Sta++ler

SDK

Импортозамещение кроссплатформенной разработки

Роман Катунцев. Дмитрий Муканин

Что видит пользователь?

Qt

Flutter

Kotlin Multiplatform Mobile

Jetpack Compose

React Native

Sta++ler SDK

Один показательный случай

```
<manifest
  android:versionCode="185"
  android:versionName="3.10.0"
  package="org.stappler.off"
  xmlns:android="http://schemas.android.com/apk/res/androi
  <uses-sdk
    android:minSdkVersion="24"
    android:targetSdkVersion="33" />
  <uses-permission
    android:name="android.permission.INTERNET" />
  <uses-permission
    android:name="android.permission.ACCESS_NETWORK_STATE" />
  <uses-permission
    android:name="android.permission.WAKE_LOCK" />
  <uses-permission
    android:name="com.google.android.finsky.permission.BIND_GET
  <uses-permission
    android:name="com.google.android.gms.permission.AD_ID" />
  <uses-permission
    android:name="android.permission.POST_NOTIFICATIONS" />
```

ActivityResultCaller

Artifact: androidx.activity:activity

View Source

Added in 1.2.0

Kotlin | **Java**

```
public interface ActivityResultCaller
```

Known direct subclasses

ComponentActivity, Fragment

- play-services-ads-identifier:18.0.0 manifest
- play-services-base:18.0.1 manifest
- play-services-basement:18.1.0 manifest
- play-services-measurement-api:21.2.0 manifest
- play-services-measurement:21.2.0 manifest
- transport-backend-cct:3.1.8 manifest
- transport-runtime:3.1.8 manifest
- build.gradle injection

Sta++ler SDK

Догоняющее развитие
(А давайте сделаем наш Флаттер!)

- Быть догоняющим небезопасно! (Патчи CVE будут запаздывать)
- Никогда не сделаем лучше оригинала
- Куча “бюрократических” подделок
- Никаких новых подходов
- Завязаны на чужие инструменты

Sta++ler SDK

Почему нет своих фреймворков?

- Они довольно большие
- Не приносят прямой прибыли
- Сложны для внедрения и маркетинга

Почему едет Flutter, React Native, Kotlin и далее?

- Передача разработки для своих нужд в OpenSource
- Налоговый вычет и прочая господдержка
- Обучение новых специалистов сразу под себя

Это стратегическое решение!

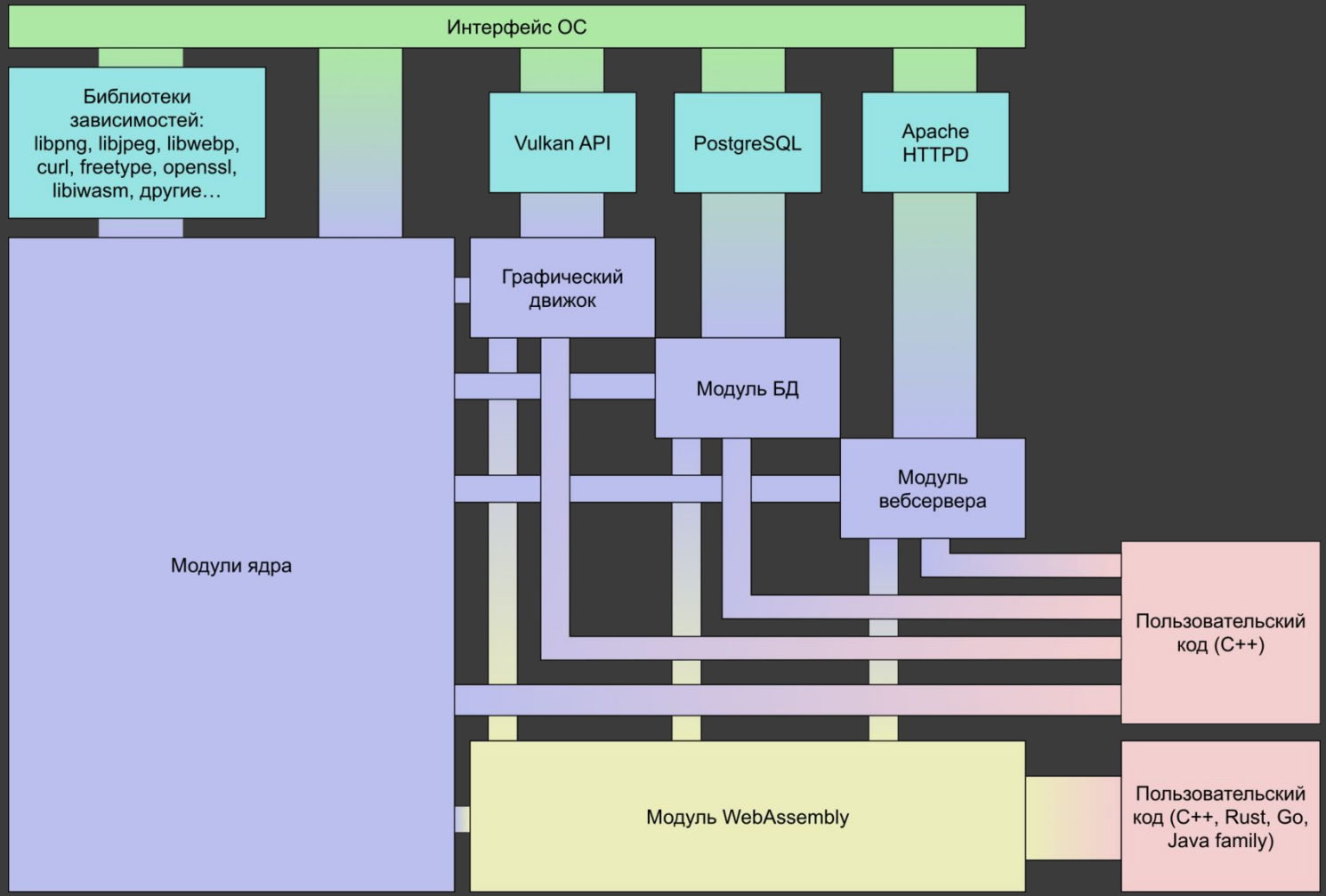
Sta++ler SDK

Приоритеты:

1. Открытость
2. Расширяемость
3. Платформонезависимость/ эффективность
4. Простота освоения

ФОНД СОДЕЙСТВИЯ ИННОВАЦИЯМ

Sta++ler SDK


Интерфейс ОС
Библиотеки зависимостей: libpng, libjpeg, libwebp, curl, freetype, openssl, libiwasm, другие…
Vulkan API
PostgreSQL
Apache HTTPD
Модули ядра
Графический движок
Модуль БД
Модуль вебсервера
Пользовательский код (C++)
Модуль WebAssembly
Пользовательский код (C++, Rust, Go, Java family)

Криптография

ГОСТ 34.10-2018, ГОСТ 34.11-2018, ГОСТ 34.12-2018

- Модифицированный JsonWebToken
- Пользовательские криптохранилища
- Упрощённый интерфейс

Проблема криптографии - её не используют!

```
Bytes signature;
crypto::PrivateKey privKey(s_GostPrivKey);

privKey.sign([] (BytesView sig) {
    signature = sig.bytes<Interface>();
}, s_GostPrivKey, crypto::SignAlgorithm::GOST_512);

auto pubKey = privKey.exportPublic();

bool success = pubKey.verify(s_GostPrivKey, signature,
        crypto::SignAlgorithm::GOST_512);
```

Sta++ler SDK

Базы данных

- Никто не будет писать SQL, проще взять Firebase

SQL-база это:

- Собственный сервер
- Администрирование
- Новый язык
- Самим писать биндинги и api
- Лишние расходы на бэкэнд

А что мешает упростить интерфейс SQL-баз?

Базы данных

Что у нас есть:

- PosgreSQL и SQLite
- Автоматическая настройка
- Полнотекстовый поиск
- Контроль доступа
- Вычислимые поля
- Виртуальные поля
- Триггеры
- Автоматический REST API

```
_objects.define(Vector<Field>({
        Field::Text("text", MinLength(3)),
        Field::Extra("data", Vector<Field>{
                Field::Array("strings", Field::Text("")),
        }),
        Field::Set("subobjects", _subobjects),
        Field::File("image", MaxFileSize(1_MiB)),
        Field::Text("alias", Transform::Alias),
        Field::Integer("mtime", Flags::AutoMTime | Flags::Indexed),
        Field::Integer("index", Flags::Indexed),
        Field::View("refs", _refs, ViewFn([] (const Scheme &objScheme, const Value &obj) -> bool {
                return true;
        }), FieldView::Delta),

        Field::Set("images", _images, Flags::Composed),
}),
AccessRole::Admin(AccessRoleId::Authorized));
```

Sta++ler SDK

```
_images.define(Vector<Field>({
	Field::Integer("ctime", Flags::ReadOnly | Flags::AutoCTime | Flags::ForceInclude),
	Field::Integer("mtime", Flags::ReadOnly | Flags::AutoMTime | Flags::ForceInclude),

	Field::Text("name", Transform::Identifier, Flags::Required | Flags::Indexed | Flags::ForceInclude),

	Field::Image("content", MaxImageSize(2048, 2048, ImagePolicy::Resize), Vector<Thumbnail>({
		Thumbnail("thumb", 380, 380)
	})),
}),
	AccessRole::Admin(AccessRoleId::Authorized)
);

_hierarchy.define(Vector<Field>({
	Field::Text("name", MinLength(3)),
	Field::Integer("id", Flags::Indexed),
	Field::Object("root", _hierarchy, Linkage::Manual, ForeignLink("sections")),
	Field::Set("sections", _hierarchy, Linkage::Manual, ForeignLink("root")),

	Field::View("pages", _pages, ViewFn([] (const Scheme &, const Value &obj) -> bool {
		return obj.getBool("hidden") ? false : true;
	}), Vector<String>({ "hidden" })),

	Field::Set("all_pages", _pages)
}));
```

```
Field::Data("callback", DefaultFn([] (const Value &val) {
        return Value({
                pair("name", val.getValue("name")),
                pair("time", Value(Time::now().toMicros()))
        });
})),
Field::Virtual("computed", VirtualReadFn([] (const Scheme &, const Value &value) {
        Value tmp(value);
        tmp.erase("__oid");
        tmp.setInteger(Time::now().toMicros(), "time");
        return tmp;
}), Vector<String>({"name"})),

Field::Virtual("virtual", VirtualReadFn([] (const Scheme &, const Value &value) {
        auto path = filesystem::writablePath<Interface>(toString(value.getString("name"), ".cbor"));
        if (filesystem::exists(path)) {
                return data::readFile<Interface>(path);
        }
        return Value();
}), VirtualWriteFn([] (const Scheme &objScheme, const Value &obj, Value &data) {
        auto path = filesystem::writablePath<Interface>(toString(obj.getString("name"), ".cbor"));
        data::save<Interface>(data, path);
        return true;
}), Vector<String>({"name"}))
```

```
Field::FullTextView("tsv", db::FullTextViewFn([this] (const db::Scheme &scheme, const db::Value &obj) -> db::FullTextVector {
        size_t count = 0;
        db::FullTextVector vec;

        count = _search.makeSearchVector(vec, obj.getString("key"), db::FullTextRank::A, count);
        for (auto &it : obj.getArray("text")) {
                count = _search.makeSearchVector(vec, it.getString(), db::FullTextRank::B, count);
        }

        return vec;
}), /* db::FullTextQueryFn([this] (const db::Value &data) -> db::FullTextQuery {
        return _search.parseQuery(data.getString());
}), */ _search, Vector<String>({"key", "text"})),
```

Sta++ler SDK

Сложный C++? WebAssembly + любой язык

- WebAssembly System Interface
- WASI SDK
- WIT: API + ABI
- wit-bindgen (C/C++, Rust, Go, Java)
- WAMR

Мало скорости - скомпилируем WebAssembly в нативный код!

Мало безопасности? Подпишем модули

```
/* @import */ interface webserver {
    use data.{value};

    record callback {
        callback: u32,
        userdata: u32,
    }

    record host-component-data {
        on-child-init: u32,
        on-storage-init: u32,
        on-heartbeat: u32,
        userdata: u32,
    }

    resource host-component-info {
        get-name: func() -> string;
        get-version: func() -> string;
        get-file: func() -> string;
        get-symbol: func() -> string;
        get-data: func() -> borrow<value>;
    }
```

Веб

Облачные сервисы:

- Удобны
- Дёшевы (но сильно не всегда)
- Распространены
- Владеют вашими данными
- Могут превратиться в тыкву

Своя инфраструктура:

- Сложная
- Непонятная
- Нет привычных инструментов
- Данные принадлежат вам
- Никто не отберёт

Мы предлагаем:

- Шаблонизатор
- Общий код и инструменты клиент/сервер
- Интеграция C/C++ и другого кода, Vulkan на сервере
- Нативная производительность - низкая стоимость ресурсов
- Возможность нетиповых решений

Sta++ler SDK

Графика и вычисления (Vulkan API)

Vulkan - зрелая и всеобъемлющая инфраструктура

- С другими API: MoltenVK, Zink, DXVK
- Vulkan vs. Cuda
- Устройства: NVK, RADV, V3DV, Tumip, Lavapipe
- Технологии: Bindless, BDA, Descriptor Buffer, RayTracing
  - В первую очередь - экономия рабочего времени и ресурсов. Для чипов 15-18 года
  - То есть, отзывчивый и современный интерфейс на старых устройствах
- Отказ от устаревших подходов

Sta++ler SDK

Графика и вычисления

Не вулканом единым:

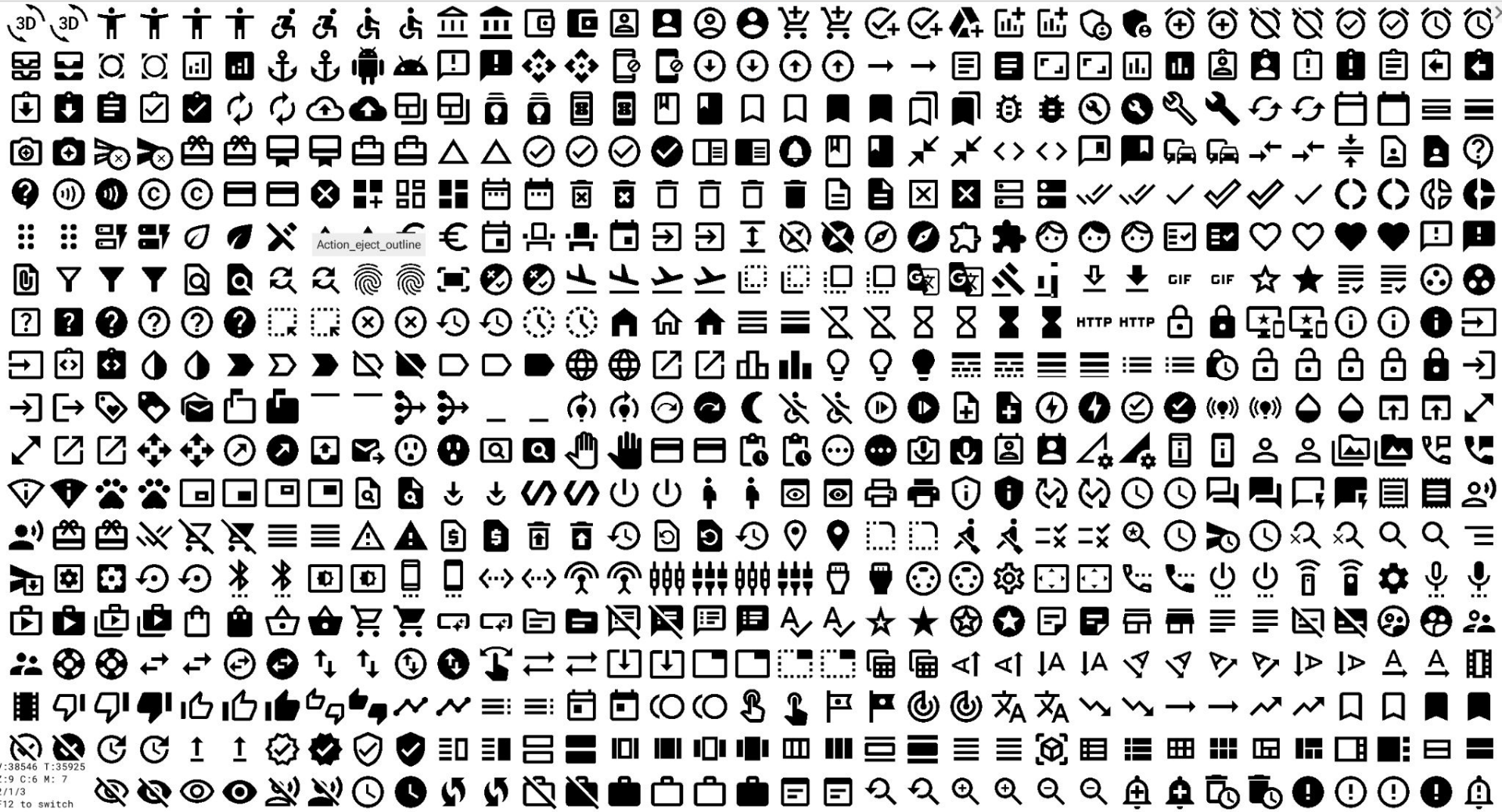
- RenderGraph не зависит от API
- Возможность инференса и обучений нейросетей
- Возможность решать вычислительные задачи

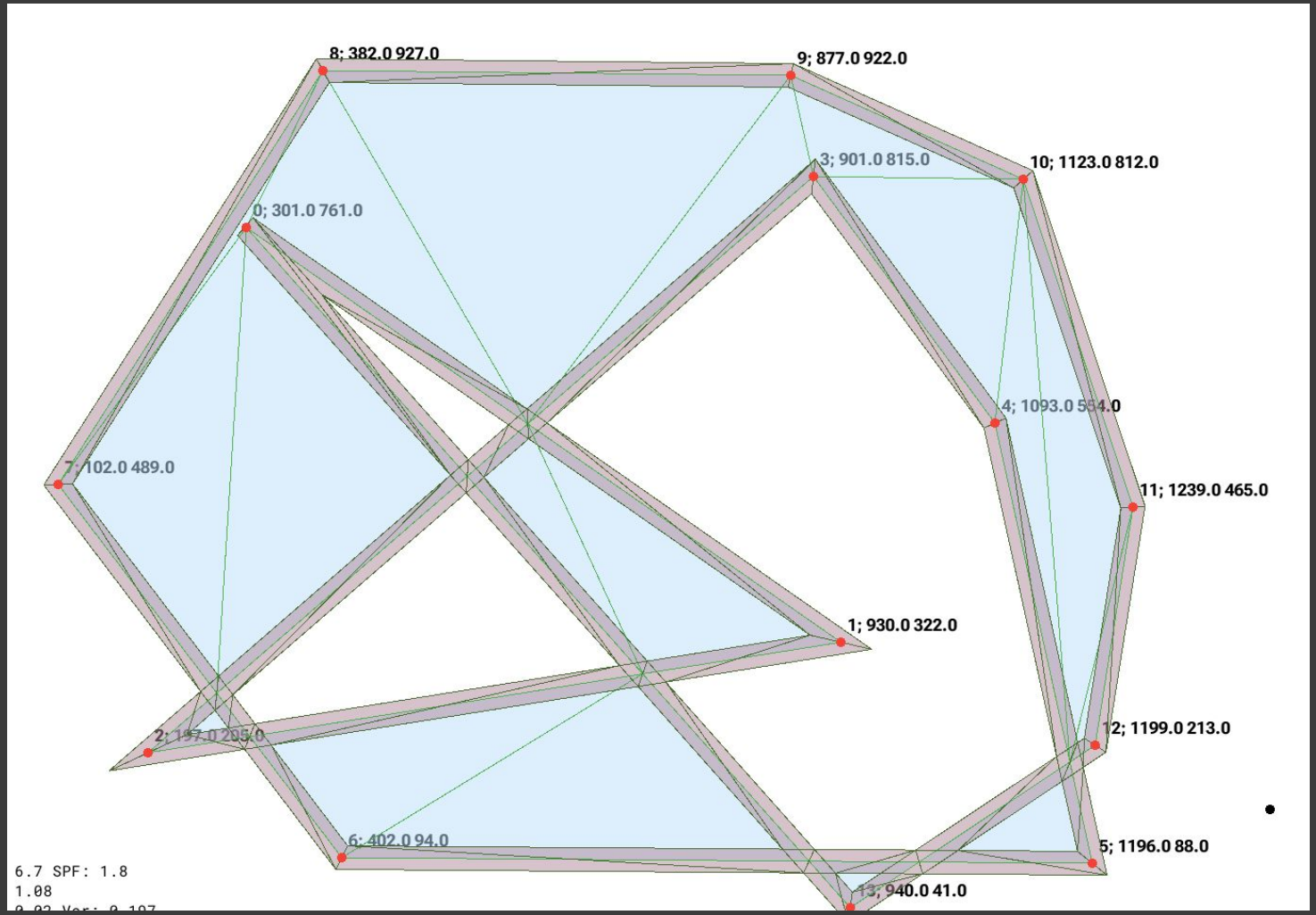
```
auto dataAttachment = builder.addAttachemnt("NoiseDataAttachment",
        [&] (AttachmentBuilder &attachmentBuilder) -> Rc<Attachment> {
    attachmentBuilder.defineAsInput();
    auto a = Rc<vk::BufferAttachment>::create(attachmentBuilder, core::BufferInfo(
        core::BufferUsage::UniformBuffer, sizeof(NoiseData)
    ));

    a->setValidateInputCallback([] (const Attachment &, const Rc<AttachmentInputData> &data) {
        return dynamic_cast<NoiseDataInput *>(data.get()) != nullptr;
    });

    a->setFrameHandleCallback([] (Attachment &a, const FrameQueue &queue) {
        auto h = Rc<vk::BufferAttachmentHandle>::create(a, queue);
        // ...
        return h;
    });
    return a;
});

auto imageAttachment = builder.addAttachemnt("NoiseImageAttachment",
        [&] (AttachmentBuilder &attachmentBuilder) -> Rc<Attachment> {
    attachmentBuilder.defineAsOutput();
    return Rc<vk::ImageAttachment>::create(attachmentBuilder,
        ImageInfo(Extent2(1024, 768), ImageUsage::Storage | ImageUsage::TransferSrc,
                ImageTiling::Optimal, PassType::Compute, ImageFormat::R8G8B8A8_UNORM),
        ImageAttachment::AttachmentInfo{
            .initialLayout = AttachmentLayout::Undefined,
            .finalLayout = AttachmentLayout::General,
            .clearOnLoad = true,
            .clearColor = Color4F(0.0f, 0.0f, 0.0f, 0.0f)}
    );
});

builder.addPass("NoisePass", PassType::Compute, RenderOrdering(0),
        [&] (QueuePassBuilder &passBuilder) -> Rc<core::QueuePass> {
    return Rc<NoisePass>::create(builder, passBuilder, dataAttachment, imageAttachment);
});
```

Графика: векторные изображения

Собственный универсальный тесселятор вместо libtess2, Skia, cairo и так далее. Рендерит что угодно векторное:

- С самопересечениями и без
- Монотонные и нет
- Антиалиасинг
- Обводки
- (экспериментально) порождает Acceleration Structure для решения задачи точки внутри полигона на GPU.
- Работает там, где libtess2 падает и ошибается
- Работает быстро и многопоточно
- Адаптирован для ввода данных на GPU




8; 382.0 927.0
9; 877.0 922.0
3; 901.0 815.0
10; 1123.0 812.0
0; 301.0 761.0
4; 1093.0 584.0
7; 102.0 489.0
11; 1239.0 465.0
1; 930.0 322.0
12; 1199.0 213.0
2; 197.0 205.0
6; 402.0 94.0
5; 1196.0 88.0
13; 940.0 41.0
6.7 SPF: 1.8
1.08

Понятно, что копить деньги надо не в чулке и не под матрасом, а в виде каких-то финансовых инструментов (либо же организовывать свой собственный бизнес, но предположим, что вы не чувствуете в себе предпринимательских талантов). Вы решаете формировать диверсифицированный инвестиционный портфель (см. главу 8 «Фондовый рынок»), в котором 25 % средств будет вложено в рублевый депозит в надежном банке, 25 % - в облигации федерального займа (ОФЗ), 25 % - в акции и 25 % - в корпоративные облигации. По вашим расчетам, ожидаемая средняя доходность такого портфеля составит 9 % годовых, или 0,72 % ежемесячно, - весьма оптимистичный прогноз, но вам он кажется реалистичным (о расчете сложных процентов см. главу 7 «Кредиты и займы»). Какую сумму надо откладывать ежемесячно, чтобы достичь поставленной цели? Если не ответить на этот вопрос, то ведение бюджета само по себе к этой цели вас не приведет.

Из формулы

$$S_n = A \times \frac{(1+r)^n - 1}{r},$$

где A - величина аннуитетного платежа (в данкладываемой суммы), онце периода, n - чисдаемой доходности в олучаем не очень ра-

FPS: 3.76 SPF: 1.99
GPU: 0.679
Dir: 0.038 Ver: 0.424
F12 to switch

Савелий Крамаров о семейном бюджете (СССР, 1971 год)

В этом ролике персонаж замечательного советского артиста Савелия Крамарова рассуждает о семейном бюджете, выражая доходы и расходы... в бутылках водки. Понятно, что даже при приличной по тем временам зарплате семье было бы крайне трудно сводить концы с концами, если более трети (!) своих доходов она бы тратила на водку. Впрочем, приведенные в монологе сведения не надо воспринимать буквально - это все-таки юмористическое произведение с элементами социальной сатиры.

Подробнее

Анализ доходов и их оптимизация

А что тут анализировать? - скажете вы. - Вот моя стипендия, вот зарплата, вот премия, вот «пособие» от родителей. Что есть, то и есть, больше не станет, как ни переставляй эти цифры. Думать про расходы, какие из них урезать - неприятно, но можно; а доходы просто записал - и все дела.

Разве не так?

Так, но не совсем

FPS: 15.1 SPF: 1.64
GPU: 0.506
Dir: 0.029 Ver: 0.279
F12 to switch

, несмотря на все по-
вы не можете свести
берете в долг, либо
ежения - то, может

ЗАКЛАДКИ | ИСТОРИЯ | ГЛОССАРИЙ

исходи
ности.

И,
жета я
ки», п
напитк
коголе
но и
так ка
ровье,
а со вр

Са
бюдж

В э
ветско
о семе
ды... в
прилич
ло бы
если б
ла на
сведен
все-та

today с 21:15

Глава 4. Личный бюджет и финансовое планирование

Раздел 4.3. Техника и технология ведения личного бюджета

21:18 — 4.3.3. Принцип «Поставь перед собой Большую Цель»

13 may с 19:30

Глава 9. Валюта

Раздел 9.4. Заработать на валюте

19:30 — 9.4.3. Инфляция и обменный курс

13 may с 13:59

Глава 9. Валюта

Раздел 9.3. Операции с валютой

14:00 — 9.3.4. Занимать?

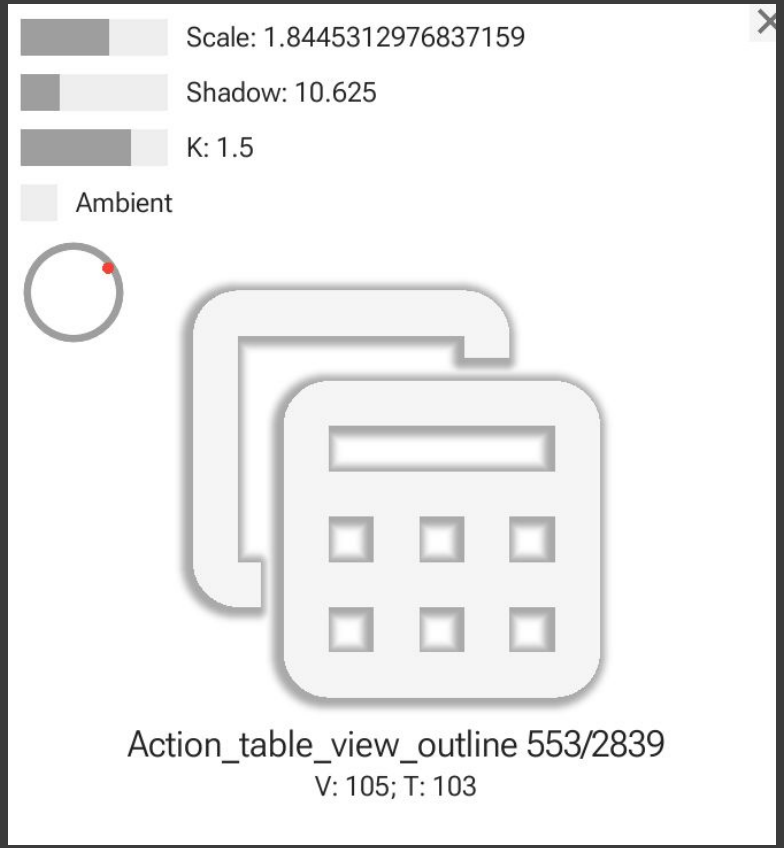
Глава 3. Доходы

Дополнительные материалы

13:59 — Расчет показателей рентабельности бизнеса

стирование

FPS: 1.77 SPF: 2.66
GPU: 1.29
Dir: 0.064 Ver: 0.421
F12 to switch



Немного экспериментальщины: SDF-тени

Что есть:

- Linux, Windows, Android
- x86, amd64, armv7, arm64, e2k
- Интерфейс к БД
- WebAssembly
- 2D-движок
- Минибраузерный движок
- Небольшой набор виджетов
- Куча вспомогательных функций
- Сериализованный код для обработки ИИ и инструментами

Чего нет:

- Биндингов ко всему для WebAssembly
- Больше виджетов
- Больше ОС и архитектур

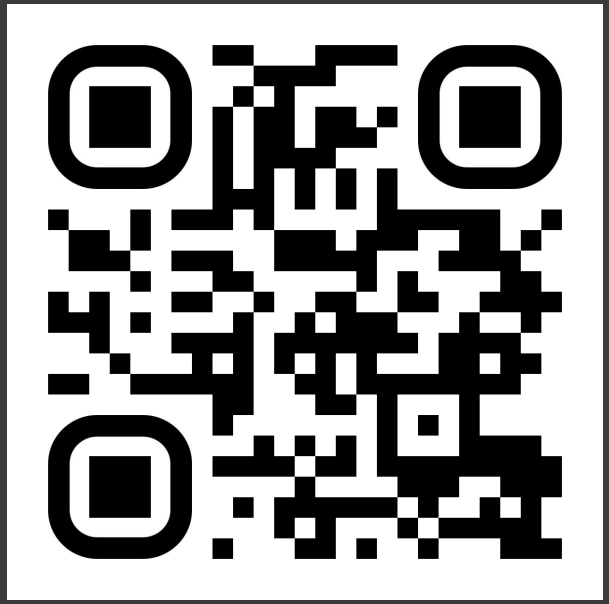
Что мы хотим:

- Единый фреймворк и SDK для отечественных ОС
- Открытую среду для мобильного линукса (No more Plasma!)
- Своё сообщество, завязанное на свои технологии и свои компании
- Выход за пределы России: Иран, Индия, полинезия, Африка, Южная Америка

Sta++ler SDK

Stappler

SDK



stappler.dev

Конкретный случай - реализация медкарты

Клиент:

- Хранит свои данные в виде шифроблока
- Не может вскрыть шифроблок
- Может подтвердить источник шифроблока

Сервер:

- Не хранит критических пользовательских данных
- Получает и расшифровывает шифроблок
- Создаёт и отправляет новый шифроблок

Только вместе имеют полные данные!

Sta++ler SDK