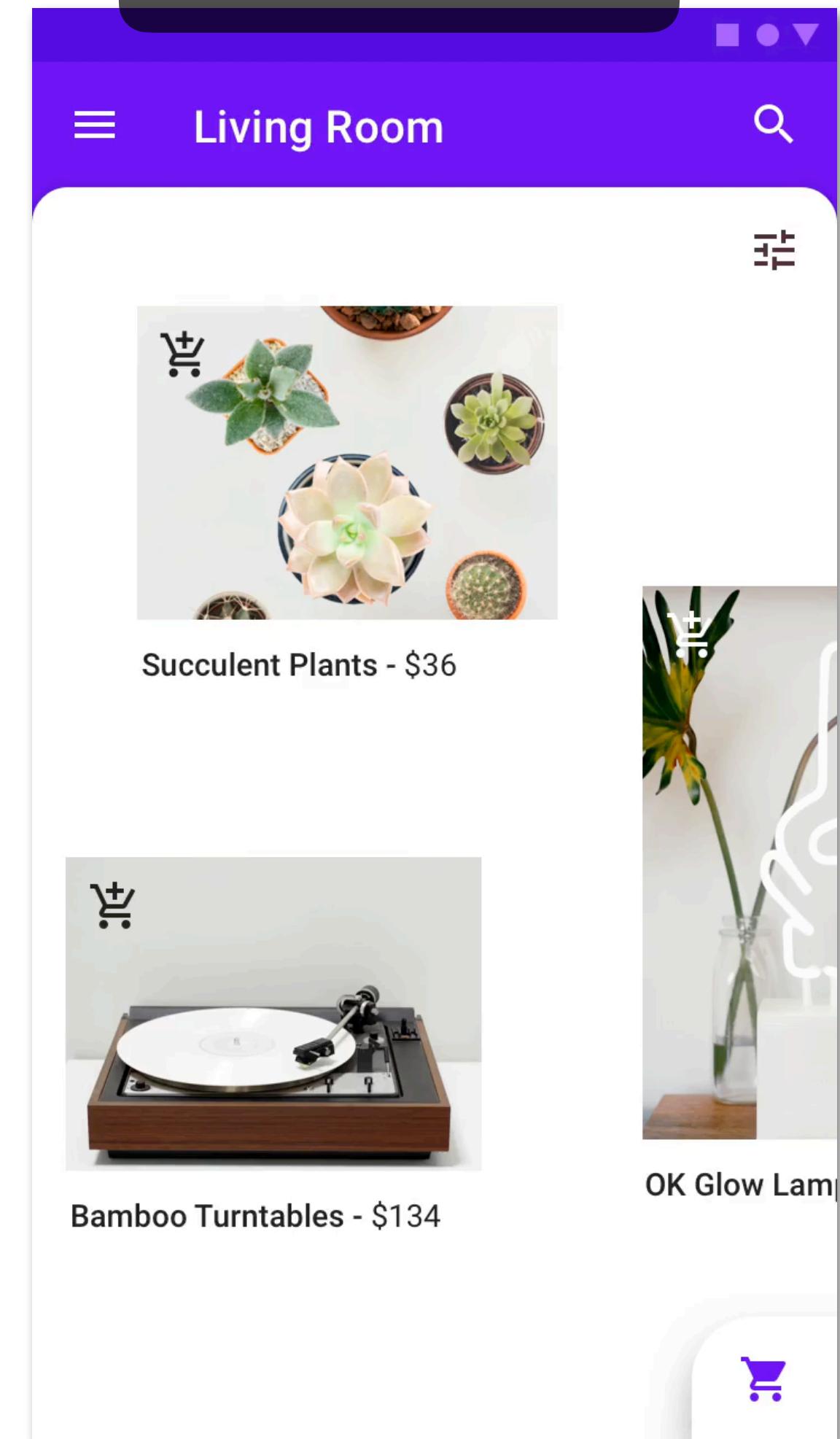


增强图标含义

功能指引



功能指引



传递品牌色彩

motion系统

过渡展示关系

“Motion system是一系列的过渡模式；它可以帮助用户理解app，并提供良好的app导览效果”

Material design的motion system由4种模式组成，这几种模式旨在通过加强UI元素之间的关系来帮助用户理解以及更好的导览app。

这四种过渡模式是：

① 容器转换 Container transform

强调容器之间的衔接关系

② 共享轴 Shared axis

强调关联的级别

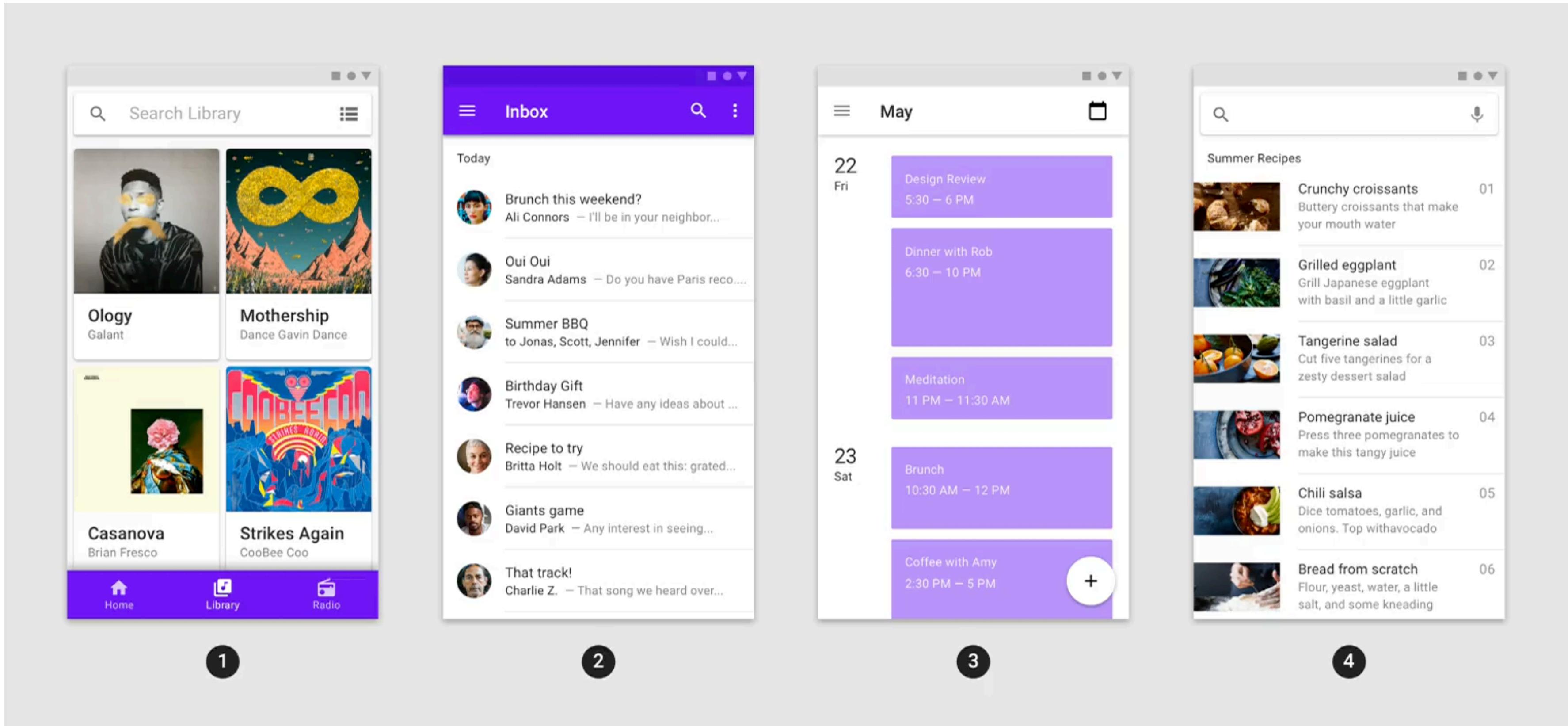
③ 淡入 Fade through

强调无关联的进入

④ Fade

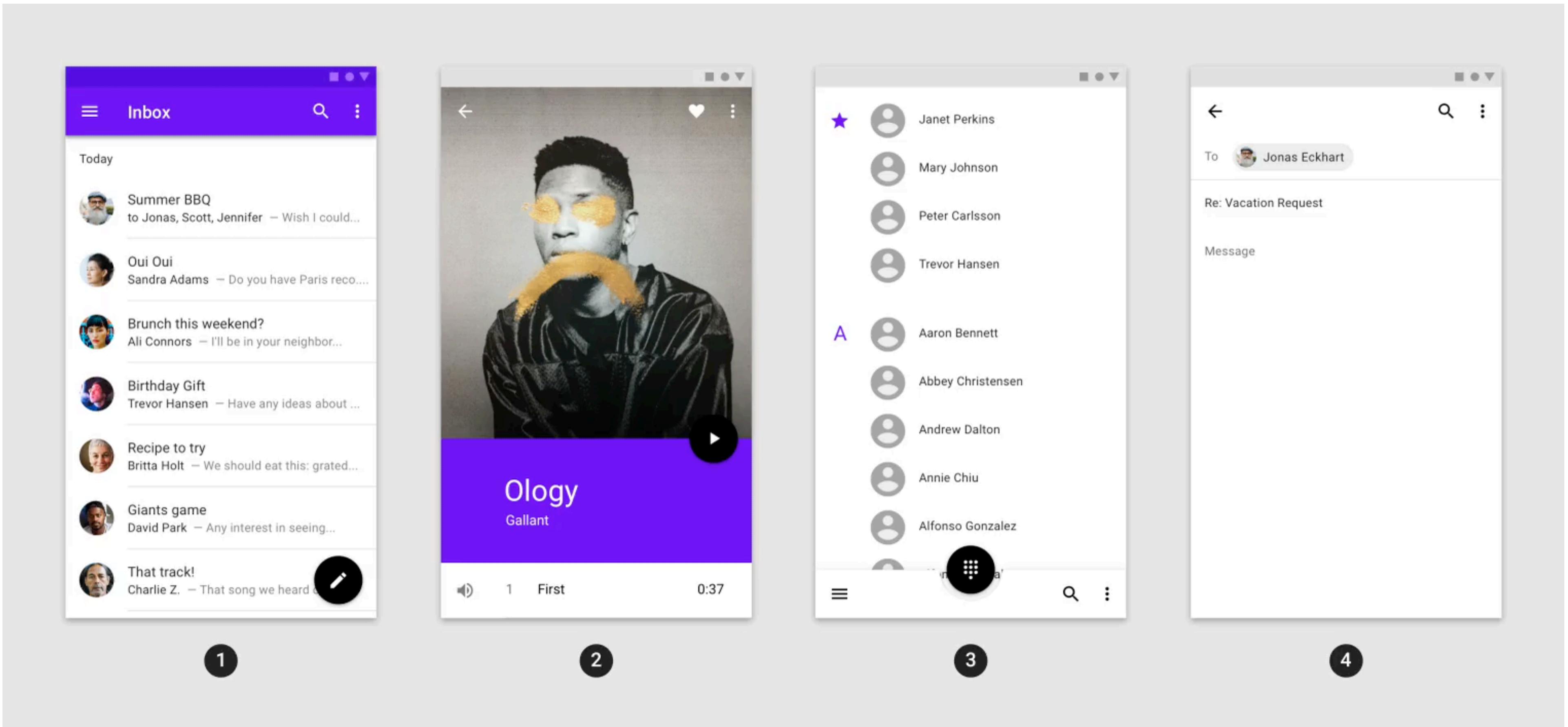
强调元素入场&退场

1 容器转换Container transform



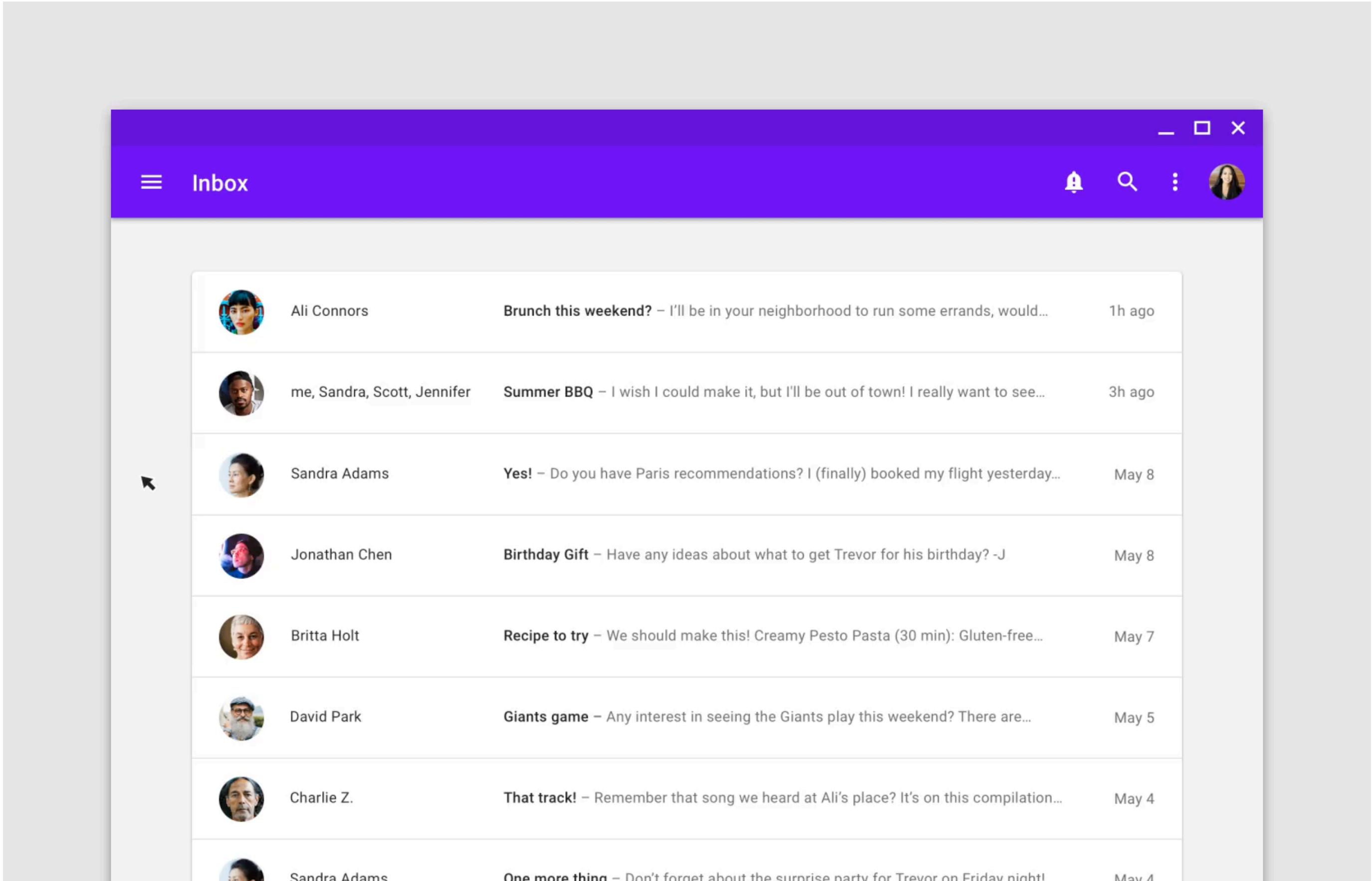
容器转换模式设计用于包含容器的UI元素之间的转换。此模式在两个UI元素之间创建可见连接。
通过将一个元素无缝转换为另一个元素，可以增强两个元素之间的关系。
例如，当卡片转换为详细信息页面时，通过运动中的焦点让用户识别该详细信息页面是卡片的扩展版本。

1 容器转换Container transform



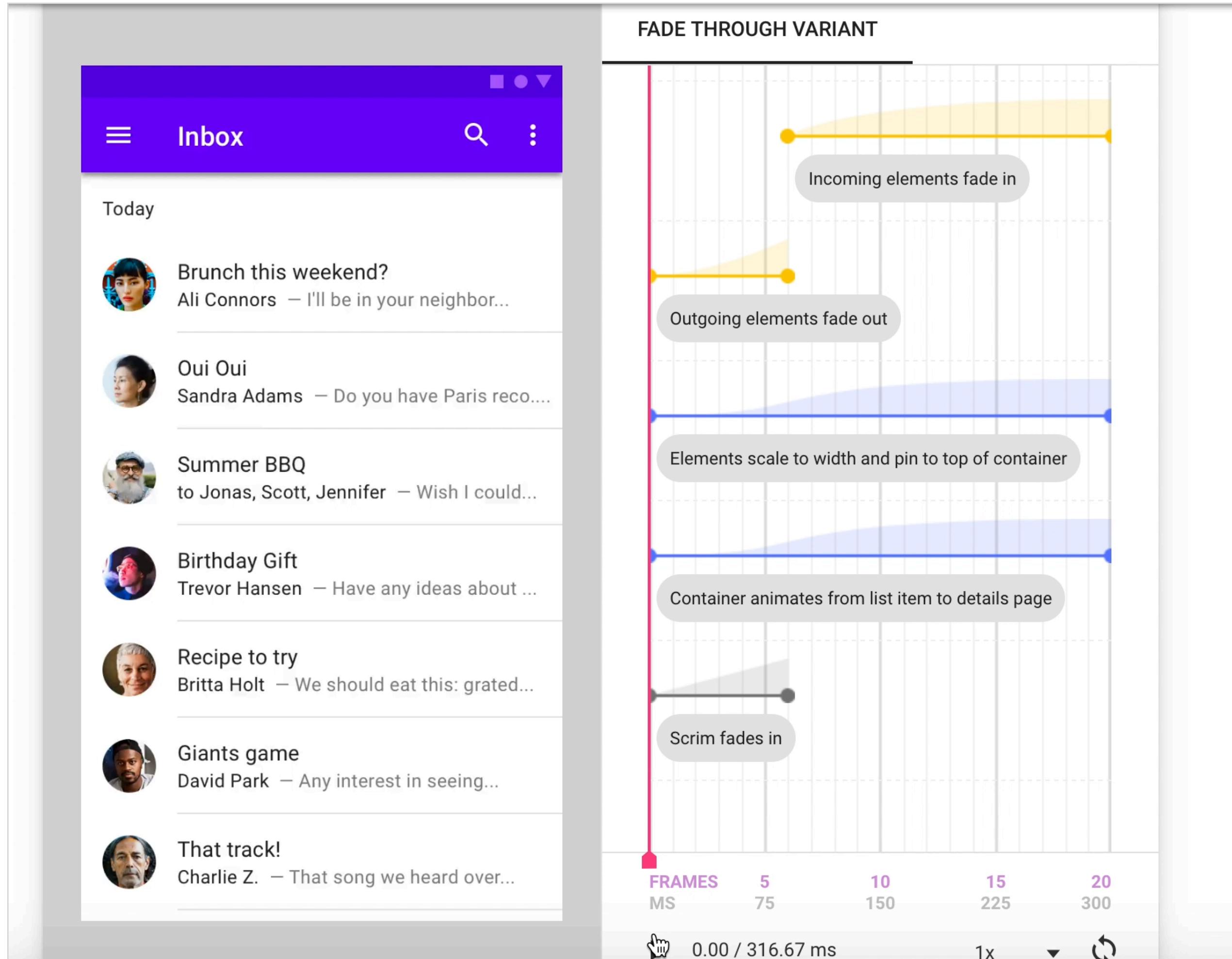
容器转换模式也可以应用于仅占据屏幕一部分且不会扩展为全屏视图的转换。

1 容器转换Container transform



桌面应用案例：容器转换模型让列表和内容之间的关联更紧密

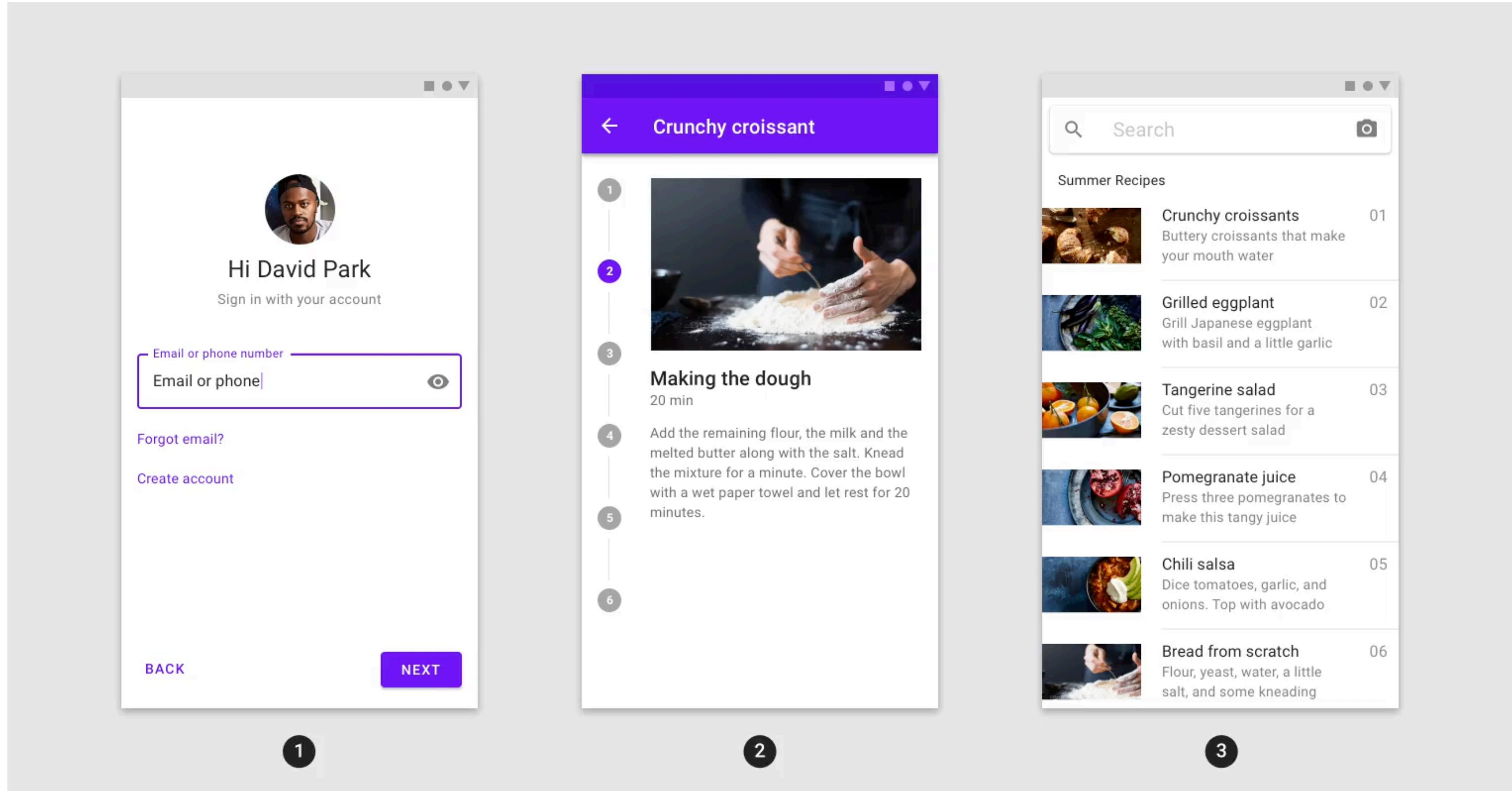
1 容器转换Container transform



体验demo：

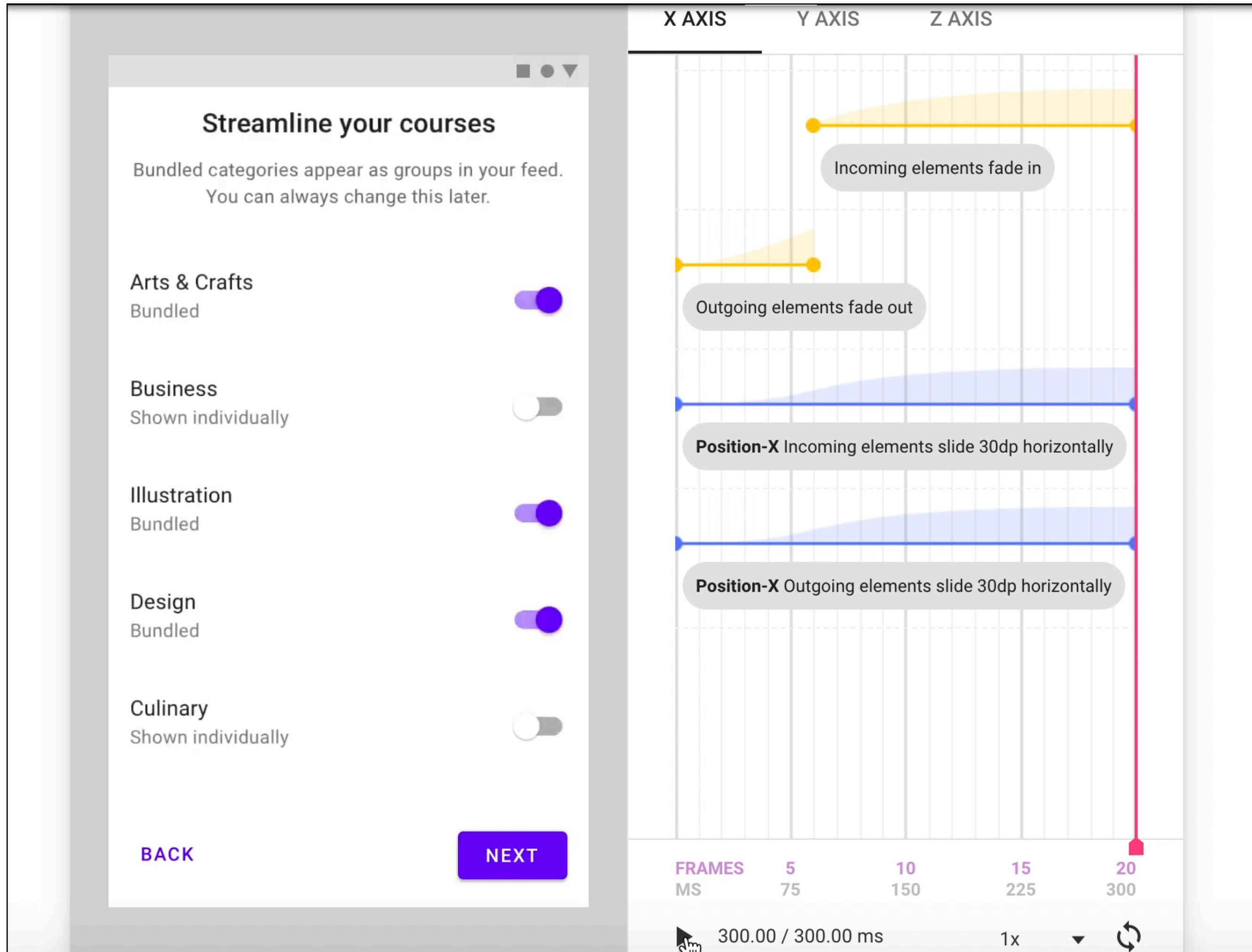
在过渡期间，容器的尺寸，位置和形状会串联起来，而容器则充当持久元素。容器的内容固定在容器的顶部边缘，并缩放以匹配其变形时的宽度。此外，也经常搭配【淡入】【淡出】效果来更好的展示容器转换；

2 共享轴 Shared axis



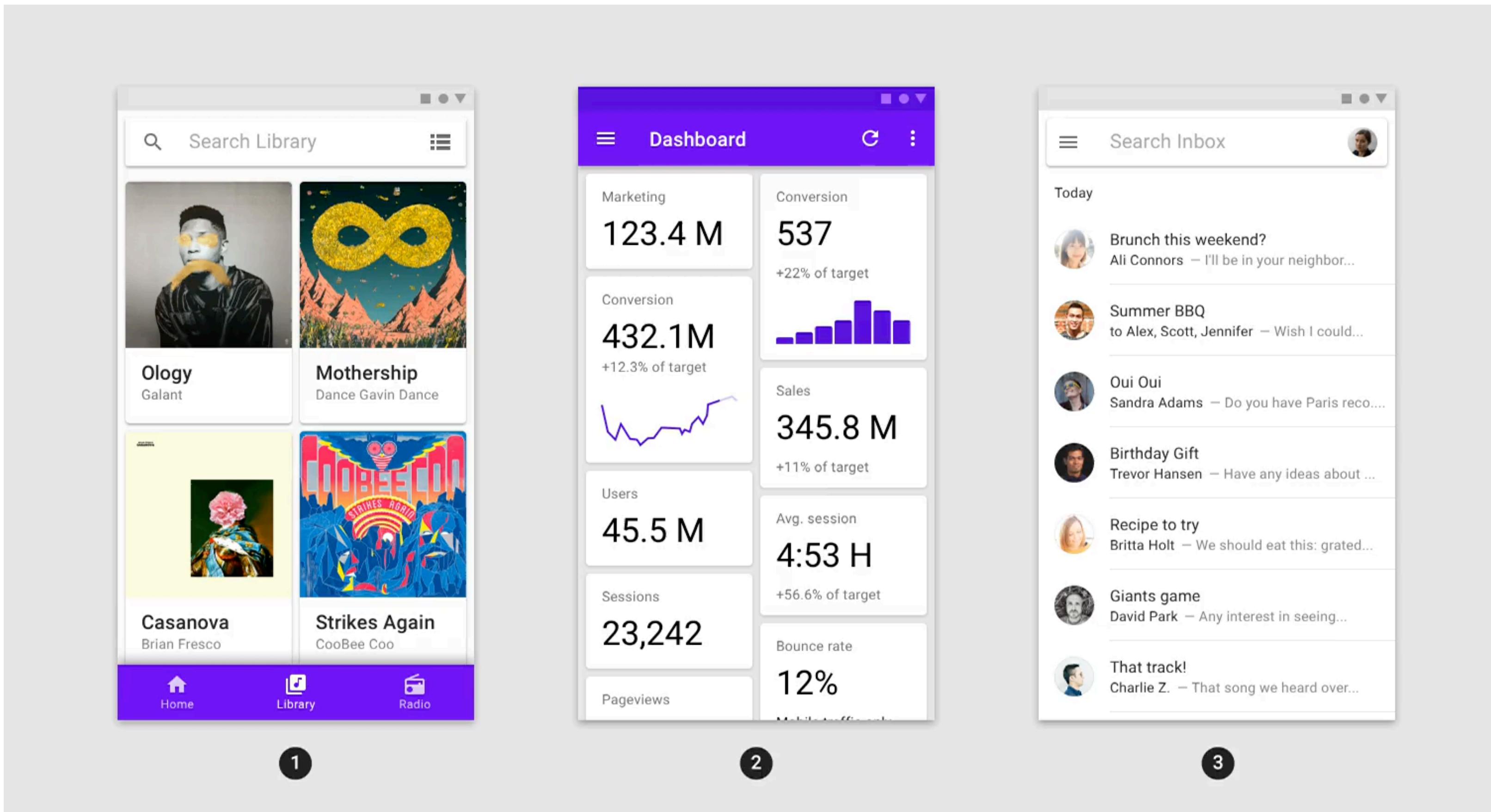
共享轴模式用于具有空间或导航关系的UI元素之间的过渡。该模式在x, y或z轴上使用共享的空间轴变换来加强元素之间的关系。例如（1）中的登陆流程，左右侧按钮触发的操作分别从左、右横向切换，表明元素间的相关与平级。

2 共享轴 Shared axis



体验demo：

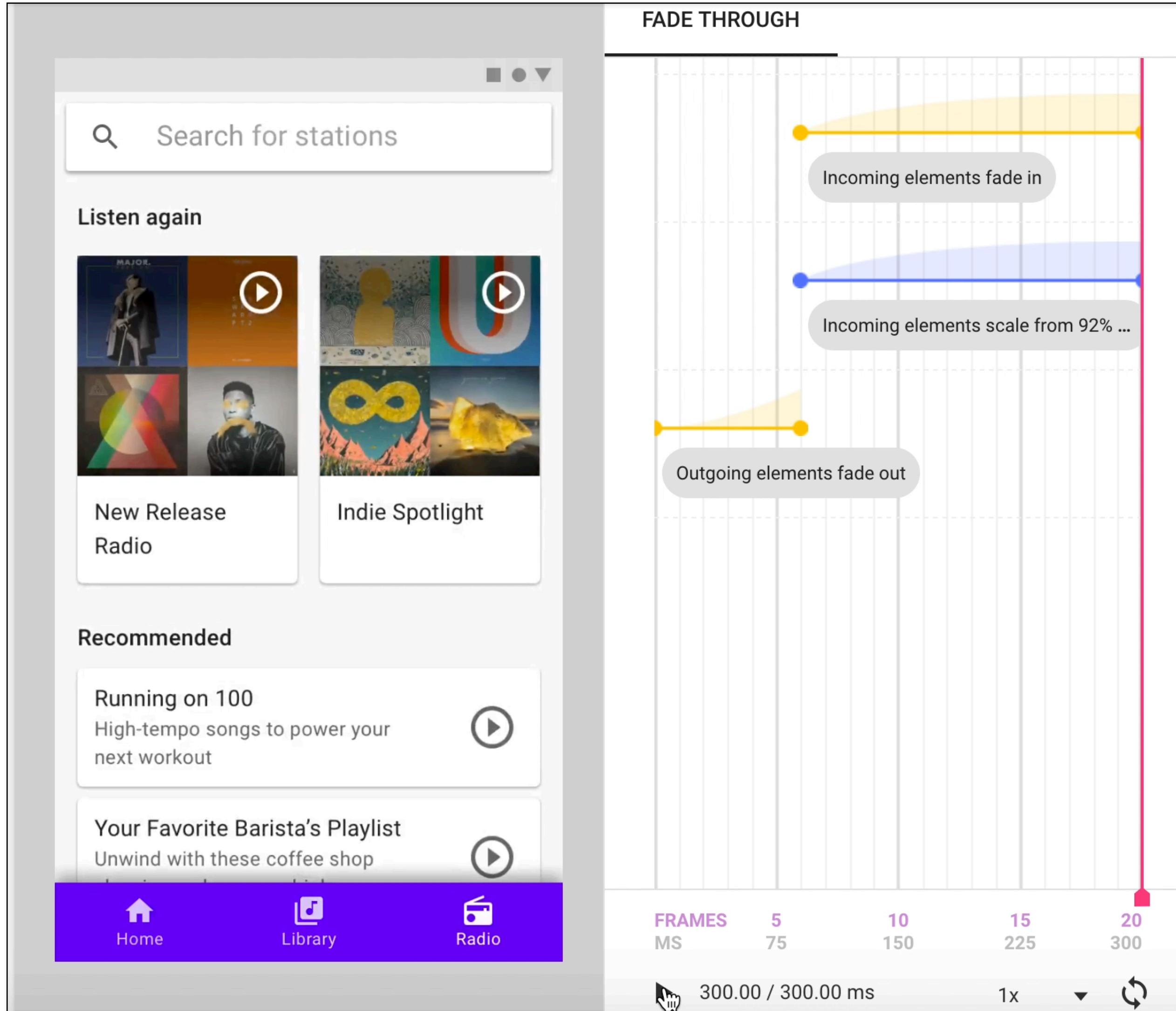
3 淡入Fade through



淡入模式用于彼此之间没有密切关系的UI元素之间的过渡。

例如，通过点击底部导航栏触发的过渡使用淡入模式。淡入是最佳选择，因为底部导航中的目的地通常被分组为可能彼此不相关的主要任务。此外，淡入不会让用户以为他们可以在目的地（页面/tab）之间水平滑动。

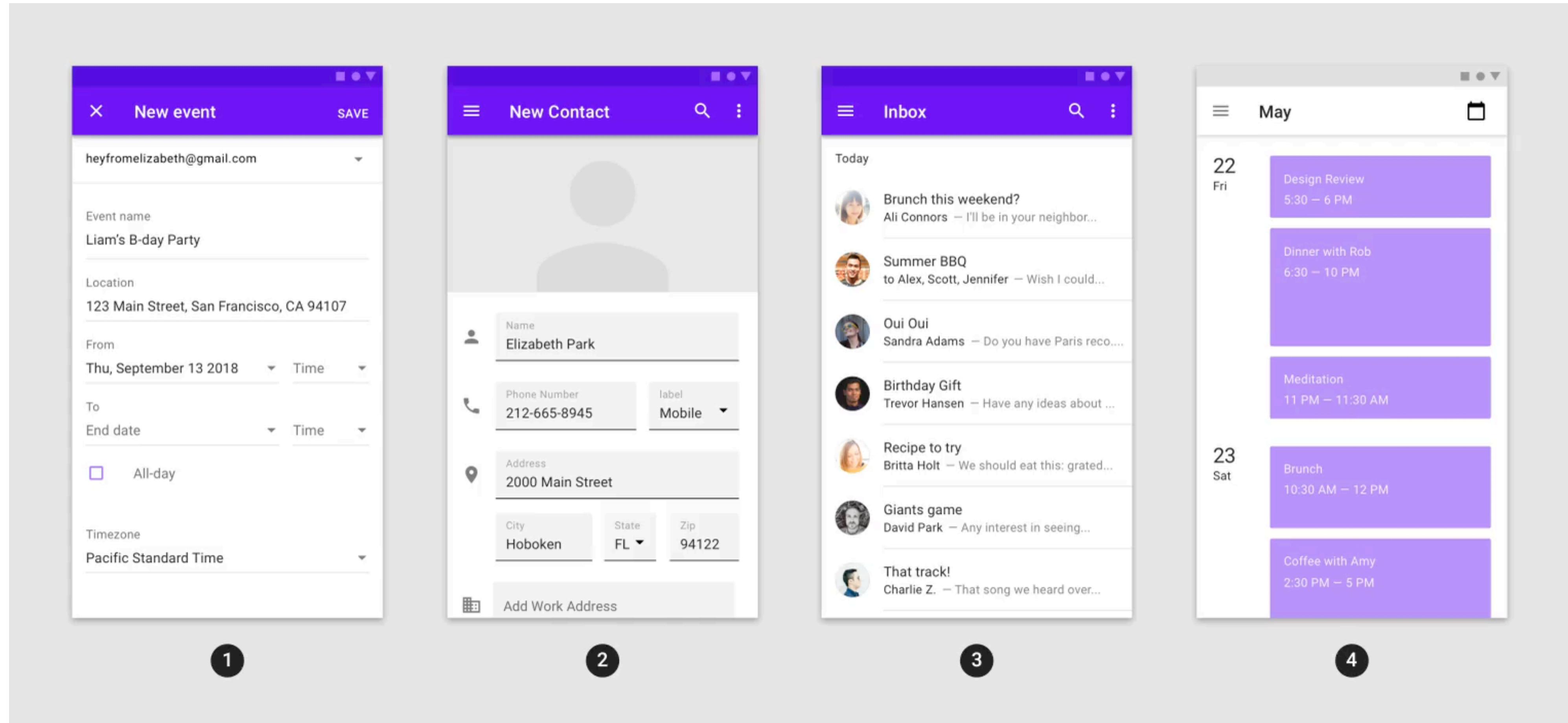
3 淡入Fade through



体验demo：

在淡入过渡中，输出元素首先淡出。接下来，传入元素逐渐淡入，同时将整体大小从92%缩放到100%。元素缩放比例从92%而不是0%开始，以避免过度注意过渡。此处比例仅应用于传入的元素，以便在旧内容上强调新内容。

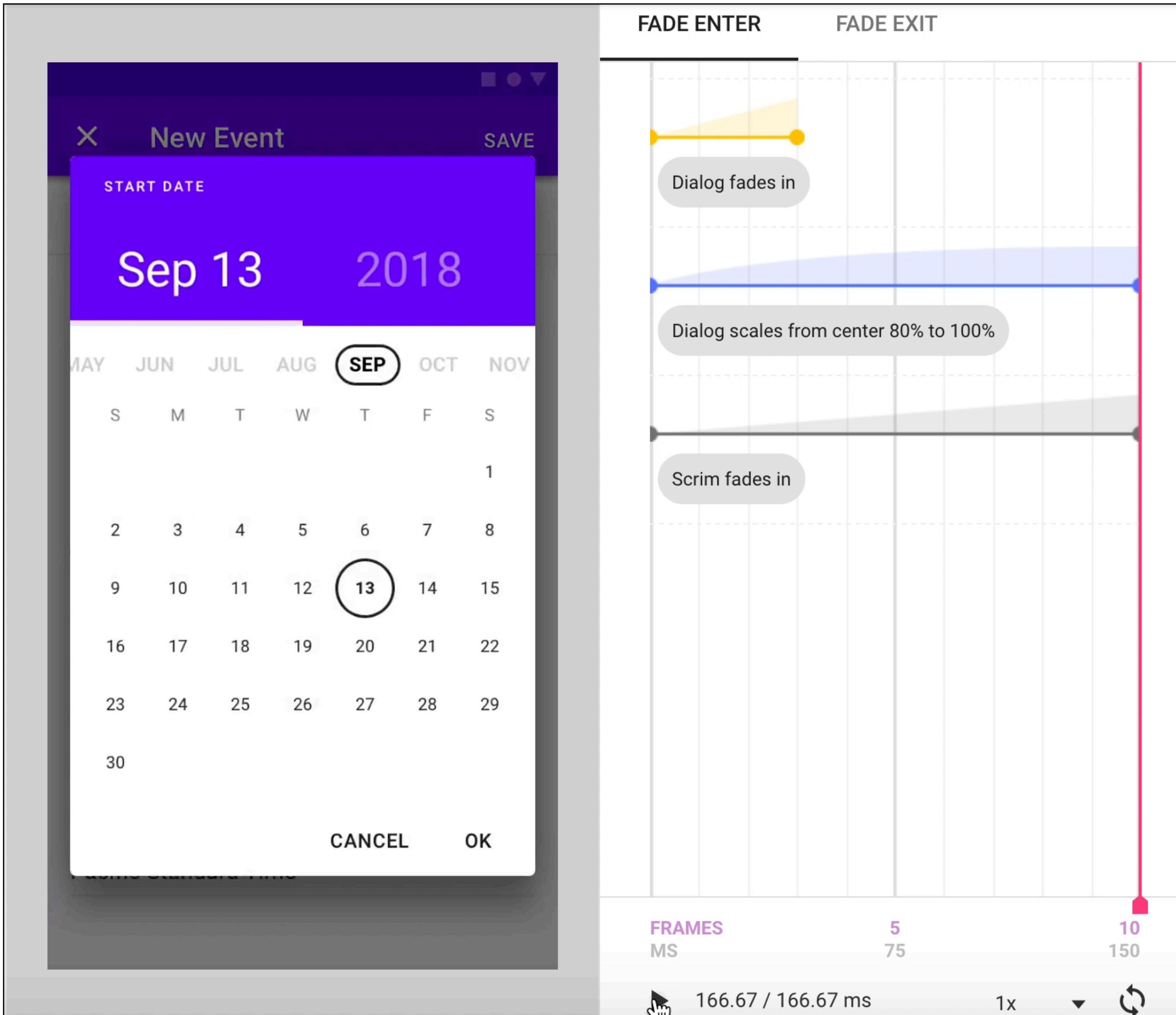
4 Fade



案例：【1-对话框】 【2-顶部菜单】 【3-底部snackbar】 【4-FAB按钮】

【淡入淡出】运动模式用于在屏幕范围内，进入或退出的UI元素，例如从屏幕中心淡入和淡出视图的对话框。

4 Fade

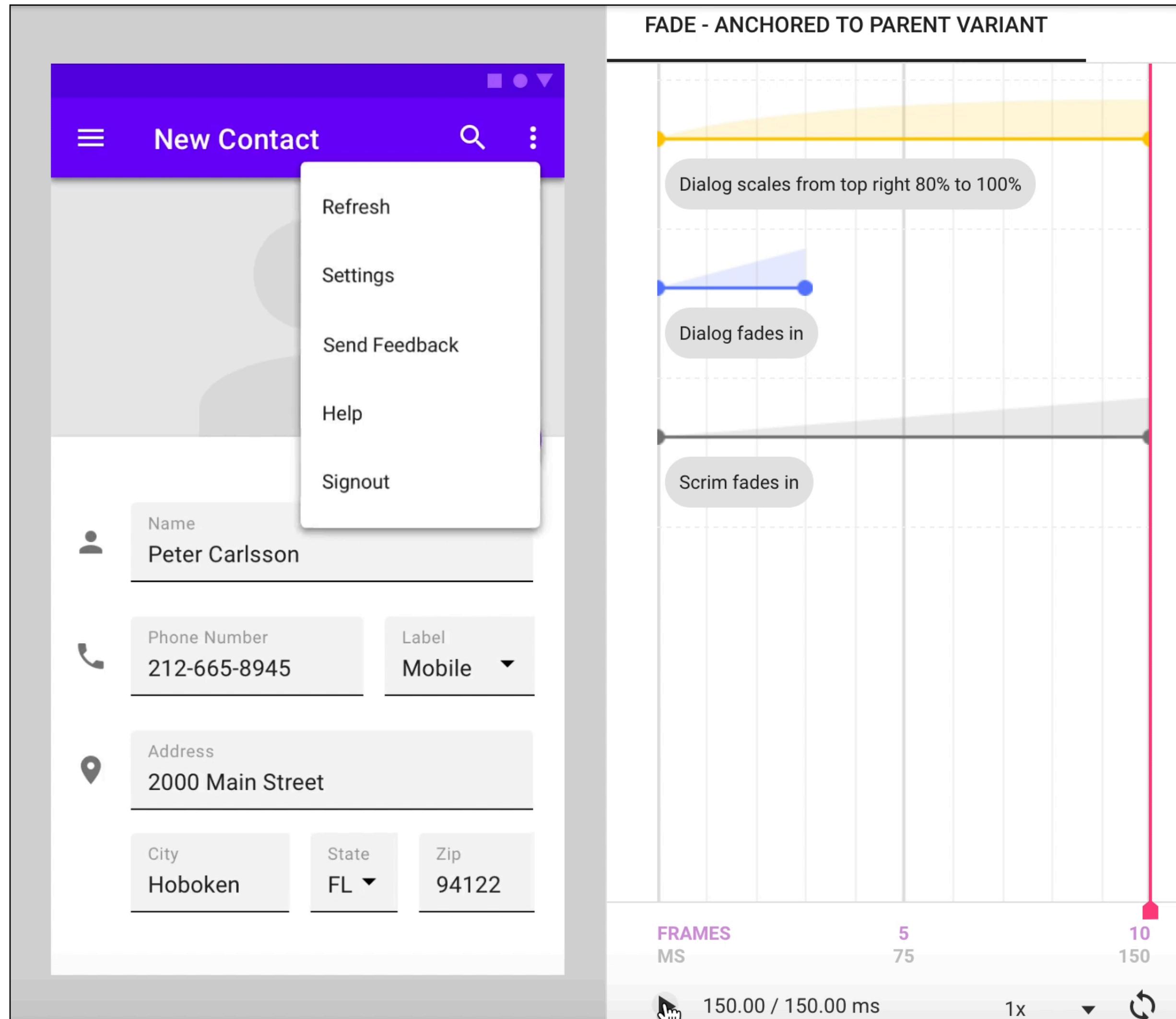


体验demo：

元素输入时，会使用淡入淡出的方式，整体大小从80%（出场时）放大到100%。从80%（而不是0%）开始是为了避免过度的视觉刺激。元素退出时，元素只会淡出。

为了将重点放在新内容（输入元素）上而不是旧内容（现有元素）上，比例动画（80%~100%）仅应用于输入的元素。

4 Fade



体验demo：

比例变换+锚点定位

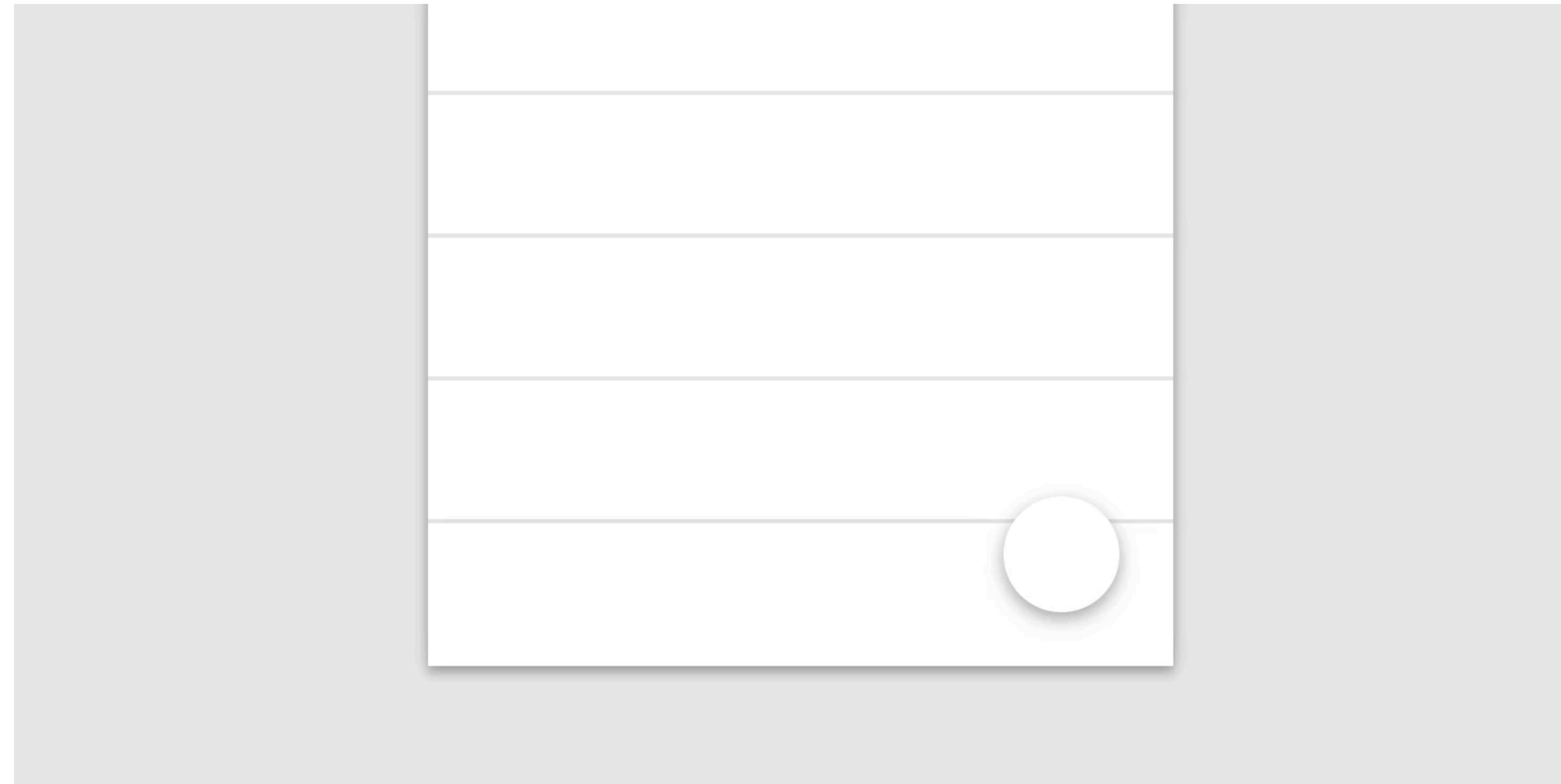
默认情况下，淡入时输入元素会从其中心点开始缩放。但在某些情况下，可以将缩放锚点从元素的中心重新放置到元素内的另一个位置。

例如，当点击图标（父组件）触发菜单（子组件）时，菜单的定位点可以位于菜单的右上角，以便在过渡期从点击图标处缩放。这种变化在视觉上将两个组件（父&子）结合在一起。

如何从四种过渡模型中选择适合的？

① 容器转换Container transform ② 共享轴Shared axis ③ 淡入Fade through ④ Fade

1. 【判断容器本身】这个永久性的容器是过渡的一部分吗？



容器是用来表示封闭区域的形状，例如按钮，列表，卡片表面。如果在过渡中有一个容器是一个持久性元素，请选择容器转换模式。

案例case:

【容器转换模式】适用于永久性容器，例如【FAB按钮转换为工作表弹窗】或【列表项转换为详情页面】。

如何从四种过渡模型中选择适合的？

① 容器转换Container transform ② 共享轴Shared axis ③ 淡入Fade through ④ Fade

2. 【判断元素之间的关系】UI元素之间是否存在空间或导航关系？



如果过渡中的元素没有持久性容器，则可能存在需要强调的空间关系，例如垂直或水平布局中的元素。或者，元素之间可能存在导航关系，例如在同级之间或应用程序中的关卡之间移动。 **如果元素之间存在空间或导航关系，请选择【共享轴】模式。**

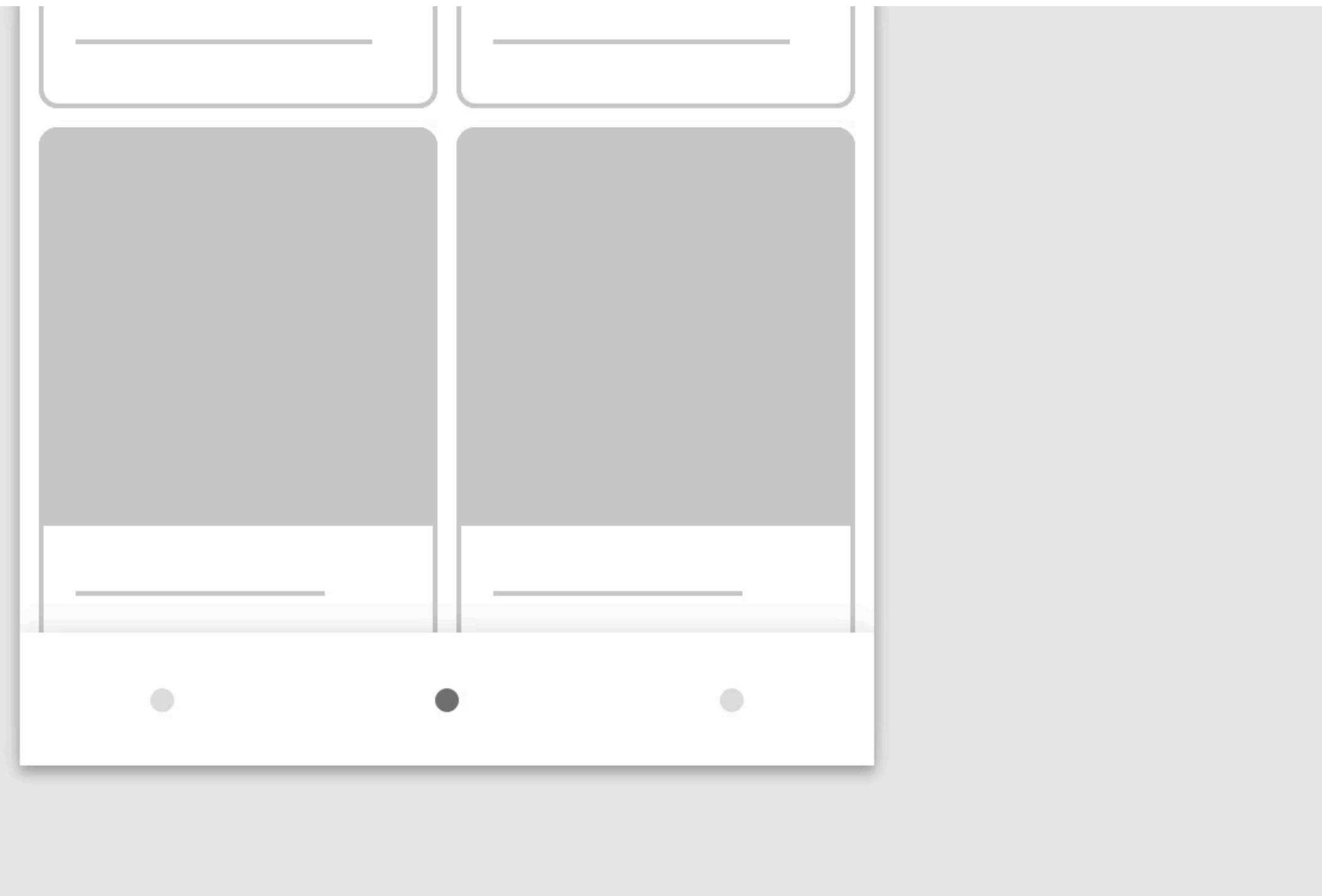
案例case:

【共享轴】模式可以应用于具有空间或导航关系的UI元素，例如在登陆流程中进行【上一步】【下一步】的指引。

如何从四种过渡模型中选择适合的？

① 容器转换Container transform ② 共享轴Shared axis ③ 淡入Fade through ④ Fade

3. 【判断元素之间的关系】UI元素之间是否存在空间或导航关系？



当元素之间的关系不明显或不存在关联时，淡入模式更合适。

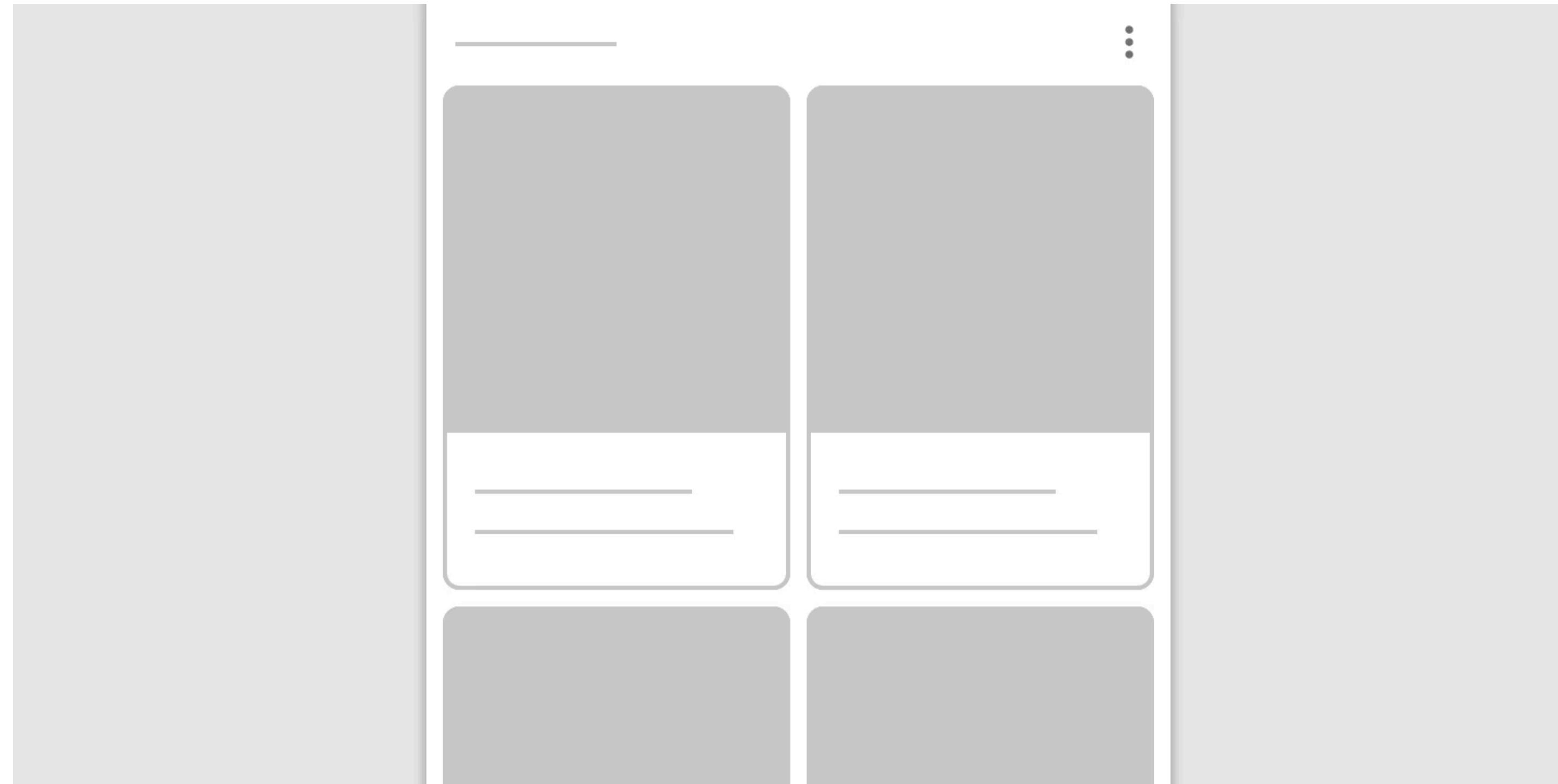
案例case：

彼此之间没有很强关联的UI元素之间可使用【fade through淡入模式】；
例如底部导航中的目标。

如何从四种过渡模型中选择适合的？

① 容器转换Container transform ② 共享轴Shared axis ③ 淡入Fade through ④ Fade

4. 【进场退场】 UI元素是否进入或退出屏幕中？



对于出现进场&退场的元素，适合使用fade
(淡入淡出) 模式；

案例case：

有进出场的UI元素适合使用fade模式，
如【对话框】

速度

通过对速度的控制，让过渡更加顺滑。

速度=持续时间&缓动曲线：

0 控制速度

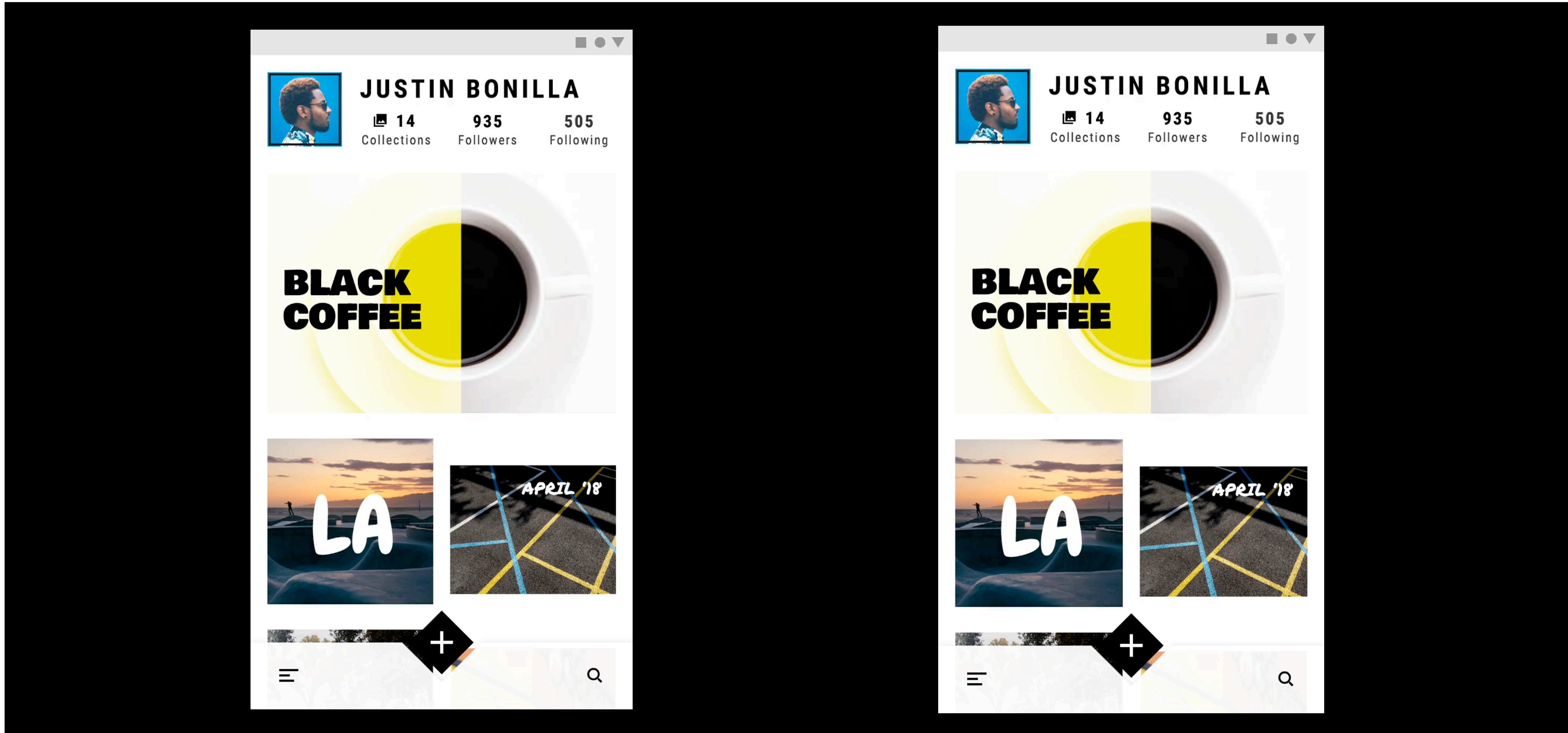
1 持续时间： (1) 退出&关闭 (2) 过渡区域 (小/中/大)

2 缓动曲线Easing (种类： 标准曲线/强调曲线/减速曲线/加速曲线)

情感&映射

0 控制速度

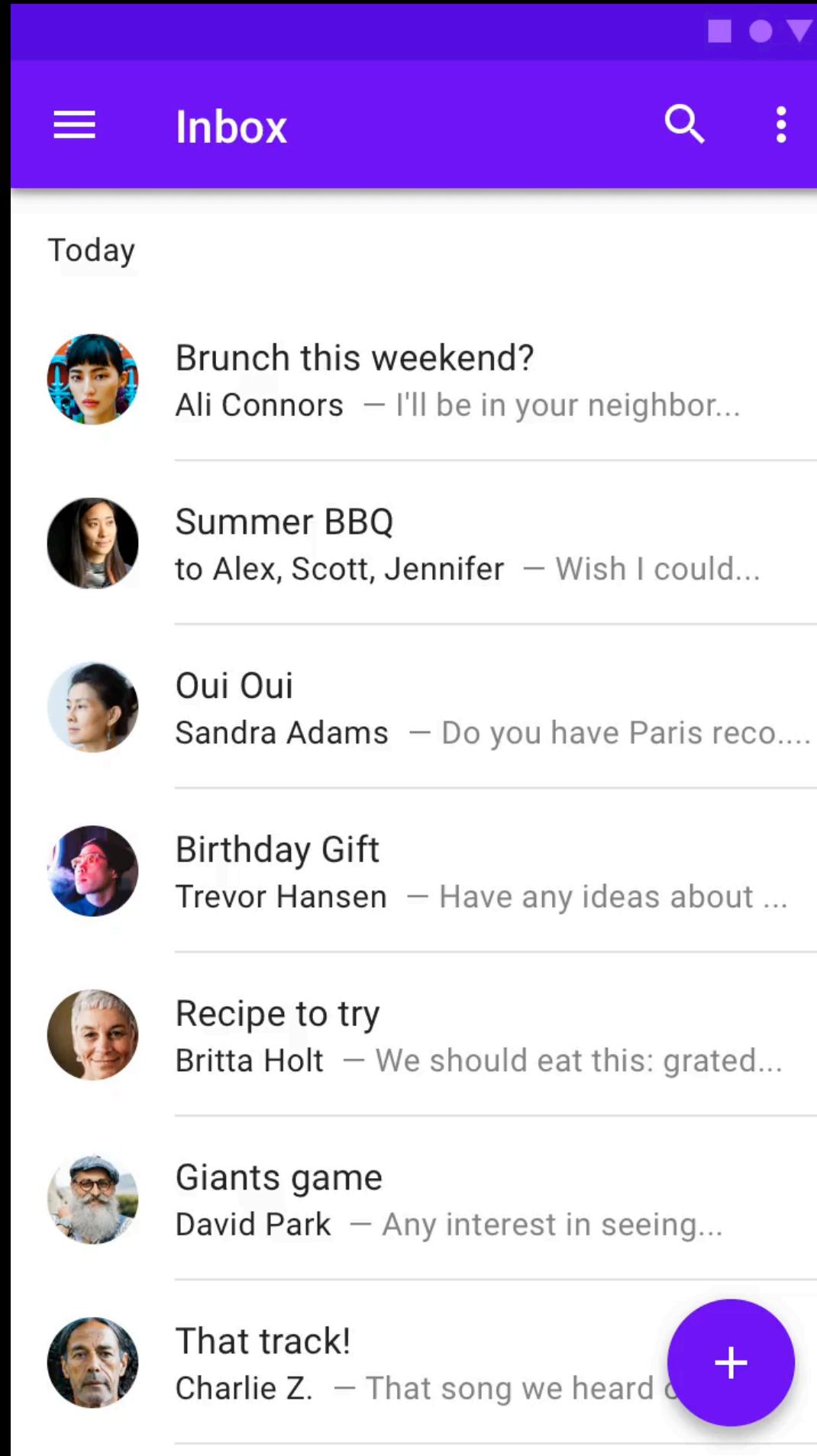
速度可通过【持续时间】和【缓动曲线】来控制。通过调整速度来控制过渡有助于用户理解UI的变化。



无论是什么风格的动画，过渡都不应过快或过慢，**过慢会让用户感觉好像在等待**。持续时间和缓动曲线的正确组合会产生平稳清晰的过渡。

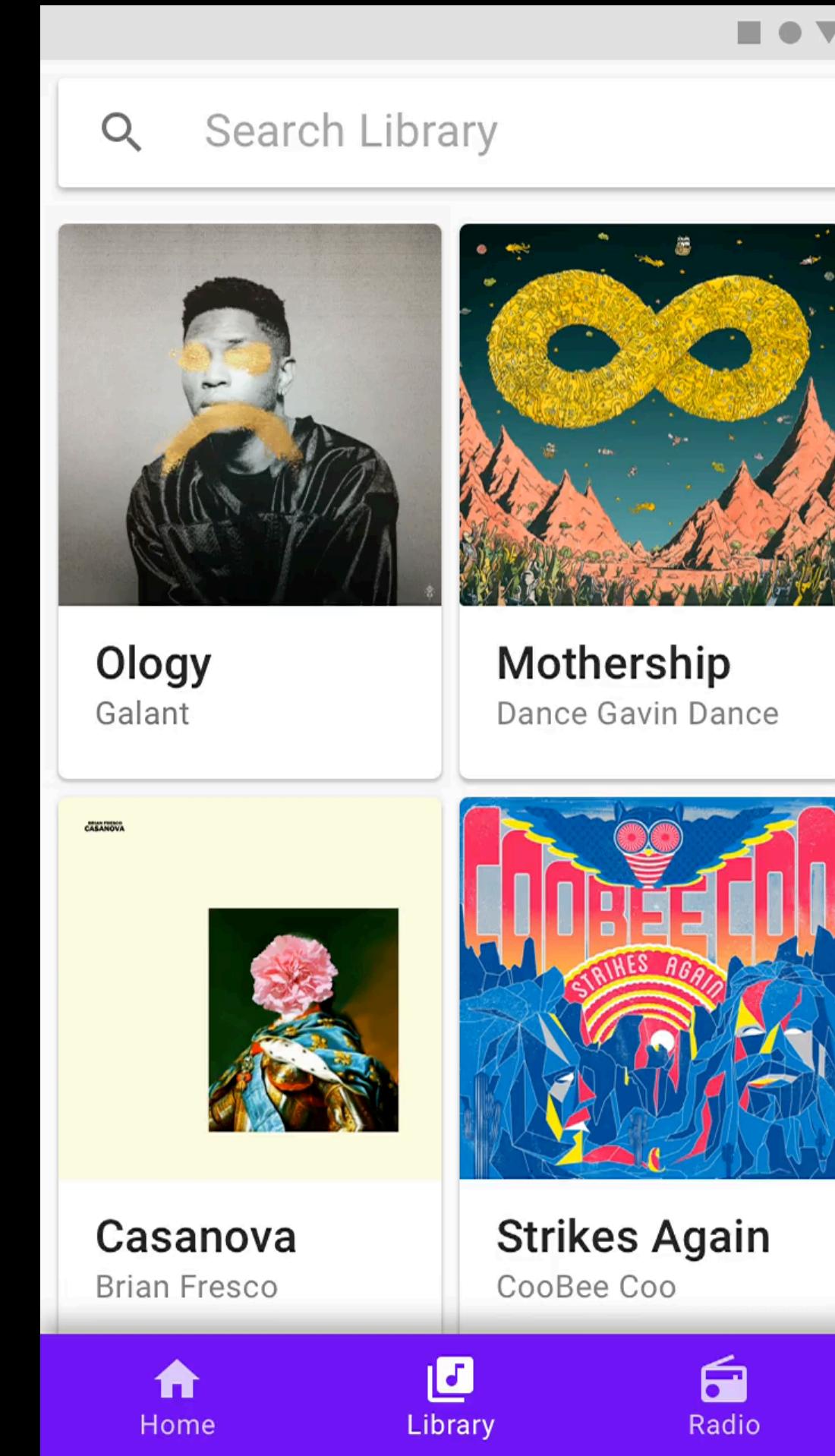
1 持续时长 (duration)

【关闭】 【退出】 【折叠收起】 这些运动的持续时长会比 【展开】 【出现】 时的时长要短；因为与用户的下一个任务相比，退出的环节所需的注意力更少。



【展开>收起】

导航展开250ms
但收起200ms

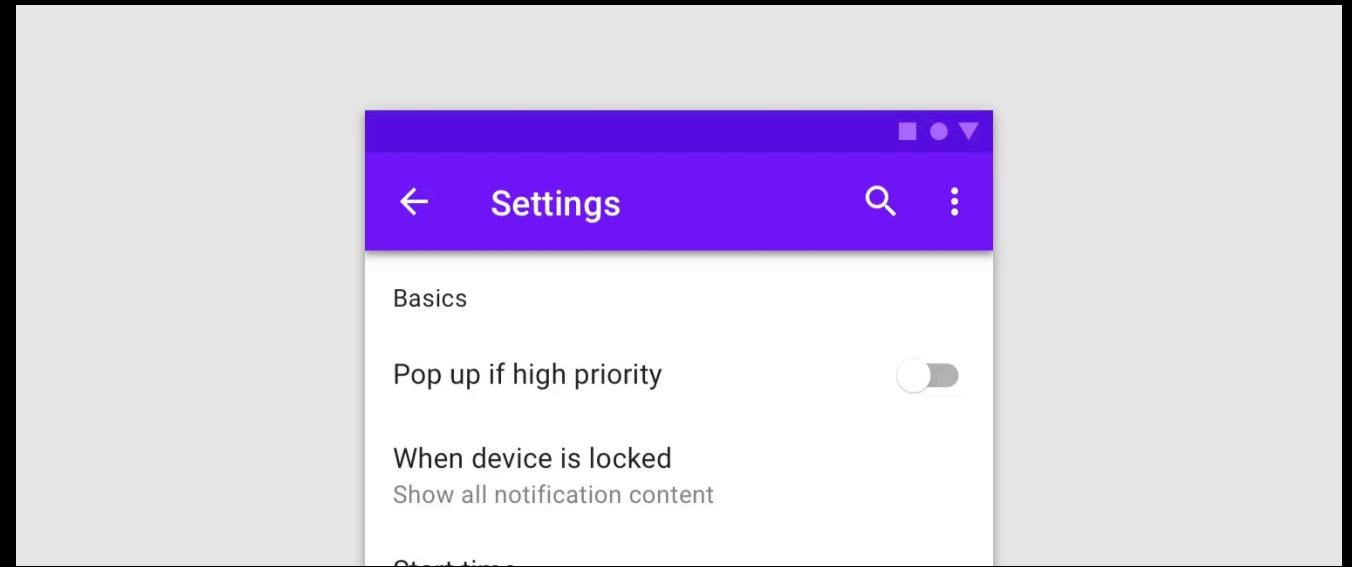


【展开>收起】

卡片扩展300ms
但收起250ms

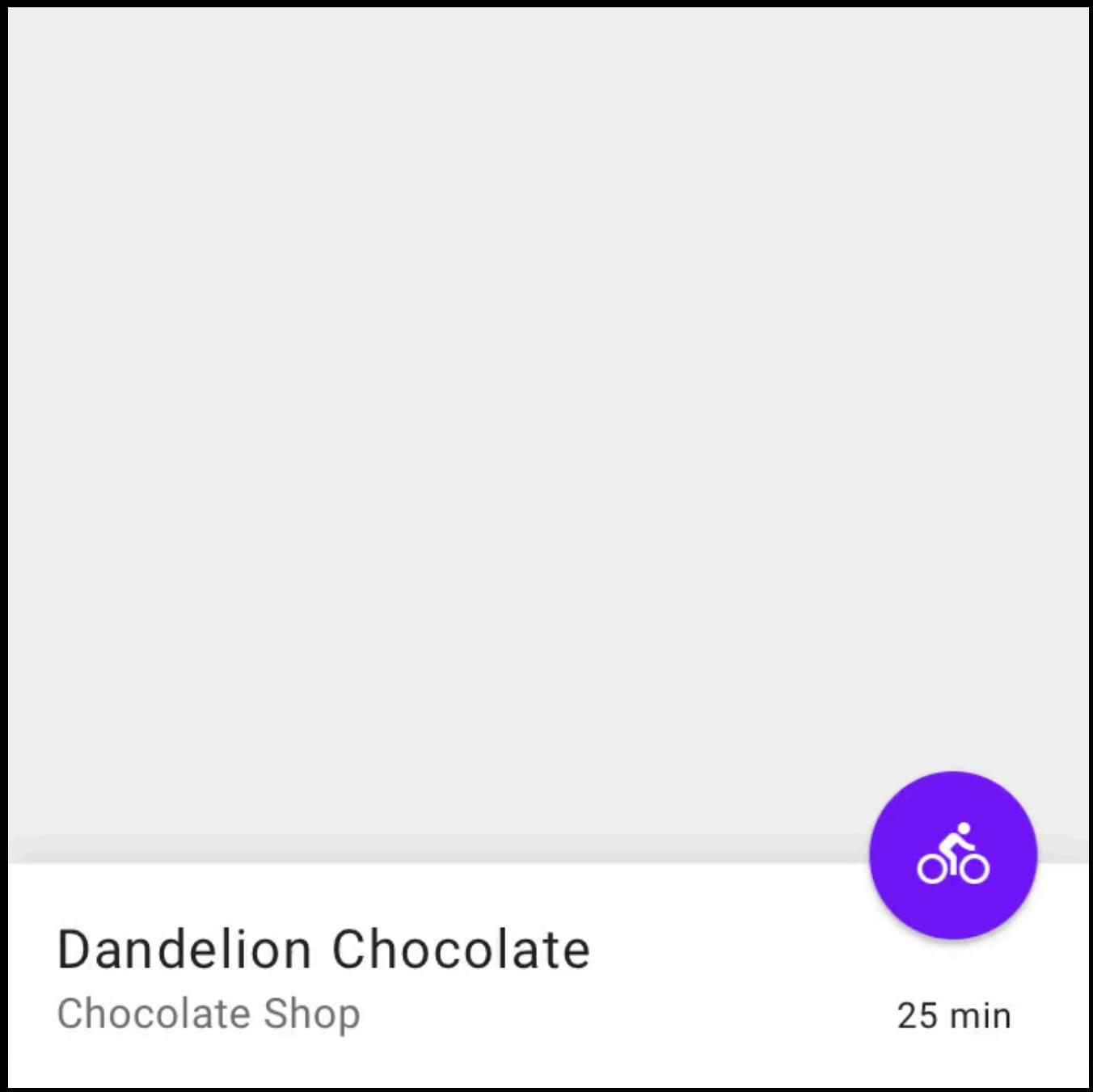
1 持续时长 (duration)

【过渡面积】与【持续时长】成正比；发生转换的面积越大，持续时间越长；



【小面积组件】

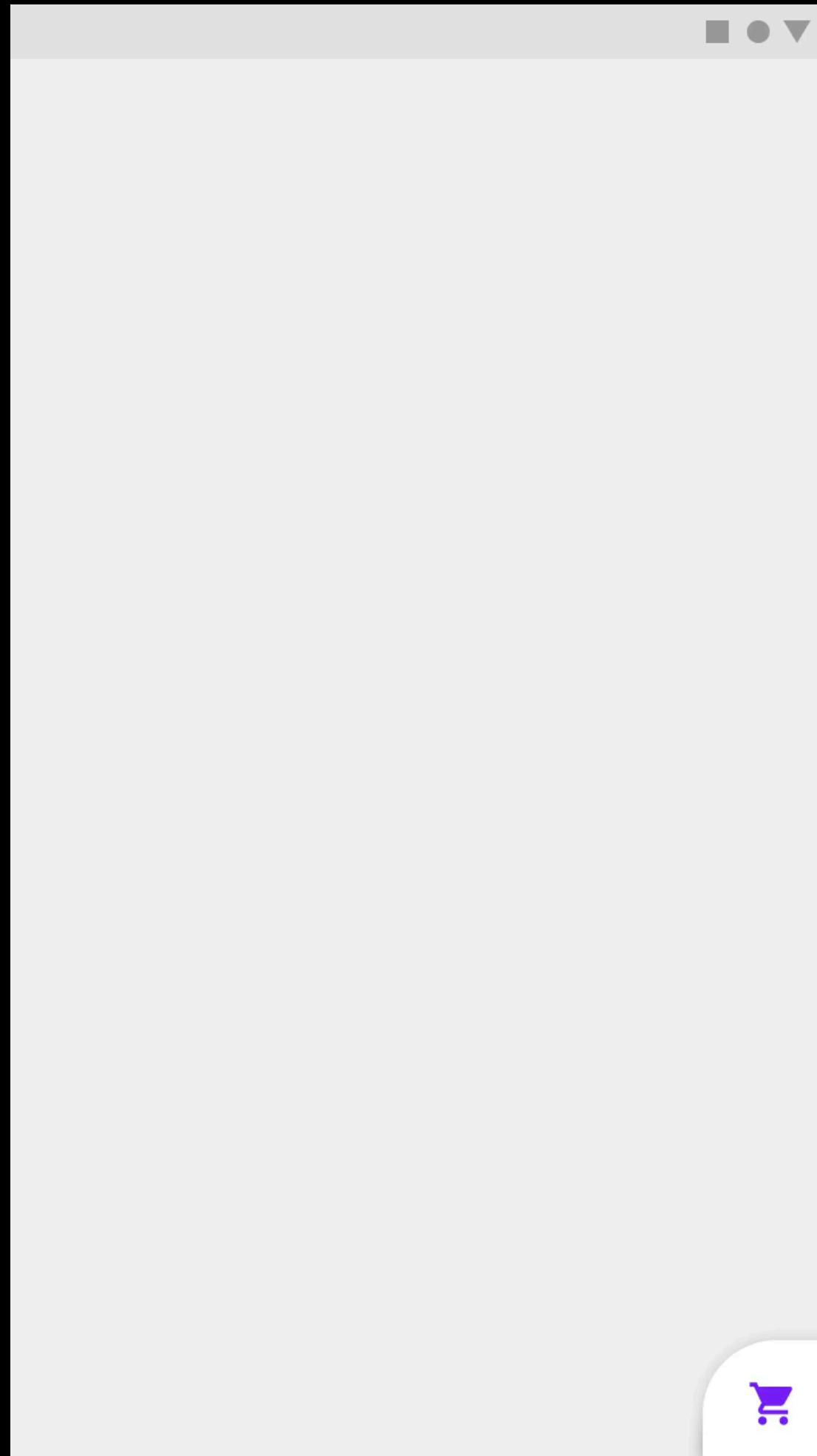
开关时长: 100ms



【中面积组件】

展开: 250ms

关闭: 200ms



【大面积组件】

展开: 300ms

关闭: 250ms

2 缓动曲线

缓动曲线可以调整动画的运动趋势，使过渡的元素可以加速和减速，而不是以恒定的速度运动。

在现实世界中，物体不会瞬间启动或停止，运动过程中存在加速和减速。

在UI元素运动中增加缓动就像摩擦，重力和质量之类的自然力在对元素起作用一样。



2 球动曲线有哪些类型

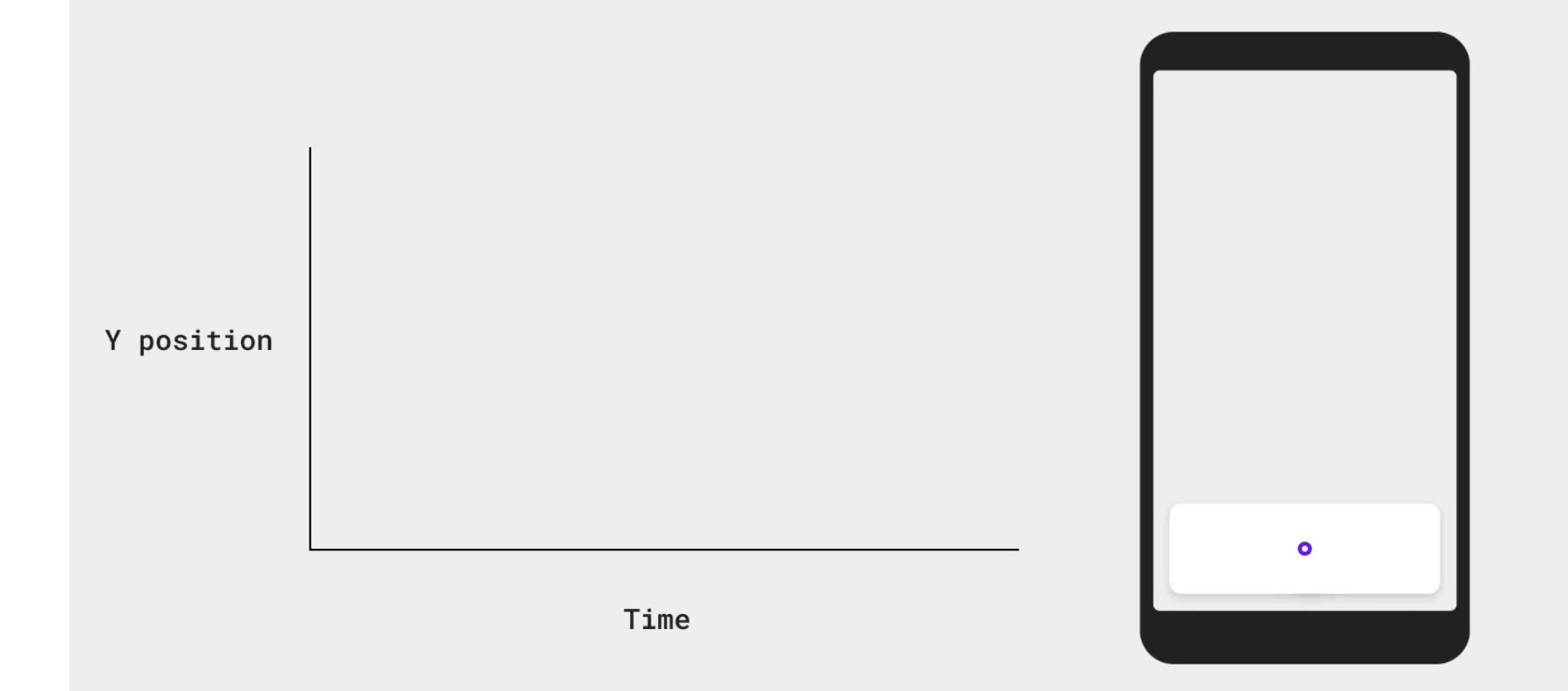
【标准曲线】 *最常用的曲线，轻微强调结尾



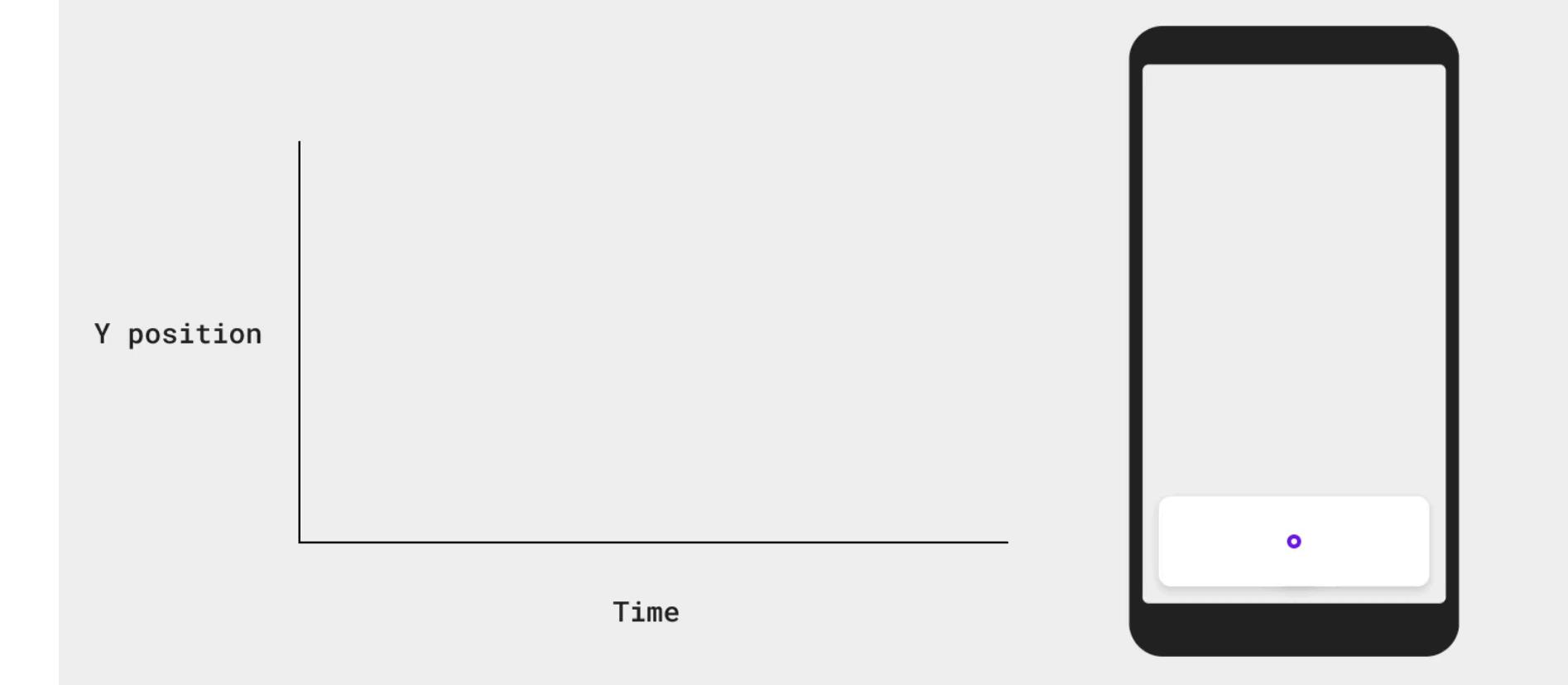
【减速曲线】 *元素进入、出现时使用减速曲线



【强调曲线】 *通过极速加速和较长的缓动强调结束部分



【加速曲线】 *元素离开屏幕时，可能使用加速曲线



编排Choreography

流畅度

【transition choreography过渡编排】是对一系列动效进行排序。
从而让用户的焦点始在界面中顺畅移动；

① 排序 Sequencing

- (1) 元素类型 (2) 消失类型 (3) 峰值速度

通过排序去掉不顺畅的帧

② 动画变化 Transformation:

- (1) 简单动画 (2) 复杂动画 (3) 有容器限制的过渡动画 (4) 无容器限制的过渡动画

给复杂动画设置简单规则

③ 焦点元素 Focal elements

依靠关键元素增强连贯性

① 排序 Sequencing

- (1) 元素类型 (2) 消失类型 (3) 峰值速度

排序良好的“动画”可以帮助用户了解屏幕上发生了什么变化（比如用户添加或删除了元素），并将用户的焦点集中在与交互有关的重要内容上。

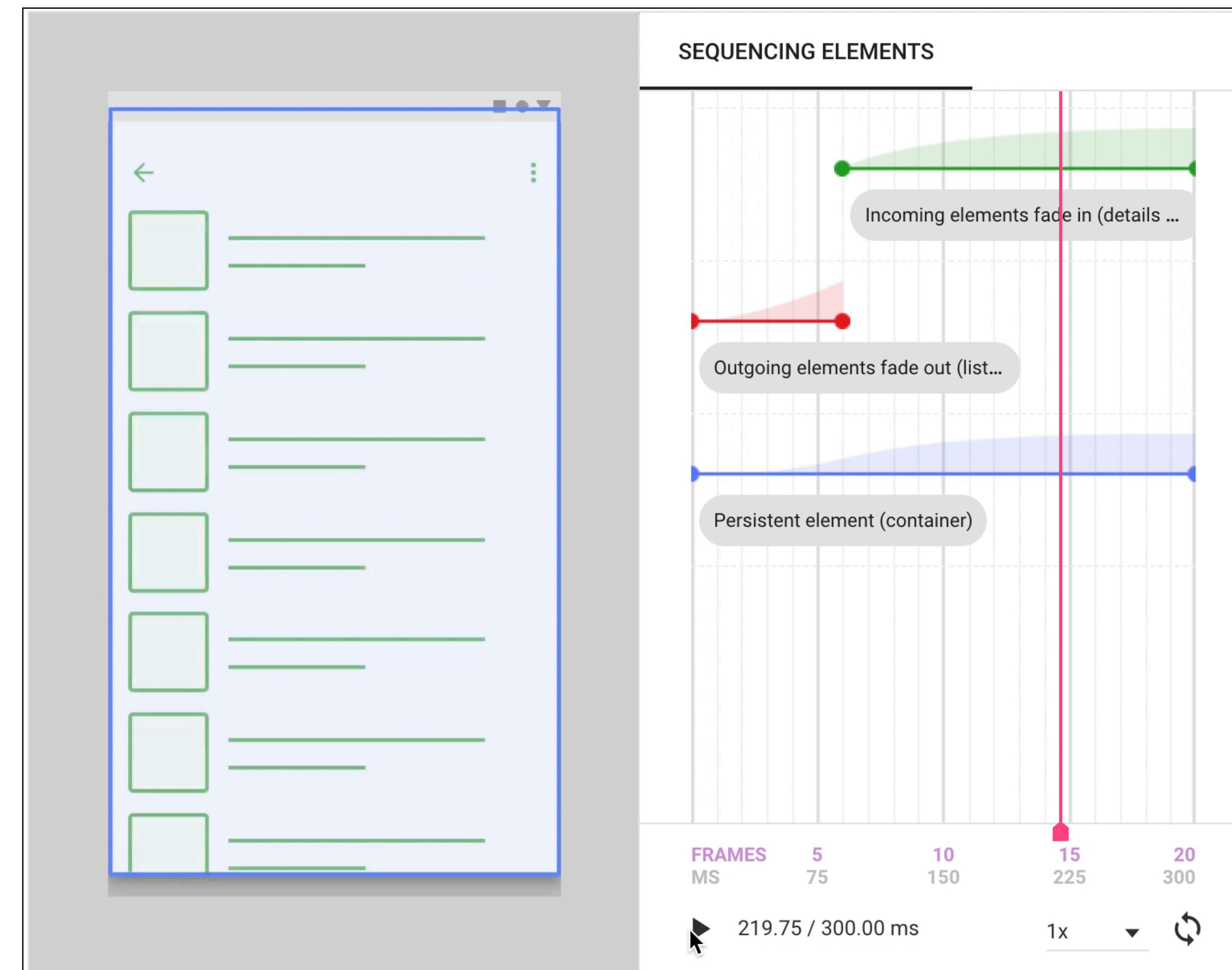
元素类型

排序中主要的三种元素类型：

(1) 进入屏幕的元素

(2) 离开屏幕的元素

(3) 固定存在屏幕中的元素



消失类型

当元素之间重叠时候我们经常使用fade淡入淡出来实现良好的动画排序过渡；

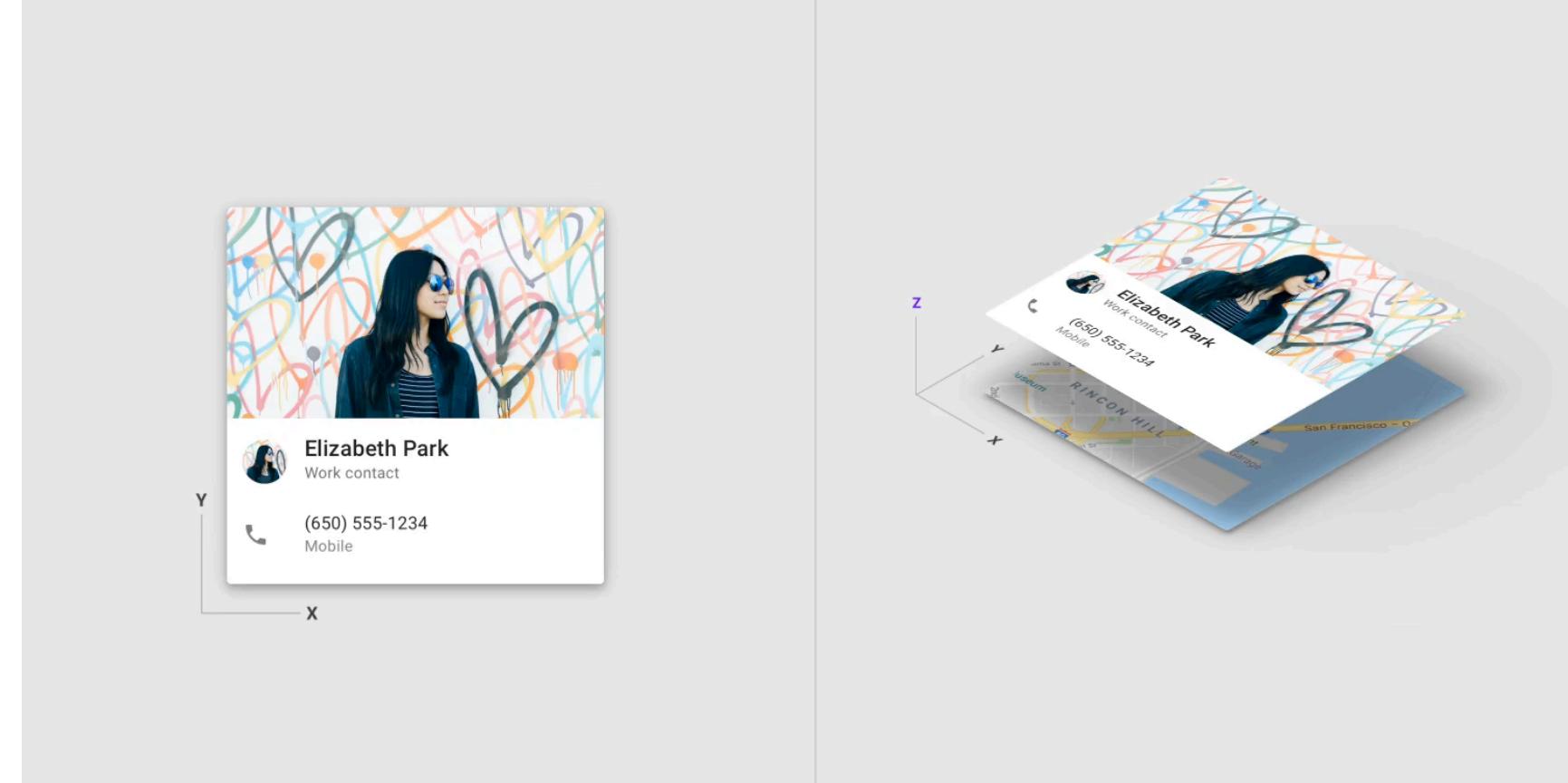
常用的3种消失过渡类型：

(1) fade (单层渐变)



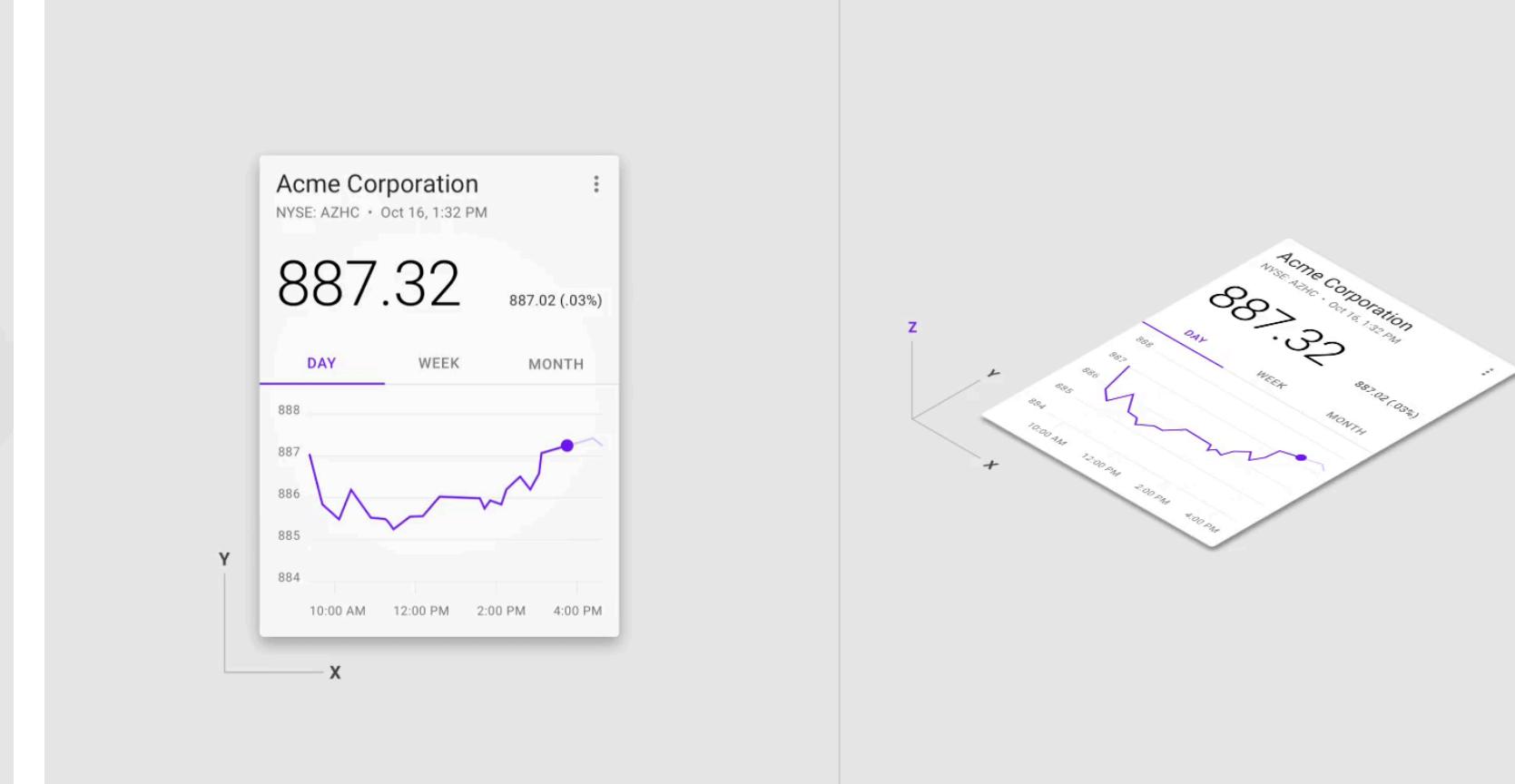
- 进入元素展现&消失
- 下层的固定元素保持不变

(2) Cross-fade (交叉渐变)



- 进入元素渐现（同时）
 - 消失元素渐隐（同时）
 - 固定存在的元素不变
- *不好之处是同时多层发生透明变化
容易视觉混乱

(3) Fade through (衔接渐变)



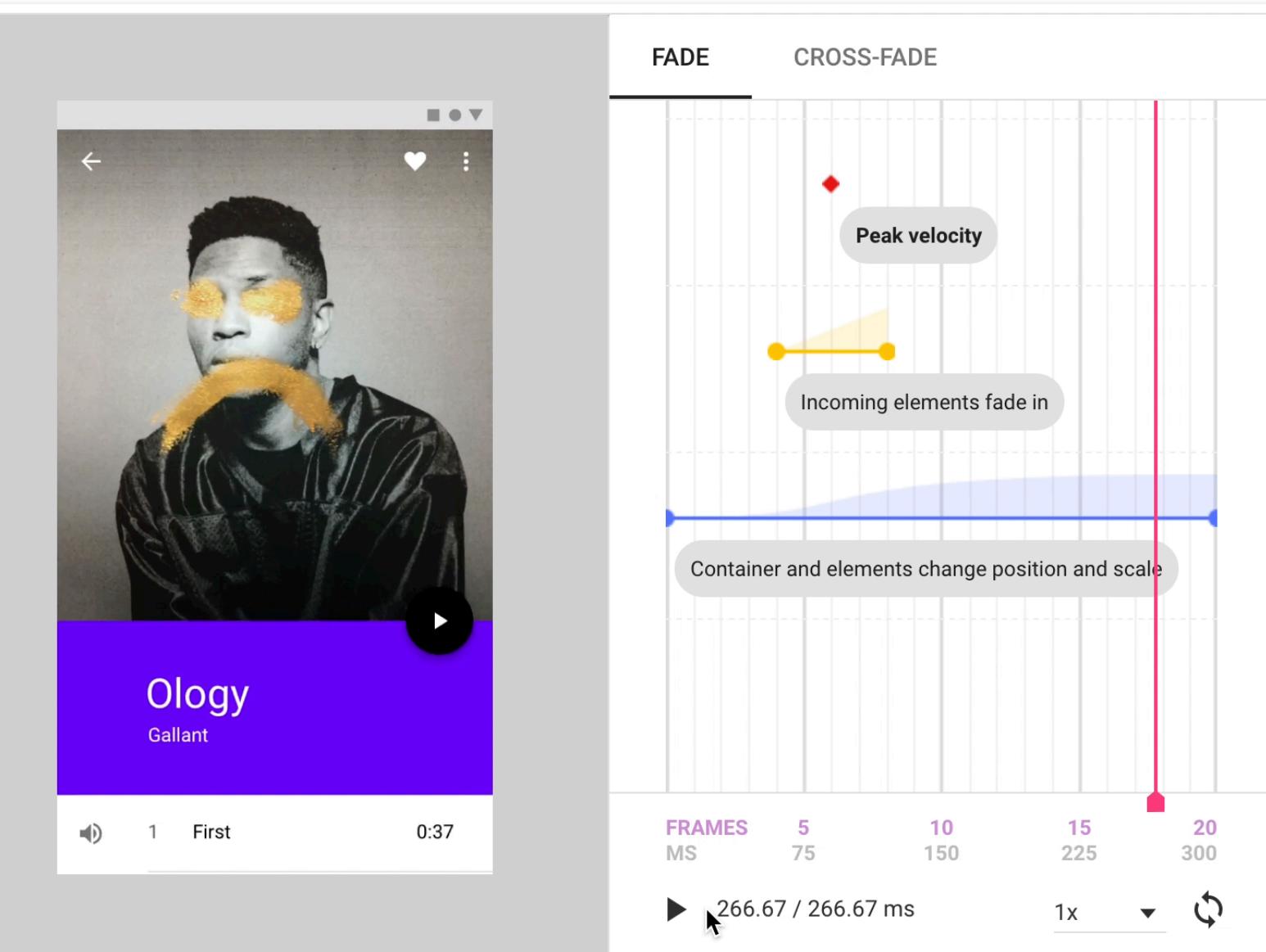
- 消失元素渐隐（先）
 - 进入元素渐现（后）
- *好处是最大可能的避免透明度重叠
带来的混乱

峰值速度

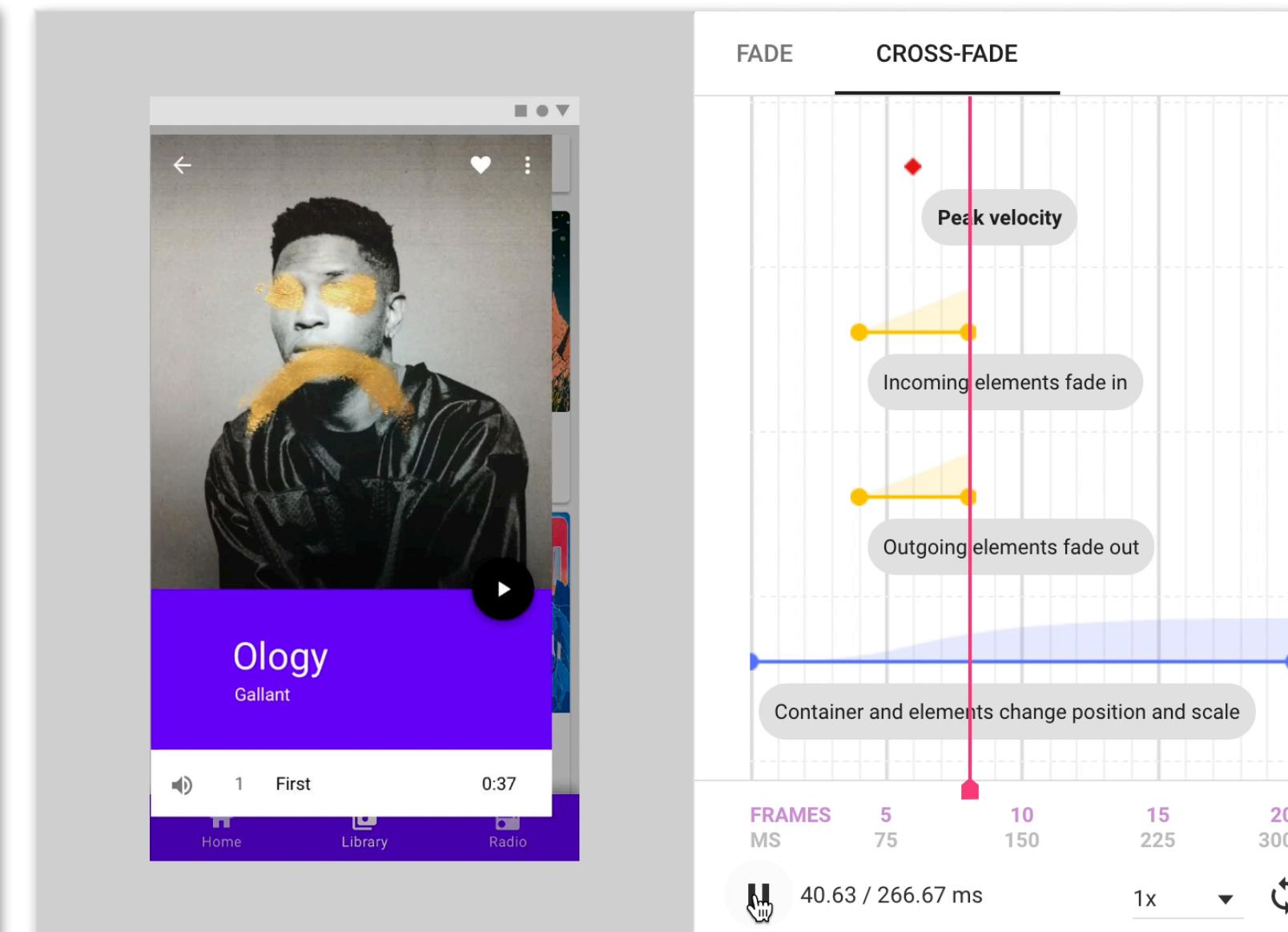
峰值速度指的是运动中最快的时刻；

峰值速度时刻可能会出现一些不好的展示效果（如多个透明层重叠效果混乱，两个层衔接中出现白色控白帧）
将峰值速度的效果不好的帧去掉，以优化整体的过渡效果，让效果看起来更加流畅、无缝衔接。

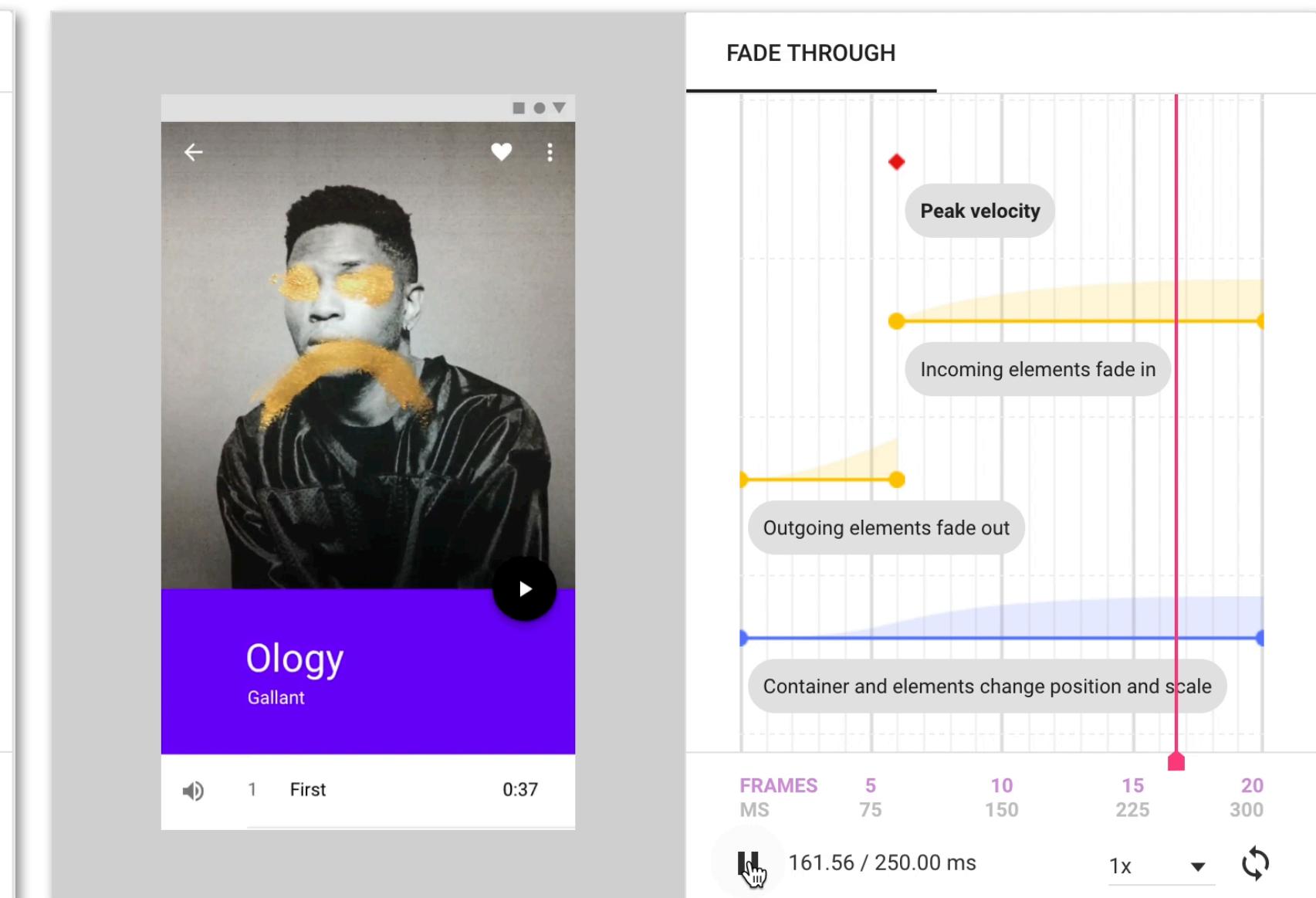
(1) fade (单层渐变)



(2) Cross-fade (交叉渐变)



(3) Fade through (衔接渐变)

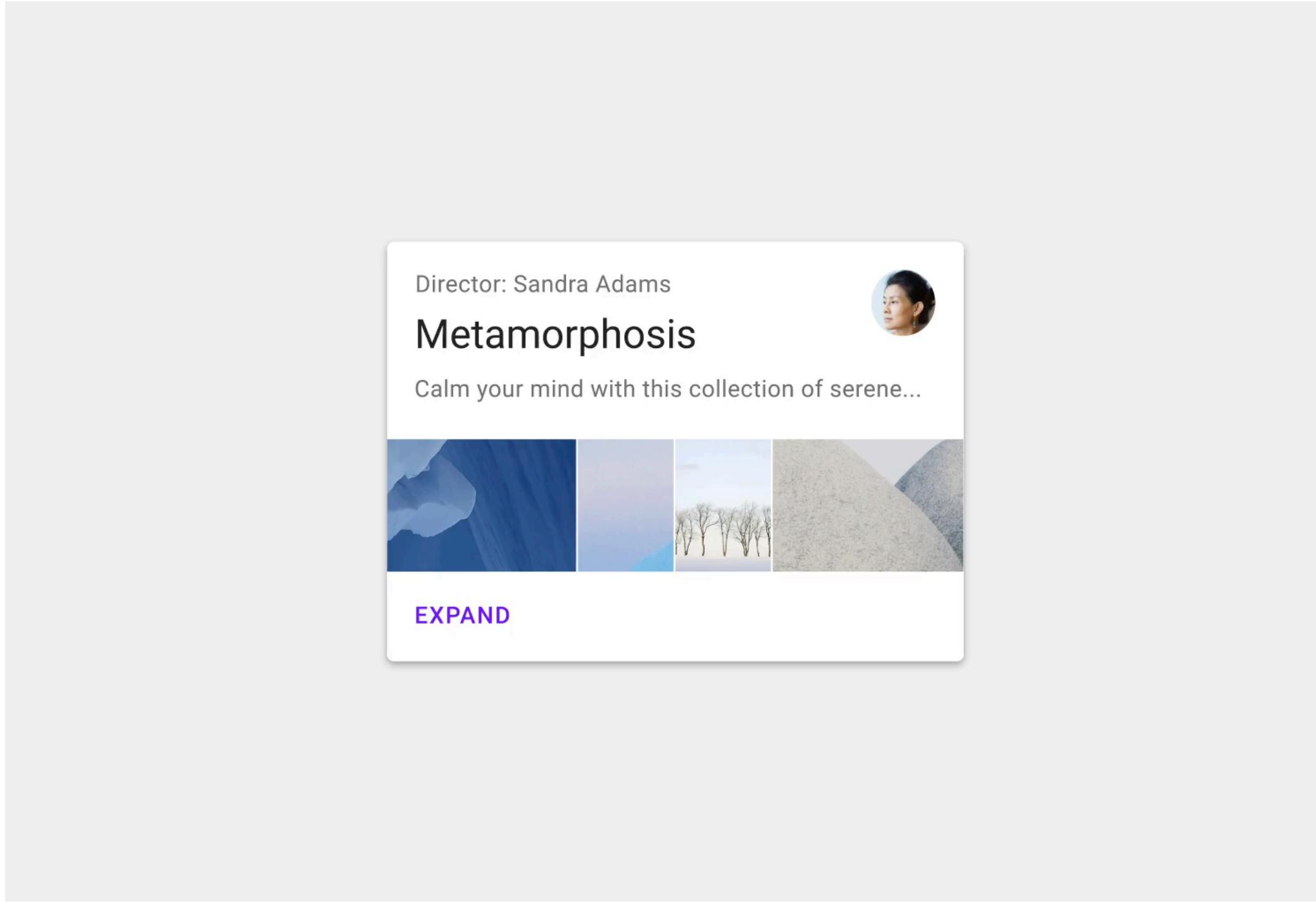


② 变形动画转换 Transformation:

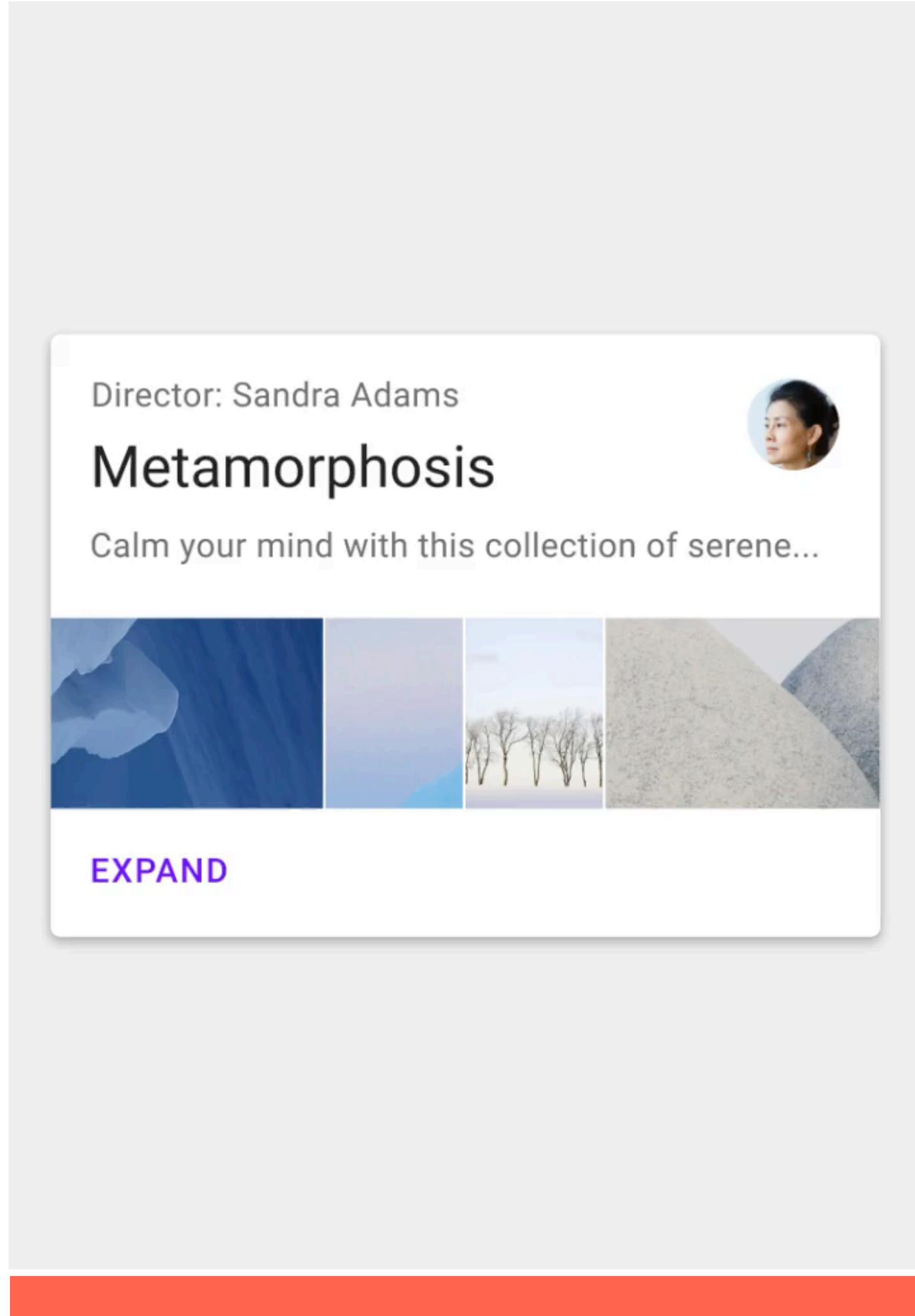
- (1) 简单动画 (2) 复杂动画 (3) 有容器限制的过渡动画 (4) 无容器限制的过渡动画

元素过渡变化时发生的特定动画叫做transformation。

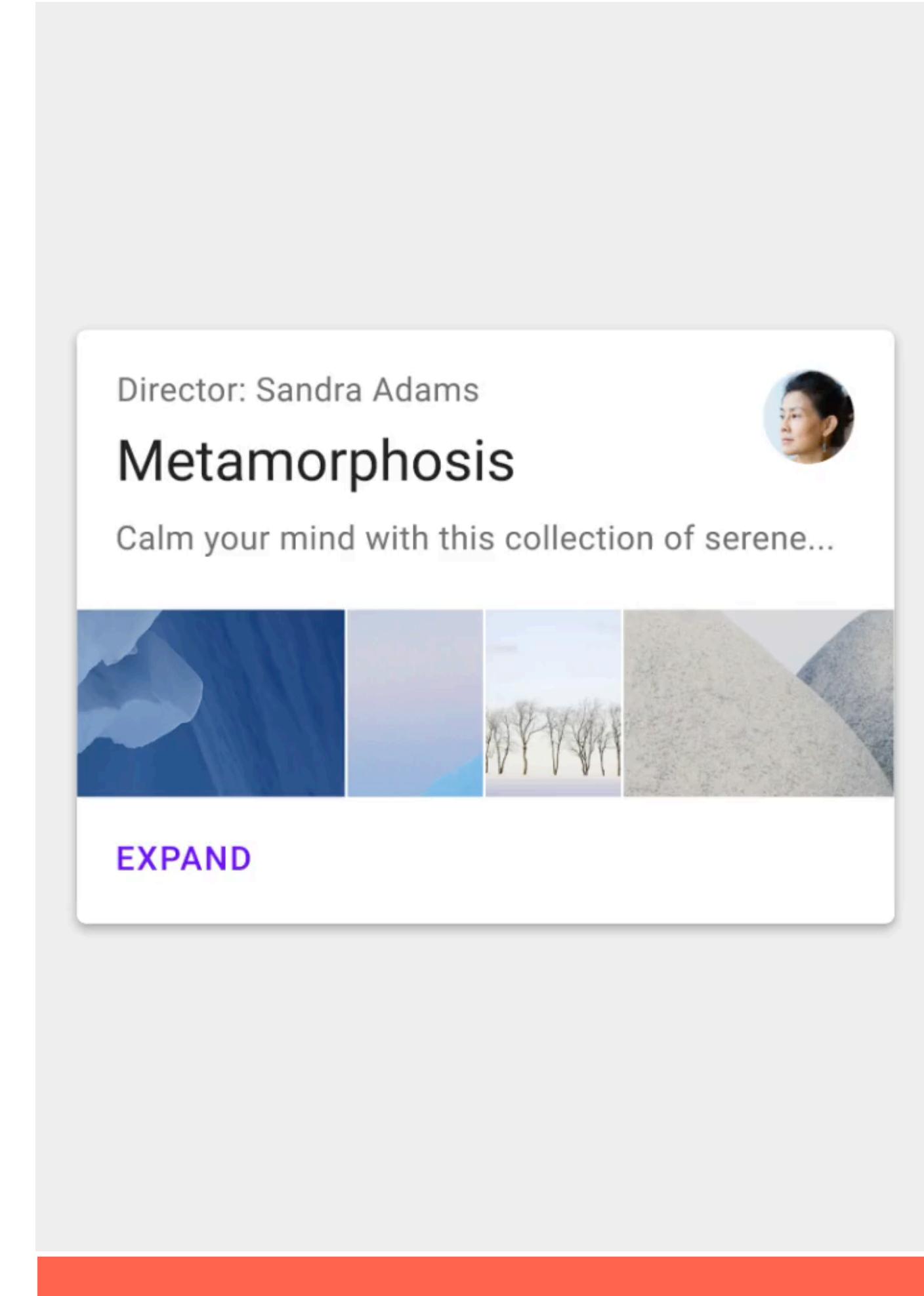
复杂动画case1



Do: 元素打包成组进行淡入淡出



Don't: 不要给有关联的多个元素分别设置动画
会分散用户的注意力



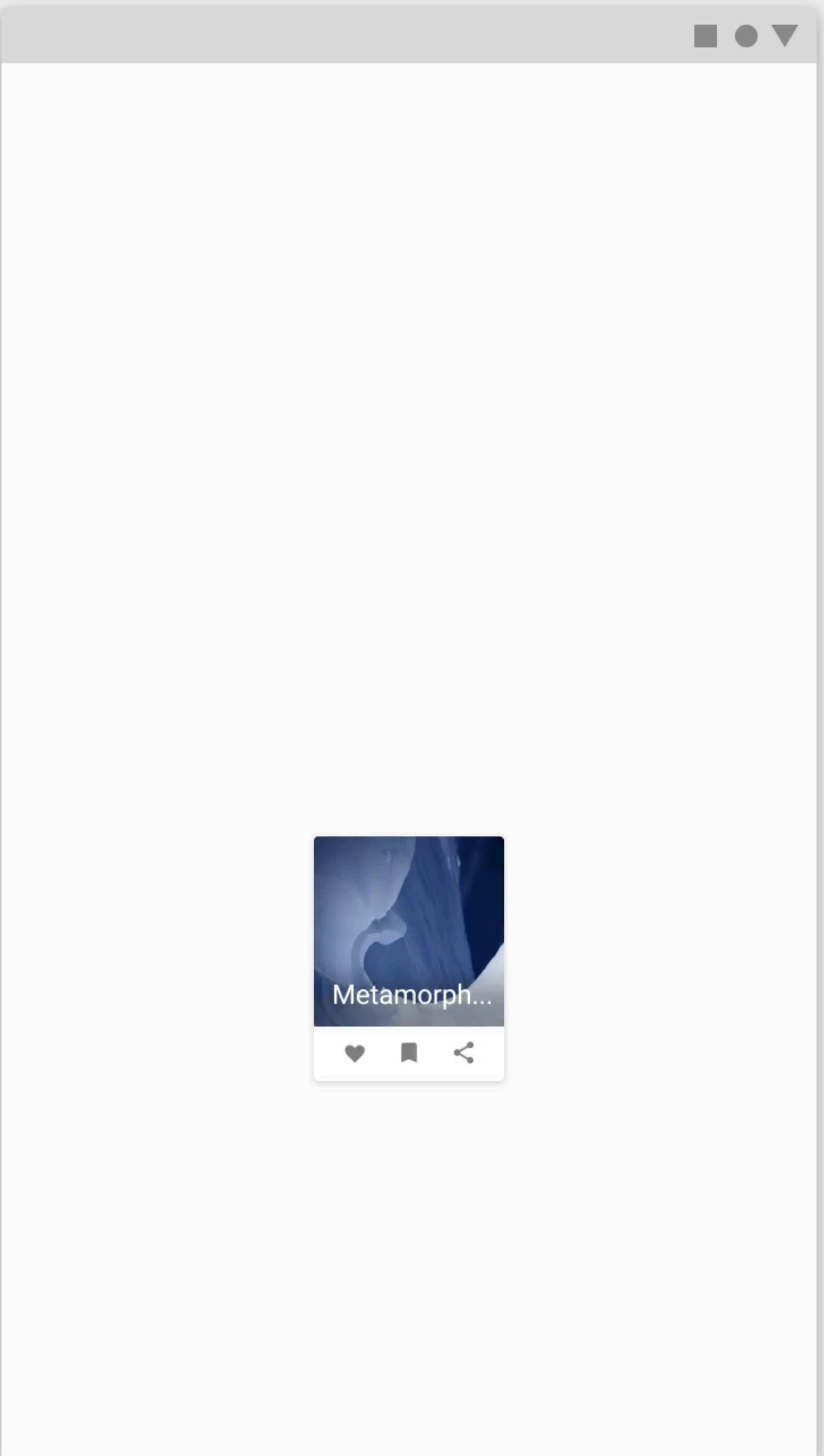
不要独立移动多个元素，
各部件的突然运动会使用户难以聚焦

有容器限制的过渡动画

“过渡的顺畅连续性”

当一组需要过渡的元素有明显边界时（比如卡片）
卡片内部元素可以按卡片整体缩放的比例去同步缩放

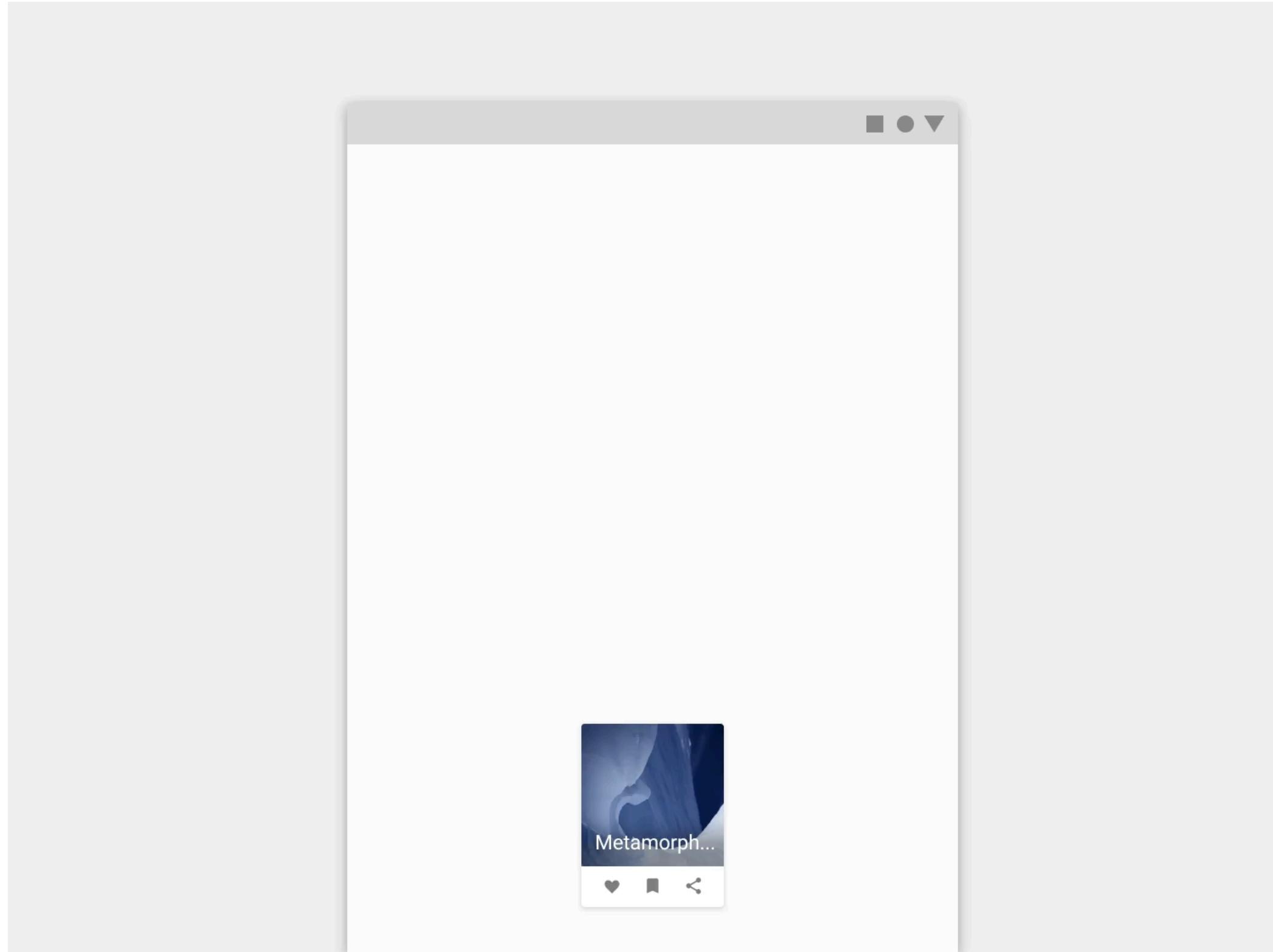
缩放时保持顶部对齐；



3 焦点元素 Focal elements

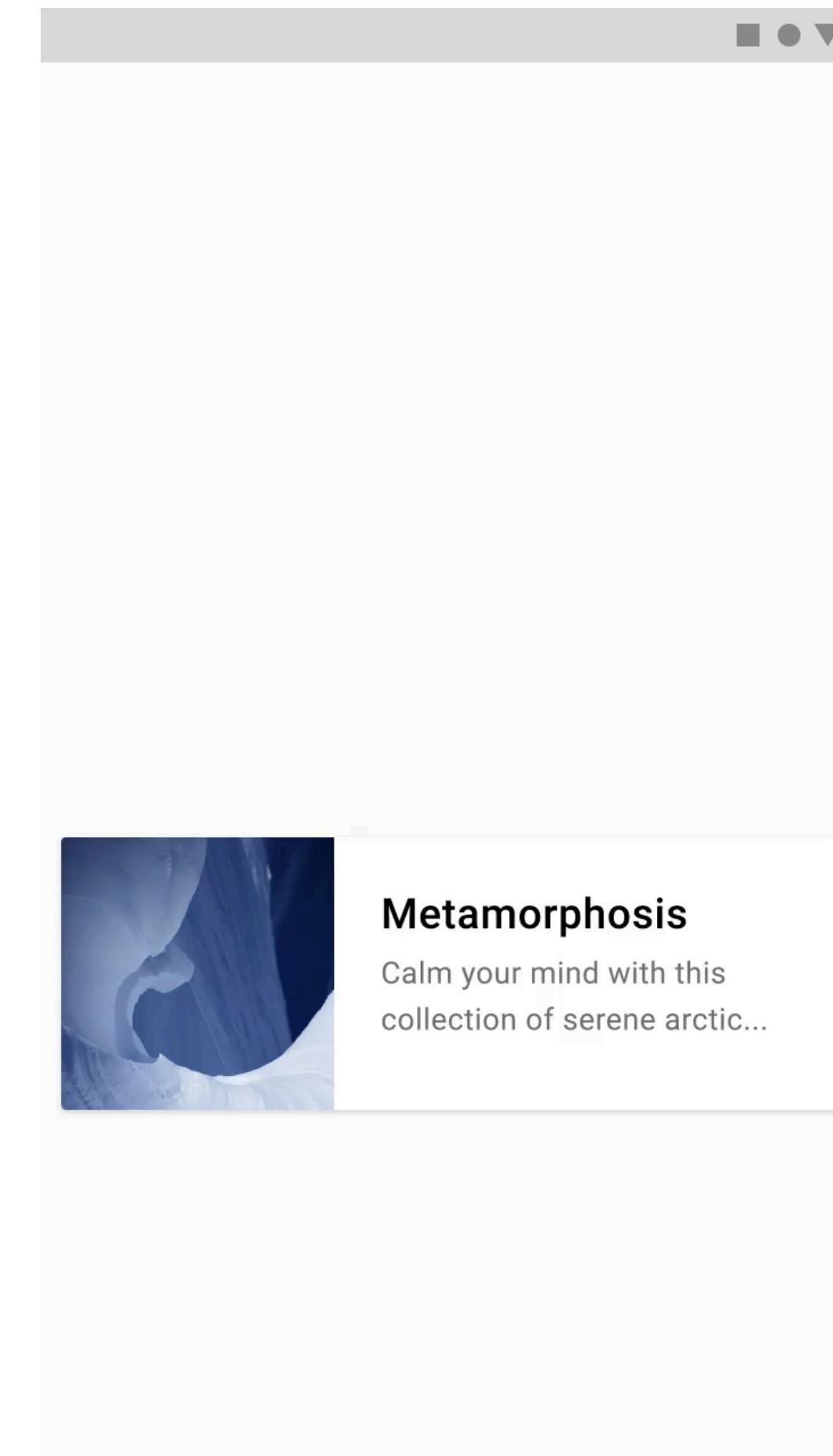
过渡中的焦点元素是一个持久存在的元素，对过渡中元素的整体层次结构很重要。
像动画的容器一样，焦点元素通过无缝转换外观来增强连续性。

焦点元素case1



Do: case中的头图作为焦点元素始终在变化中保持可见
通过头图的一致性增强过渡的连贯与一致性

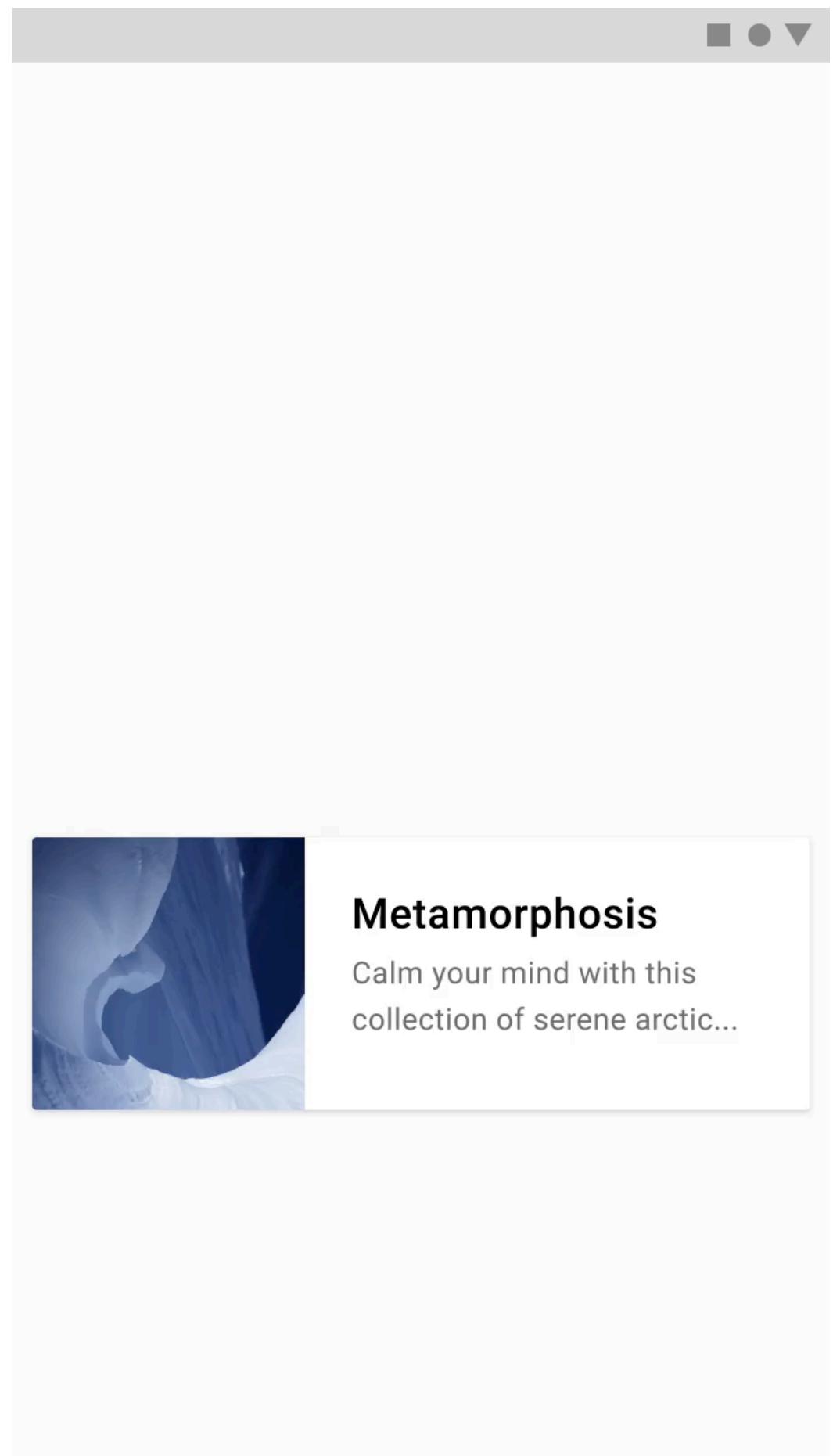
焦点元素case2



Caution:

如果焦点元素与运动中的其他元素重叠,
请注意, 因为它会造成混乱的过渡。

焦点元素case3



Do:

对焦点元素使用简单的fade模式,
可以淡化运动中出现的重叠混乱感

定制化Customization

通过自定义motion传达品牌的风格

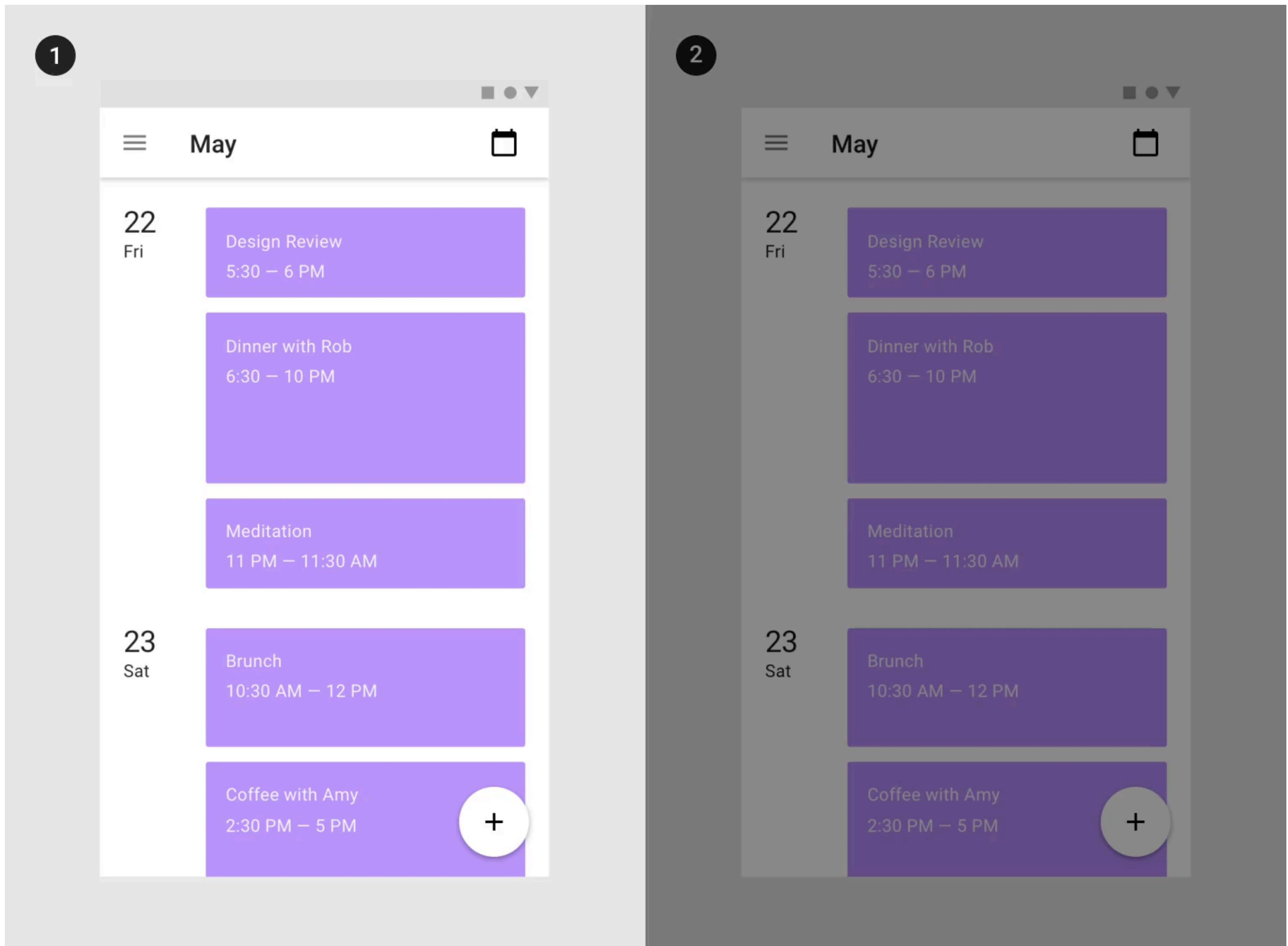
- 1 应用
- 2 速度
- 3 运动路径
- 4 震荡（操作反馈）
- 5 过渡模式
- 6 强化空间感
- 7 蹤跚（连续性的缓动） Stagger

生动&灵活
传递特有信息

3 运动路径

直线运动路径简单实用；
弧线运动路径带有强调的用意
更加生动；

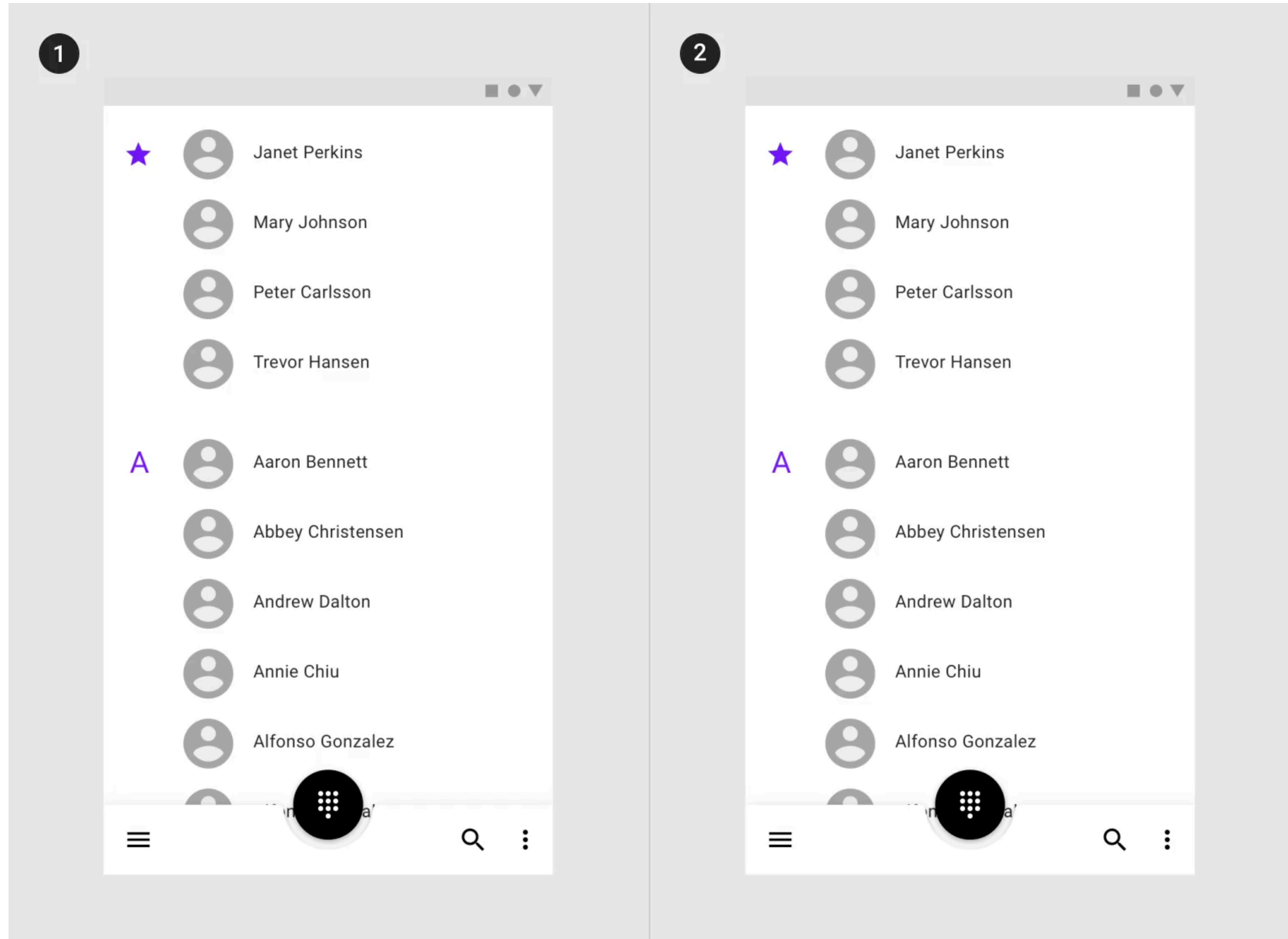
- (1) 直线运动
- (2) 弧线运动



5 过渡模式

一般包含固定容器的过渡都使用普通的容器模式（1）
但也可尝试使用共享轴模式让变化更加微妙

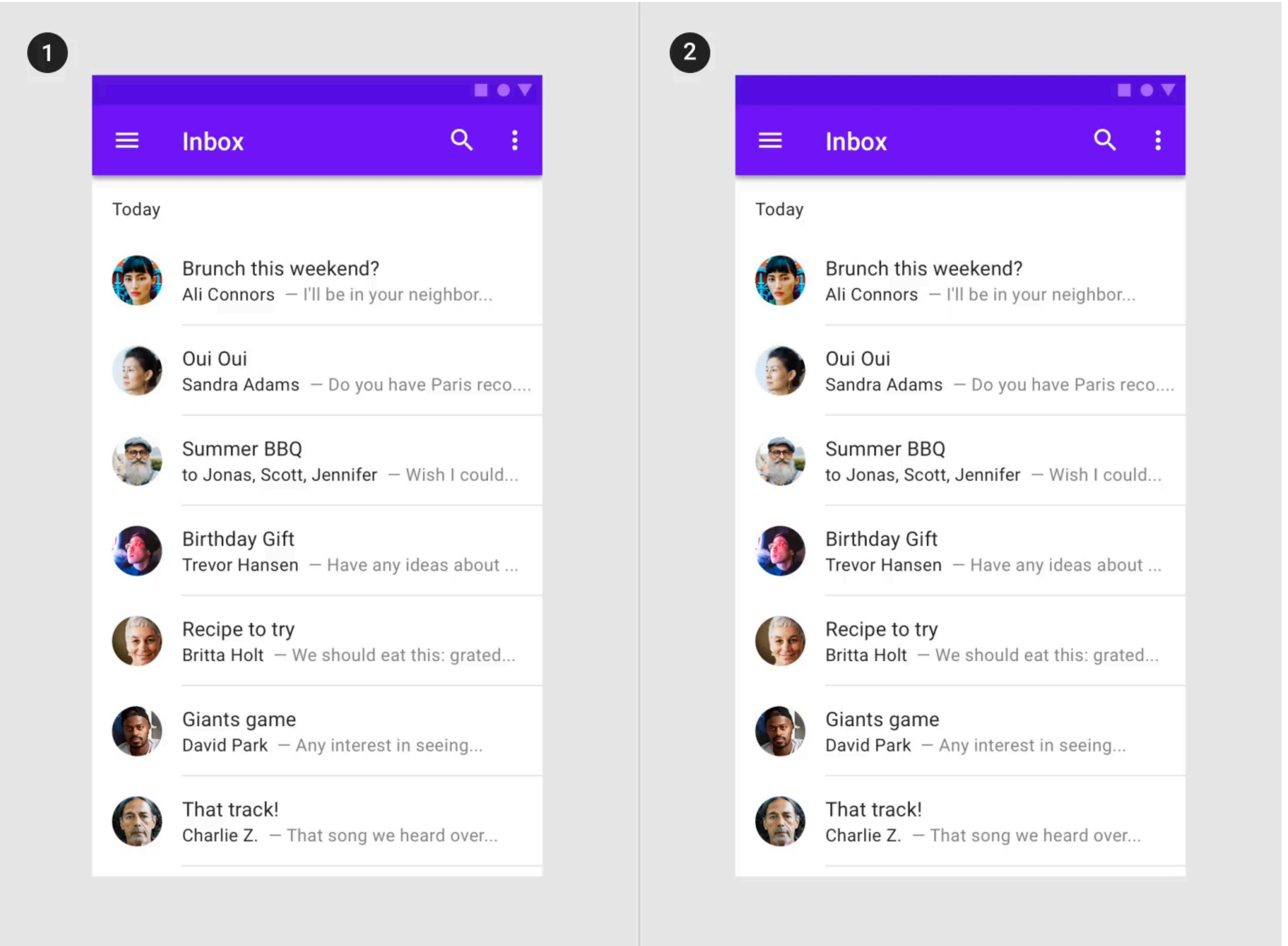
- (1) 容器过渡模式
- (2) 共享y轴过渡模式



6 强化空间感

默认情况下，背景内容在容器转换过渡期间保持静态。但可以通过设置背景内容的大小来强调空间高度。通过强调前景和背景内容之间的距离，这种自定义设置会使过渡效果更加生动。

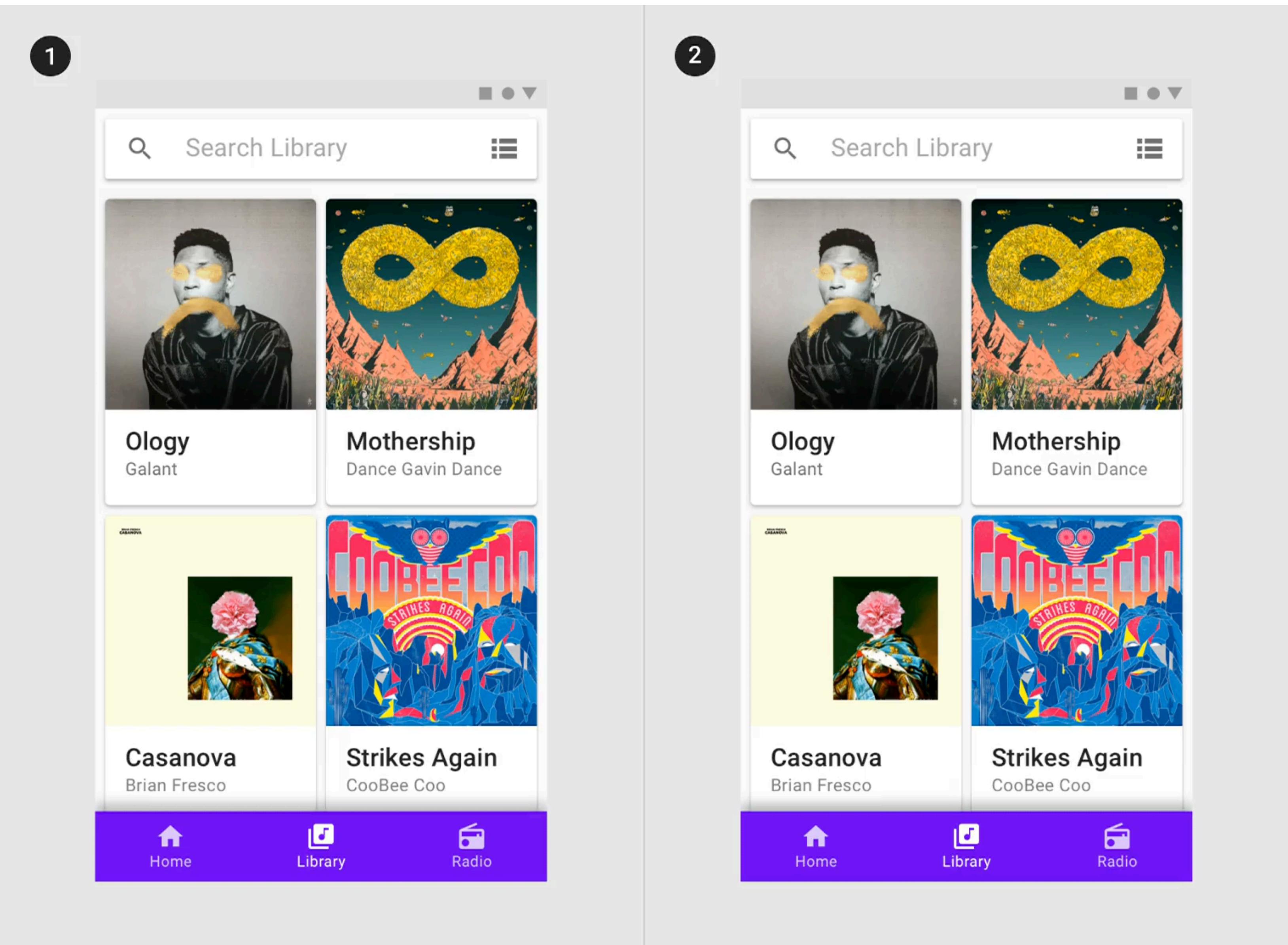
- (1) 默认容器转换效果
- (2) 强化空间感后

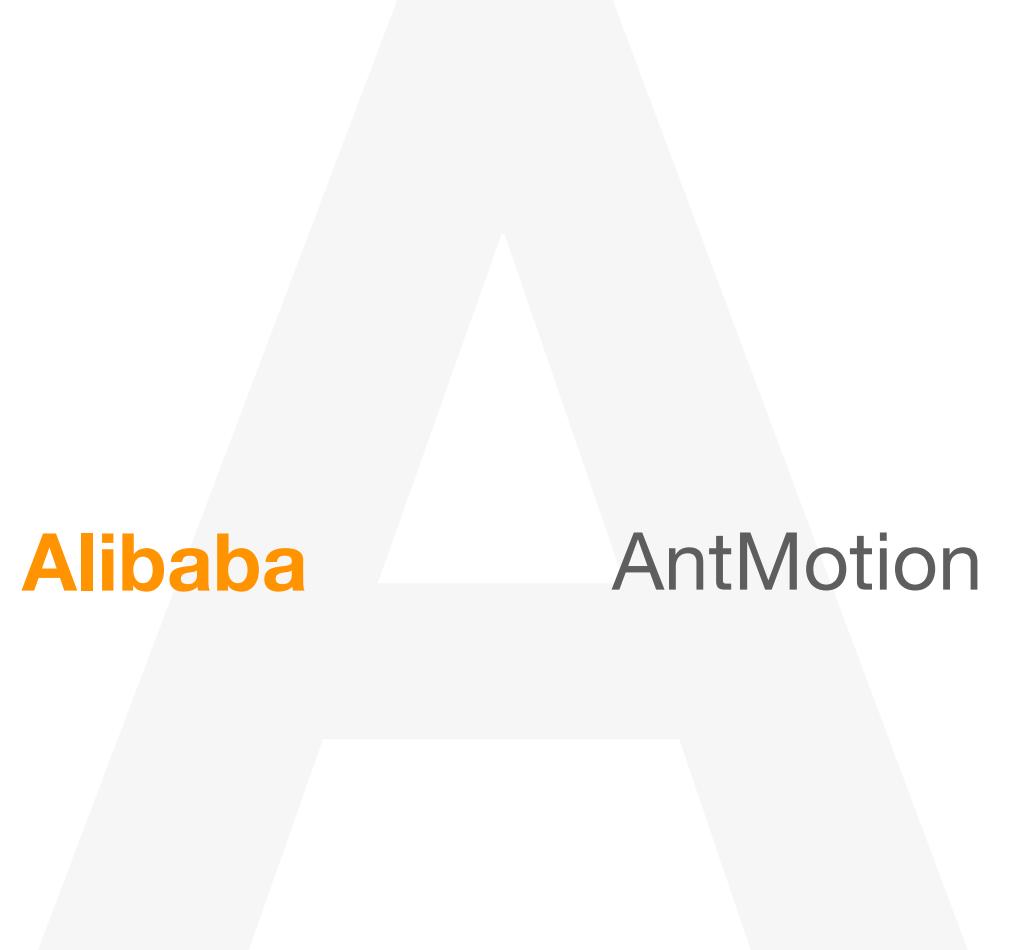


7 stagger 蹤跚（依序连续缓动）

通过有序的延迟制造级连感，
让用户的注意力短暂的集中在每个元素上

- (1) 无连续效果
- (2) 增加stagger效果





原则

速度

空间

组合

过渡

自然 提效

Motion 动效

- 动效价值
- 衡量动效意义

界面动效能加强用户认知且增加活力。

动效价值

- 增加体验舒适度：让用户认知过程更为自然。
- 增加界面活力：第一时间吸引注意力，突出重点。
- 描述层级关系：体现元素之间的层级与空间关系。
- 提供反馈、明确意向：助力交互体验。

衡量动效意义

衡量一个动效是否有意义，我们可以通过以下几个标准来考核：

- 一个动效的存在是否合理：是否带有明确的目的性，助力交互体验，没有多余的动效。
- 动效与性能：不能出现大幅度波动丢帧或者卡顿现象，动效的体验须是流畅的，并且不影响产品的性能。

原则

在企业级应用的产品设计中，使用动效和前台类产品有很大的不同，助力交互行为和增强信息认知显得尤为重要，在 ant design 设计价值观的基础之上，我们衍生出动效设计的三原则：



自然

自然运动规律，保证视觉连贯，让用户感知到动作是自然的



高效

尽量节省过渡的时间，快速完成过渡的动画效果



克制

做有意义的动效，不去做太多的修饰而干扰用户

速度 时间&缓动曲线

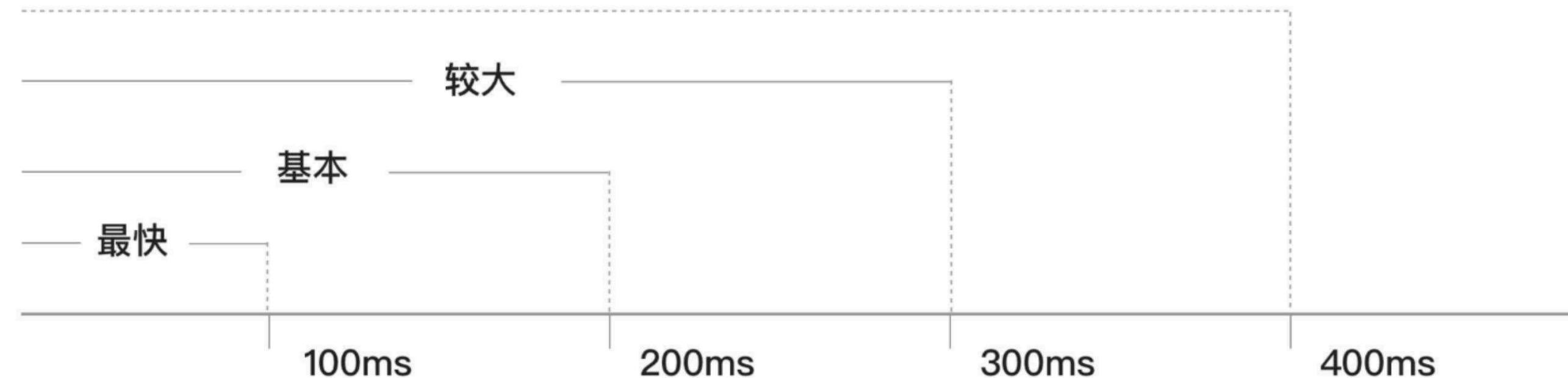
速度的快或慢取决于时间与缓动，相同的距离，时间越短速度则越快，而缓动则是能将同一段时间划分快与慢的区域。

时间

在企业级应用的设计中，动效需要在尽可能短的时间内有效的完成过渡。那么如何把握这个时间的长短呢？伦敦城市学院神经学创立者 [Davina Bristow](#) 曾提出，眨眼是人体最快的潜意识动作，大多数人每分钟眨眼 15 次，眨眼平均持续 100–150 毫秒；所以我们将这个时间段定义成 Ant Design 动效的基本时间单位。

Ant Design 组件里的基本时间单位

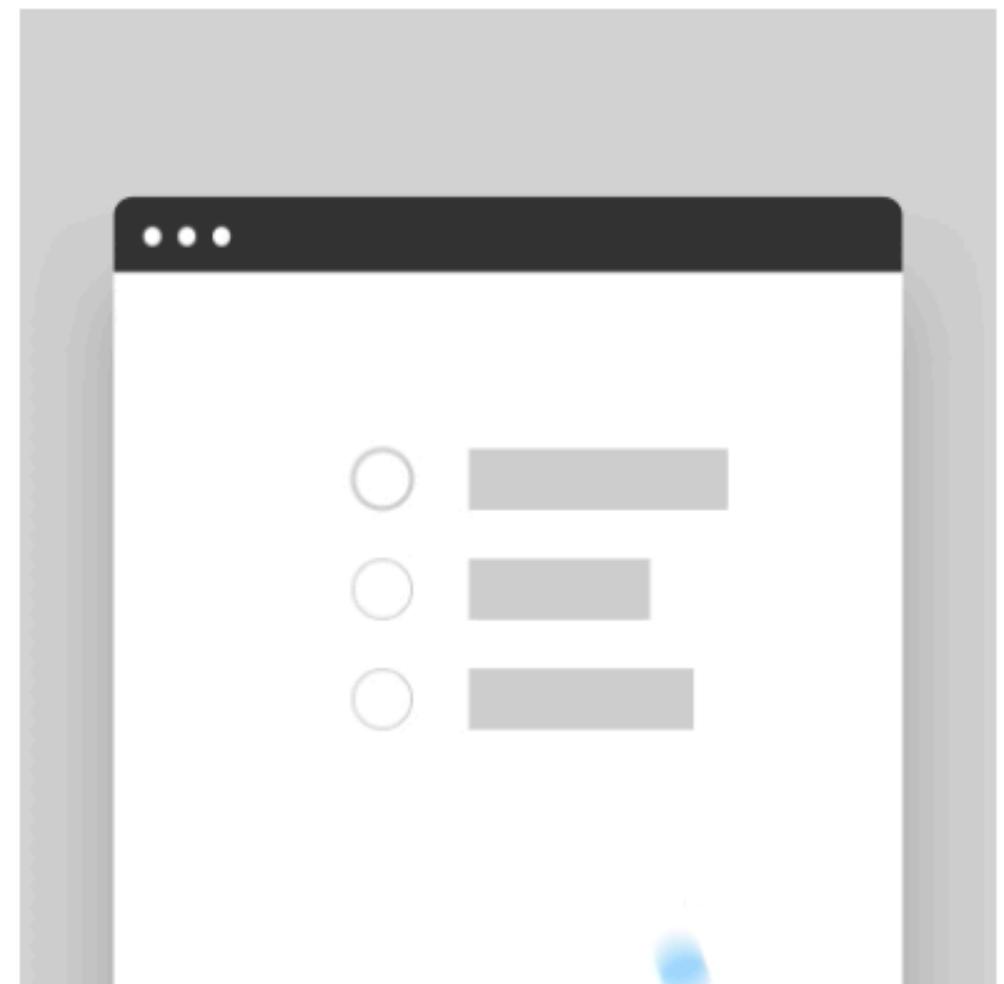
最快时间: 100ms, 基本时间: 200ms, 较大时间: 300ms... 100 的倍增方式。



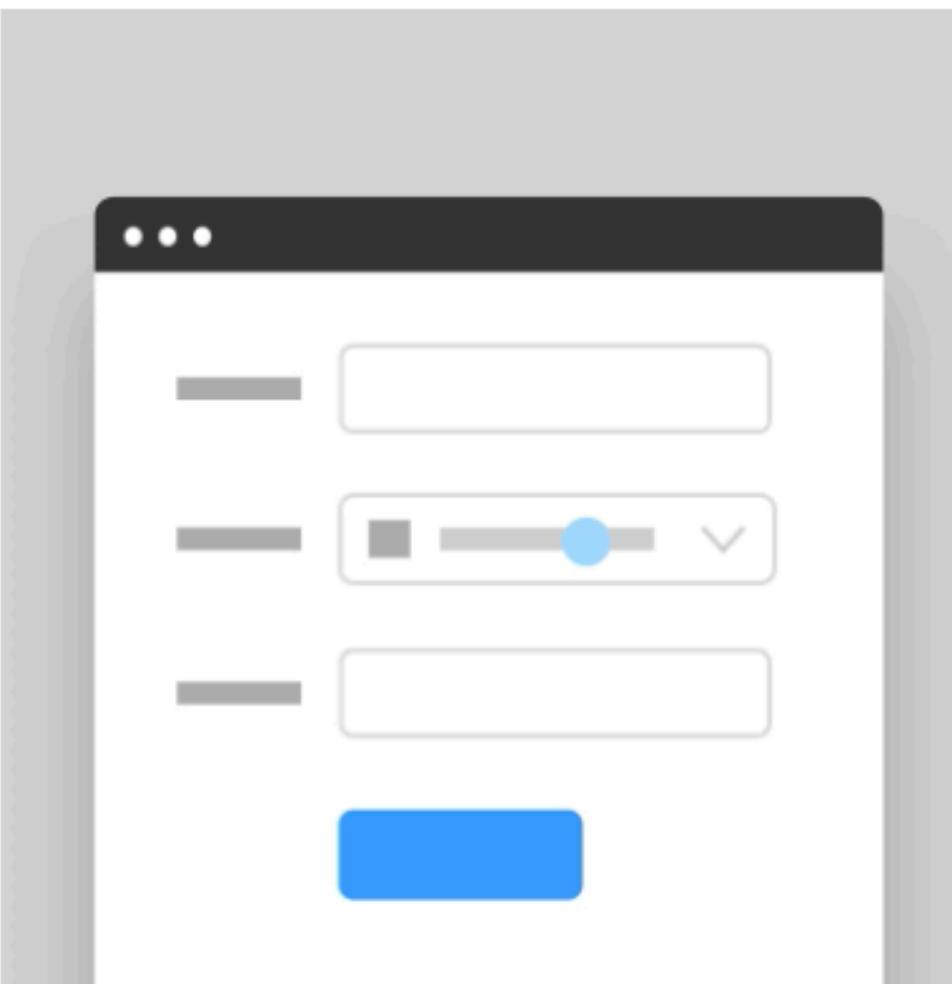
面积越大耗时越长

过渡距离与过渡时间成正比

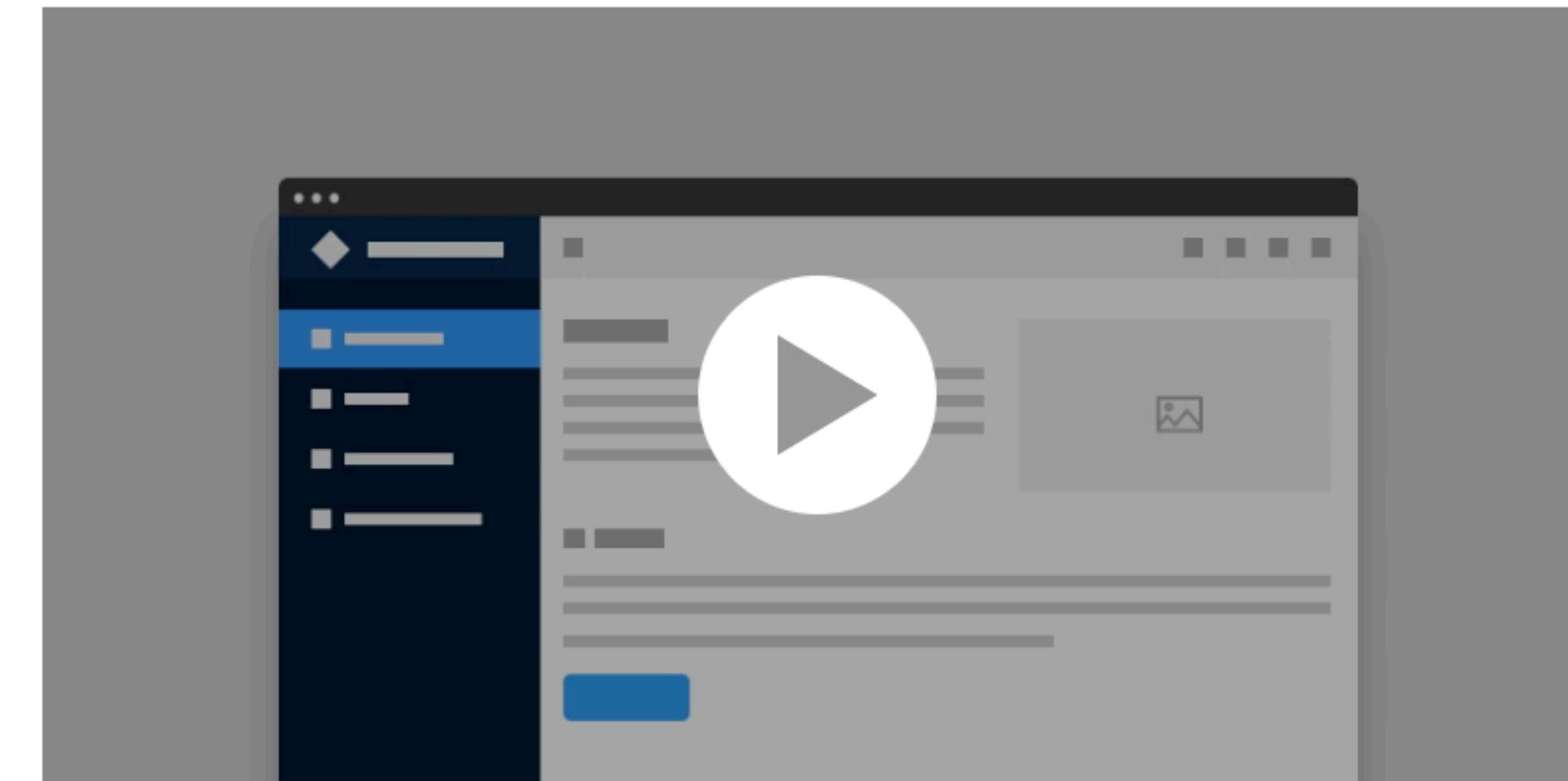
示例演示



radio 的时间为 100ms

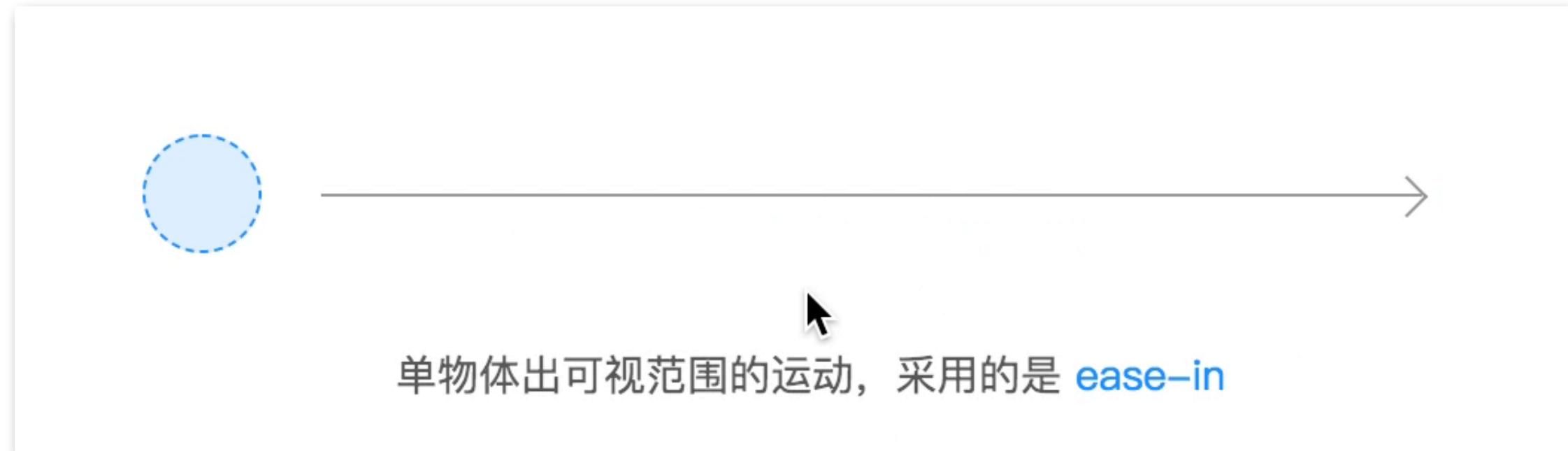


select 的时间为 200ms

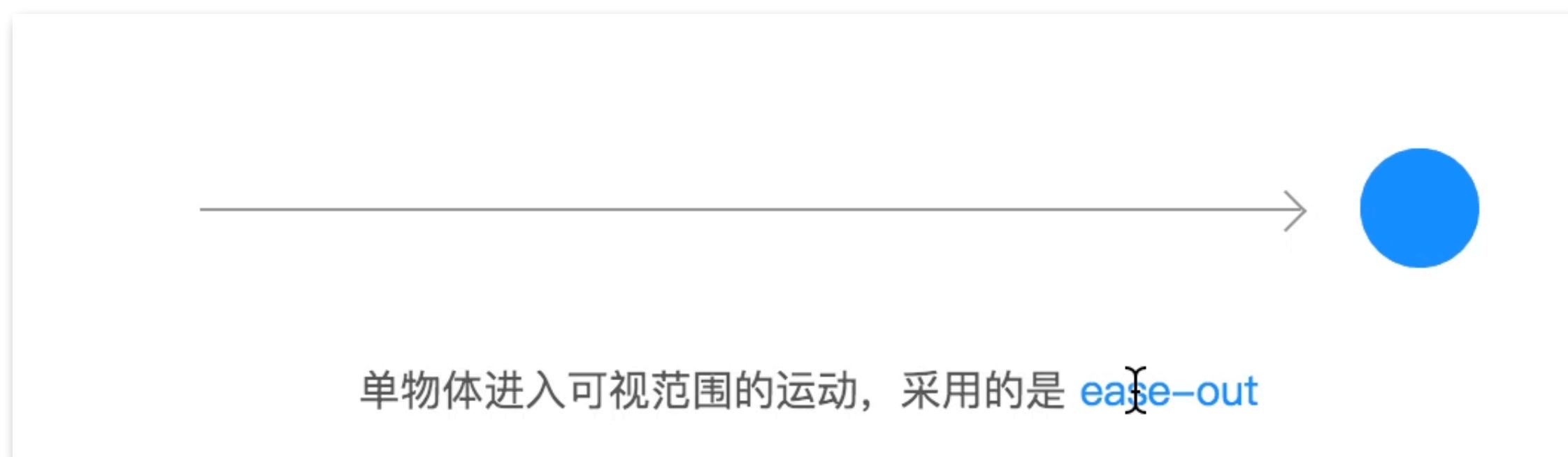


modal 的时间为 300ms

缓动：缓动与元素入场方式相关



离场动画：缓入

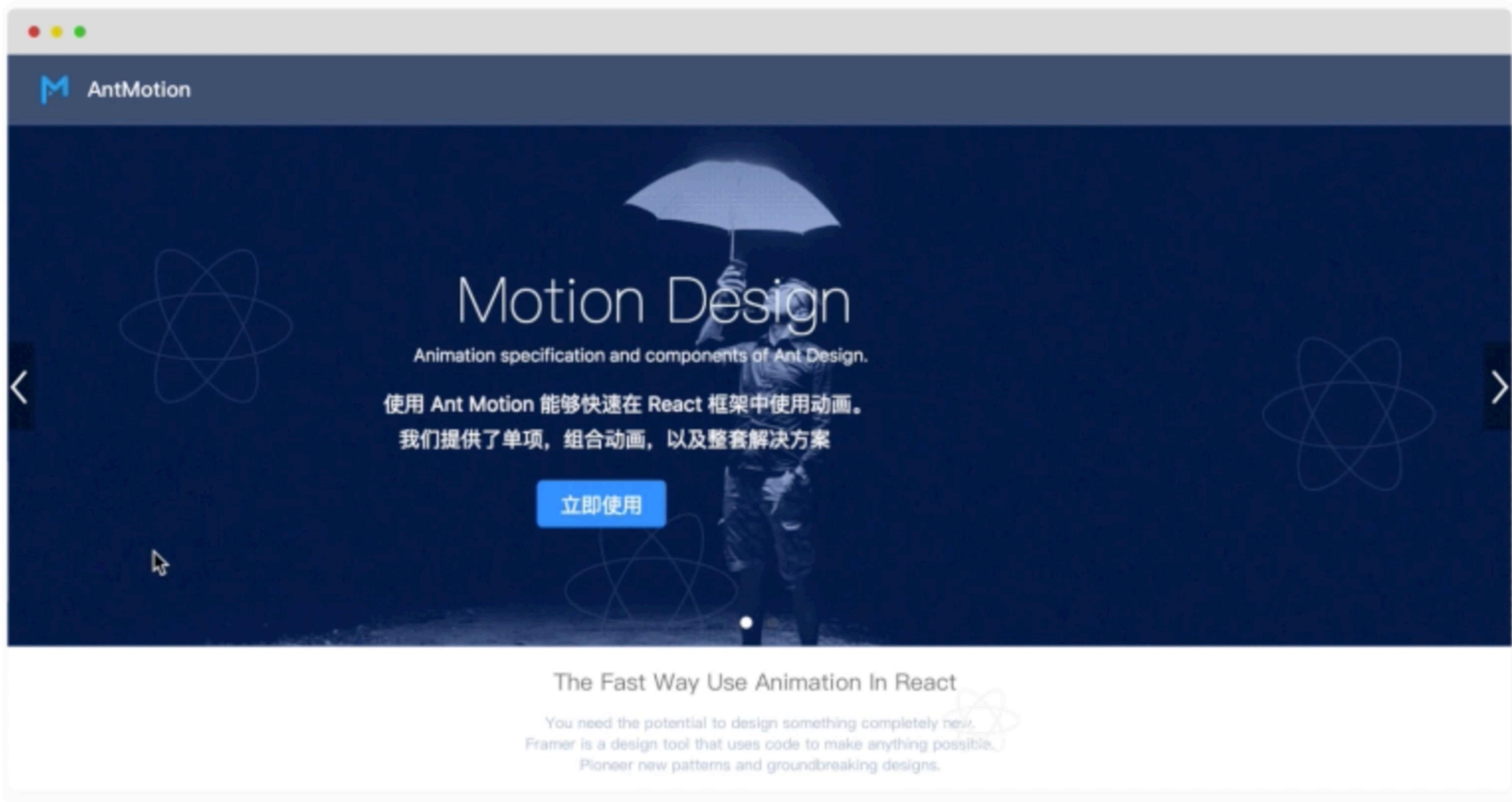


入场动画：缓出

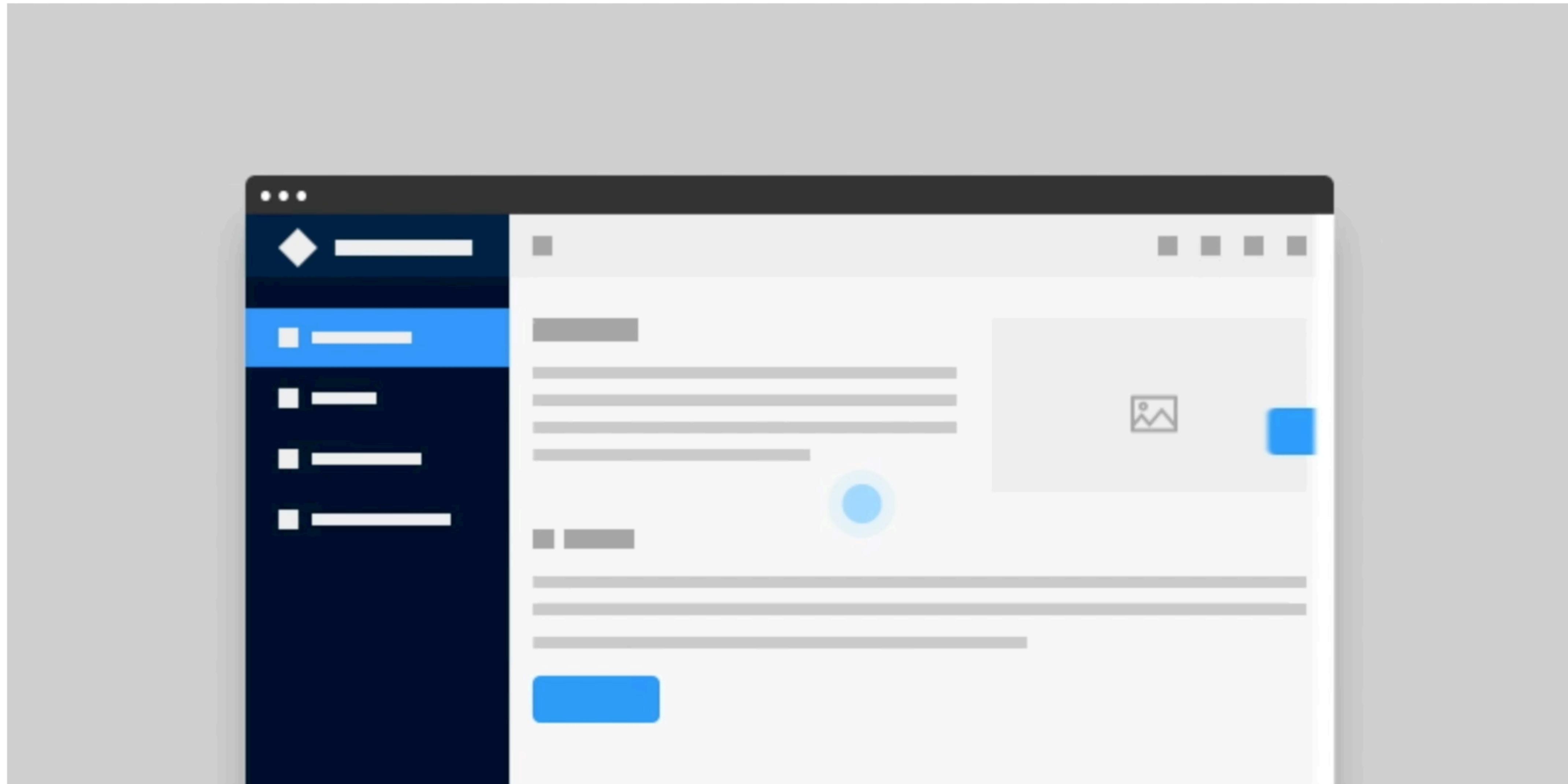


屏幕间的运动：缓入缓出

空间感：空间视差



空间感：大小视差



Combined 组合

- 单元素组合动效
- 多元素队列动效
- 多元素聚散动效

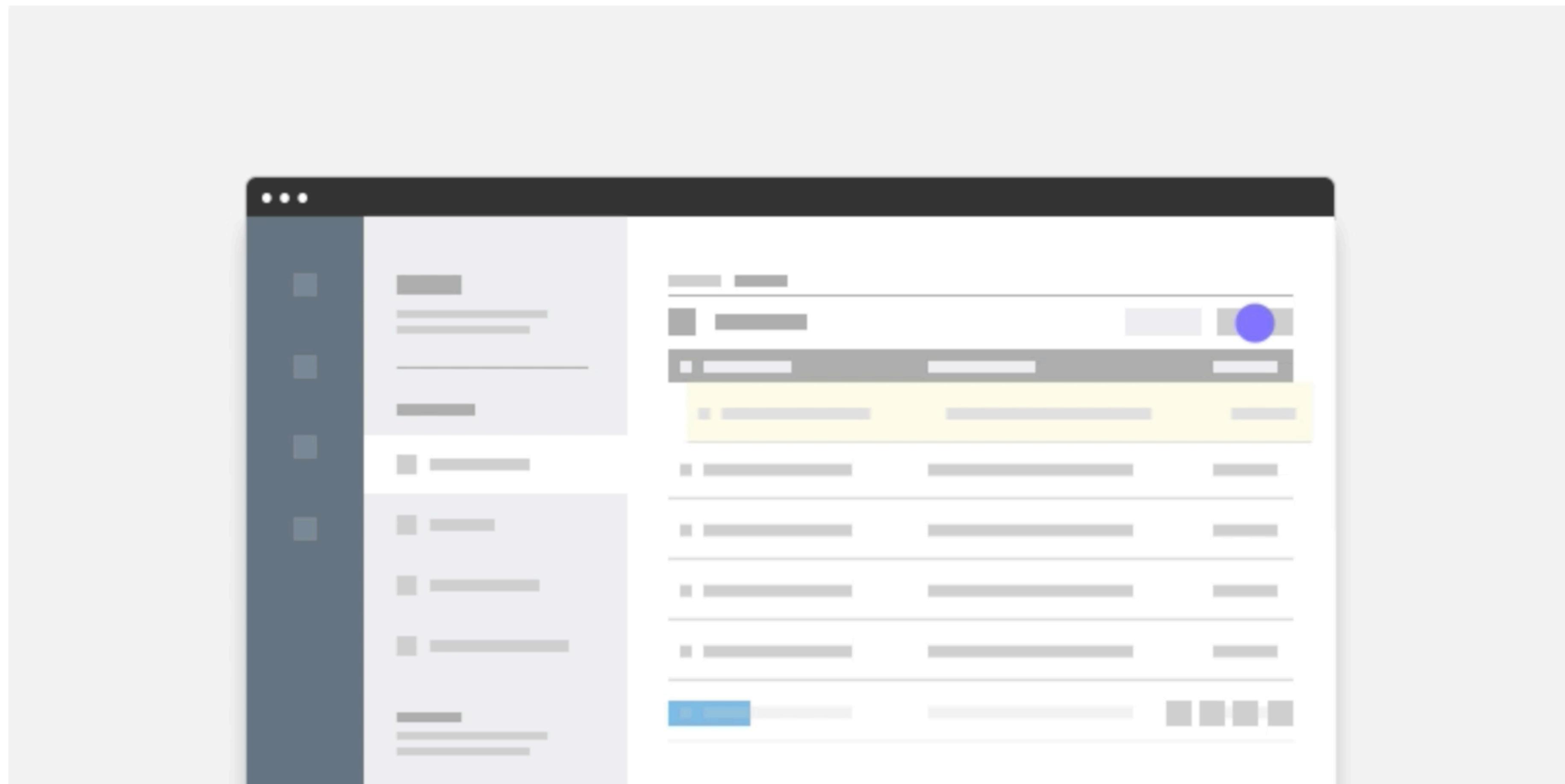
组合动效是将多种动画效果组合起来运用在单个元素或者界面中的表现方式，吸引用户的注意，让用户快速了解信息的主次关系。ant design 最常用的组合动效有以下三种：

- 单元素组合动画
- 多元素列队动画
- 多元素聚散动画

过渡：解释刚刚发生了什么

对象增加

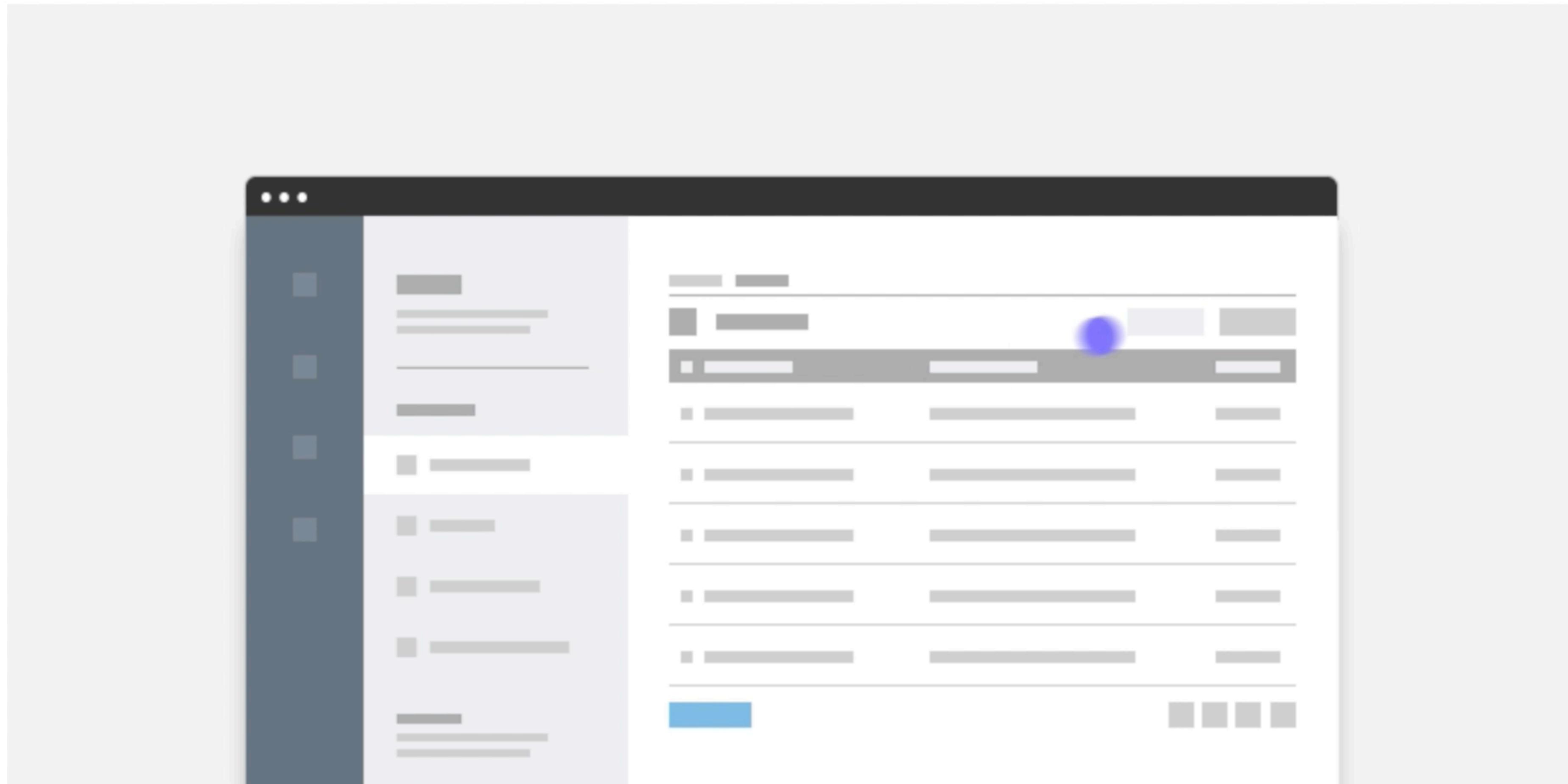
增加后，用一个动画和背景色来区分新增元素，过一段时间再恢复正常。



过渡：解释刚刚发生了什么

对象删除

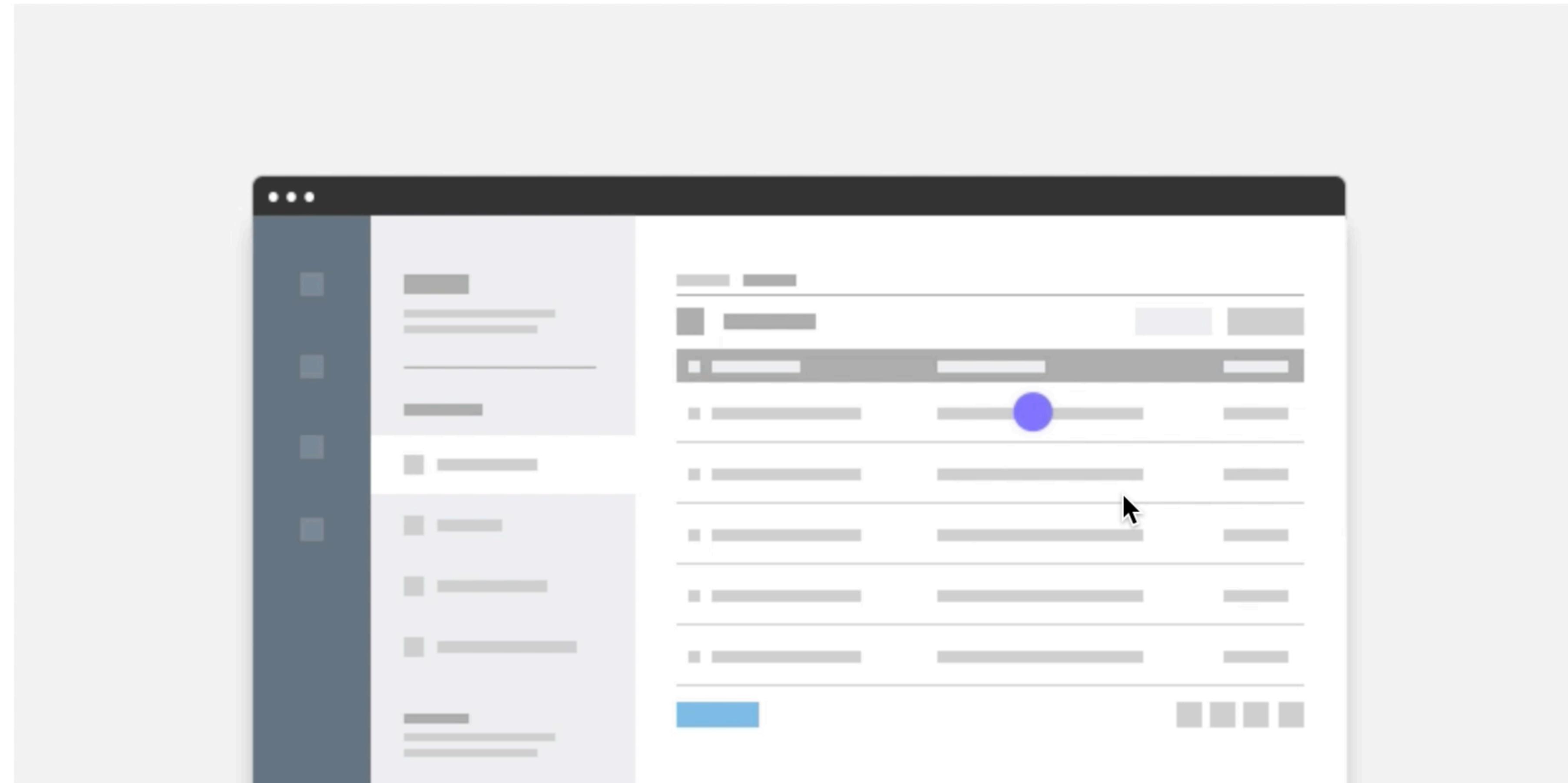
删除后，用移出的效果来做删除的效果。



过渡：解释刚刚发生了什么

对象更改

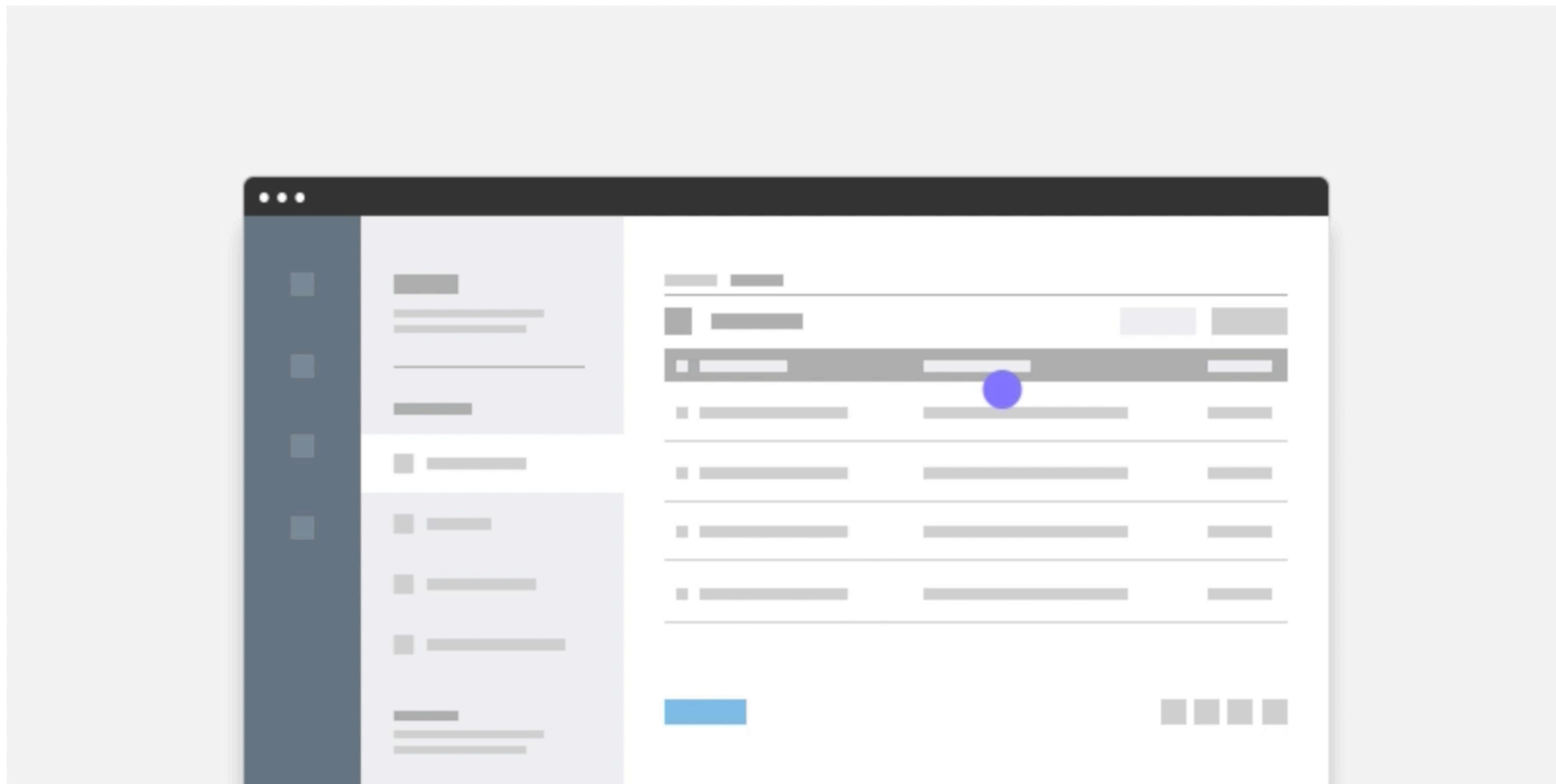
用户更改了内容时，在保存后，在修改过的位置出现背景色，表示该对象发生过变更，然后背景色持续一断时间再消失，恢复正常。



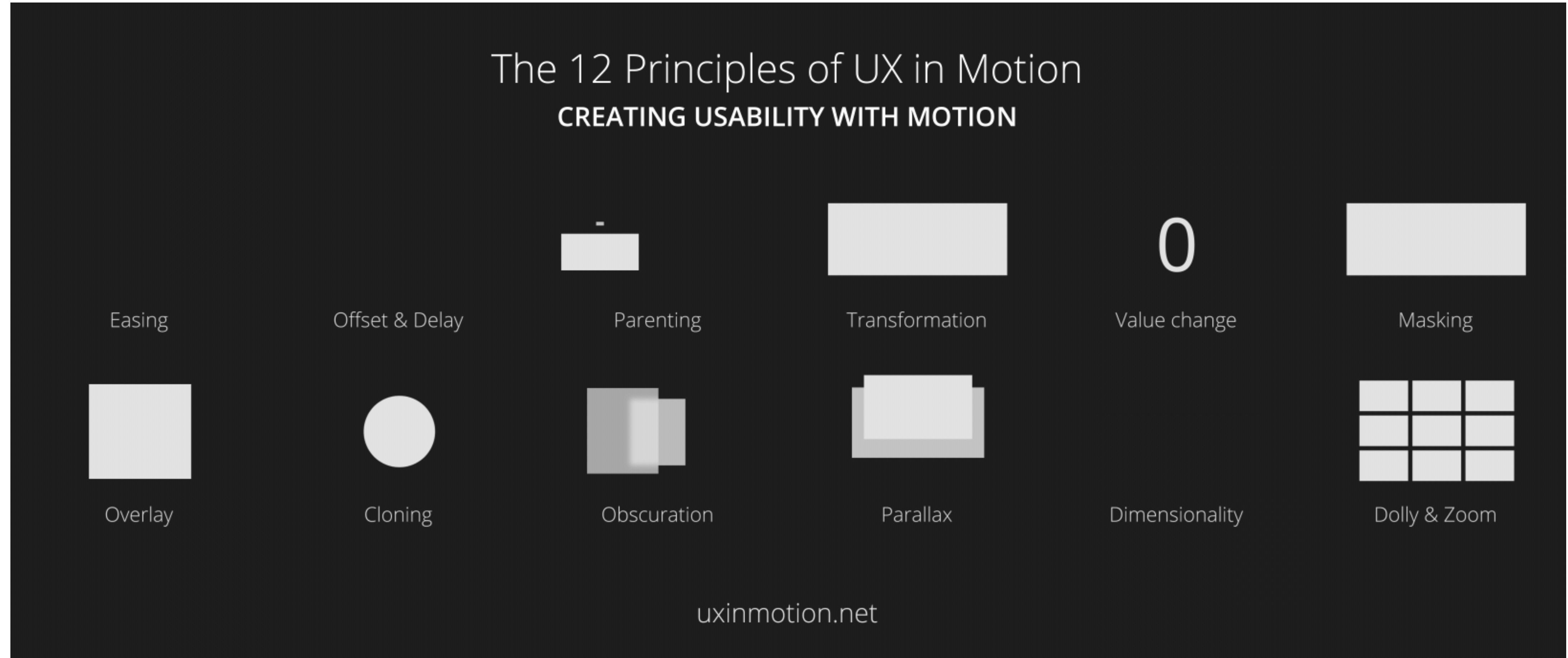
过渡：解释刚刚发生了什么

弹出框呼出

从页面的某个按钮呼出弹出框时，弹框从按钮处呼起，可提示用户弹框与按钮的关系。



Medium专题：“UX In Motion” — Creating usability with motion.



<https://medium.com/ux-in-motion>

motion带来什么？

定义与“用户发生交互”的物体的运动行为。

让用户的交互操作更加自然、流畅、清晰，并能够提升交互质量。