



Liquid Glass

СВОИМИ РУКАМИ:

разбираем системный эффект и воссоздаём на Metal

Константин Москаленко

Старший iOS разработчик

- Команда UX-развития Мобильного банка
- 7 призовых мест в Telegram Contests
- Бакалавр CS, магистр AI
- Бегун и 3+ года streak в Duolingo



Содержание

01 Разбор Liquid Glass

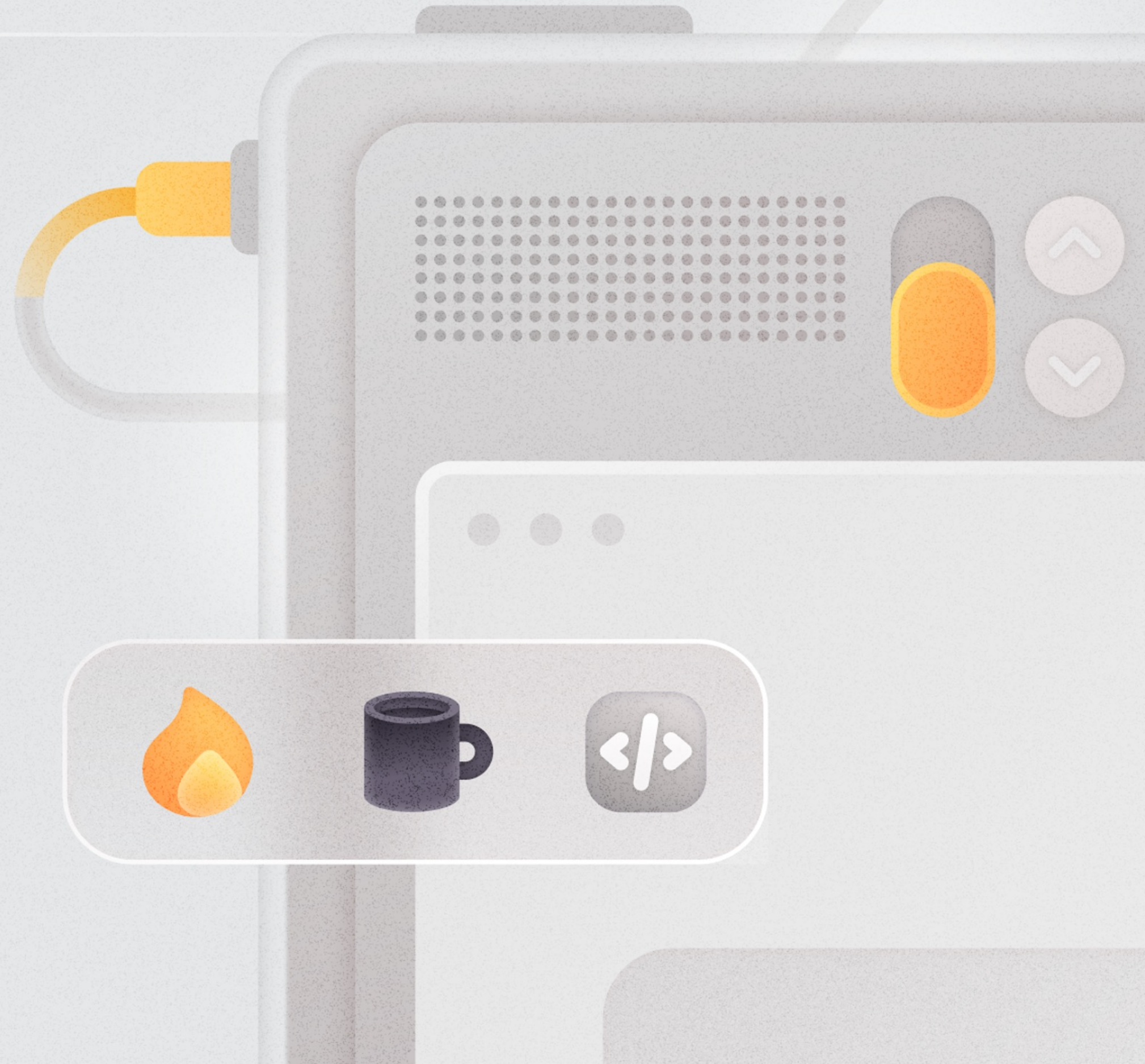
02 Реализация на Metal

03 Внедрение в Telegram

04 Анимации и итоги

01

Разбор Liquid Glass



Telegram iOS Contest (2025)

Telegram Contests, <https://t.me/contest/436>



Telegram Contests

Prize fund: Up to \$50,000

Deadline: 23:59 Dubai time on December 26, 2025

Who can participate: Everyone

Telegram is hosting a contest for iOS developers to bring Liquid Glass effects to older versions of iOS. The winner will get a chance to join the Telegram Team.

Developers at Telegram can earn up to \$1,000,000 per year before taxes.

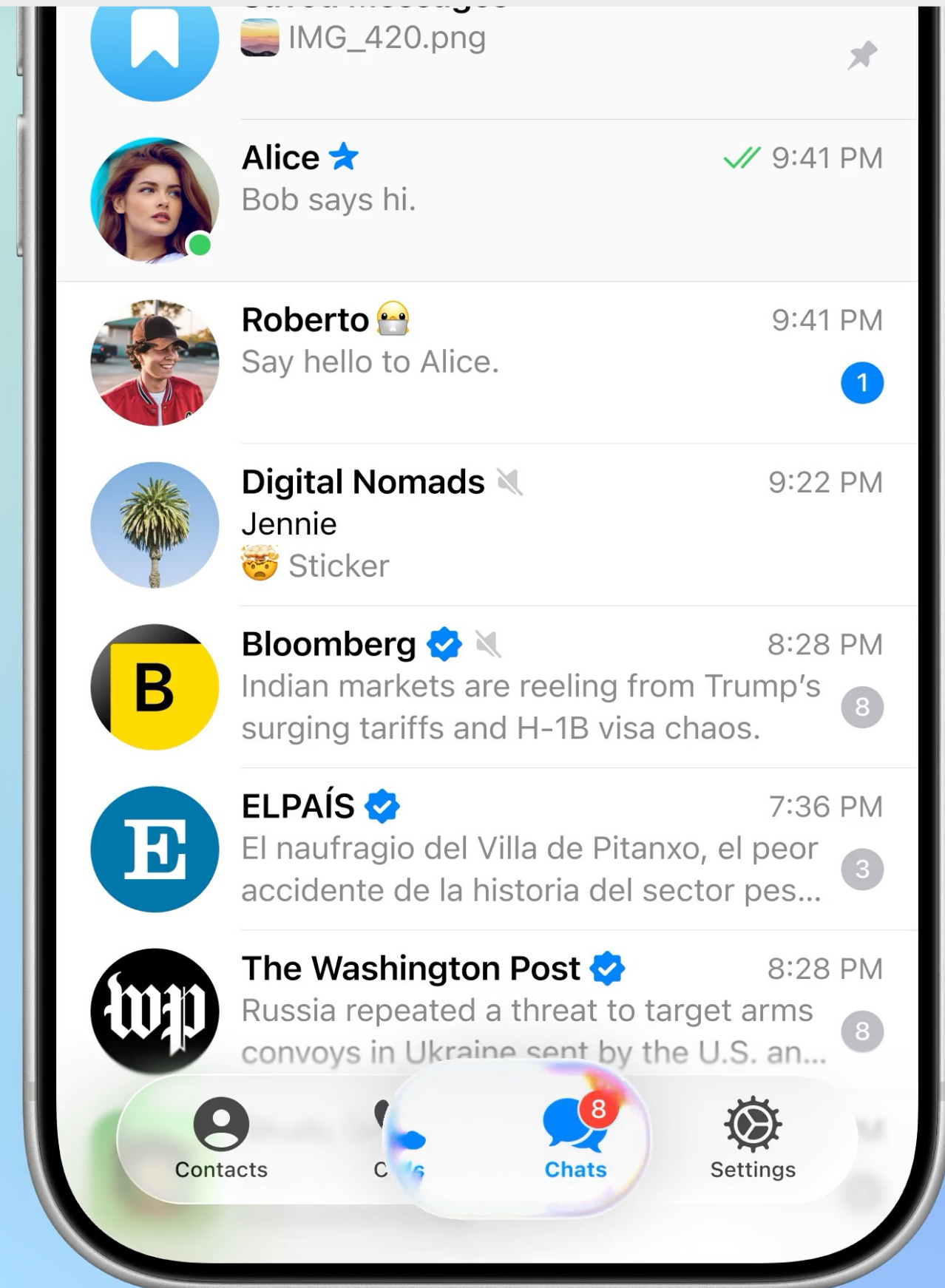
THE TASK

Implement *custom versions* of some of the **Liquid Glass** effects and the relevant interface flows into **Telegram for iOS** so these appearances work on iOS 18 and older.

You must precisely replicate the animations and look and feel (*highlight on tap, scale-up, bounce, and stretching*) on the glass elements.

t.me/contest/436

235K 👁 edited Dec 4, 2025 at 10:59



НАТИВНЫЙ

Liquid Glass

Нативный Liquid Glass в iOS 26

```
let container = UIGlassContainerEffect()
let containerView = UIVisualEffectView(
    effect: container
)

let glassEffect = UIGlassEffect()
let view1 = UIVisualEffectView(effect: glassEffect)
let view2 = UIVisualEffectView(effect: glassEffect)

containerEffectView.contentView.addSubview(view1)
containerEffectView.contentView.addSubview(view2)
```

Apple Inc., <https://developer.apple.com/videos/play/wwdc2025/284/>



Обзор иерархии UIVisualEffectView

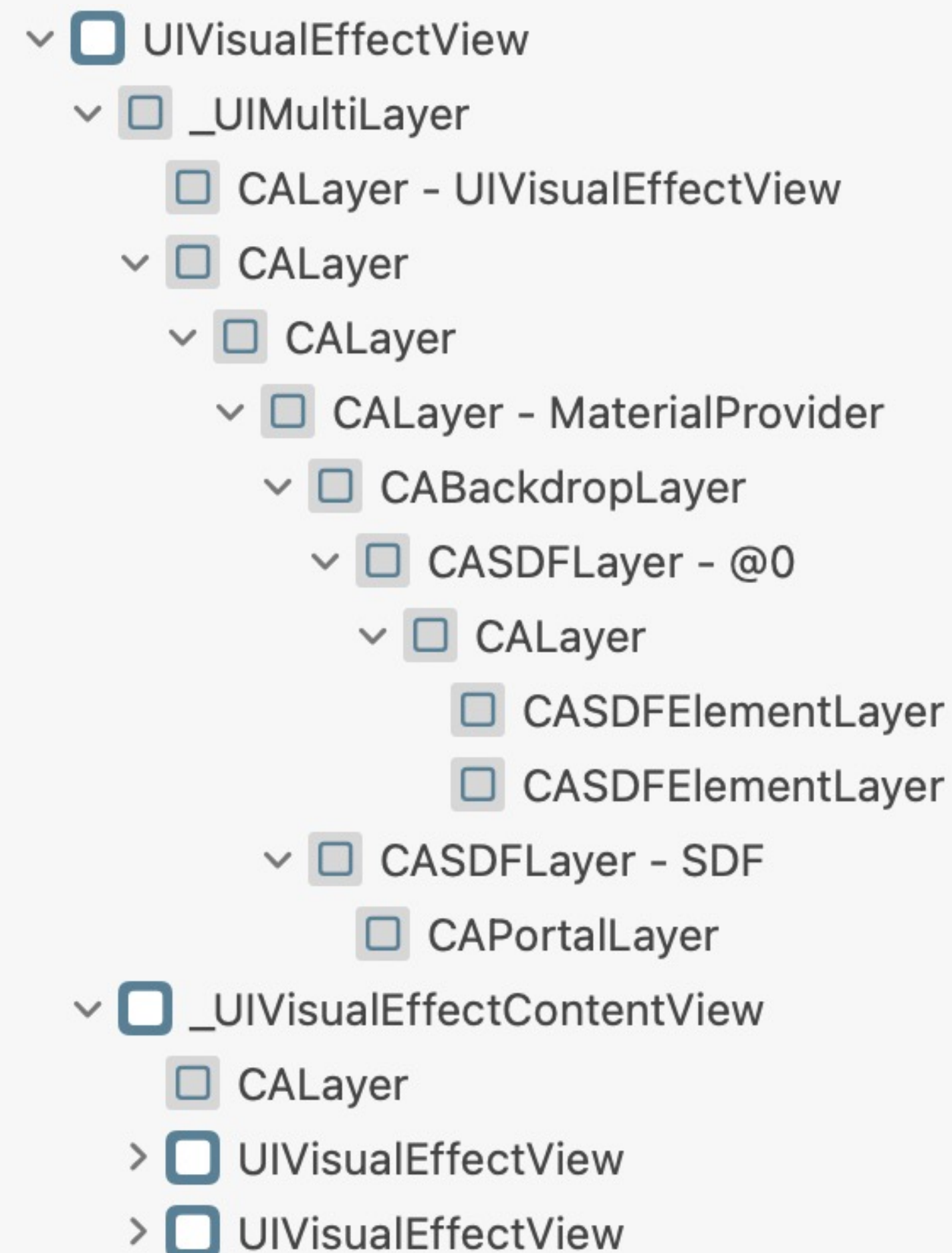
CABackdropLayer: Отрисовывает стеклянный фон с эффектом преломления (фильтр glassBackground)

CASDFLayer (@0): Предоставляет SDF текстуру для backdrop слоя через CASDFOutputEffect

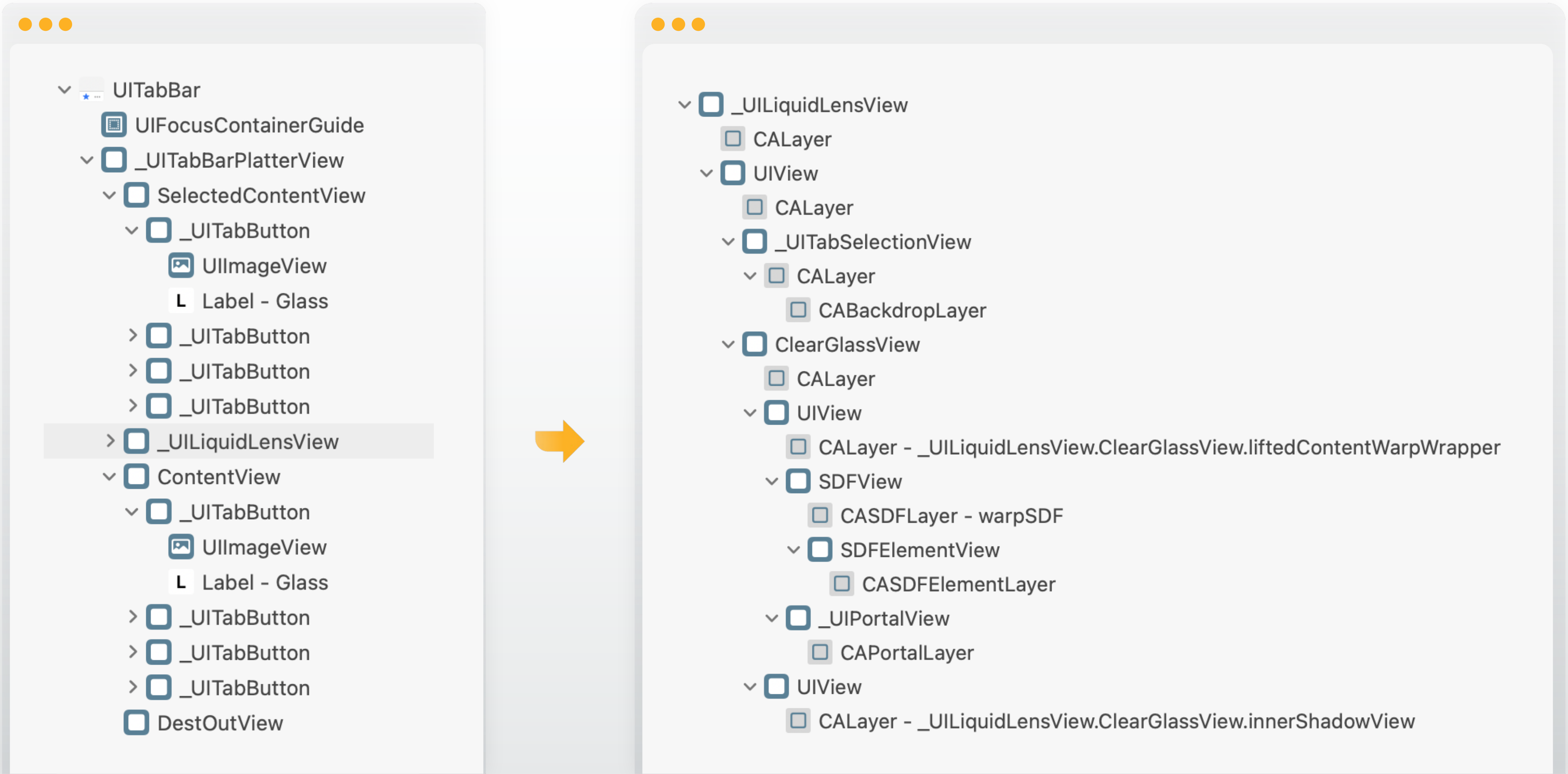
CASDFElementLayer:
Определяет форму SDF

CASDFLayer (SDF): Добавляет эффект подсветки на рамке стекла

CAPortalLayer: Вариант CAPortalLayer для повтора контента CASDFElementLayer в эффекте подсветки



Обзор иерархии UITabBar



Анатомия эффекта

Преломление

Закон Снеллиуса:

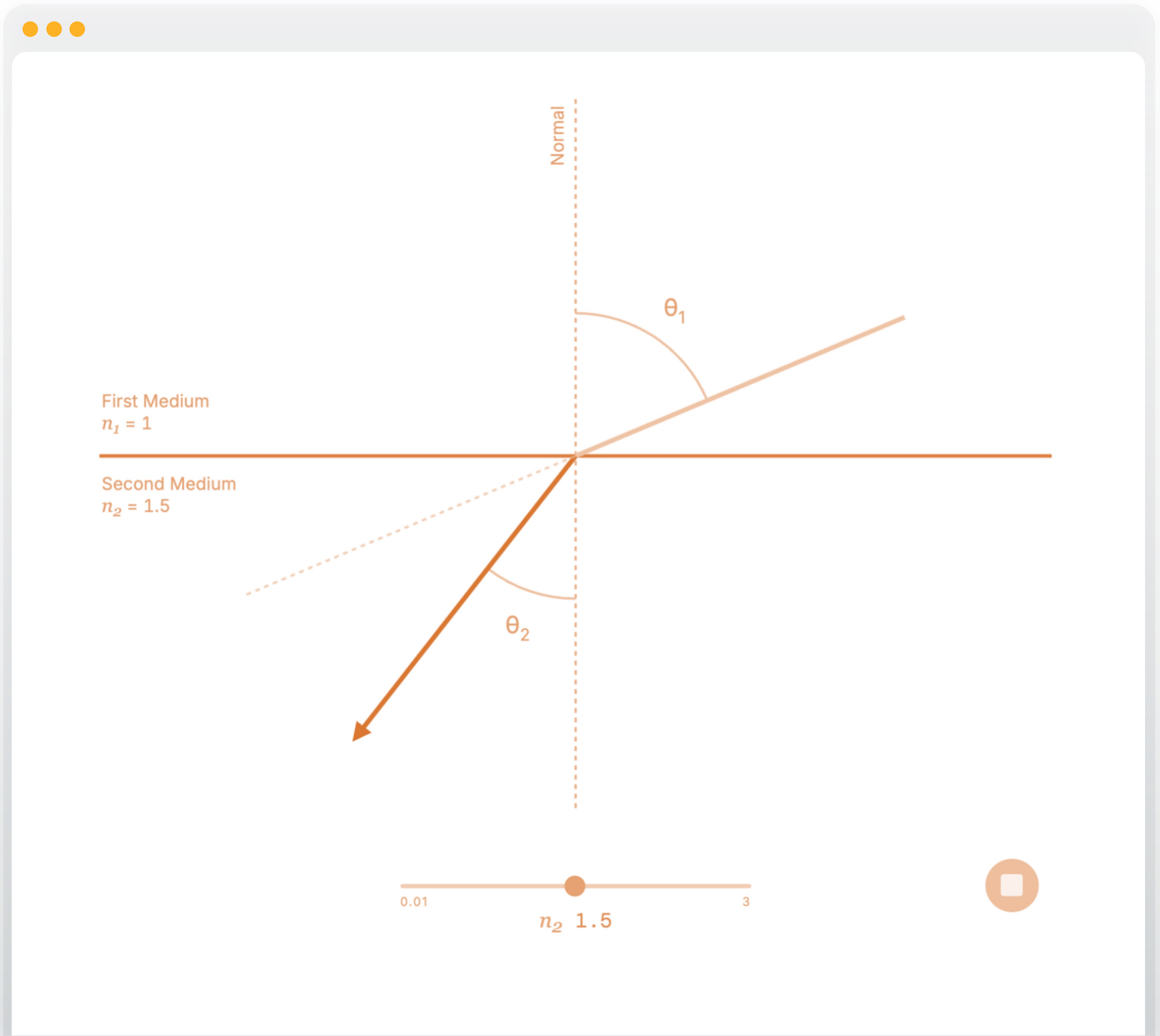
$$n_1 \sin(\theta_1) = n_2 \sin(\theta_2)$$

n_1 — показатель преломления
первой среды (воздух = 1)

θ_1 — угол падения света

n_2 — показатель преломления
второй среды (стекло = 1.5)

θ_2 — угол преломления света



Функция поверхности

Функция поверхности принимает значение от 0 (внешний край) до 1 (начало плоской части) и возвращает толщину стекла в заданной точке.

Chris Feijoo, <https://kube.io/blog/liquid-glass-css-svg>

Convex Circle

$$y = \sqrt{1 - (1 - x)^2}$$



Convex Squircle

$$y = \sqrt[4]{1 - (1 - x)^4}$$

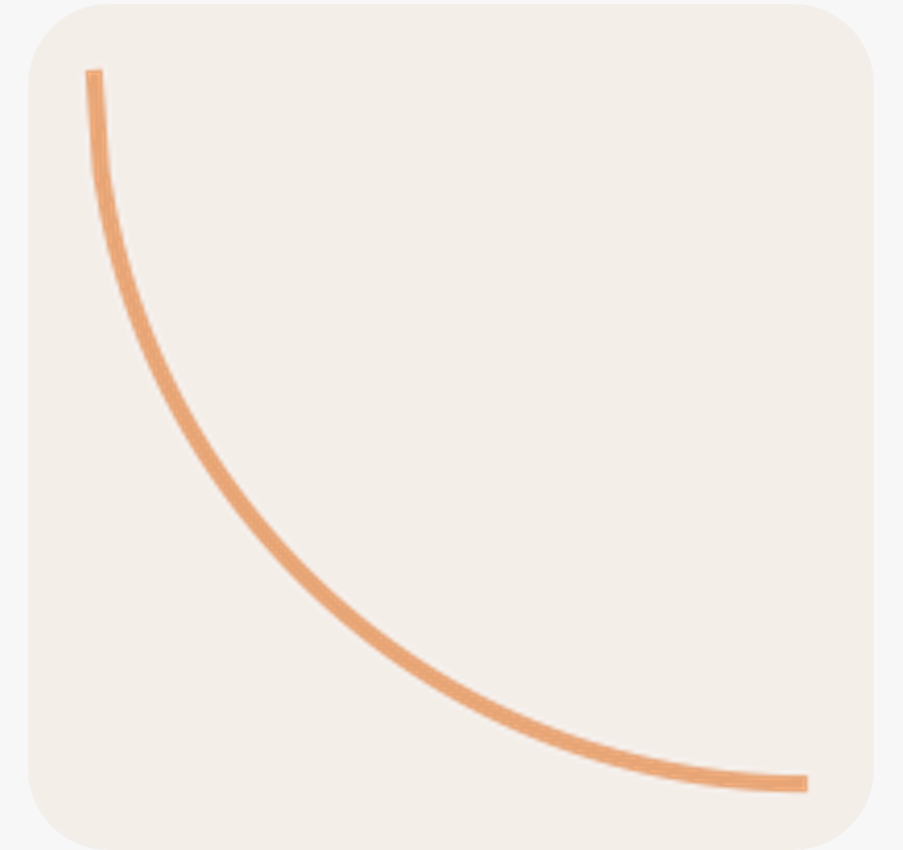


Функция поверхности

Функция поверхности принимает значение от 0 (внешний край) до 1 (начало плоской части) и возвращает толщину стекла в заданной точке.

Concave

$$y = 1 - \text{Convex}(x)$$



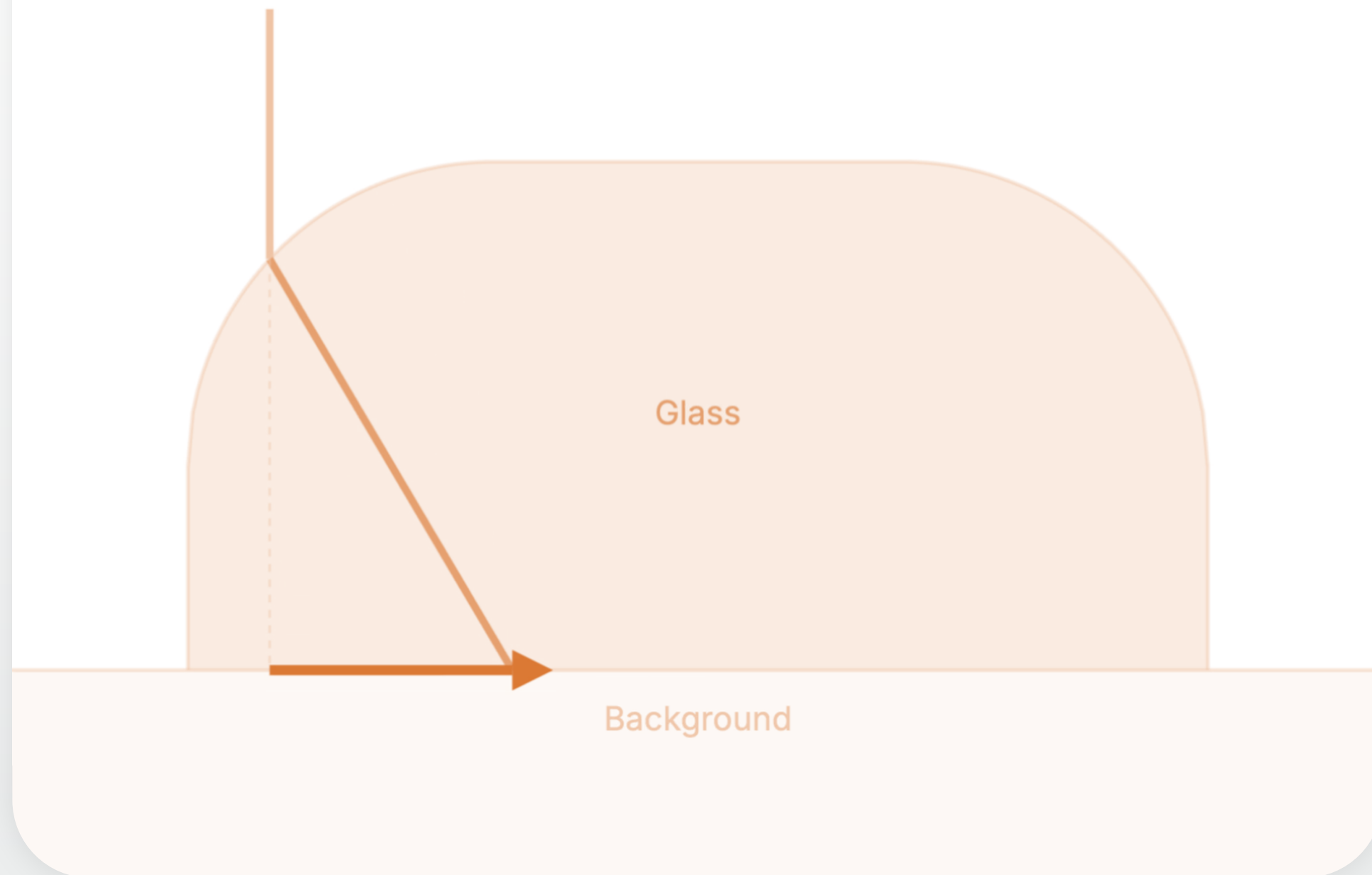
Lip

$$y = \text{mix} \left(\begin{array}{c} \text{Convex}(x), \\ \text{Concave}(x), \\ \text{Smootherstep}(x) \end{array} \right)$$

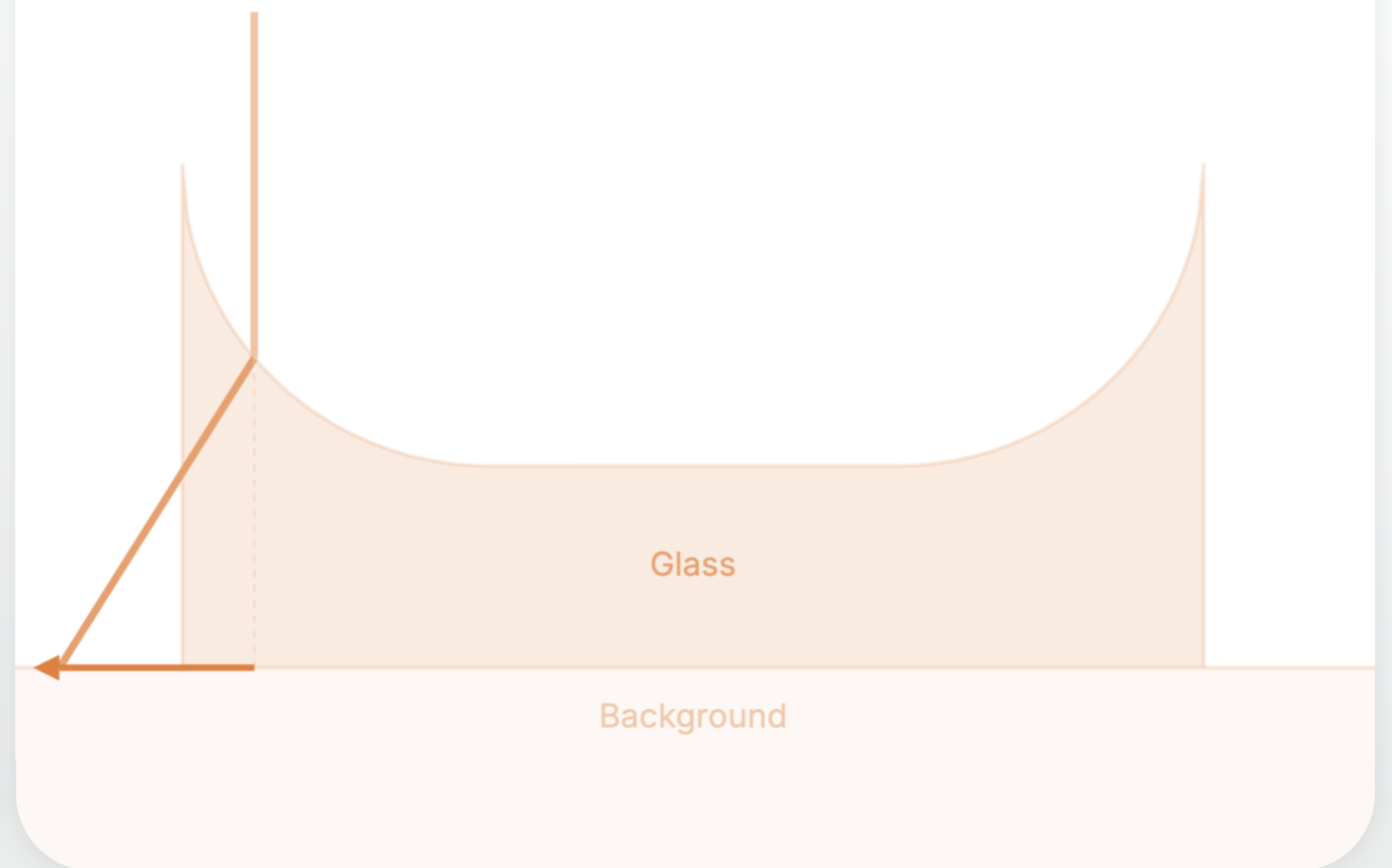


Функции поверхности и преломление

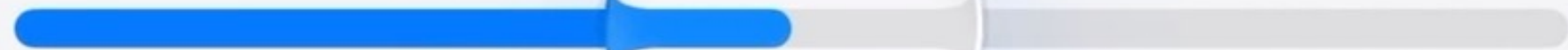
Выпуклые



Вогнутые

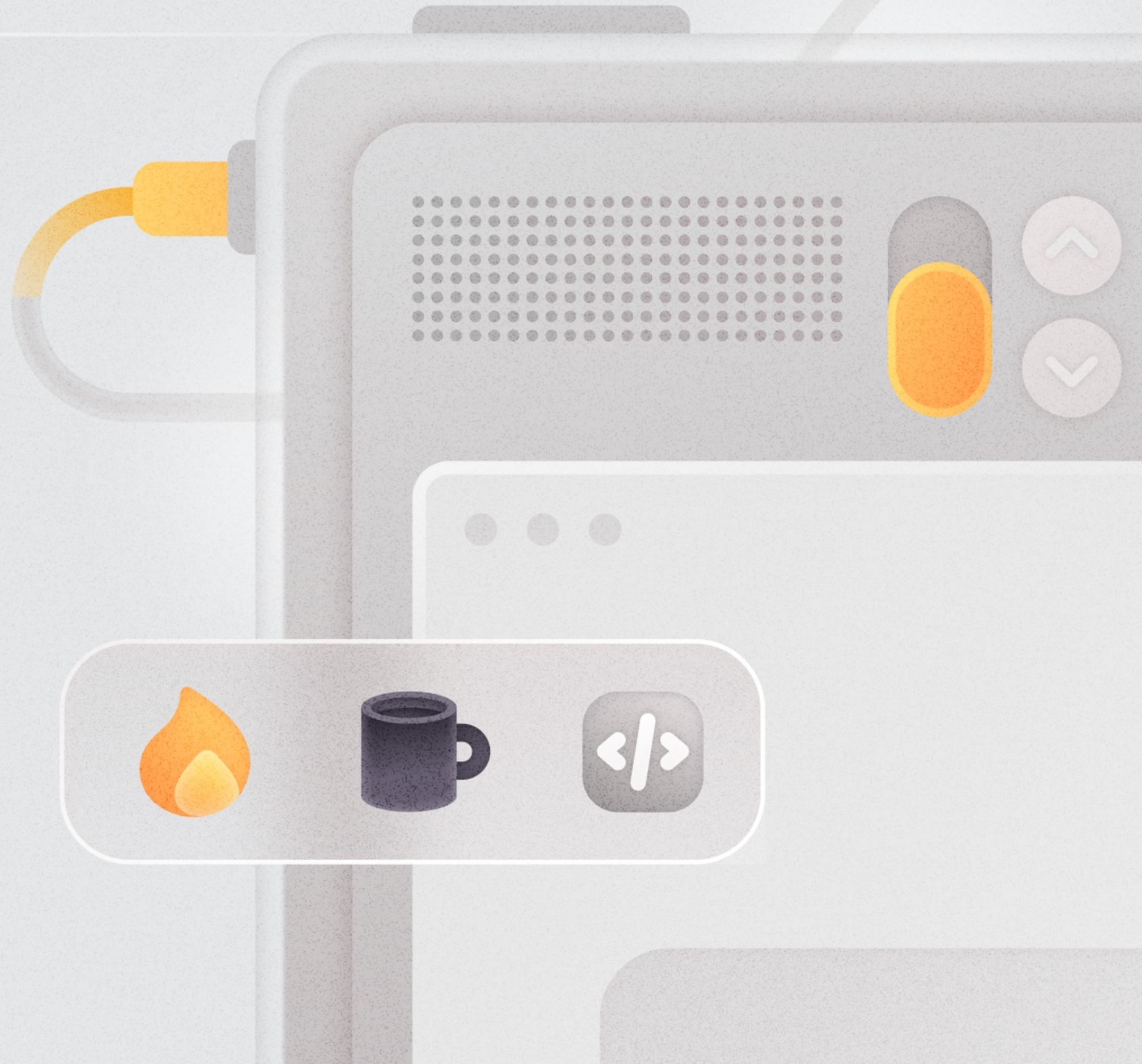


Системные компоненты



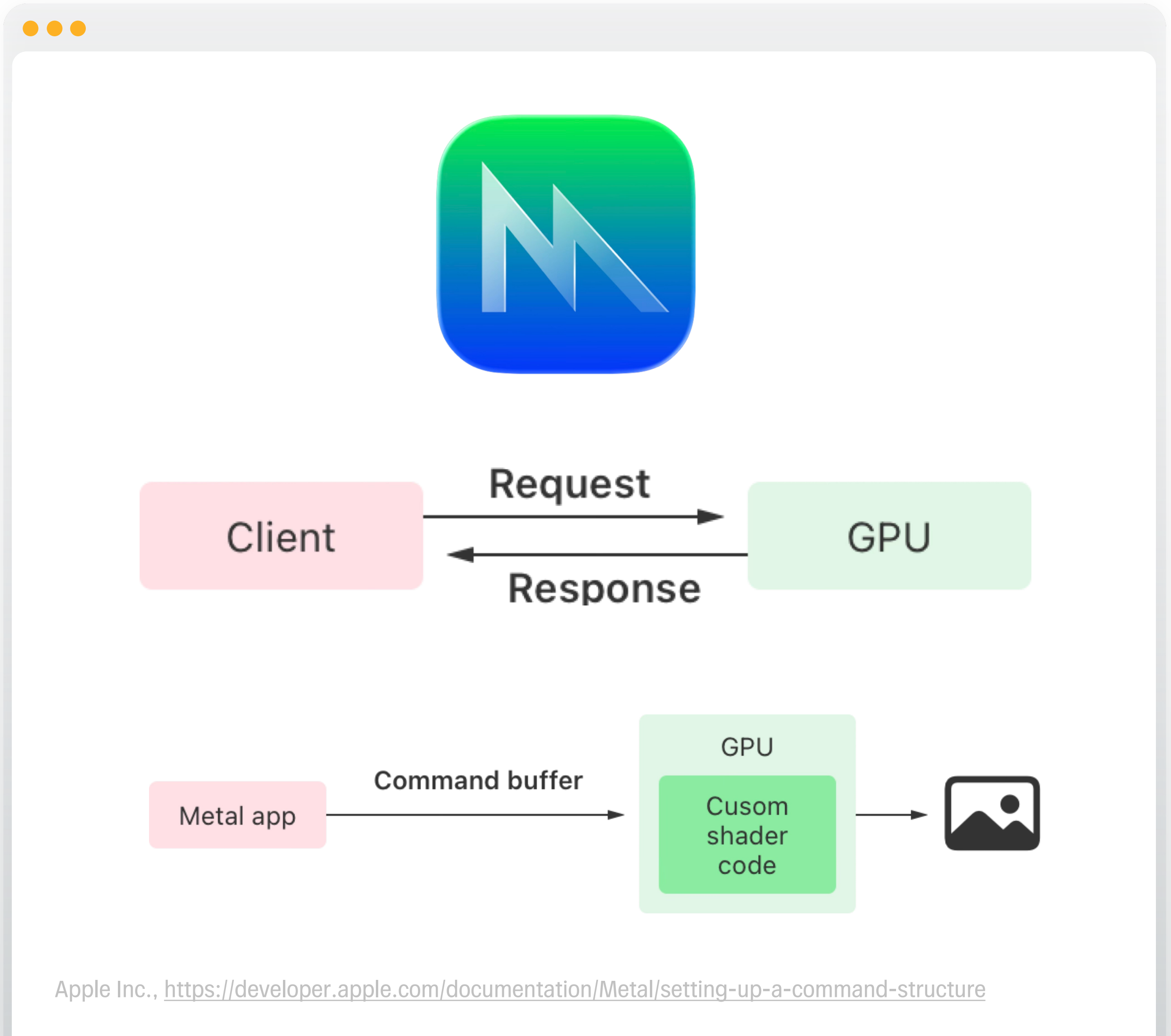
02

Реализация на Metal



Введение в Metal

- ➔ Низкоуровневый GPU API от Apple
- ➔ Клиент-серверная архитектура
- ➔ Единая модель для графики и вычислений
- ➔ Глубокая интеграция с платформой



Виды шейдеров



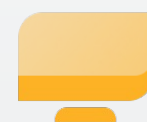
Вершинный

Обрабатывает каждую вершину входных данных, преобразует координаты



Фрагментный

Определяет итоговый цвет каждого пикселя на экране

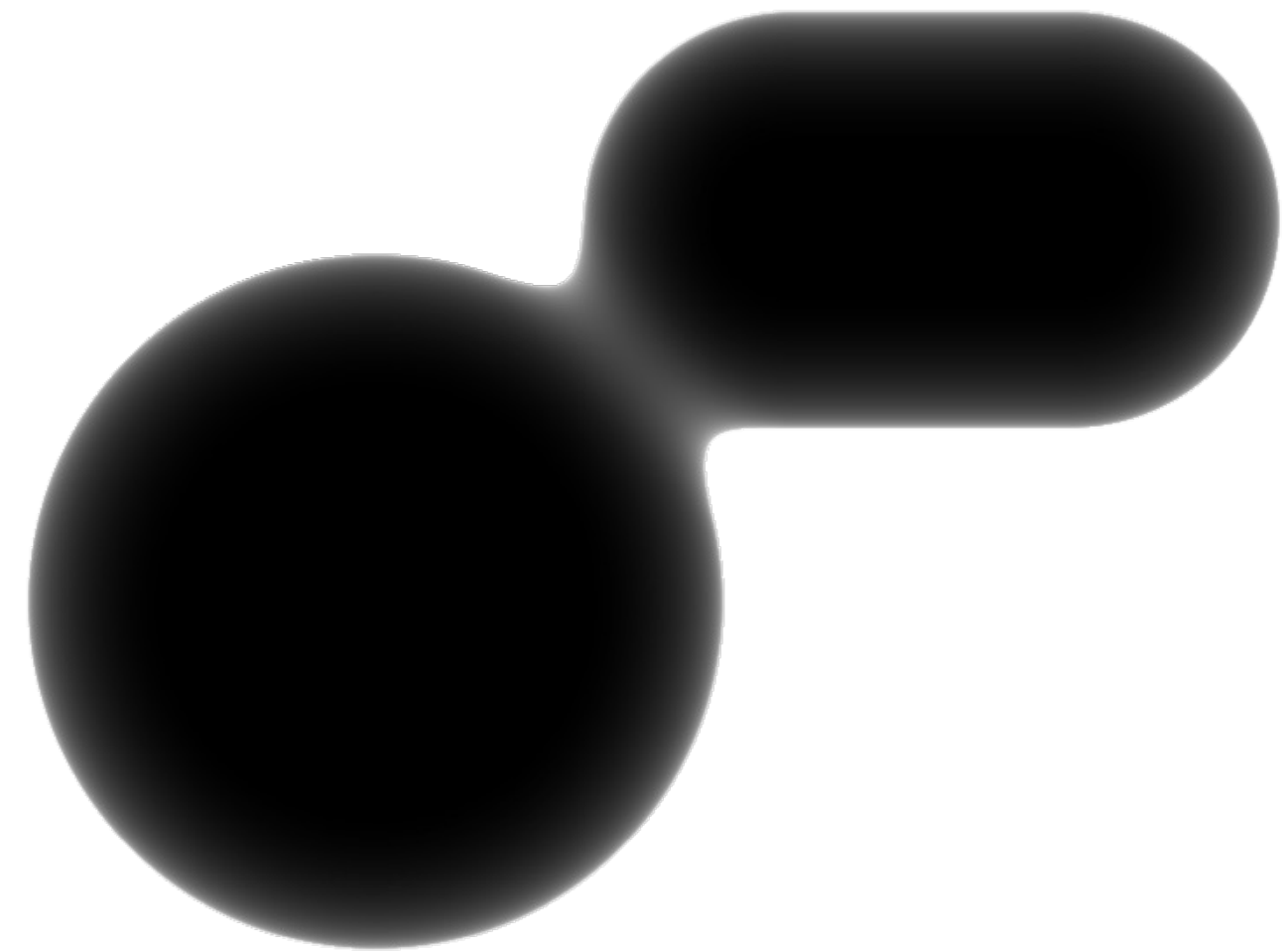


Вычислительный

Выполняет произвольные вычисления на GPU, без привязки к рендерингу


```
float mainSDF(float2 p1, float2 p2)
{
    float d1 = roundedRectSDF(
        p1, size, radius);
    float d2 = roundedRectSDF(p2, ...);
    return smin(d1, d2, mergeRate);
}

float roundedRectSDF(
    float2 p, float2 s, float r)
{
    float2 d = abs(p) - s * 0.5 + r;
    return length(max(d, 0.))
        + min(max(d.x, d.y), 0.) - r;
}
```



```
float d = fabs(mainSDF(
    p1 + shadowOffset,
    p2 + shadowOffset,
    fragCoord
));

float shadow =
    shadowFactor *
    exp(-d / shadowExpand);
float shadowAlpha =
    clamp(shadow, 0.0, 1.0);

bgColor *= 1.0 - shadowAlpha;
```




```
float4 color = sample(uv)
               * blurWeights[0];

for (int i = 1; i <= radius; i++)
{
    float w = blurWeights[i];
    float2 offset = i * texelSize;

    color += sample(uv + offset) * w;
    color += sample(uv - offset) * w;
}
```



```
float edge = computeEdgeFactor(  
    merged,  
    refThickness,  
    refFactor  
);  
  
float2 delta =  
    normal * edge * distortionScale;  
  
float4 glassColor =  
    sampleBlurredBg(uv - delta);
```




```
float fresnel = pow(
    edgeTerm + fresnelHardness,
    5.0);
float glare = pow(
    angleTerm(normal, glareAngle)
    * glareFactor,
    glarePower);

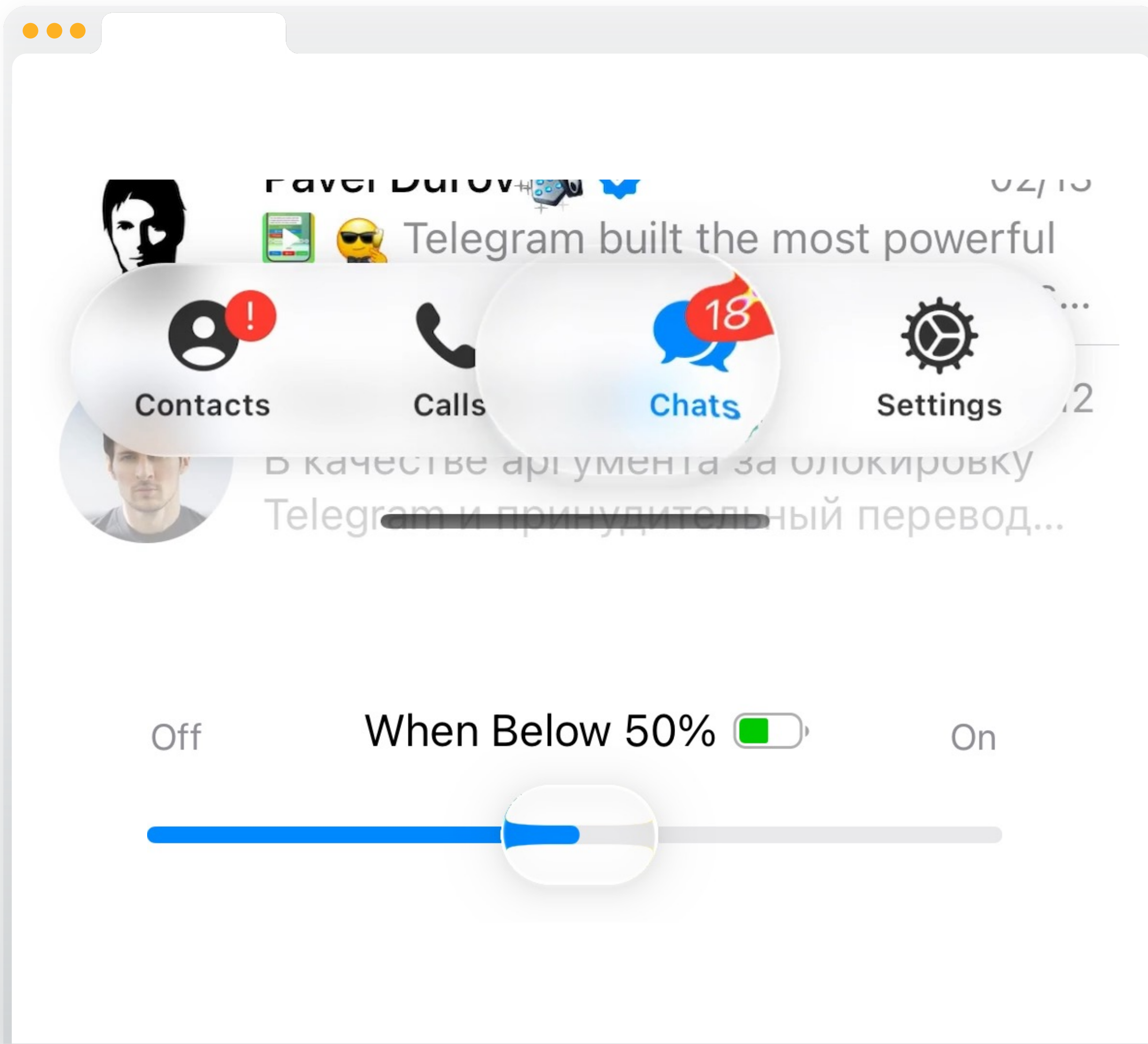
color = mix(
    color, white,
    fresnel * fresnelAmount);
color = mix(
    color, white,
    glare * glareGeoMask);
```



```
float4 glass = sampleDispersion(  
    blurredBg,  
    uv - delta,  
    refDispersion  
);  
glass = applyTint(  
    glass, tintRGB, tintAmount  
);  
  
float edgeBlend = smoothstep(  
    -edgeEps, edgeEps, merged  
);  
outColor = mix(  
    glass, bgColor, edgeBlend  
);
```

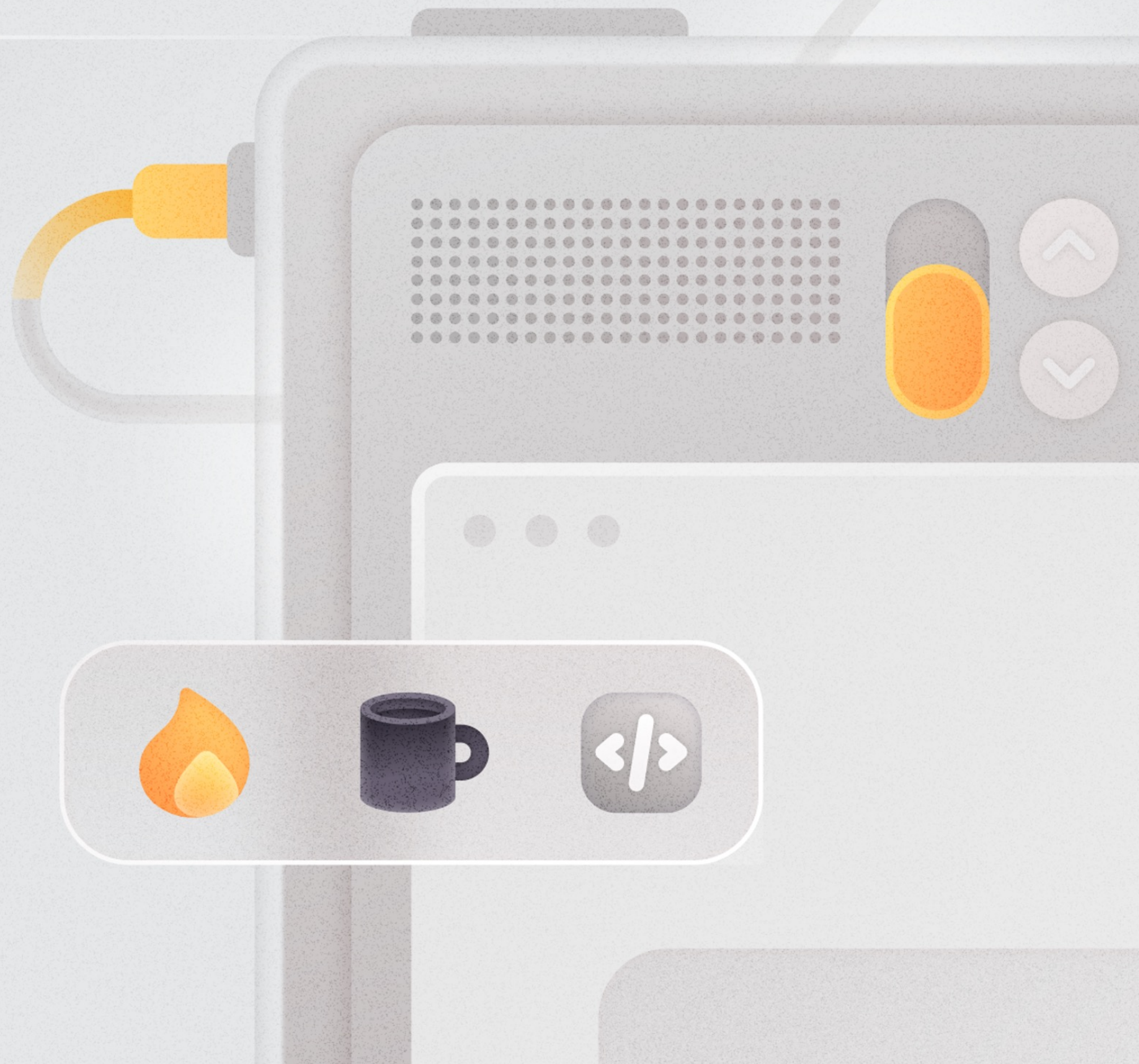


Демонстрация компонентов



03

Интеграция в Telegram





Особенности кодовой базы



Open-source без документации и внешних контрибуций



Собственный UI-фреймворк вместо стандартного UIKit



Многоязычная кодовая база (Swift, Objective-C, C, C++)



Монолитные классы на тысячи строк и сотни связанных модулей



Bazel как система сборки и медленная работа Xcode

Точки интеграции



Tab Bar

Основная панель вкладок на главном экране (контакты, звонки, чаты, настройки)



Switch / Slider

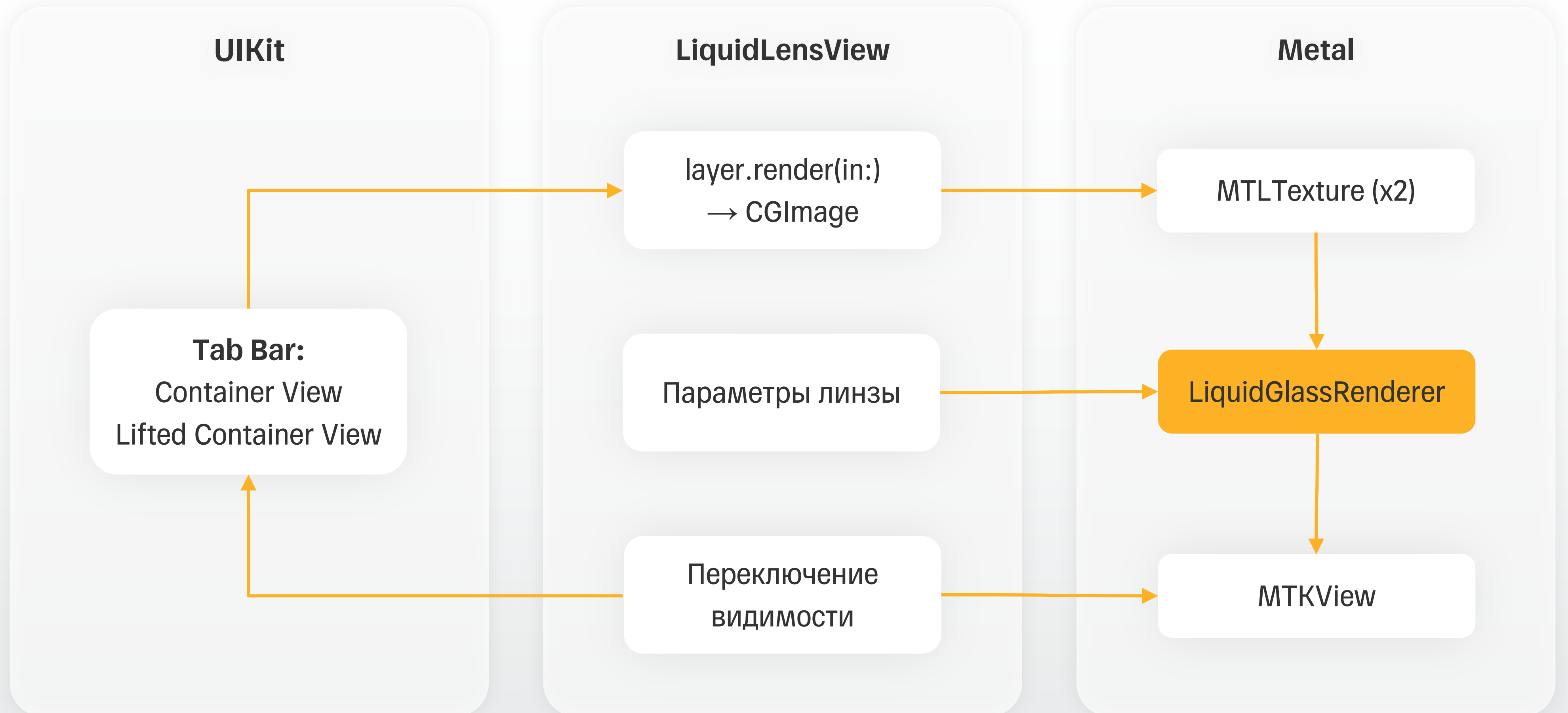
Переключатели и ползунки в настройках приложения



Buttons

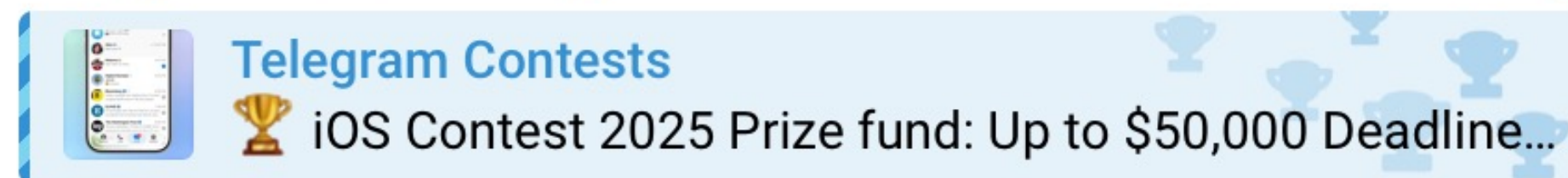
Кнопки и стеклянные панели в приложении (внутри каналов и чатов)

Общая схема интеграции





Telegram Contests



So, fellas.

We never required an exact replication of the Glass core; however, as you can see, true refraction is fully achievable even on older iOS versions.

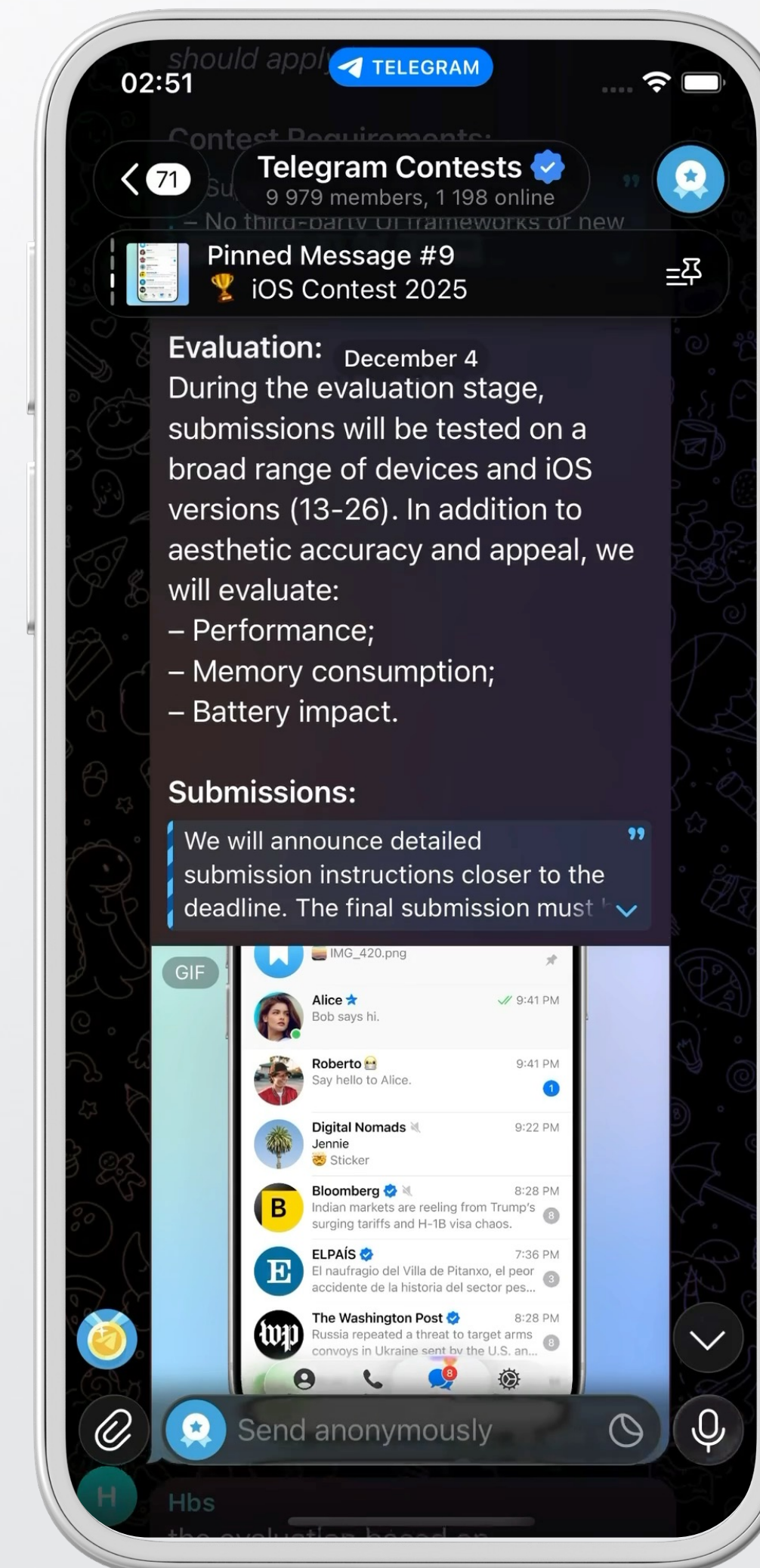
Moreover, it is not only achievable – it has already been successfully implemented into Telegram for iOS and is currently live on TestFlight.

Go check it out. For appreciation and praising, please use the reaction buttons.



t.me/contests/264893

Dec 16, 2025 at 10:36



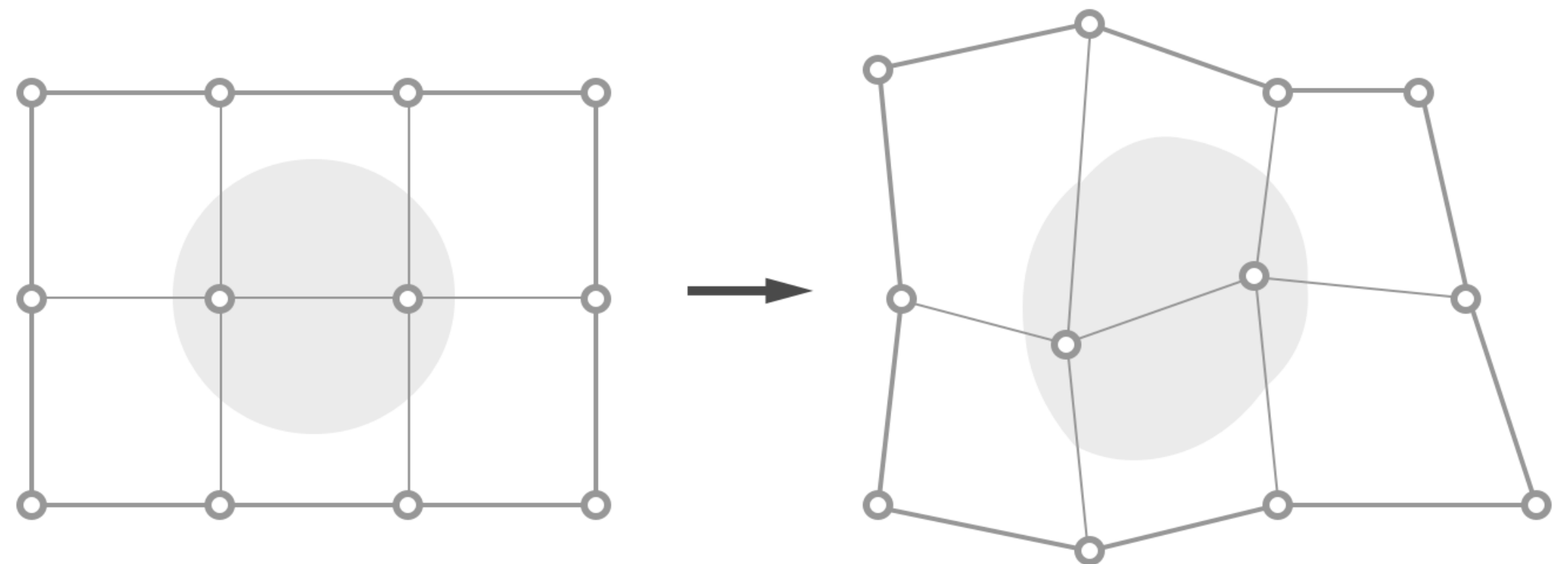
Как Telegram это сделали?

Private API

➡ `CABackdropLayer`

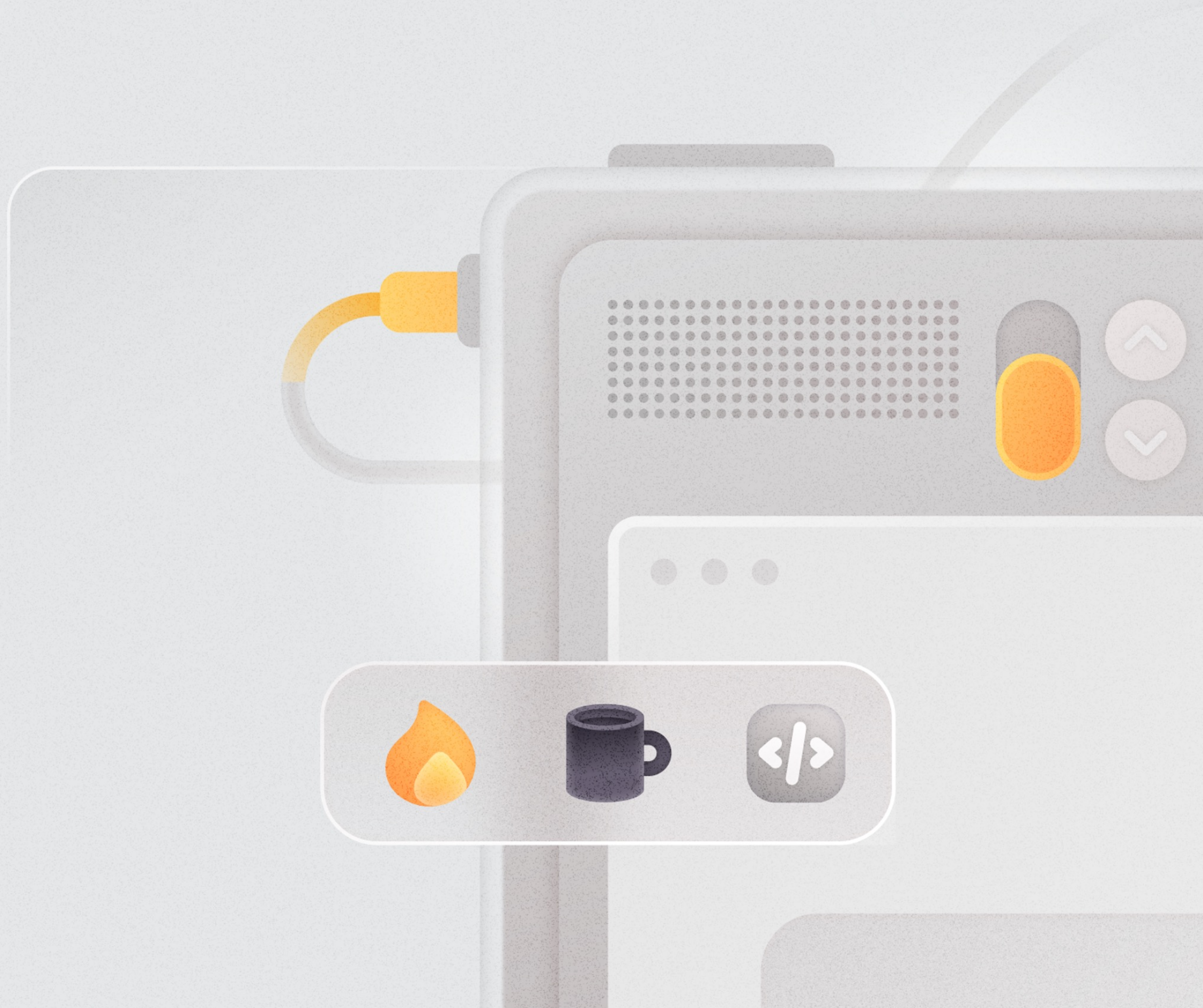
➡ `CAFilter(name:
kCAFilterGaussianBlur)`

➡ `CAMeshTransform`



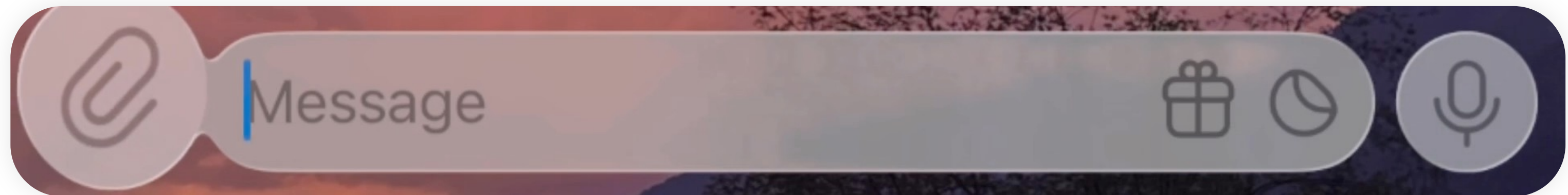
04

Анимации и итоги





Highlight on tap, scale-up, bounce, and stretching



Telegram Contests, <https://t.me/contests/263785>

Интерактивные анимации



UILongPressGestureRecognizer

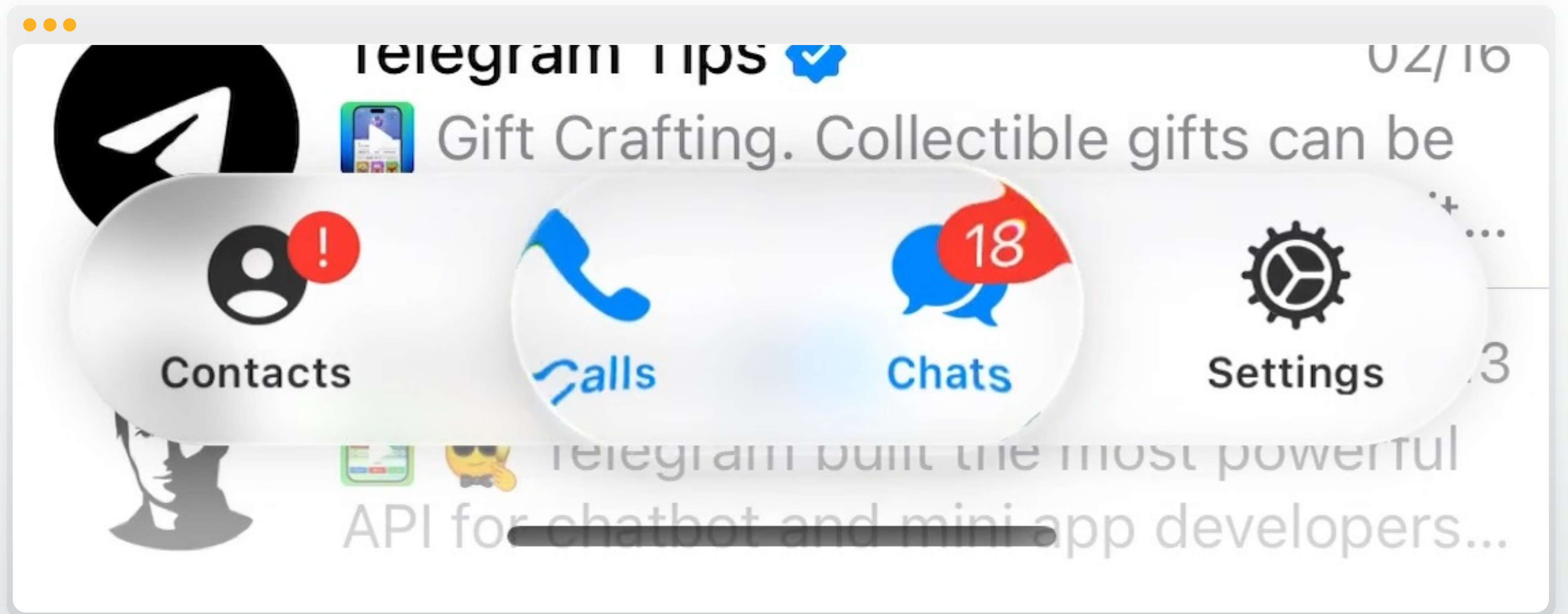
- .began → press(); .ended → release()
- Drag translation → Transform (scale)
- Радиальный градиент в точке касания



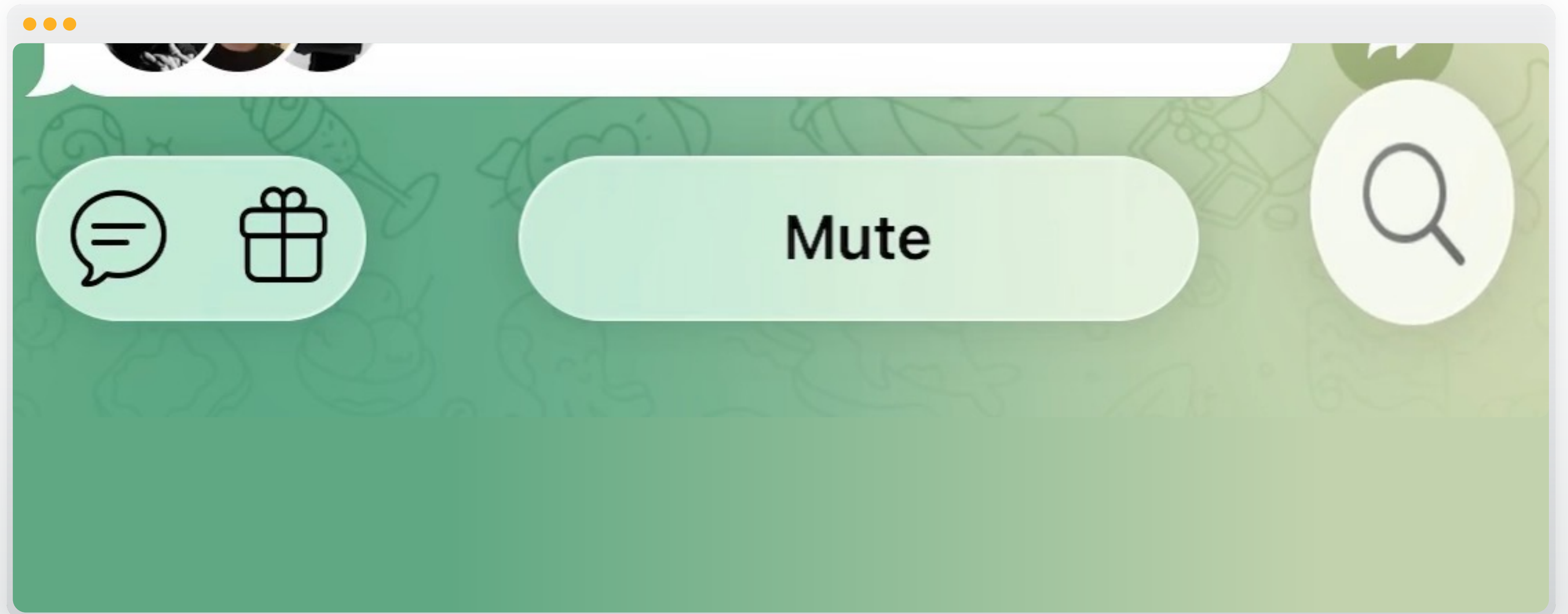
CADisplayLink

Измеряем скорость перемещения точки касания и симулируем физику пружин на каждом кадре

Демонстрация интерактивных анимаций



Демонстрация интерактивных анимаций



Итоги конкурса



2nd

iOS Contest 2025

\$4,500 • Dec 4 – Dec 26, 2025

4th

ML Competition, Round 2

\$1,000 • Nov 10 – Nov 20, 2023

[VIEW](#)

—

ML Competition

no prize • Oct 5 – Oct 15, 2023

[VIEW](#)

—

iOS UI Competition

no prize • Aug 23 – Sep 3, 2023

[VIEW](#)

1st

Call UI iOS

\$4,500 • Feb 17 – Mar 4, 2023

[VIEW](#)

3rd

iOS Contest 2022: Round 1

\$2,000 • Oct 9 – Oct 31, 2022

[VIEW](#)

2nd

C++ Contest 2022

\$4,000 • Mar 4 – Mar 13, 2022

[VIEW](#)

3rd

Sticker App Contest

\$750 • Jun 25 – Jul 4, 2021

[VIEW](#)

B

watchOS Contest 2021

\$1,000 • Mar 5 – Mar 21, 2021

[VIEW](#)



Fair Quokka Mar 12 at 19:23

Конкурсное задание выполнено практически полностью, за исключением поддержки нового дизайна тумблеров.

В ходе тестирования конкурсной работы мы выявили несколько недоработок:

– Параметры линзы подобраны близко к эталонным, однако фокусировка линзы при выборе раздела в панели навигации могла бы быть плавнее. В частности, сейчас при переключении между двумя крайними разделами заметен резкий прыжок линзы на финальном этапе анимационного цикла.

Отметим отличную реализацию компонентов внутри чата, в частности поля ввода сообщения и сопутствующих кнопок.

Медиа: <https://t.me/c/3824920627/42>

Желаем удачи в следующих конкурсах!

iPhone XR, iOS 18.3





Общий подход к конкурсам



Внимательно изучить требования, сфокусироваться на самом главном



Провести исследование возможных решений до начала реализации



Декомпонировать задачу и составить план



Собрать быстрый, желательно изолированный прототип



Итеративно интегрировать, чтобы быстрее выявлять проблемы



Стабилизировать в реальном UI, тщательно тестировать на девайсе

Материалы доклада



@konstantin_moskalenko





**Декомпозиция —
ключ к любому
сложному эффекту
(и не бойтесь Metal)**

Ваши вопросы?

No-code

Python

Swift

Process

Development

Planning

Java

Analysis

Golang

Mobile App

JavaScript

Node.js

Innovation

