



МойОфис

Как мы веб-редактор документов строили



Сергей Симонов
Веб-разработчик

О чем поговорим

- Ключевые архитектурные решения наших редакторов
- Унификация функционала для разных платформ
- Использование кода на «неродном» языке на стороне JavaScript
- Рендеринг на canvas
- Нюансы при работе со шрифтами
- Существующие проблемы и их решения



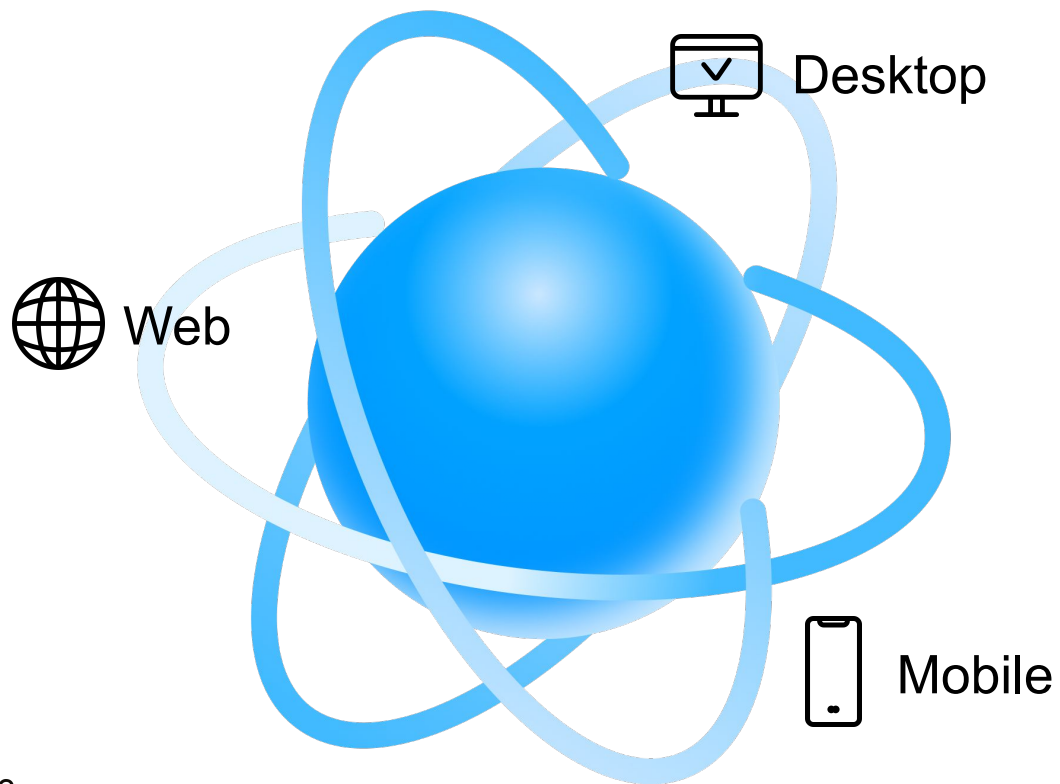
Требования и ключевые архитектурные решения

Базовые требования

- Открытие документов – сторонних и собственных
- Набор инструментов, аналогичный сторонним редакторам
- Возможность совместного редактирования
- Мультиплатформенность

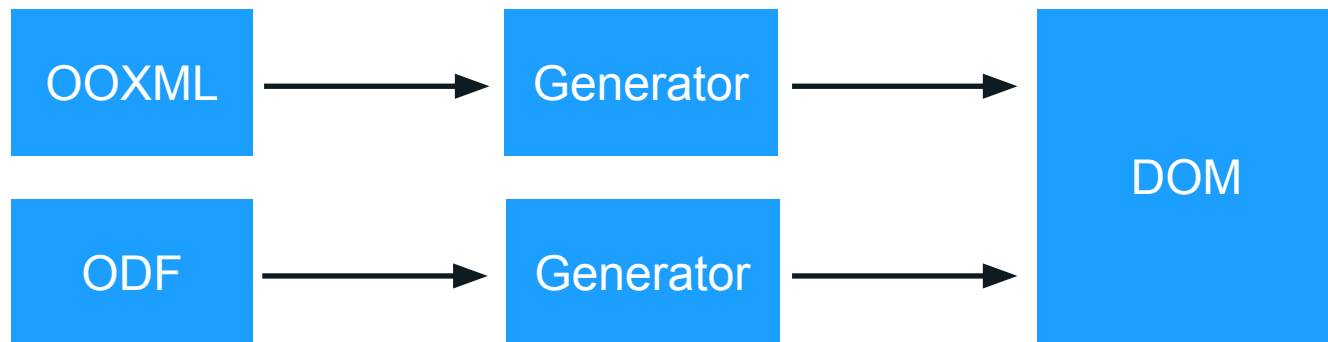


Ядро

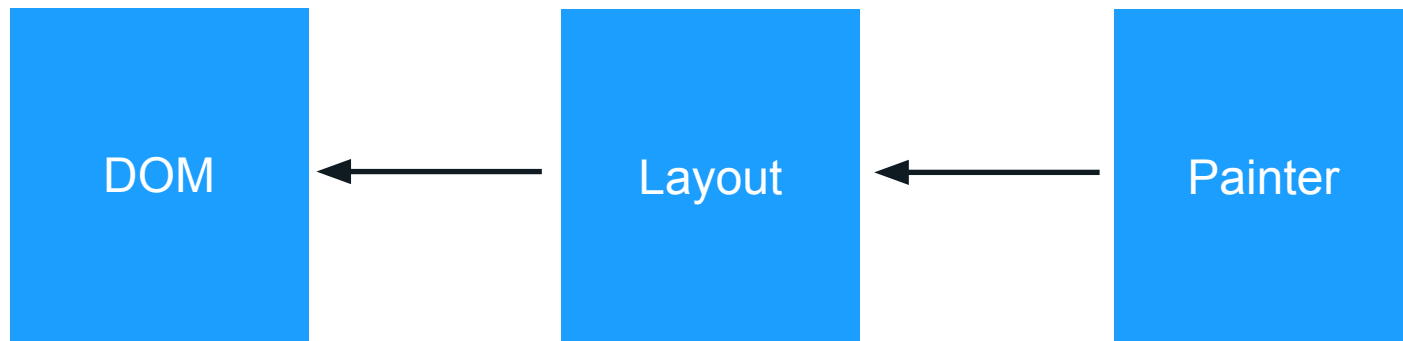


МойОфис

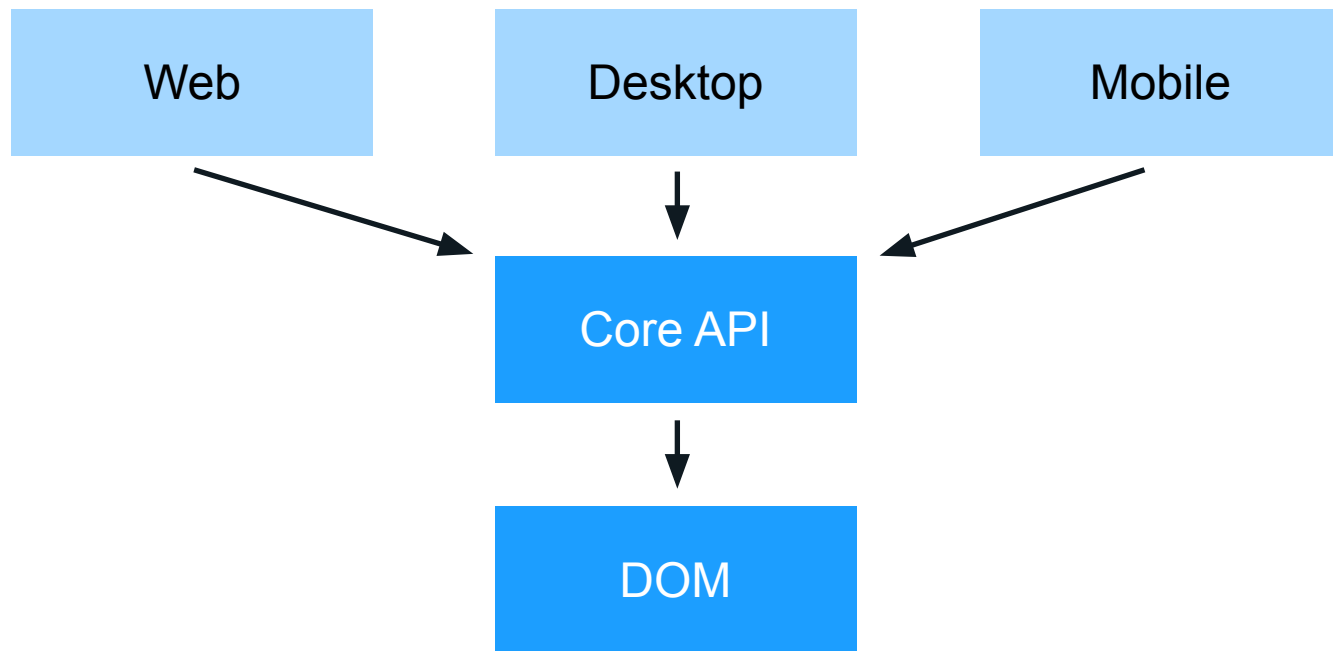
Ядро. DOM



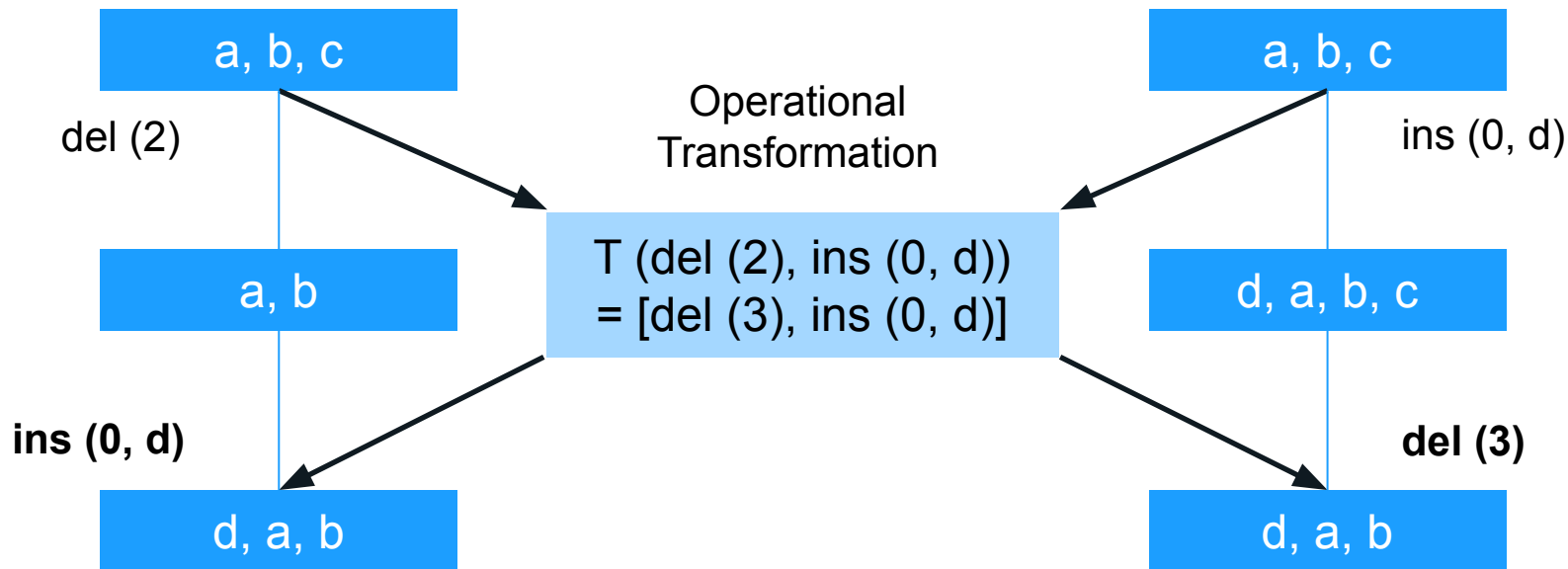
Ядро. DOM



Ядро. API



Ядро. Совместное редактирование



Ядро. Плюсы

- Одинаковое отображение документов на всех платформах
- Унификация основного функционала
- Новые фичи приезжают практически «из коробки»
- Транзитивные цели от бизнеса



Ядро. Минусы

- Полная зависимость от другой команды
- Обновления ядра – иногда бывает очень больно



Что в нашей команде.
Про web

Как мы работаем с ядром



WEBASSEMBLY



Как мы работаем с ядром

- Создаем статическую библиотеку из исходного кода ядра
- Пишем биндинги на сущности ядра
- Компилируем ядро в Wasm
- Публикуем npm-пакет
- Используем на стороне редактора



Как мы работаем с ядром

```
class_<EditorsCore::CellHelper>("CellHelper")  
    .smart_ptr<EditorsCore::CellHelperPtr>("std::shared_ptr<EditorsCore::CellHelper>")  
    .function("setMode", &EditorsCore::CellHelper::setMode)  
    .function("moveToPoint",  
        select_overload<void(const RTE::TouchPoint &)>(&EditorsCore::CellHelper::moveTo))  
    .function("moveToCell",  
        select_overload<void(const CellIndex &)>(&EditorsCore::CellHelper::moveTo))  
    .function("apply", &EditorsCore::CellHelper::apply)
```



Как мы работаем с ядром



Почти готово, так ведь?

АБСОЛЮТЛИ



МойОфис

Как мы рендерим

- Интерфейс Painter в ядре – содержит набор методов для отрисовки
- Наша реализация – WebCanvasPainter – формирует ArrayBuffer
- Разбиваем ArrayBuffer на последовательность команд и параметров
- Вызываем соответствующие методы Web Canvas API



WebCanvasPainter

```
class WebCanvasPainter final : public Gfx::Painter
{
public:
    WebCanvasPainter();
    void setLineWidth(Gfx::LineWidth lineWidth) override;
    void setLineType(Gfx::LineType type) override;
    void setStrokeColor(const ColorRGBA& strokeColor) override;
    void setFillColor(const ColorRGBA& fillColor) override;
    ...
}
```

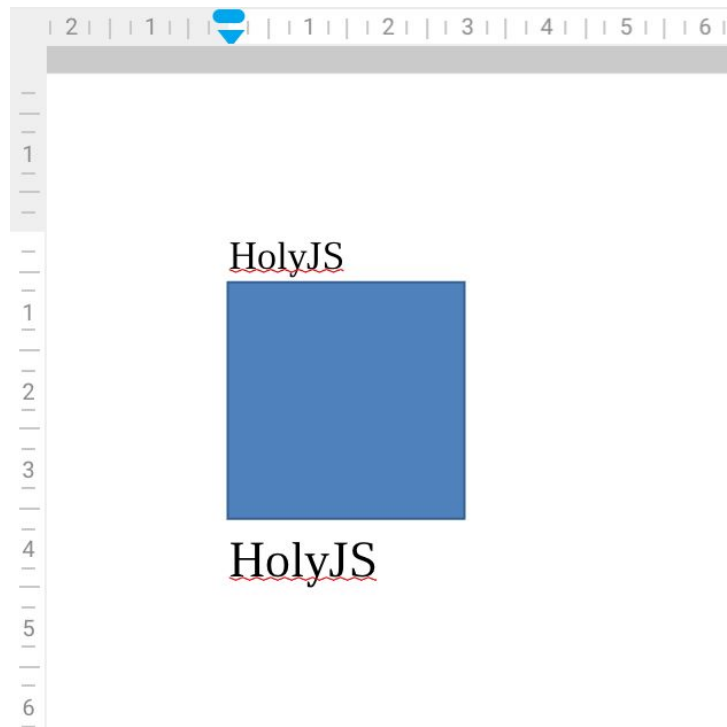


WebCanvasPainter

```
void WebCanvasPainter::fillRect(const Rect& rect)
{
    checkFlush();
    _renderingBuffer.setValue((uint8_t)PainterActions::fillRect);
    _renderingBuffer.setValue(rect.getUpperLeft().x);
    _renderingBuffer.setValue(rect.getUpperLeft().y);
    _renderingBuffer.setValue(rect.getWidth());
    _renderingBuffer.setValue(rect.getHeight());
}
```



Содержимое ArrayBuffer



```
drawText('HolyJS', 14, ...),  
setFillColor(79, 129, 189, 1),  
fillRect(80, 90, 100, 100),  
drawText('HolyJS', 18, ...),
```



Это основа



МойОфис

Рендеринг текста. Пример



```
inputHandler.onInput( callback: (text) => {  
    PerformanceHelper.api.startMeasuring(PerformanceActionType.TextInsert);  
    this.core.insertText(text);  
});
```



Рендеринг текста. Пример



Рендеринг текста. Пример

```
pages.addEventListener(coreUi.events.documentPages.pageDataChanged, (changes) => {  
    Object.keys(changes).forEach((pageId : string ) => {  
        if (changes[pageId].size) {  
            this.pages[pageId].updateSize(changes[pageId].size);  
        }  
    });  
  
    this.#redrawVisible();  
});
```



Рендеринг текста. Пример

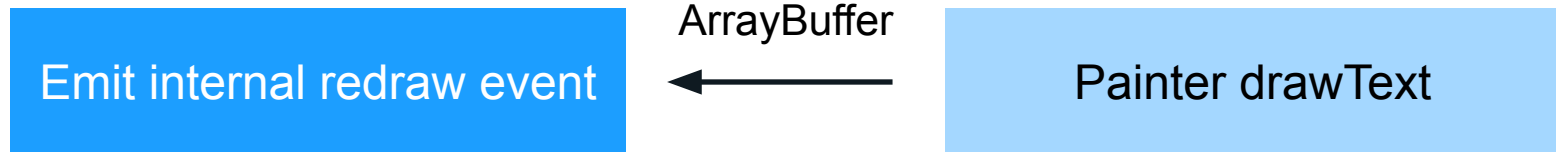


Рендеринг текста. Пример

```
#drawPage(viewport: IXYRect, pageId: number, drawingSettings: CoreDrawingSettings): void {  
    const webCanvasPainter = new this.#core.WebCanvasPainter();  
    webCanvasPainter.initDraw();  
    webCanvasPainter.setClipBoundingBox(  
        viewport.x,  
        viewport.y,  
        viewport.x + viewport.width + BLOCKS_SUPERPOSITION_OFFSET,  
        viewport.y + viewport.height + BLOCKS_SUPERPOSITION_OFFSET,  
    );  
    this.#core.drawPage(webCanvasPainter, pageId, drawingSettings);  
    webCanvasPainter.flush();  
}
```



Рендеринг текста. Пример

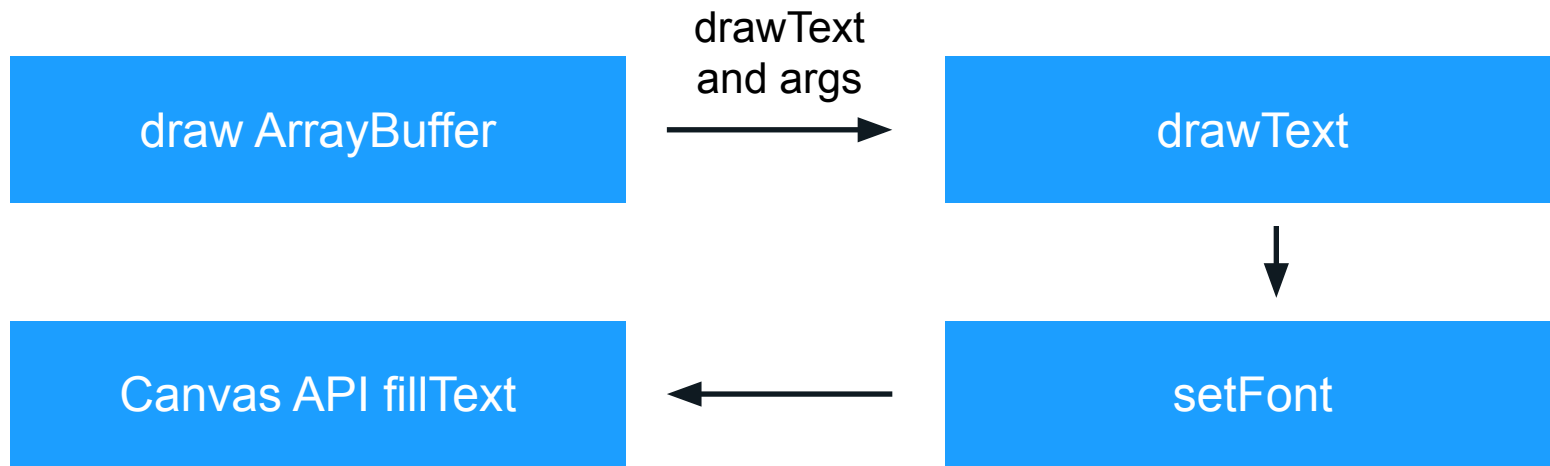


Рендеринг текста. Пример

```
emitRedrawEvent(pageId) {  
  const coreRenderResult = this.#aggregateBuffer();  
  const renderedBlocks = coreRenderResult.map((block) => ({  
    id: block.id || coreUi.defaultBlockId,  
    data: block.data,  
  }));  
  eventChannel.emit(coreUi.events.uiRendering.pageDraw, {  
    pageId,  
    blocks: renderedBlocks,  
  });  
}
```



Рендеринг текста. Пример



Рендеринг текста. Пример

```
#renderToCanvas(canvasData: CanvasData, canvasRef: HTMLCanvasElement) {  
    const context = canvasRef.getContext( contextId: '2d', options: { desynchronized: true });  
    const canvasAdapter = new CanvasAdapter(context, {  
        width: canvasData.width,  
        height: canvasData.height,  
        scale: canvasData.scale,  
    });  
    BufferPainter.drawBuffer(canvasData.canvas.buffer, canvasAdapter);  
}  
  
drawText(text, font, size, bold, italic) {  
    this.setFont(font, size, bold, italic);  
    this._ctx.fillText(text, x: 0, y: 0);  
}
```

Откройте форточку, спикер наспикерил



МойОфис

Немного про шрифты

- Фиксированный список шрифтов
- Начертания: regular, bold, italic, bold-italic

XO Caliburn

XO Courser

XO Oriel

XO Oriel Condensed

XO Tahion

XO Thames

Cormorant

PT Astra Sans

PT Astra Serif

PT Sans

PT Serif



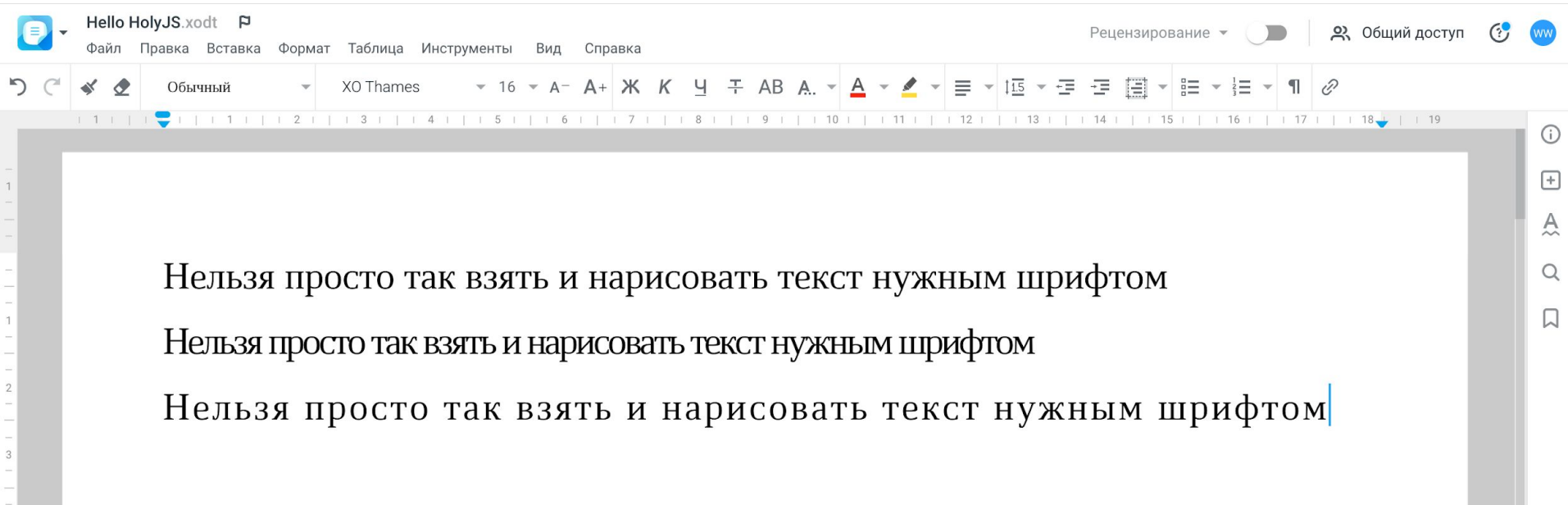
Малппинг шрифтов

Calibri	XO Caliburn
Courier	XO Courser
Arial	XO Oriel
Arial Narrow	XO Oriel Condensed
Tahoma	XO Tahion
Times New Roman	XO Thames

В иных случаях – XO Thames



Нельзя просто так взять и...



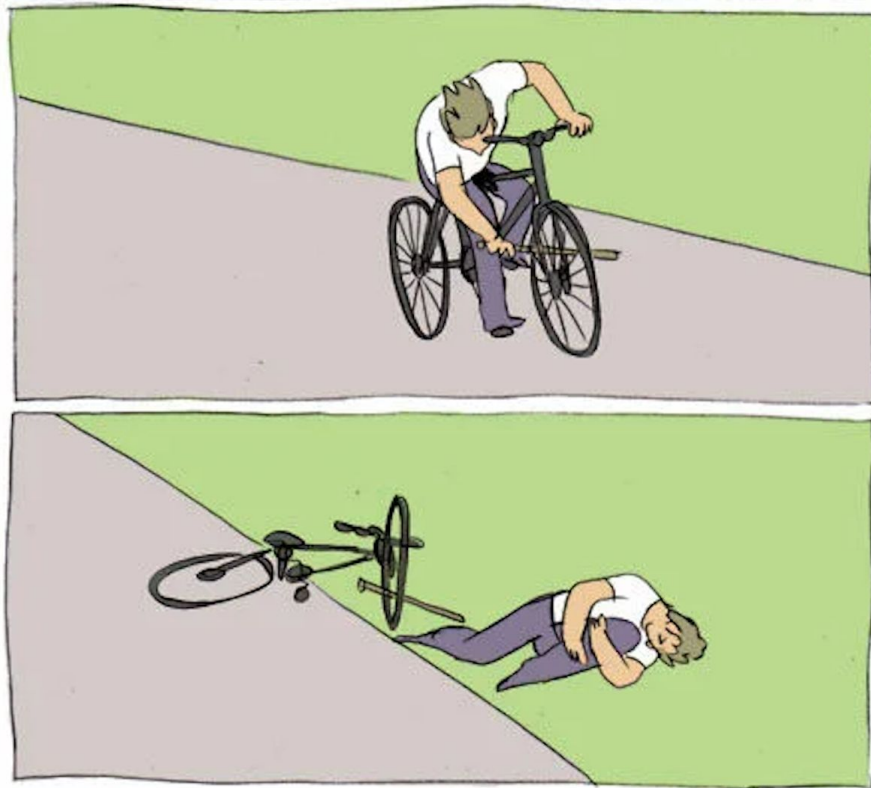
Посимвольный рендеринг

```
for (let i = 0, offsetX = 0; i < text.length; i++) {  
    const charWidth = parseFloat(charsWidth[i]);  
    if (charWidth) {  
        this._ctx.fillText(text[i], offsetX, y: 0);  
        offsetX += charWidth;  
    }  
}
```

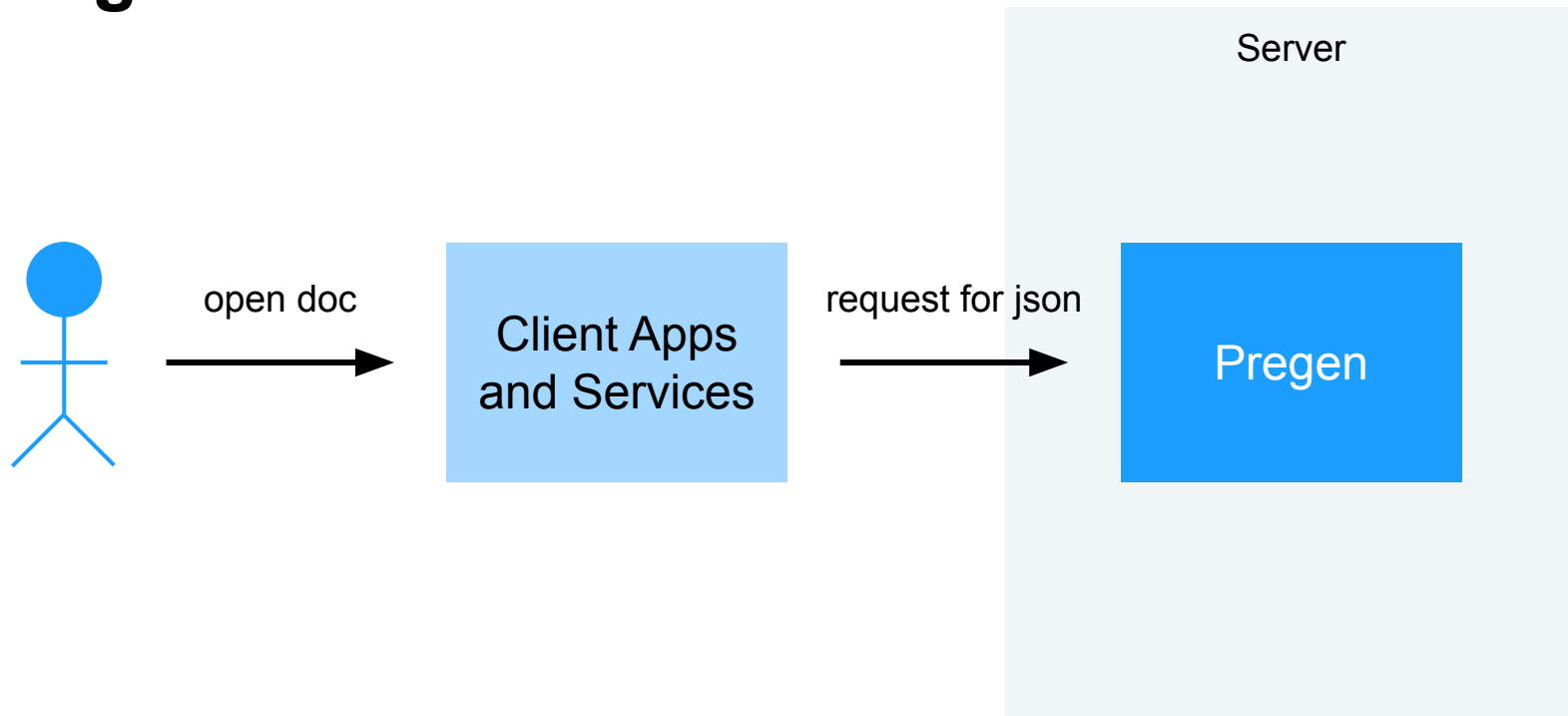


**Все работает отлично,
но есть нюанс...**

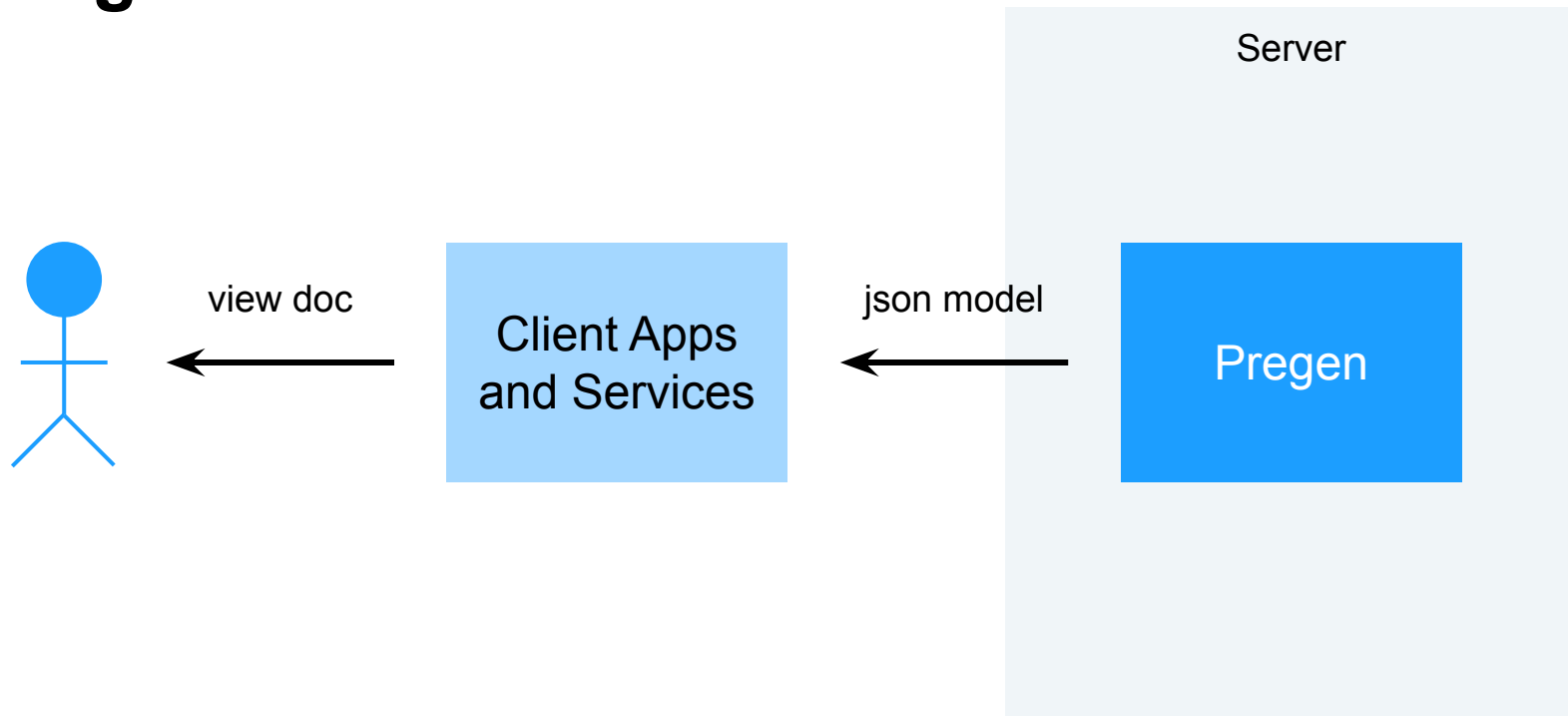
Ядро и просмотр документа



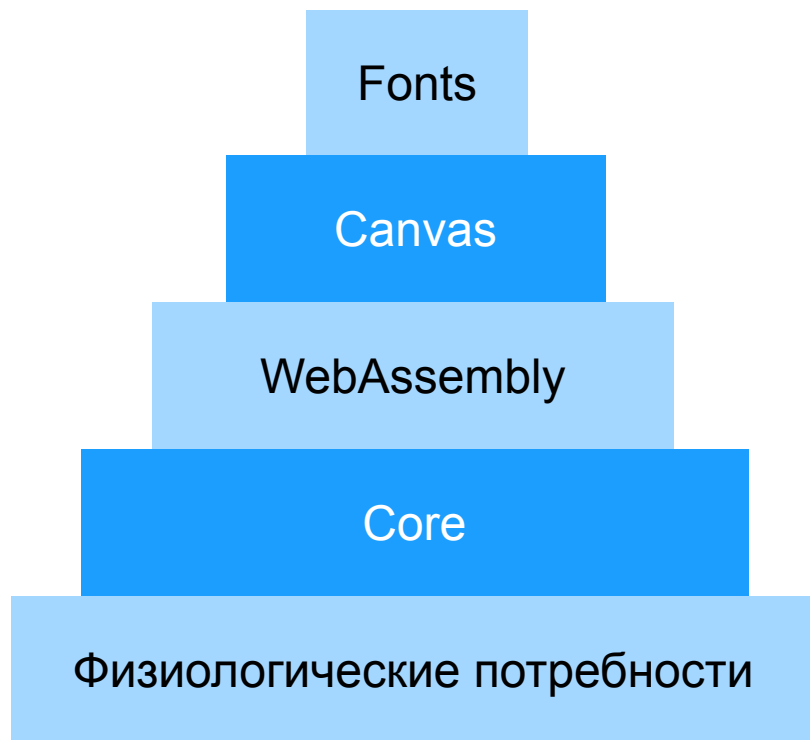
Pregen



Pregen



Итоги



спасибо

