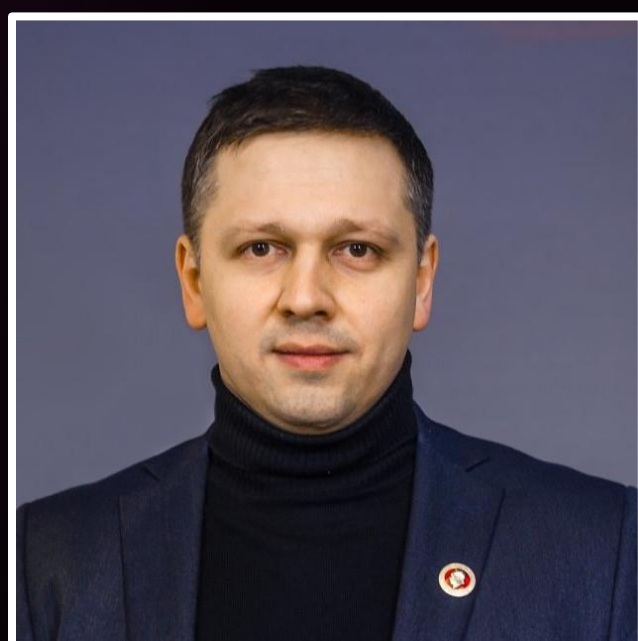


Node.js

Меньше сложности,
больше надежности

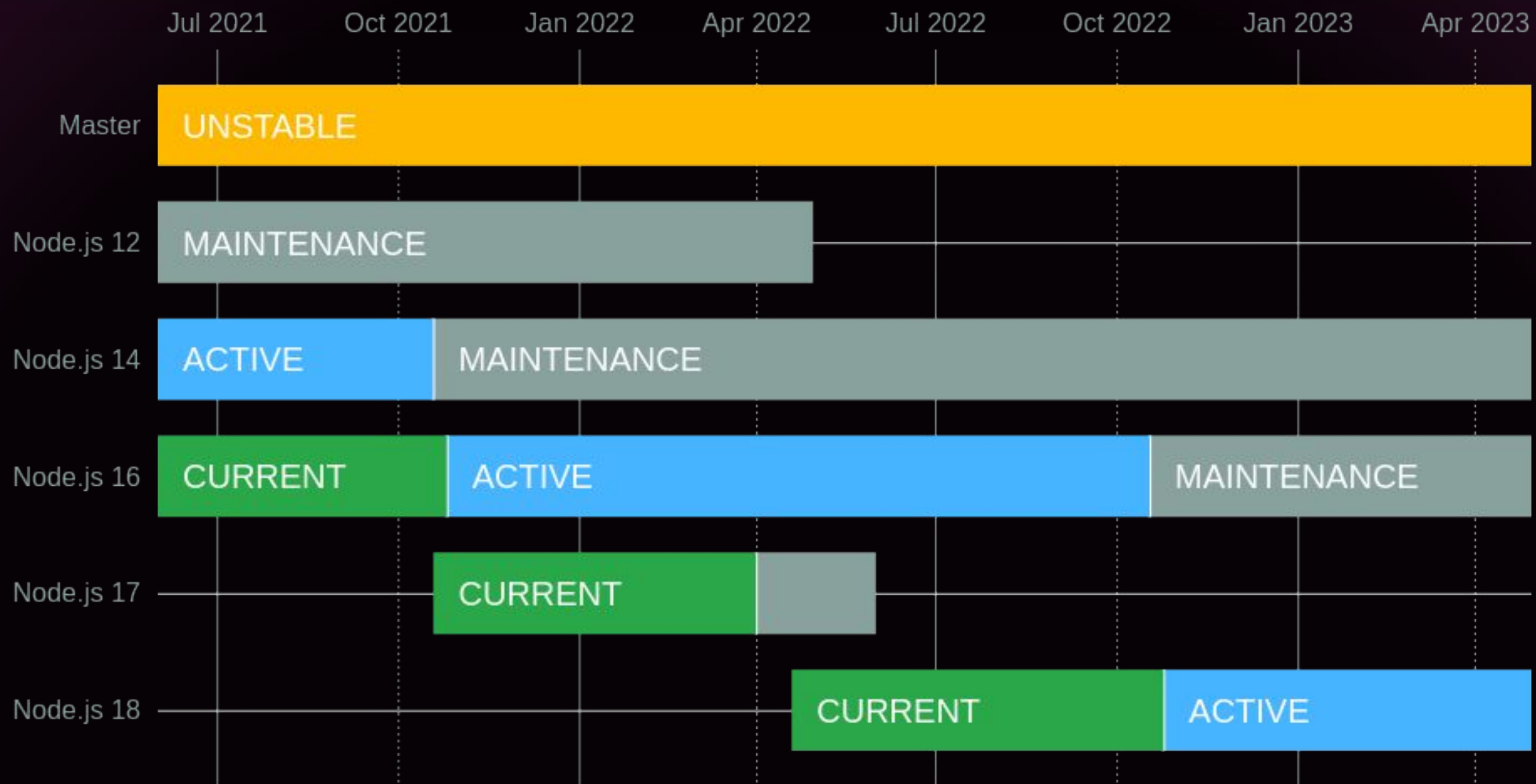


Timur Shemsedinov

Chief Architect at Metarhia, CTO at Salucyber
Lecturer at KPI & KSE

github.com/tshemsedinov

Куда мы идем?



МНОГО НОВОГО

ПОТОКИ

Promises API

ПОТОКИ

Promises API

ПОТОКИ

Больше Web API

ESM

Promises API

ПОТОКИ

Больше Web API

ESM

Promises API

ПОТОКИ

WASI

Больше Web API

Когда fetch завезут?

ESM

Promises API

ПОТОКИ

WASI

Больше Web API

Когда fetch завезут?

ESM

Promises API

ПОТОКИ

WASI

Че там со стримами?

Больше Web API

Когда fetch завезут?

ESM

Promises API

ПОТОКИ

WASI

Че там со стримами?

Больше Web API

http/3

Но подождите

**Но подождите
Куда мы несемся?**

У нас же все сломано

Но подождите

Куда мы несемся?

Фатальные недостатки

Какие проблемы?

Какие проблемы?

- Надежсодноточный

Какие проблемы?

- Нодежсоднопоточный
- Джаваскриптоднопоточный

Какие проблемы?

- Нодежсоднопоточный
- Джаваскриптоднопоточный
- Модель параллелизма в go гораздо лучше

Какие проблемы?

- Нодежсоднопоточный
- Джаваскриптоднопоточный
- Модель параллелизма в go гораздо лучше

github.com/metarhia/noroutine

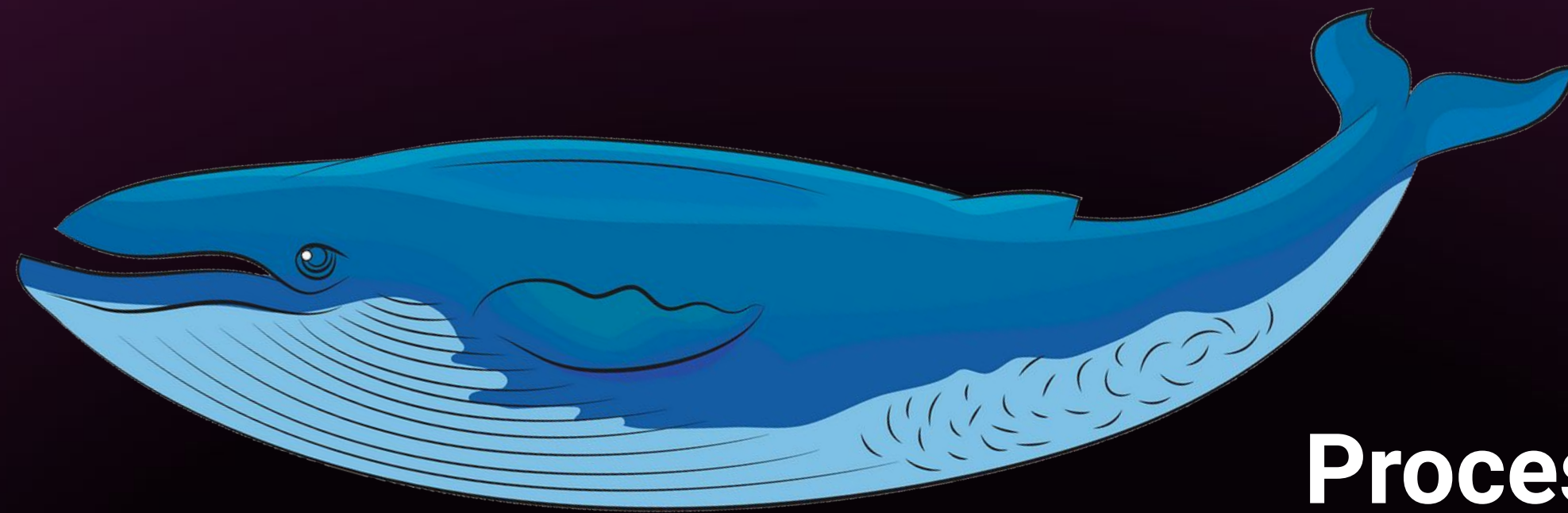
Какие проблемы?

Какие проблемы?

- Habr: 2011
“**Node.js — раковая опухоль**”
Александр Кавун

Какие проблемы?

- Habr: 2011
“**Node.js — раковая опухоль**”
Александр Кавун
- Holy.js 2017
“**Фатальный недостаток Node.js**”
Алексей Охрименко



Process



Thread



v8::Isolate



v8::Context

Какие проблемы?

Какие проблемы?

- Память течет

Какие проблемы?

- Память течет
- Callback hell

Какие проблемы?

- Память течет
- Callback hell
- Promise hell

Какие проблемы?

- Память течет
- Callback hell
- Promise hell
- Блокирующие операции

Какие проблемы?

- Память течет
- Callback hell
- Promise hell
- Блокирующие операции
- Неотловленные исключения

Какие проблемы?

- Память течет
- Callback hell
- Promise hell
- Блокирующие операции
- Неотловленные исключения
- Процесс крашится

Какие проблемы?

- Память течет
- Callback hell
- Promise hell
- Блокирующие операции
- Неотловленные исключения
- Процесс крашится
- Дыры в безопасности

Ваш код в окружении node_modules



< 50 Kb


> 500 Mb

В чем же проблема?

КОРА
МЫ

ТРМEXAM

Разруха в ГОЛОВОАХ



ПРЕЖДЕ ЧЕМ ЛЕЗТЬ
К ЗВЕЗДАМ, ЧЕЛОВЕКУ
НУЖНО НАУЧИТЬСЯ
ЖИТЬ НА ЗЕМЛЕ!

SHOTS · РУСЛАН · АНДРЕЙ · ВАРС

Разруха в головах

- Слабое знание Node.js

Разруха в головах

- Слабое знание Node.js
- Сотни мегабайт зависимостей

Разруха в головах

- Слабое знание Node.js
- Сотни мегабайт зависимостей
- Устаревшие фреймворки

Разруха в головах

- Слабое знание Node.js
- Сотни мегабайт зависимостей
- Устаревшие фреймворки
- Недостаток знаний и культуры в сообществе

Разруха в головах

- Слабое знание Node.js
- Сотни мегабайт зависимостей
- Устаревшие фреймворки
- Недостаток знаний и культуры в сообществе
- непонимание асинхронщины

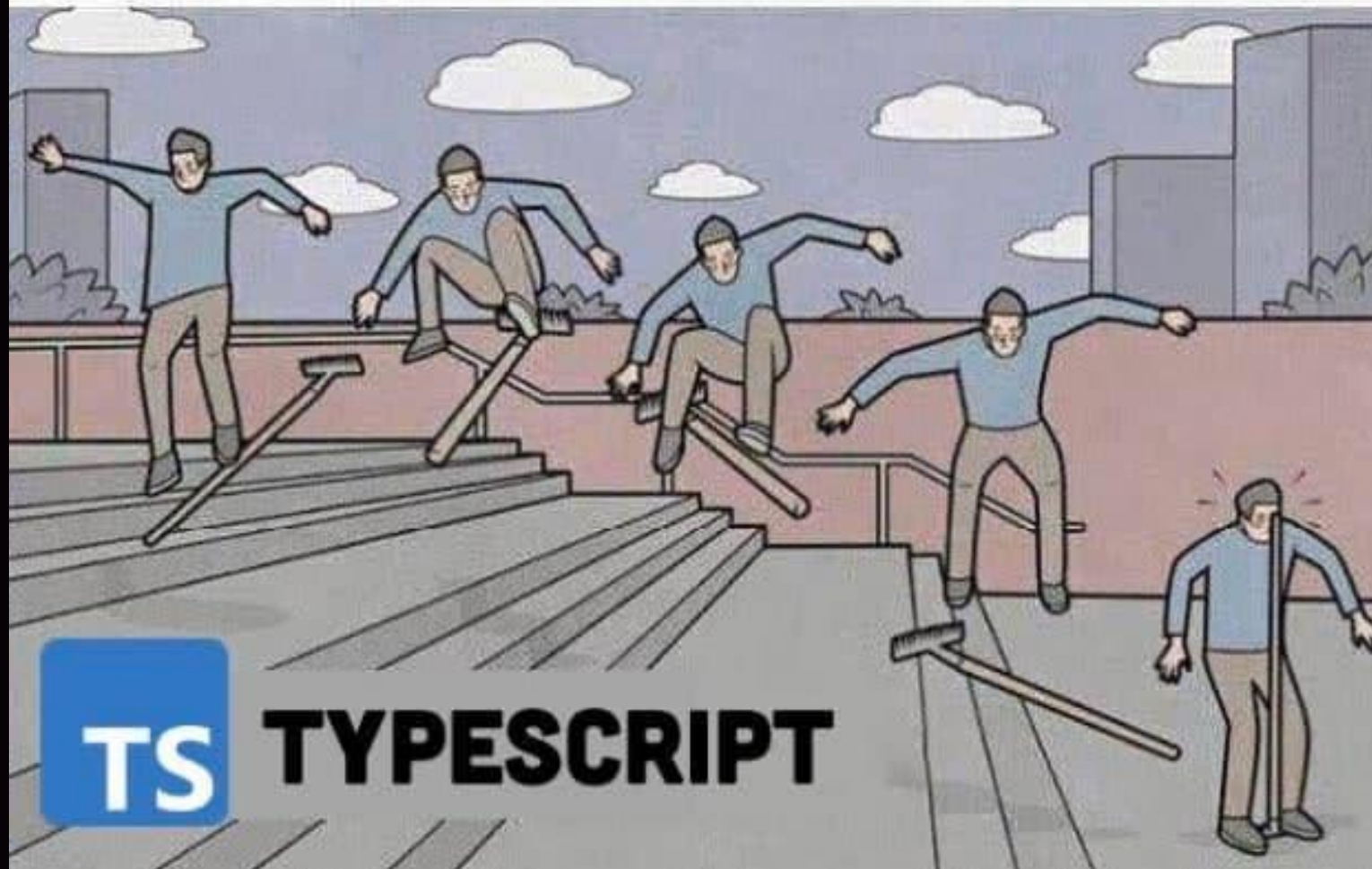
Разруха в головах

- Слабое знание Node.js
- Сотни мегабайт зависимостей
- Устаревшие фреймворки
- Недостаток знаний и культуры в сообществе
- Непонимание асинхронщины
- Быстрый старт, но ...
разработка вязнет в болоте

Что делать?



JS JAVASCRIPT



TS TYPESCRIPT

Будет будет GraphQL

ВЫ ТОЛЬКО НЕ ВОЛНУЙТЕСЬ!

Пишите просто и
понятно! Потом
заливайте, ибо
Отцу угодно, чтоб
все было как у нас
на небе, без магии



Что он
сказал?

Говорит, можно
писать просто
как нам
понятно, а Отец
потом в облаке
это какой-то
магией зальет...





Me

React.js

**Data structures
and algorithms**

Неизвестное внутреннее API

fs, v8, vm, child_process,
worker_threads, perf_hooks, crypto,
assert, stream/web,
async_hooks, async_context,
events, wasi

Как выбирать зависимости

axios, express, lodash, async.js,
bcrypt, argon2, mkdirp, bluebird,
socket.io, request, cluster

Как выбирать зависимости

axios, express, lodash, async.js,
bcrypt, argon2, mkdirp, bluebird,
socket.io, request, cluster

undici, pino, fastify, ws, llparse

Как выбирать зависимости

~~axios, express, lodash, async.js,
bcrypt, argon2, mkdirp, bluebird,
socket.io, request, cluster~~

undici, pino, fastify, ws, llparse

Как выбирать зависимости

Как выбирать зависимости

- Возможно, встроенные модули это умеют

Как выбирать зависимости

- Возможно, встроенные модули это умеют
- Пакеты с похожими именами

Как выбирать зависимости

- Возможно, встроенные модули это умеют
- Пакеты с похожими именами
- Зараженные пакеты

Как выбирать зависимости

- Возможно, встроенные модули это умеют
- Пакеты с похожими именами
- Зараженные пакеты
- Перед использованием смотрим код

Как выбирать зависимости

- Возможно, встроенные модули это умеют
- Пакеты с похожими именами
- Зараженные пакеты
- Перед использованием смотрим код
- Смотрим вложенные зависимости

Как выбирать зависимости

- Возможно, встроенные модули это умеют
- Пакеты с похожими именами
- Зараженные пакеты
- Перед использованием смотрим код
- Смотрим вложенные зависимости
- Проверяем по базам уязвимостей

Как выбирать зависимости

- Возможно, встроенные модули это умеют
- Пакеты с похожими именами
- Зараженные пакеты
- Перед использованием смотрим код
- Смотрим вложенные зависимости
- Проверяем по базам уязвимостей
- Проверяем сообщество и поддержку

Недостаток культуры разработки

Недостаток культуры разработки

- Отрицание DDD и Clean Architecture

Недостаток культуры разработки

- Отрицание DDD и Clean Architecture
- Слабое проникновение GRASP, SOLID, GoF

Недостаток культуры разработки

- Отрицание DDD и Clean Architecture
- Слабое проникновение GRASP, SOLID, GoF
- Перемешивание слоев абстракций

Недостаток культуры разработки

- Отрицание DDD и Clean Architecture
- Слабое проникновение GRASP, SOLID, GoF
- Перемешивание слоев абстракций
- Повсеместное использование антипаттернов:
mixin, middleware,
reference pollution, prototype pollution ...



Уязвимости роутинга

- ReDOS
- Path traversal

Уязвимости роутинга

- ReDOS
- Path traversal

Вместо этого,
можно искать эндпоинт по URL
в коллекции (Map)

Уязвимости

XSS, SQLI, XSRF, заражения из NPM...

CSP (Content Security Policy)

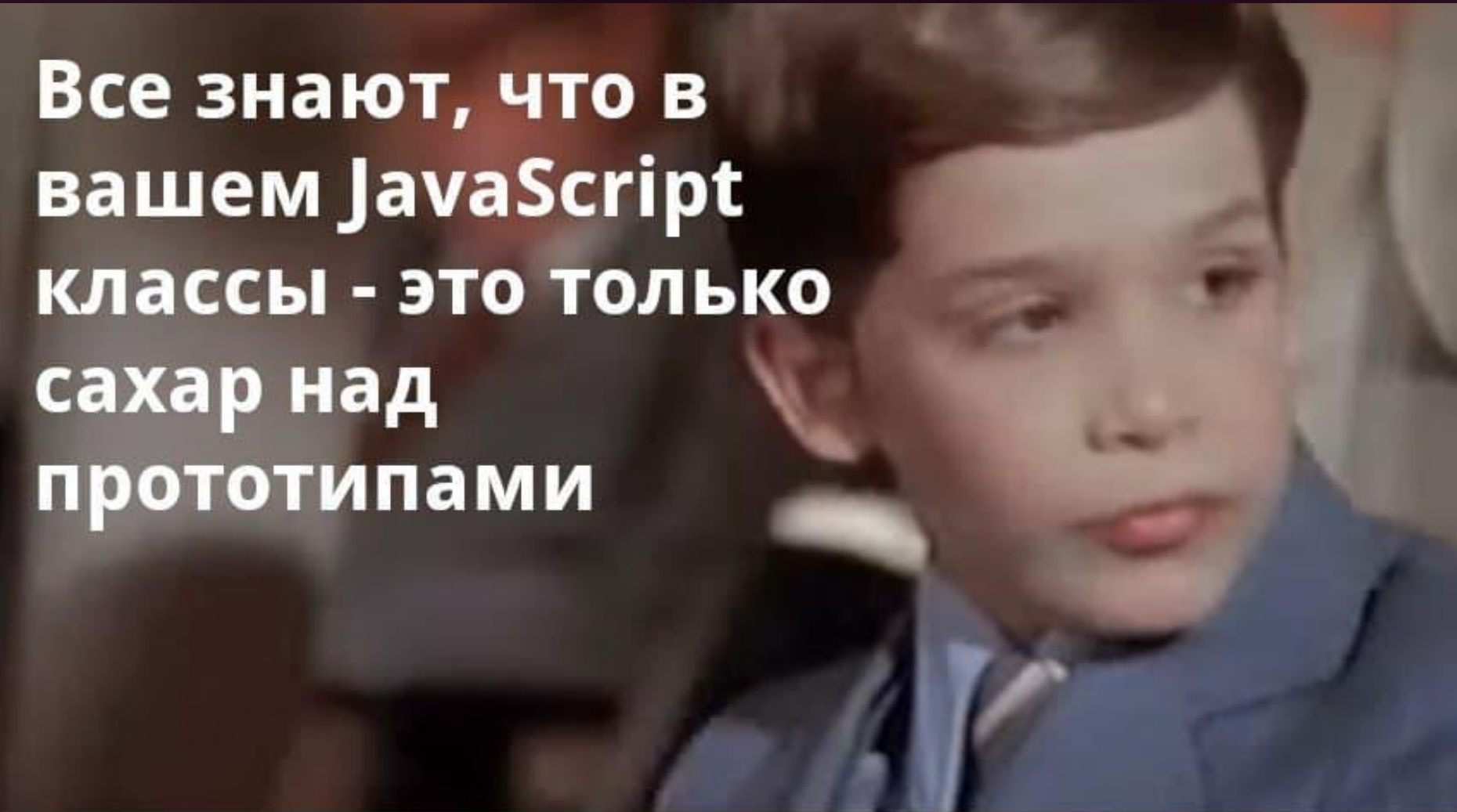
[https://developer.mozilla.org/en-US/docs/
Web/HTTP/CSP](https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP)

npm audit, snyk, Github Security Alert

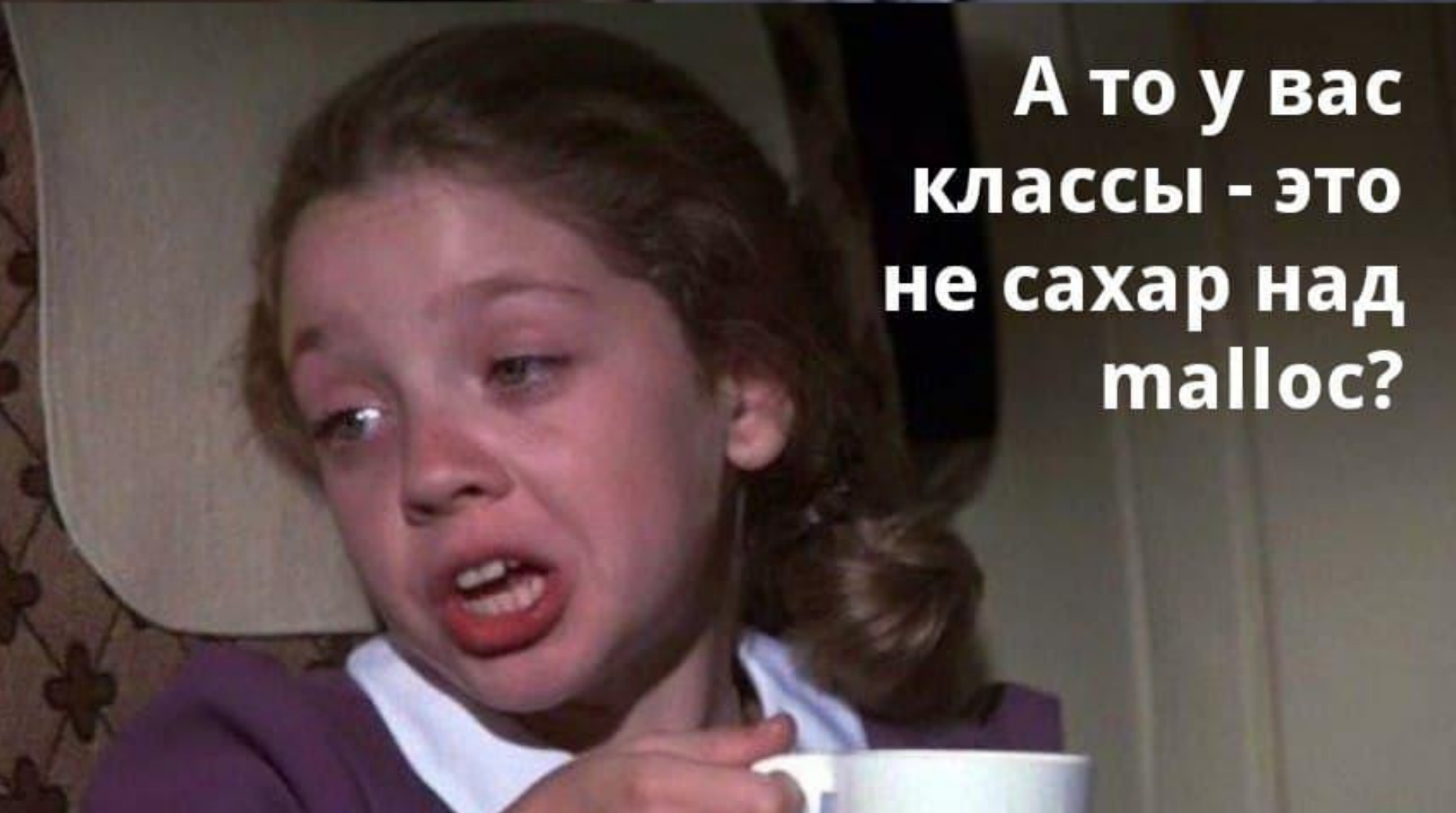
Нестабильность процессов

- Утечки памяти и других ресурсов (приводящие к падению)
- Игнорирование ошибок
- Сильная связанность модулей
- Массовый манкипатчинг структур данных

Все сложно

A young boy with short brown hair, wearing a blue suit jacket and a light-colored tie, looking slightly to the right with a serious expression.

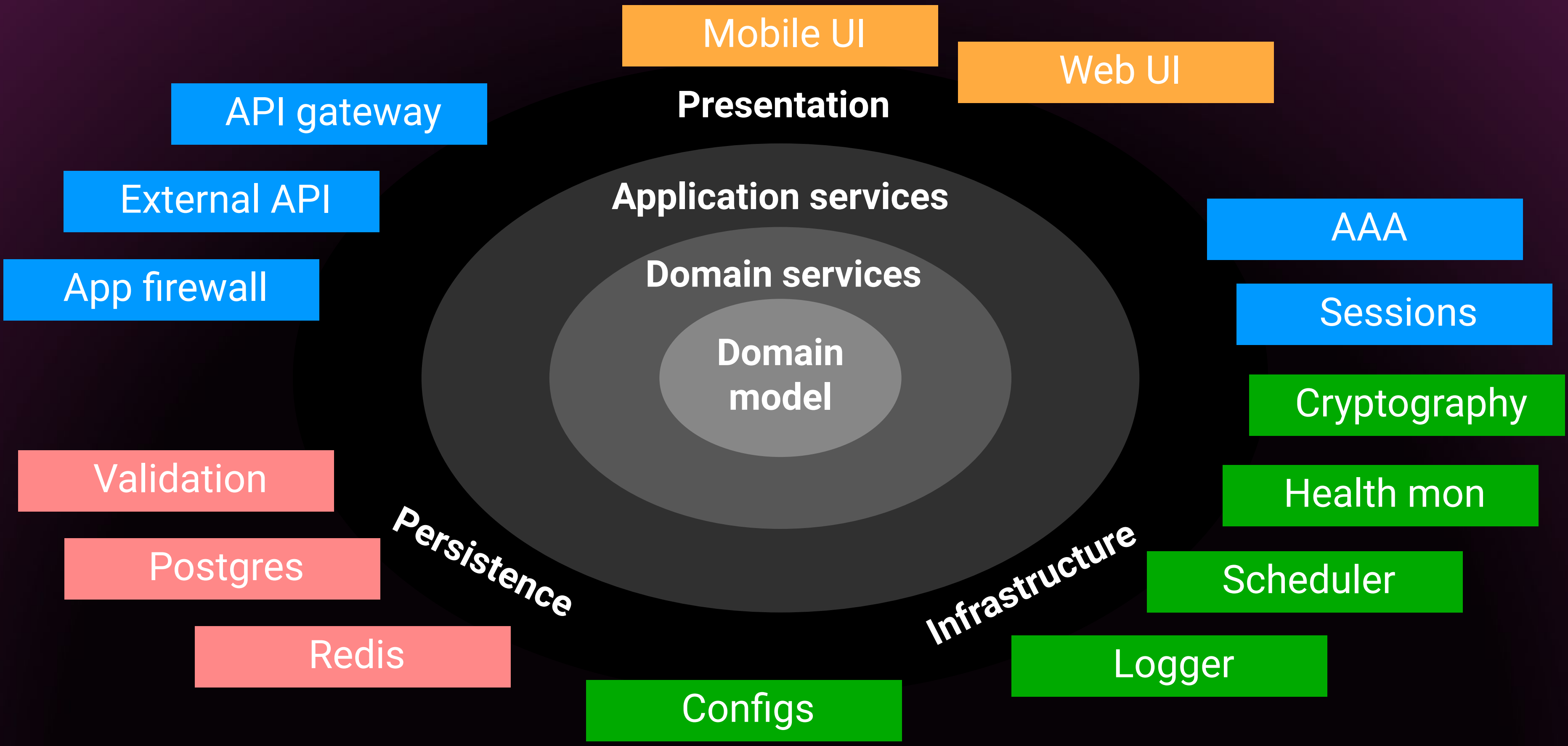
**Все знают, что в
вашем JavaScript
классы - это только
сахар над
прототипами**

A young girl with curly brown hair, wearing a purple sweater over a white collared shirt, looking upwards with a surprised or questioning expression.

**А то у вас
классы - это
не сахар над
malloc?**

КТО ВИНОВАТ?

Clean Architecture



Mobile UI

Web UI

API gateway

Presentation

External API

Application services

AAA

App firewall

Domain services

Sessions

Domain model

Cryptography

Validation

Health mon

Postgres

Persistence

Infrastructure

Scheduler

Redis

Logger

Configs

Внедрение зависимостей

- Через передачу в конструктор класса
- Модуль экспортирует функцию, в которую передаются зависимости
- Через DI контейнеры
- Через `vm: createContext, createScript`
- Зависимости импортируются в одном модуле и оттуда экспортируются

Что интересует программистов

Язык: любимый синтаксис, крутые фишки

IDE: подсказки, автокомплит, финтифлюшки

Привычные и популярные фреймворки

Наслушались: нагрузки, микросервисы, арх...

Инструменты: оптимизация, дебаг, либы

Экосистема: сообщества, группы, S-Overflow

Должно быть важно программистам

Моделирование предметной области

Язык и парадигмы: синтаксис и семантика

Простота и сложность, поддерживаемый код

Как не накапливать технический долг

Как минимизировать протекание абстракций

Как избавиться от зависимостей

Снизить зацепление, повысить связность

Важные аспекты архитектуры

- Приложение это место для прикладного кода
- Декларативный код лучше императивного
- Состояние: управляемое и явное
- Конфигурация и инфраструктура в коде
- Изолируем подсистемы и модули
- Используем слоеную архитектуру

Важные аспекты для Node.js

- Избегаем состояния гонки, ограничиваем конкурентность очередями и семафорами
- Закон Деметры
- framework-агностик
- Асинхронное и параллельное программирование
- Реализуем graceful shutdown

Enterprise development requirements

- Корпоративные требования: надежность, поддерживаемость, высокая доступность, безопасность, изоляция, совместимость
- Приложение это место для прикладного кода
- Бизнесу нужны прототипы быстро (TTM)
- Избавляемся зависимостей, качество и безопасность которых сомнительна

Как понять, что что-то пошло не так

- Вы чувствуете себя формошлепом
- Вы постоянно разрабатываете API
- Вы все время решаете задачи тайпскрипта
- Вы мыслите роутингом
- Вы концентрируетесь на фреймворках

Мы заменим тебя
этим роботом

А что он
умеет делать?

Ничего)))



Наступит время, сам поймешь, наверное.



Лекции

Курс по ноде

github.com/HowProgrammingWorks/Index/blob/master/Courses/NodeJS.md

Курс асинхронщине

github.com/HowProgrammingWorks/Index/blob/master/Courses/Asynchronous.md

Все лекции

github.com/HowProgrammingWorks/Index

Доклады

Индекс докладов

github.com/HowProgrammingWorks/Index/blob/master/Courses/Talks.md

Презентация Метархии

youtu.be/PHyl4b8Fj5A

Node.js в 2021

youtu.be/nnB7ADYso8s

Ссылки

Гитхаб

github.com/tshemsedinov

Ютуб

youtube.com/c/TimurShemsedinov

Телеграм

github.com/HowProgrammingWorks/Index/blob/master/Links.md