

# Иконки и как ИХ ГОТОВИТЬ 🍳

**Жемчугов Петр**  
Head of frontend  
SberHealth



# Про что поговорим?

## Наша история

Переоценка работы с иконками  
в компании sberHealth

# Про что поговорим?

## Наша история

Переоценка работы с иконками  
в компании sberHealth

## Существующие подходы

Немного истории

# Про что поговорим?

## Наша история

Переоценка работы с иконками  
в компании sberHealth

## Дизайн-системы

И интеграция с figma

## Существующие подходы

Немного истории

# Про что поговорим?

## Наша история

Переоценка работы с иконками в компании sberHealth

## Существующие подходы

Немного истории

## Дизайн-системы

И интеграция с figma

## Компонент Icon

Реализация компонента в дизайн-системе

# С чего начинали:

Дизайн

**Более 400 иконок**

И количество постоянно растёт

# С чего начинали:

Дизайн

**Более 400 иконок**

И количество постоянно растёт

**3 размера у каждой иконки**

Нельзя использовать  
одно изображение  
для разных размеров

# С чего начинали:

Дизайн

**Более 400 иконок**

И количество постоянно растёт

**3 размера у каждой иконки**

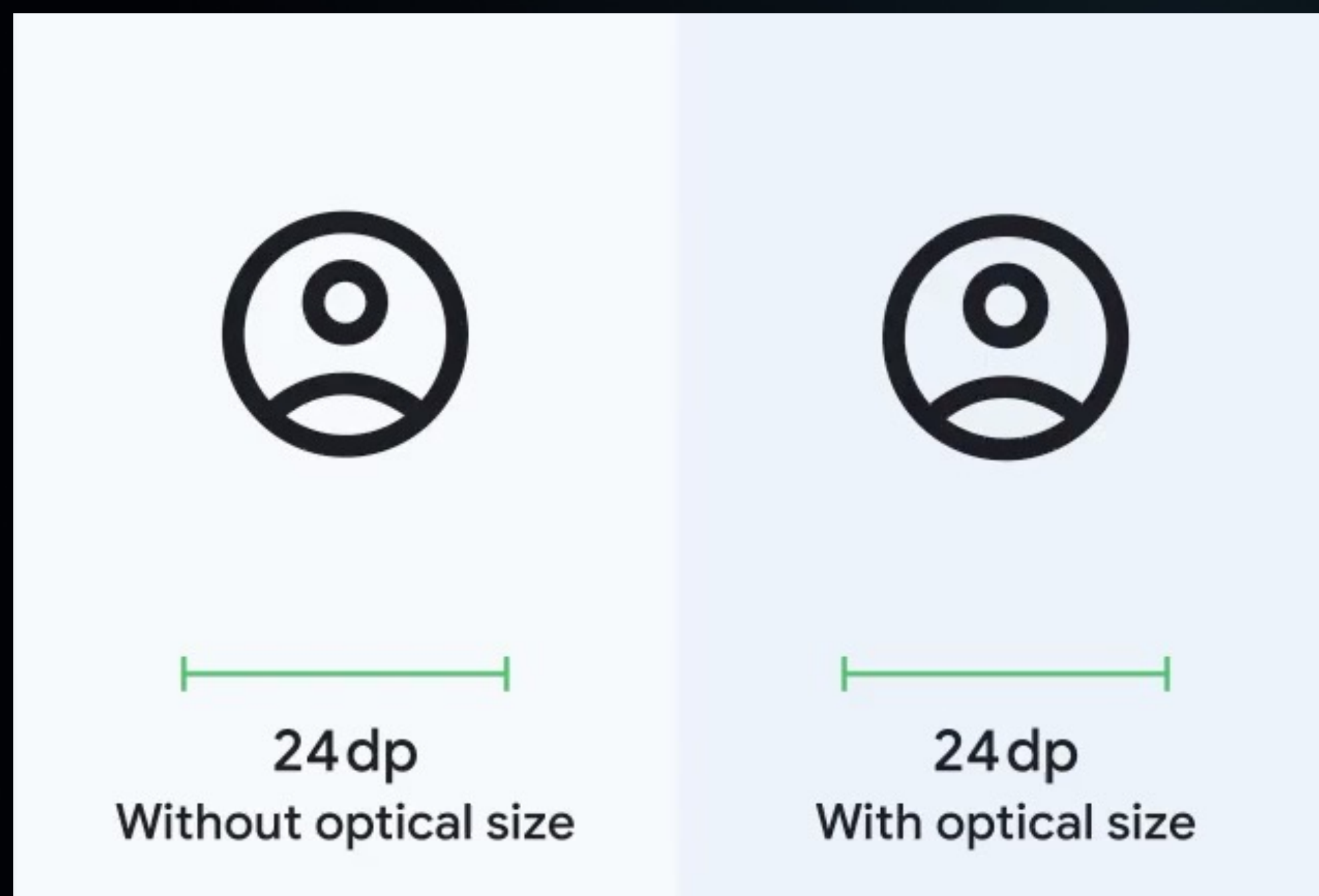
Нельзя использовать  
одно изображение  
для разных размеров

**Около 1200 изображений**

3 размера для каждой иконки



# Одно изображение для разных размеров иконки



# С чего начинали:

Разработка

**Выгружали иконки руками**

SVG файлы вручную складывали  
в репозиторий

# С чего начинали:

Разработка

## **Выгружали иконки руками**

SVG файлы вручную складывали  
в репозиторий

## **Свой механизм генерации компонентов**

Из каждого svg файла  
создавался компонент

# С чего начинали:

Разработка

## Выгружали иконки руками

SVG файлы вручную складывали в репозиторий

## Свой механизм генерации компонентов

Из каждого svg файла создавался компонент

## Одна иконка = один компонент

1200 компонентов в npm-пакете

# С чего начинали:

Разработка

## **Выгружали иконки руками**

SVG файлы вручную складывали в репозиторий

## **Свой механизм генерации компонентов**

Из каждого svg файла создавался компонент

## **Одна иконка = один компонент**

1200 компонентов в npm-пакете

## **Отсутствие единого нейминга**

В Figma 1 компонент на все иконки, в коде отдельные компоненты

# Классификация иконок

## SVG

SVG inline

SVG спрайт

SVG файл

## PNG

PNG файл

PNG Спрайт

## Шрифты

Лигатуры

Глифы

# Классификация иконок

**SVG**

**SVG inline**

**SVG спрайт**

**SVG файл**

**PNG**

**PNG файл**

**PNG Спрайт**

**Шрифты**

**Лигатуры**

**Глифы**

# SVG inline

## Плюсы

Векторное изображение  
высокой четкости

currentColor



# CurrentColor



```
<div class="a">
  <div class="icon"></div>

  <div class="b">
    Hello world
  </div>
</div>
```



```
.a {
  color: red;
}
```

# currentColor



```
<div class="a">
  <div class="icon"></div>

  <div class="b">
    Hello world
  </div>
</div>
```



```
.a {
  color: red;
}

.c {
  background: currentColor;
}
```

# SVG inline

## Плюсы

Векторное изображение  
высокой четкости

CurrentColor

## Минусы

Большое количество DOM-узлов

Увеличивают размер бандла

Повторяющиеся иконки  
рендерятся многократно

Не подходят для сложных  
растровых изображений

# SVG inline минусы


Размер бандла, количество DOM-узлов





```
<svg width="20" height="20" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg">  
  <path d="M1.09901 10.5995C2.1537 9.18746 8.64554 1.99069 12.5006  
    0.323584C15.1647 -0.824256 16.8103 1.30745 16.3739 9.35473 11.0732  
    10.3276 9.52452C11.3914 7.83009 12.7461 5.54352 13.3826  
    3.86731C13.519 3.52114 13.1826 3.24784 12.8643 3.46648C11.5823  
    4.37746 9.00923 7.15597 7.94544 8.42223C6.61798 10.0073 4.73589  
    12.4852 4.03579 13.2869C3.33569 14.0886 2.00823 14.2434 1.14447  
    13.4235C0.198876 12.5126 0.389813 11.556 1.09901 10.59957" fill="#F9F9F9" />  
</svg>
```

# SVG inline минусы

Проблема повторяющихся иконок



**Христорождественская Анастасия Владимировна** 


**5.0**  159 ОТЗЫВОВ


**ЭКСПЕРТ** Д.М.Н. Стаж 6 лет


Стоимость приёма  
**2 250 ₽** ~~2 500~~ -25% от СБЕРЗДОРОВЬЯ

**Ближайшая запись** Сегодня, 19 апреля

14:30 15:10 15:40 [Ещё](#)



**Петров Игорь Михайлович** 

**3.5**  323 ОТЗЫВА

**ПРОФИ** ПРОФЕССОР Стаж 12 лет

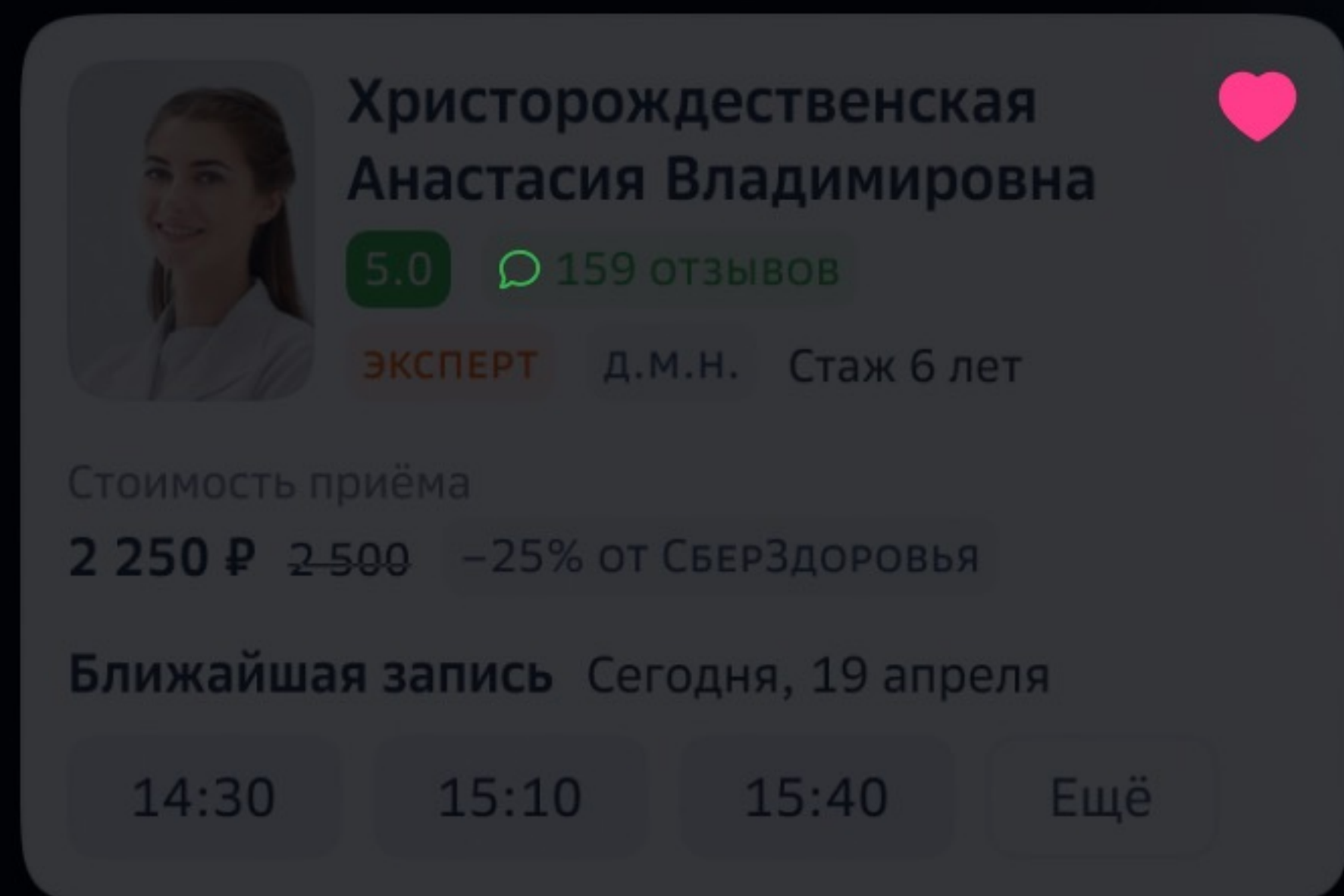
Стоимость приёма  
**2 950 ₽** ~~3 500~~ -25% от СБЕРЗДОРОВЬЯ


**Ближайшая запись** Сегодня, 19 апреля

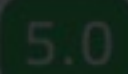

14:30 15:10 15:40 [Ещё](#)

# SVG inline минусы

Проблема повторяющихся иконок



Христорождественская Анастасия Владимировна 

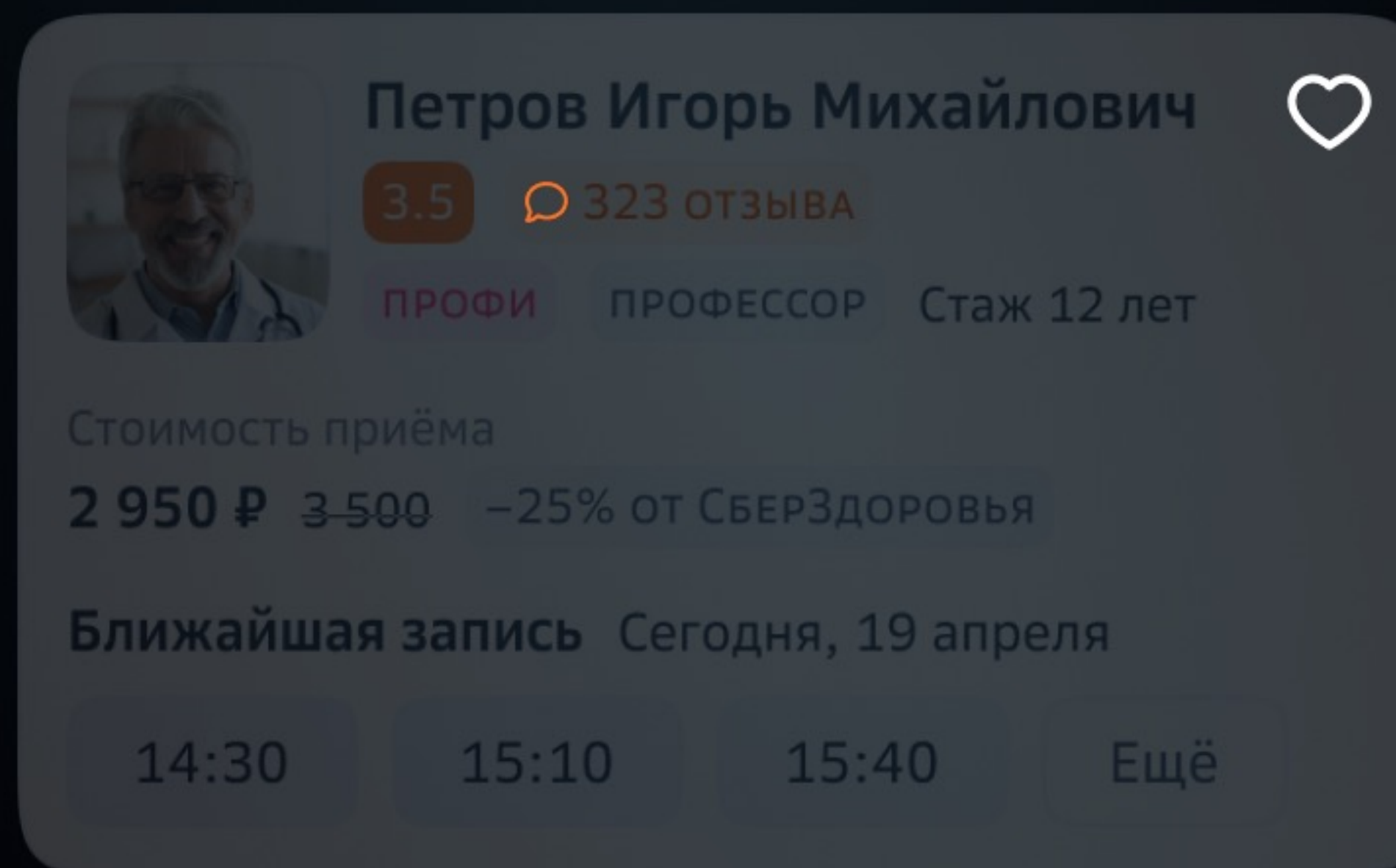
 


ЭКСПЕРТ Д.М.Н. Стаж 6 лет

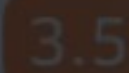

Стоимость приёма  
2 250 ₽ ~~2 500~~ -25% от СВЕРЗДОРОВЬЯ

Ближайшая запись Сегодня, 19 апреля

14:30 15:10 15:40 Ещё



Петров Игорь Михайлович 

ПРОФИ ПРОФЕССОР Стаж 12 лет

Стоимость приёма  
2 950 ₽ ~~3 500~~ -25% от СВЕРЗДОРОВЬЯ

Ближайшая запись Сегодня, 19 апреля

14:30 15:10 15:40 Ещё

# SVG спрайт

## Плюсы

Векторное изображение  
высокой четкости

currentColor

Повторяющиеся иконки  
рендерятся однократно

# SVG спрайт

## Плюсы

Векторное изображение  
высокой четкости

currentColor

Повторяющиеся иконки  
рендерятся однократно

## Минусы

Большое количество DOM-узлов

Увеличивают размер бандла

Не подходят для сложных  
растровых изображений



# SVG спрайты

Как они работают



```
<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">  
  
  <symbol id="circle">  
    <circle cx="50" cy="50" r="45" fill="#0f0" />  
  </symbol>  
  
  <symbol id="square">  
    <rect y="5" x="5" width="90" height="90" fill="#f00" />  
  </symbol>  
  
</svg>
```

# SVG спрайты

Как они работают



```
<svg viewBox="0 0 100 100" xmlns="http://www.w3.org/2000/svg">  
  <symbol id="circle">  
    <circle cx="50" cy="50" r="45" fill="#0f0" />  
  </symbol>  
  
  <symbol id="square">  
    <rect y="5" x="5" width="90" height="90" fill="#f00" />  
  </symbol>  
  
</svg>
```



```
<svg width="100" height="100">  
  <use xlink:href="sprite.svg#circle" />  
</svg>  
  
<svg width="100" height="100">  
  <use xlink:href="sprite.svg#square" />  
</svg>
```

# SVG картинки

Пример кода



```

```

# SVG картинки

## Плюсы

Векторное изображение  
высокой четкости

Не влияет на размер бандла

Не замедляет рендер

Кэшируется браузером

# SVG картинки

## Плюсы

Векторное изображение  
высокой четкости

Не влияет на размер бандла

Не замедляет рендер

Кэшируется браузером

## Минусы

currentColor не работает

Нельзя поменять цвет

Не подходят для сложных  
растровых изображений

# PNG картинка

## Плюсы

Подходят для  
растровых изображений

Не влияет на размер бандла

Не замедляет рендер

Кэшируется браузером

# PNG картинка

## Плюсы

Подходят для  
растровых изображений

Не влияет на размер бандла

Не замедляет рендер

Кэшируется браузером

## Минусы

currentColor не работает

Нельзя поменять цвет

# PNG спрайт

## Плюсы

Повышает скорость загрузки при использовании HTTP 1.1



# PNG спрайт

## Плюсы

Повышает скорость загрузки при использовании HTTP 1.1

## Минусы

currentColor не работает

Нельзя поменять цвет

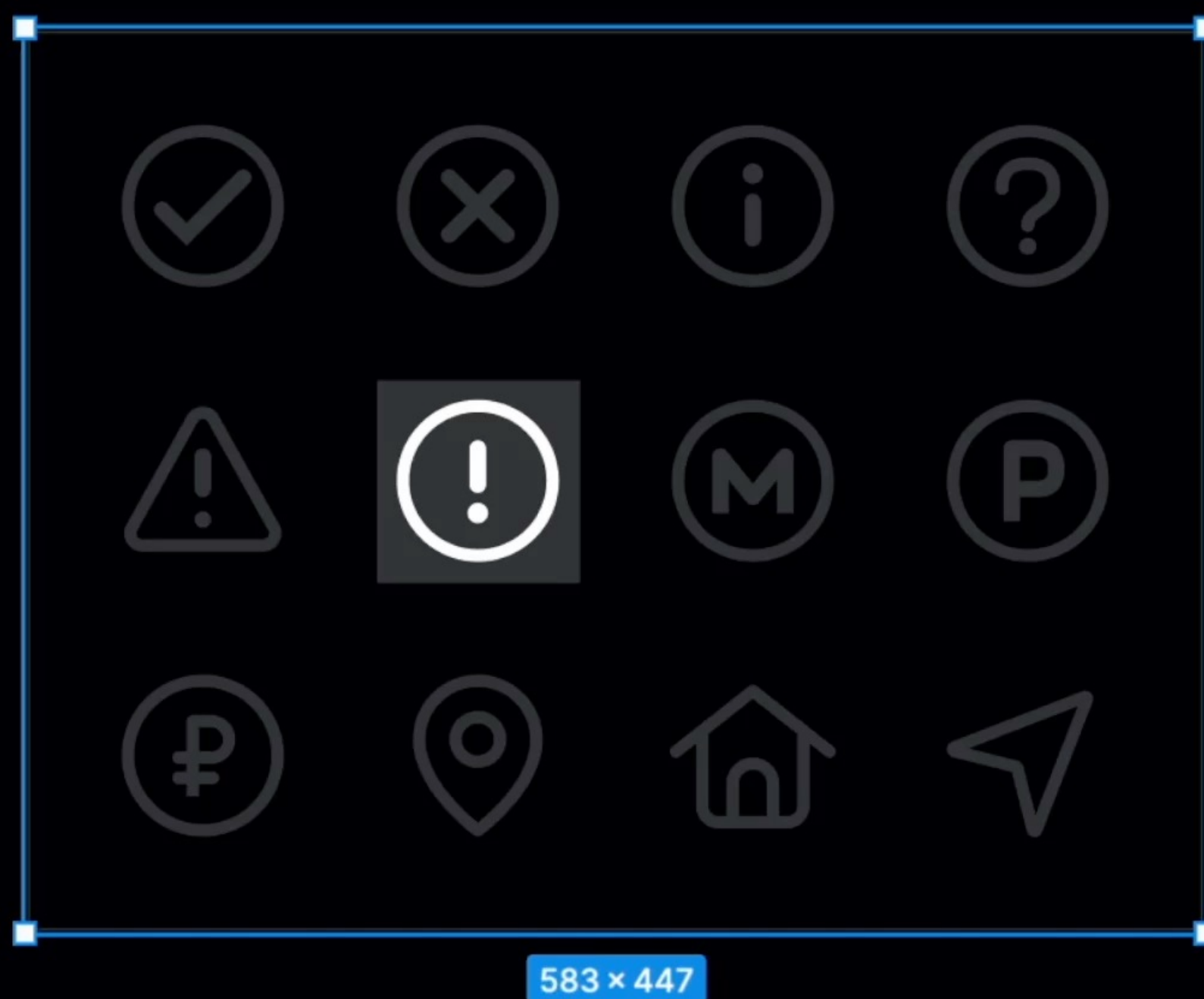
# PNG спрайт

Иллюстрация



# PNG спрайт

Иллюстрация



# Шрифты

## Плюсы

Кэшируются браузером

Не влияет на размер бандла

Не замедляет рендер

# Шрифты

## Плюсы

Кэшируются браузером

Не влияет на размер бандла

Не замедляет рендер

## Минусы

Всегда грузится весь шрифт со всеми иконками

Сложная реализация

Высокий порог входа в понимание механизма работы

# Шрифты

Лигатуры

## Лигатуры в JetBrains Mono

<> a | => b+++c ->  
<! - - d~~ > e- > >  
/ =f <=g ###h |>

## Лигатуры в Material Symbols

+ Add  
Delete  
Arrow Back

⊕ Add Circle  
⊗ Cancel  
Arrow Back iOS

# Шрифты

Глифы

**Иконка**



heart

# Шрифты

Глифы

Иконка

HTML



heart



```
<i class="fa-solid fa-heart fa-fw">  
  ::before  
</i>
```



# Шрифты

Глифы

Иконка



heart

HTML



```
<i class="fa-solid fa-heart fa-fw">  
  ::before  
</i>
```

CSS



```
.fa-heart::before {  
  content: "\f004";  
}
```

# Существующие решения

И их реализация

**Google font icons**

Шрифт + лигатуры

# Существующие решения

И их реализация

**Google font icons**

Шрифт + лигатуры

**Font awesome**

Шрифт + глифы

# Существующие решения

И их реализация

## Google font icons

Шрифт + лигатуры

## Font awesome

Шрифт + глифы

## MUI icons

SVG inline обернутый  
в React-компонент

# Существующие решения

И их реализация

## Google font icons

Шрифт + лигатуры

## Font awesome

Шрифт + глифы

## MUI icons

SVG inline обернутый  
в React-компонент

## FlatIcon / freepik

SVG/PNG изображение

# Требования к компоненту

# Требования к компоненту

## Управление цветом

Возможность менять цвет иконки

# Требования к компоненту

## Управление цветом

Возможность менять цвет иконки

## Интеграция с Figma

Автоматическое добавление  
НОВЫХ ИКОНОК



# Требования к компоненту

## Управление цветом

Возможность менять цвет иконки

## Интеграция с Figma

Автоматическое добавление  
НОВЫХ ИКОНОК

## Влияние на скорость рендера

Не должны влиять на скорость загрузки

# Требования к компоненту

## Управление цветом

Возможность менять цвет иконки

## Интеграция с Figma

Автоматическое добавление  
НОВЫХ ИКОНОК

## Влияние на скорость рендера

Не должны влиять на скорость загрузки

## Кэширование в браузере

Иконки должны кэшироваться  
на стороне клиента

# Требования к компоненту

## Управление цветом

Возможность менять цвет иконки

## Интеграция с Figma

Автоматическое добавление  
НОВЫХ ИКОНОК

## Влияние на скорость рендера

Не должны влиять на скорость загрузки

## Кэширование в браузере

Иконки должны кэшироваться  
на стороне клиента

## Сквозной нейминг

Единые правила именования пропсов

# Требования к компоненту

## Управление цветом

Возможность менять цвет иконки

## Интеграция с Figma

Автоматическое добавление  
новых иконок

## Влияние на скорость рендера

Не должны влиять на скорость загрузки

## Кэширование в браузере

Иконки должны кэшироваться  
на стороне клиента

## Сквозной нейминг

Единые правила именования пропсов

## Единый компонент

Единый компонент для всех иконок,  
а не компонент для каждой иконки

# Требования к компоненту

## Управление цветом

Возможность менять цвет иконки

## Интеграция с Figma

Автоматическое добавление  
новых иконок

## Влияние на скорость рендера

Не должны влиять на скорость загрузки

## Кэширование в браузере

Иконки должны кэшироваться  
на стороне клиента

## Сквозной нейминг

Единые правила именования пропсов

## Единый компонент

Единый компонент для всех иконок,  
а не компонент для каждой иконки

## Влияние на размер бандла

Не должны влиять на размер бандла

# Требования к компоненту

	SVG inline	SVG спрайт	SVG файл	PNG файл	Шрифты
Управление цветом	☑	☑			☑
Интеграция с Figma	☑	☑	☑	☑	
Скорость рендера			☑	☑	
Кэширование			☑	☑	☑
Размер бандла			☑	☑	☑
Единый компонент			☑	☑	☑
Простое решение	☑		☑	☑	

# Требования к компоненту

	SVG inline	SVG спрайт	SVG файл	PNG файл	Шрифты
Управление цветом	☑	☑			☑
Интеграция с Figma	☑	☑	☑	☑	
Скорость рендера			☑	☑	
Кэширование			☑	☑	☑
Размер бандла			☑	☑	☑
Единый компонент			☑	☑	☑
Простое решение	☑		☑	☑	

# Требования к компоненту

	SVG inline	SVG спрайт	SVG файл	PNG файл	Шрифты
Управление цветом	☑	☑			☑
Интеграция с Figma	☑	☑	☑	☑	
Скорость рендера			☑	☑	
Кэширование			☑	☑	☑
Размер бандла			☑	☑	☑
Единый компонент			☑	☑	☑
Простое решение	☑		☑	☑	



# SVG файл и currentColor

SVG injector



```

```

# SVG файл и currentColor

SVG injector



```

```



```
<svg width="20" height="20" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg">  
  <path d="M1.09901 10.5995C2.1537 9.18746 8.64554 1.99069 12.5006  
    0.323584C15.1647 -0.824256 16.8103 1.30745 16.3739 9.35473 11.0732  
    10.3276 9.52452C11.3914 7.83009 12.7461 5.54352 13.3826  
    3.86731C13.519 3.52114 13.1826 3.24784 12.8643 3.46648C11.5823  
    4.37746 9.00923 7.15597 7.94544 8.42223C6.61798 10.0073 4.73589  
    12.4852 4.03579 13.2869C3.33569 14.0886 2.00823 14.2434 1.14447  
    13.4235C0.198876 12.5126 0.389813 11.556 1.09901 10.59957" fill="#F9F9F9" />  
</svg>
```

# SVG файл и currentColor

CSS фильтры



```
filter:  
  invert(51%)  
  sepia (75%)  
  brightness(87%)  
  contrast(81%)  
  saturate (274%)  
  hue-rotate(9deg);
```

# SVG файл и currentColor

CSS фильтры



```
filter:  
  invert(51%)  
  sepia (75%)  
  brightness(87%)  
  contrast(81%)  
  saturate (274%)  
  hue-rotate(9deg);
```



codepen.io

# SVG файл и currentColor

SVG фильтр



```
<svg width="80" height="80">
  <filter id="filter">
    <feFlood flood-color="currentColor" />
    <feComposite in2="SourceAlpha" operator="in" />
  </filter>

  <image width="80" height="80" href="icon.svg" filter="url(#filter)" />
</svg>
```

# Реализация компонента

SVG файл + SVG фильтр



```
<Icon name="star" />
```

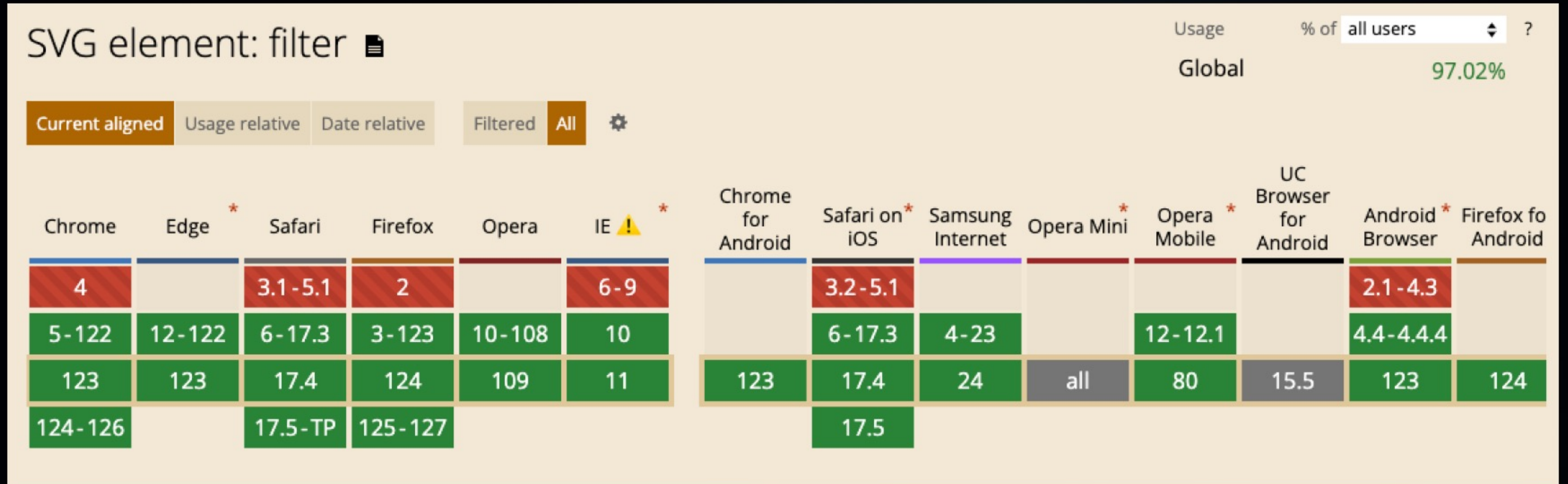


```
const Icon = ({name, size = 'm'}) = {
  const filterID = useId();
  const sizeMap = {
    's': 16,
    'm': 20,
    'l': 24
  }
  const currentSize = sizeMap[size];

  return (
    <svg width={currentSize} height={currentSize}>
      <filter id="filterID">
        <feFlood flood-color="currentColor" />
        <feComposite in2="SourceAlpha" operator="in" />
      </filter>
      <image
        filter={`url(#${filterID})`}
        href={`https://cdn.ru/icons/${name}${size}.svg`}
      />
    </svg>
  )
}
```

# Подводные камни

SVG файл + SVG фильтр



# Первые проблемы

в Safari на IOS

Safari on IOS

v6





# Первые проблемы

в Safari на IOS

Safari on IOS  
v6

Safari on IOS  
v15-16.3

# Первые проблемы

в Safari на IOS

Safari on IOS  
v6

Safari on IOS  
v15-16.3

Safari on IOS  
v16.4

27.03.2023

# Первые проблемы

в Safari на IOS

Safari on IOS  
v6

Safari on IOS  
v15-16.3

Safari on IOS  
v16.4

27.03.2023



[developer.apple.com](https://developer.apple.com)

- Fixed applying changes to `fill` with `currentColor` to other colors via CSS.
- Fixed changes to the `filter` property getting ignored.

# mask-image

Как применять



```
<span class="icon"></span>
```

# mask-image

Как применять



```
<span class="icon"></span>
```



```
.icon {  
  display: block;  
  width: 24px;  
  height: 24px;  
  background: currentColor;  
  mask-image: url("icon.svg");  
  mask-position: center center;  
}
```

# Реализация компонента

С применением mask-image



```
export const Icon = ({ name }) => {  
  const iconStyle = { maskImage: `url(https://cdn.ru/${name}.svg)` }  
  
  return (  
    <i className="icon" style={iconStyle} />  
  )  
}
```

# Реализация компонента

Добавляем пропс size



```
export const Icon = ({ name, size }) => {  
  const iconStyle = { maskImage: `url(https://cdn.ru/${size}-${name}.svg)` }  
  
  return (  
    <i className={`icon icon-size-${size}`} style={iconStyle} />  
  )  
}
```

# Реализация компонента

TypeScript

```
interface IIconProps {  
  name:  
    | 'star'  
    | 'user'  
    | 'doctor'  
    | 'like' // ...  
  size: 16 | 24  
}
```

```
const Icon: FC<IIconProps> = ({ name, size }) => {  
  const iconStyle = { maskImage: url(`https://cdn.ru/${size}-${name}.svg`) }  
  
  return (  
    <i className={`icon icon-size-${size}`} style={iconStyle} />  
  )  
}
```



# Реализация компонента

Не все иконки есть во всех размерах



heart  
size = 16px



heart  
size = 24px

# Реализация компонента

Не все иконки есть во всех размерах



heart  
size = 16px



heart  
size = 24px

# Реализация компонента

Не все иконки есть во всех размерах

```
type TIconName24 =  
  | 'arrow_left'  
  | 'calendar'  
  | 'check'  
  | 'check_circle'  
  | 'chevron_down'  
  | 'chevron_left'  
  | 'chevron_right'  
  | 'chevron_up'  
  | 'cross'  
  | 'delete_icon'  
  | 'drag_2'  
  | 'duplicate'  
  | 'edit'  
  | 'email'  
  | 'exit'  
  | 'filter'  
  // ...  
}
```

```
type TIconName16 =  
  | 'arrow_down'  
  | 'arrow_up'  
  | 'aside'  
  | 'calendar'  
  | 'check'  
  | 'cross_large'  
  | 'cross_small'  
  | 'delete_icon'  
  | 'duplicate'  
  | 'email'  
  | 'exit'  
  | 'filter_priority'  
  | 'full_settings'  
  | 'link'  
  | 'more'  
  | 'next_week'  
  // ...  
}
```

# Реализация компонента

Не все иконки есть во всех размерах

```
type TIconName24 =  
  | 'arrow_left'  
  | 'calendar'  
  | 'check'  
  | 'check_circle'  
  | 'chevron_down'  
  | 'chevron_left'  
  | 'chevron_right'  
  | 'chevron_up'  
  | 'cross'  
  | 'delete_icon'  
  | 'drag_2'  
  | 'duplicate'  
  | 'edit'  
  | 'email'  
  | 'exit'  
  | 'filter'  
  // ...  
}
```

```
type TIconName16 =  
  | 'arrow_down'  
  | 'arrow_up'  
  | 'aside'  
  | 'calendar'  
  | 'check'  
  | 'cross_large'  
  | 'cross_small'  
  | 'delete_icon'  
  | 'duplicate'  
  | 'email'  
  | 'exit'  
  | 'filter_priority'  
  | 'full_settings'  
  | 'link'  
  | 'more'  
  | 'next_week'  
  // ...  
}
```

```
interface IIcon24Props {  
  icon: TIconName24;  
  size: 24  
}  
  
interface IIcon16Props {  
  icon: TIconName16;  
  size: 16  
}
```

# Реализация компонента

Не все иконки есть во всех размерах

```
type TIconName24 =  
  | 'arrow_left'  
  | 'calendar'  
  | 'check'  
  | 'check_circle'  
  | 'chevron_down'  
  | 'chevron_left'  
  | 'chevron_right'  
  | 'chevron_up'  
  | 'cross'  
  | 'delete_icon'  
  | 'drag_2'  
  | 'duplicate'  
  | 'edit'  
  | 'email'  
  | 'exit'  
  | 'filter'  
  // ...  
}
```

```
type TIconName16 =  
  | 'arrow_down'  
  | 'arrow_up'  
  | 'aside'  
  | 'calendar'  
  | 'check'  
  | 'cross_large'  
  | 'cross_small'  
  | 'delete_icon'  
  | 'duplicate'  
  | 'email'  
  | 'exit'  
  | 'filter_priority'  
  | 'full_settings'  
  | 'link'  
  | 'more'  
  | 'next_week'  
  // ...  
}
```

```
interface IIcon24Props {  
  icon: TIconName24;  
  size: 24  
}  
  
interface IIcon16Props {  
  icon: TIconName16;  
  size: 16  
}
```

```
const Icon: FC<IIcon16Props | IIcon24Props> = {  
  // ...  
}
```

# Автоматизация



git



Выгружаем иконки из Figma  
и получаем набор SVG файлов

# Автоматизация



git



Генерируем типы для имен иконок  
из имен файлов

# Автоматизация



git



Создаем коммит и тестируем иконки



# Автоматизация



git



Выгружаем `svg` файлы  
В `S3` хранилище (или `CDN`)

# Автоматизация



git



Выгружаем в NPM  
новую версию компонента

**Применение компонента**

# Применение компонента

Было



```
import IconHeart24 from "@design-system"
```



```
<IconHeart24 />
```

# Применение компонента

Стало



```
import IconHeart24 from "@design-system"
```



```
import Icon from "@design-system"
```



```
<IconHeart24 />
```



```
<Icon name="heart" size={24} />
```

# Применение компонента

В других компонентах: было



```
<Button leftIcon={<IconHeart24/>} />
```

# Применение компонента

В других компонентах: стало



```
<Button leftIcon={<IconHeart24/>} />
```

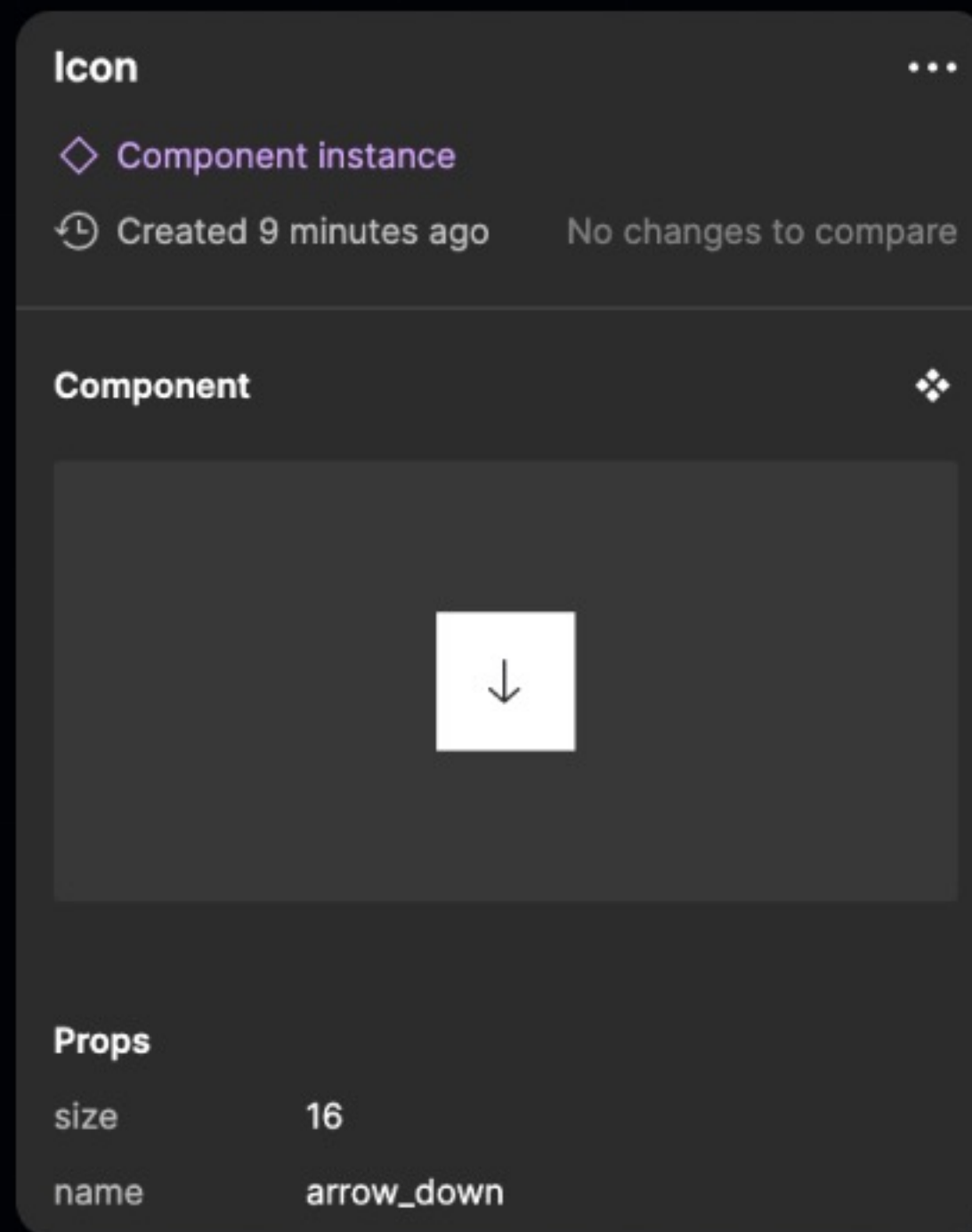


```
<Button leftIcon="heart" />
```

# Применение компонента

Единый нейминг и структура

## Figma

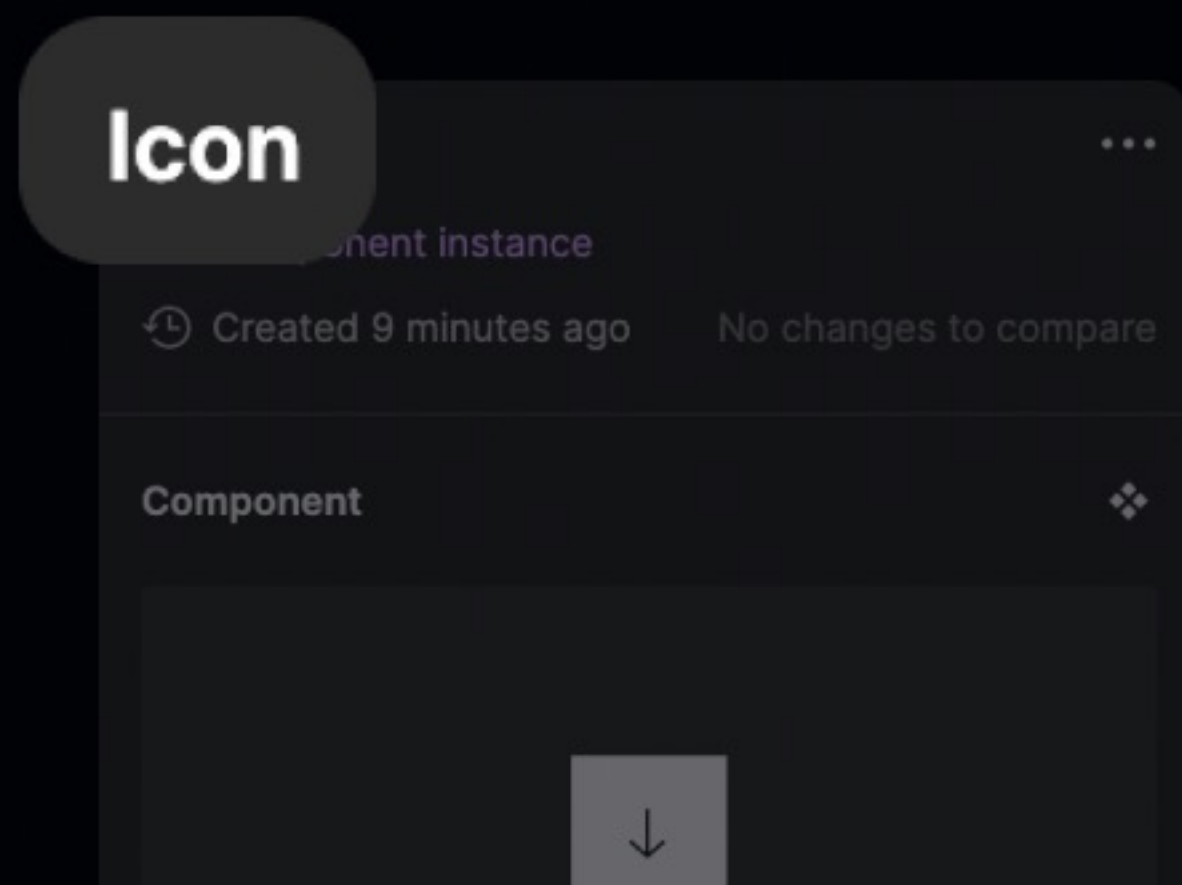




# Применение компонента

Единый нейминг и структура

## Figma



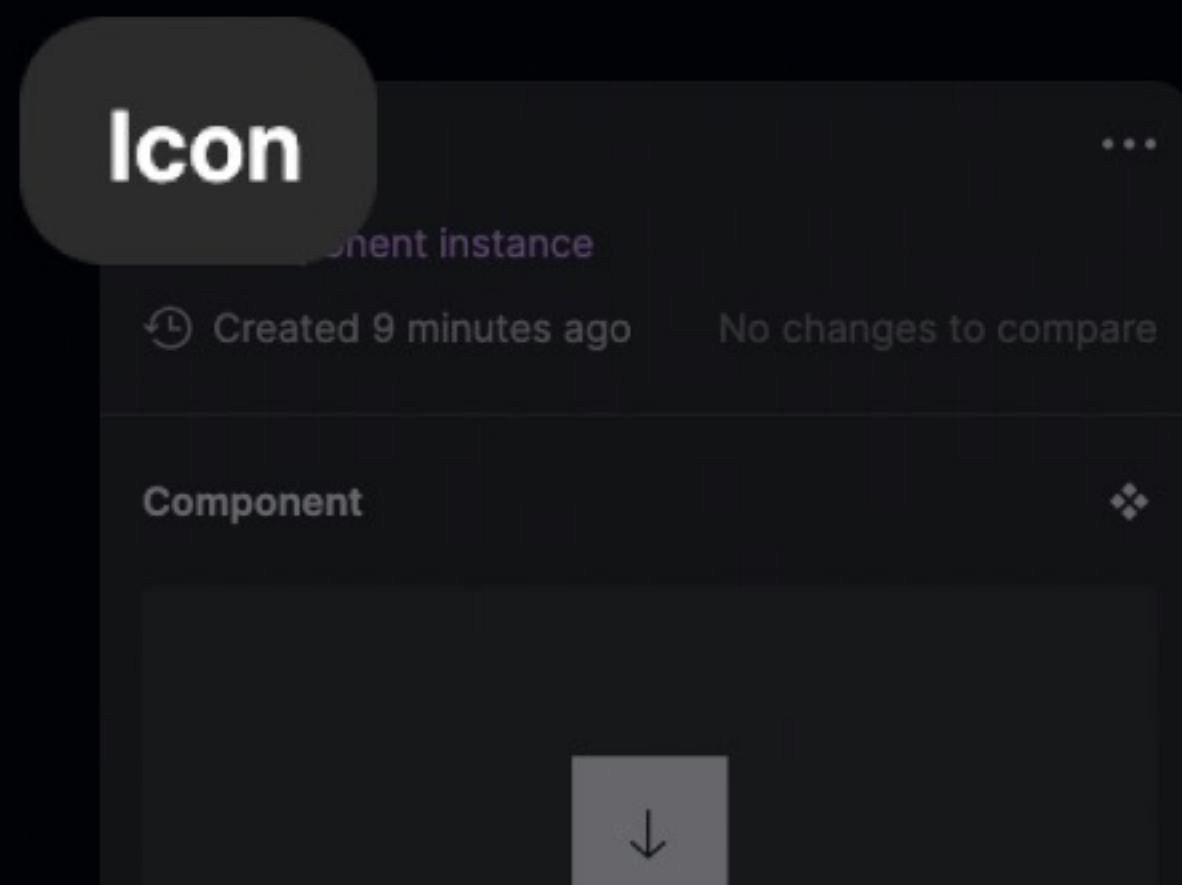
### Props

size	16
name	arrow_down

# Применение компонента

Единый нейминг и структура

## Figma



### Props

size	16
name	arrow_down

## Код



```
<Icon name="arrow_down" size={16} />
```

# Подводные камни

Cross-Origin Resource Sharing (CORS)

## Домены сайта

<https://docdoc.ru>

<https://sberhealth.ru>

## Домен s3

[s3.cdn.sberhealth.ru](https://s3.cdn.sberhealth.ru)

# Подводные камни

Cross-Origin Resource Sharing (CORS)

```
access-control-allow-origin: *  
Access-Control-Max-Age: 86400
```



w3.org

# Подводные камни

Кэширование иконок в браузере

**С версионированием иконок**

Cache-Control: max-age=31536000, immutable

# Подводные камни

Кэширование иконок в браузере

**Без версионирования иконок**

Cache-Control: max-age=86400, must-revalidate

ETag: "0d769892b536fdc20878447774e6c2fa"

# Подводные камни

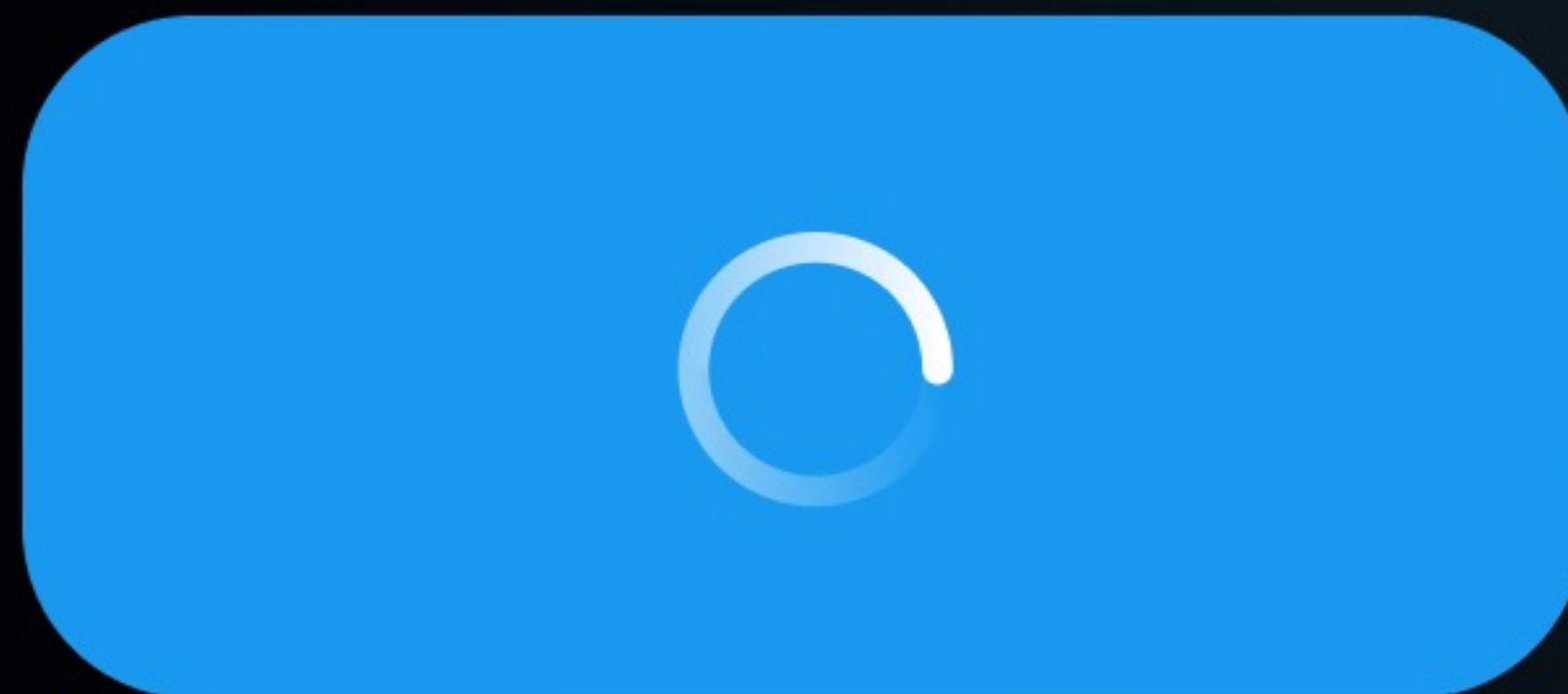
Предзагрузка иконок



Button

# Подводные камни

Предзагрузка иконок





# Результаты

☑ **Единый компонент**

# Результаты

- ☑ Единый компонент
- ☑ Не увеличивает размер бандла

# Результаты

- ☑ **Единый компонент**
- ☑ **Не увеличивает размер бандла**
- ☑ **Оптимизируется нативными возможностями браузера**

# Результаты

- ☑ **Единый компонент**
- ☑ **Не увеличивает размер бандла**
- ☑ **Оптимизируется нативными возможностями браузера**
- ☑ **Простое для понимания и поддержки решение**

# Результаты

- ☑ **Единый компонент**
- ☑ **Не увеличивает размер бандла**
- ☑ **Оптимизируется нативными возможностями браузера**
- ☑ **Простое для понимания и поддержки решение**
- ☑ **Кешируется на клиенте**

# Результаты

- ☑ **Единый компонент**
- ☑ **Не увеличивает размер бандла**
- ☑ **Оптимизируется нативными возможностями браузера**
- ☑ **Простое для понимания и поддержки решение**
- ☑ **Кешируется на клиенте**
- ☑ **Положительное влияние на метрики**



[t.me/petepearl](https://t.me/petepearl)



[github.com/petepearl](https://github.com/petepearl)

**Жемчугов Петр**

Head of frontend  
SberHealth



# Производительность

