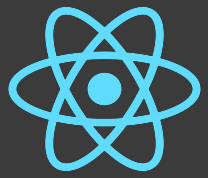
Stappler
SDK

Проблемы подготовки прикладных разработчиков в современной России

Муканин Дмитрий

Современные фреймворки и технологии

- Основной приоритет - снижение «порога входа» в разработку, а не стандартизация и систематизация решений.
- Фокусировка на одной задаче, например графической (философия Unix).



React Native

Jetpack Compose

Тотальное снижение уровня компетентности разработчиков

Система не предусматривает развитие разработчиков выше “порога входа”.

Пренебрежение оптимизацией приложений:

- рост требований к железу
- увеличение потребляемой электроэнергии

Использование “высокоуровневых” языков

Table 4: Normalized global results for Energy, Time, and Memory

Total					
	Energy (J)		Time (ms)		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

Rui Pereira, Marco Couto, Francisco Ribeiro, Rui Rua, Jácome Cunha, João Paulo Fernandes, and João Saraiva: Ranking Programming Languages by Energy Efficiency. Science of Computer Programming, volume 205. Elsevier, 2021.

Stappler SDK

Дополнительные ограничения

С помощью таких фреймворков:

- Легко и быстро делать типовые решения.
- Нестандартные решения делать, наоборот, сложнее.

Результат:

- Система не в состоянии делать по-настоящему сложные приложения, такие как: PLM, RM, PM, PDM, CAD, CAE, CAM, CAPP и тому подобные.
- Мы оказываемся завязаны на уже существующие бренды или, как минимум, уже существующие фреймворки.

Решать задачи с помощью ИИ

Задачи, которые пытаются решить с помощью ИИ, могли бы быть решены более энергоэффективно алгоритмически, будь для этого компетентные специалисты:

- статистического анализа
- кластеризации
- управления предприятием
- представления данных:
  - вместо LLM имеет смысл использовать правильно организованную объектно-ориентированную БД

Stappler
SDK

Финал предсказуем

- ИИ здесь - затычка.
- ИИ не развивается, не предлагает глобальные решения, не создаёт собственные новые подходы.
- Костыль в виде ИИ - тупик, который выбивает среднее звено прикладных специалистов. А без этого среднего звена у нас не будет высшего звена.
- Без развития собственной школы программирования, от низшего до высшего уровня, однажды мы упрёмся в энергетический кризис ИИ.

Проблемы российского рынка

Нужны специалисты способные интегрировать решения для:

- популярных западных платформ
- российских
- новых китайских

в общую систему.

Это необходимо для обеспечения экспорта наших технологий в дружественные страны.

Stappler
SDK

Проблемы российского рынка

- В условиях замкнутости разработок внутри страны, они не развиваются.
- Исчезновение отечественных разработок во многом ставит нас под контроль других государств.
- Недооценка важности мобильных технологий и проектов вроде Firebase в вопросах безопасности.

Важные условия

1. Отечественные продукты должны быть глобальными.
2. Основываться на коде и системах под нашим контролем
3. Необходимы специалисты, способные этим заниматься системно.

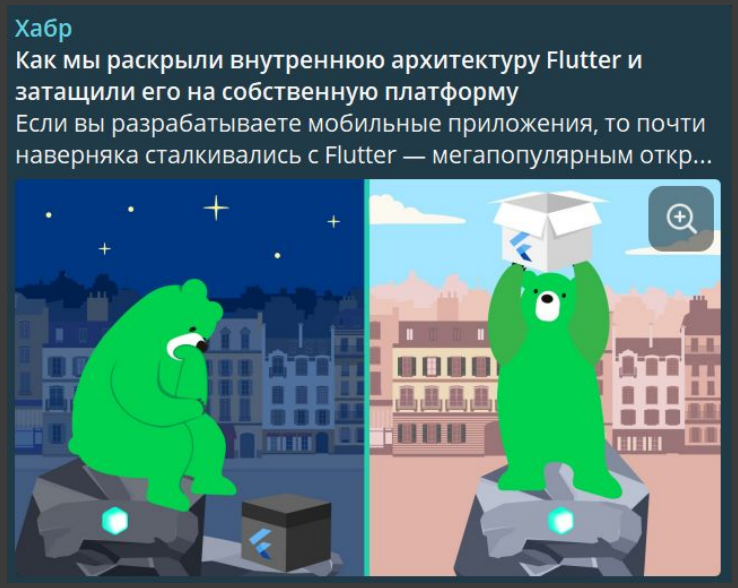
Stappler
SDK

Очевидное решение

Сделать форк необходимых технологий.

Проблемы:

- Отечественные разработчики ОС и другого большого софта не готовы поддерживать даже собственные версии зарубежных фреймворков.
- Возможность участвовать в развитий зарубежных решений становится недоступной.



Необходим системный подход

1. Необходимы собственные, отечественные решения, которые заместили бы проблемные зарубежные системы и фреймворки.
2. Необходимы специалисты, которые бы умели с ними работать.

Stappler
SDK

Заниматься образованием

- Создание образовательно-практического консорциума, в задачи которого будет входить:
  - подготовка специалистов для компаний по отечественным технологиям
  - развитие этих технологий на благо всех участников консорциума.
- Необходимо вкладываться в образование, нельзя полагаться только на FOSS.

Есть такой фреймворк!

Qt

Flutter

Kotlin Multiplatform Mobile

React Native

Stappler SDK

Открытый фреймворк для разработки кроссплатформенных приложений, сделанный в России

Stappler SDK

Приоритеты:

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

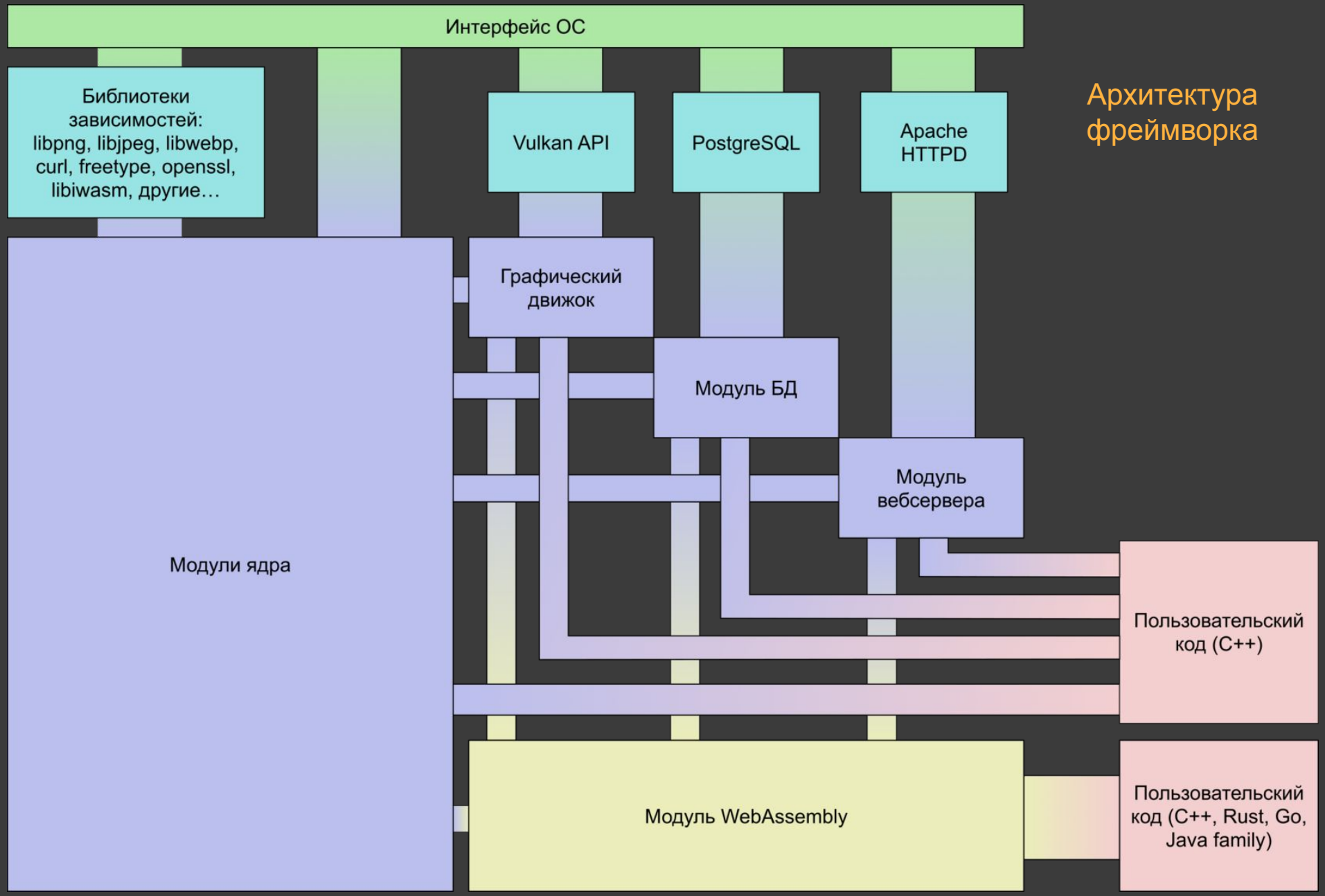
ФОНД СОДЕЙСТВИЯ ИННОВАЦИЯМ

Stappler SDK

Кратко о фреймворке

- Высокопроизводительная графика и вычисления на базе Vulkan API
- Единая база кода для всего набора инструментов, на любом языке:
  - упрощенный C++, C/C++, Rust, Go, Java family, JavaScript и др.
- Android, iOS, Linux, Windows, MacOS
  - в том числе российские ОС
- Платформы: ARM, x86, e2k (Эльбрус)
  - быстрая адаптация под новые (RISC-V, LoongArch и другие)
- Реализация всей инфраструктуры приложения на одном фреймворке:
  - встроенный модуль работы с базами данных
  - модуль веб-сервера

Stappler
SDK


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

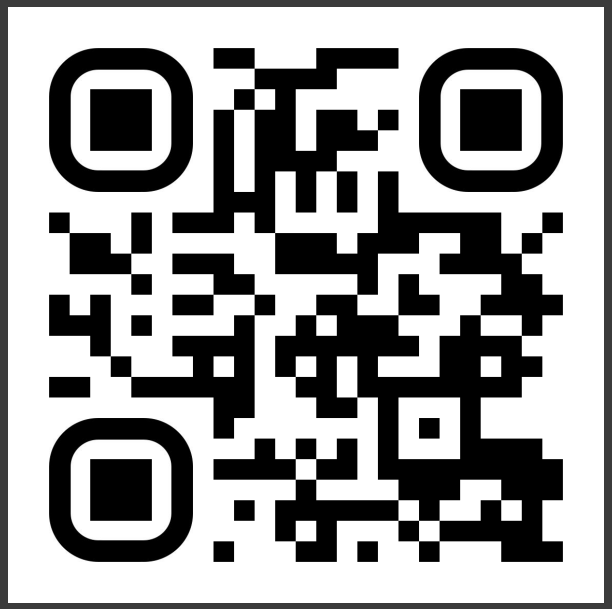
Соответствует требованиям

1. Поддержка отечественных и глобальных платформ, быстрая интеграция новых.
2. Очень высокая производительность и энергоэффективность приложений за счет жёсткой оптимизации графики, использования оптимизированных форматов обмена данными, сжатия и многопоточности.
3. Использование стандартных языков программирования.
4. Вся инфраструктура проекта в одном месте.
5. Возможность соответствия строжайшим требованиям безопасности.
6. Фреймворк прост в освоении и подходит для обучения новых разработчиков широкого профиля.

Stappler
SDK

Stappler
SDK

Спасибо за внимание!



stappler.dev