

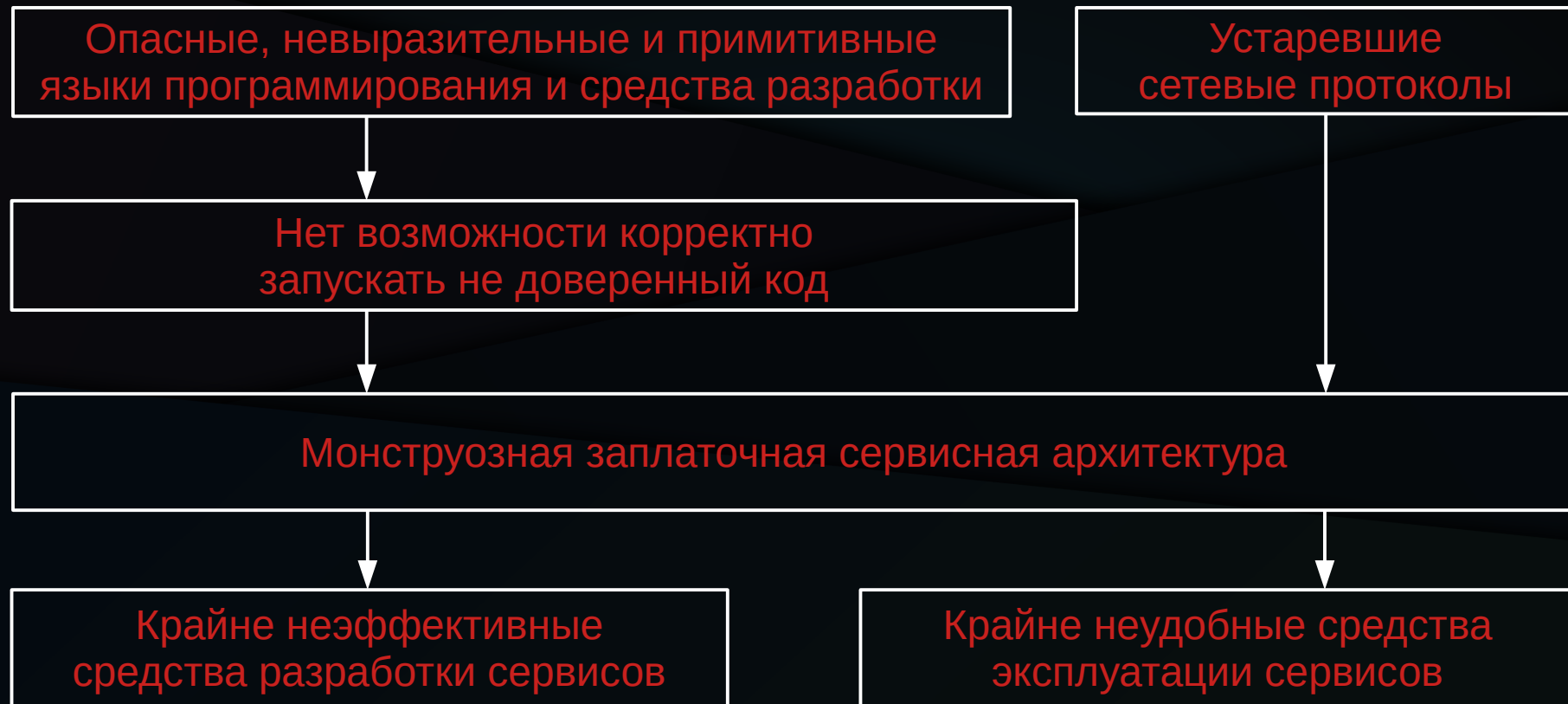
Окопник Григорий
Окопник Евгений

HardCode

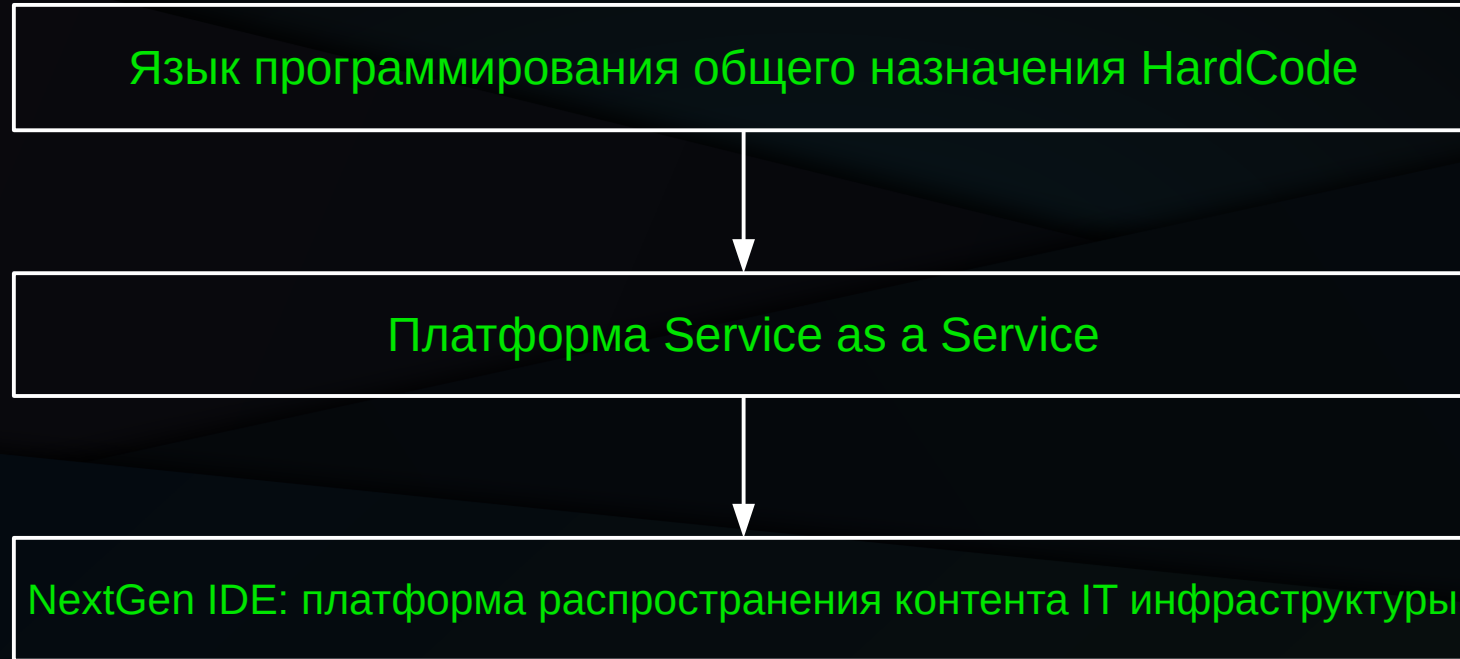
язык программирования общего назначения
платформа разработки и эксплуатации сервисов
платформа распространения контента IT инфраструктуры



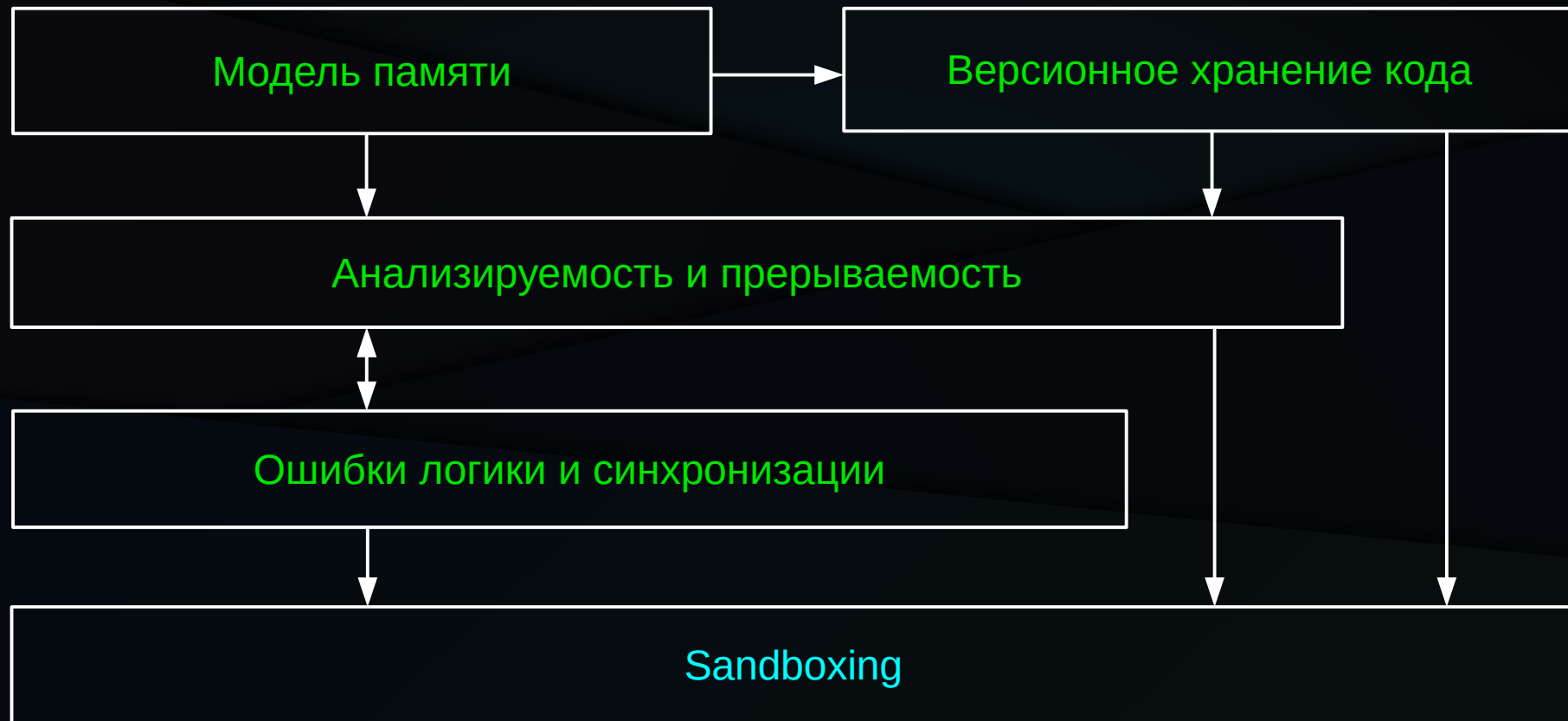
Состояние на сегодняшний день



Структура проекта



Концепция языка программирования HardCode



Концепция языка программирования HardCode

Модель памяти


- Уникальная гибридная модель памяти сочетающая RAII и GC
- Уникальный не блокирующий многопоточный Garbage Collector



- Новый уровень удобства, безопасности и изоляции
- Производительность на уровне C
- Строгая тестируемость

Концепция языка программирования HardCode

Версионное хранение кода

- Код программы хранится в версионной БД
 - Интегрированы продвинутые практики работы с VCS
 - Тестирование и документация в структуре кода в БД
 - Результаты тестирования и профилирования хранятся в БД
- 
- Версионная разработка производится “на лету” – без перезапуска программы
 - Консистентная работа с кодом: рефакторинг, ревью, анализ и т. д.
 - Предметное взаимодействие между разработчиками

Концепция языка программирования HardCode

Анализируемость

- Сложность выполнения программы и её фрагментов анализируема “из коробки”
- Прерываемость и время прерывания анализируемы “из коробки”
- Есть возможность аналитического профилирования



- Очень высокая читаемость кода
- Глубочайшие возможности целевой оптимизации

Концепция языка программирования HardCode

Ошибки логики и синхронизации

- В языке нет необрабатываемых ошибок времени выполнения
- Существенная часть логических ошибок не существует в языке
- Другую часть ошибок довольно сложно допустить
- Выразительная надёжная логика обработки ошибок
- Нет ошибок синхронизации



- Программы не падают (кроме как из-за отсутствия системных ресурсов)
- В программах в разы меньше логических ошибок
- Программы существенно короче, выразительнее и читабельнее
- В разы меньше отладки, а сама отладка в разы удобнее

Концепция языка программирования HardCode

Sandboxing

- В языке присутствует полноценный sandboxing при статической типизации за счёт изоляции, анализируемости и анализируемой прерываемости




- С серверной стороны это означает эффективный FaaS
- С клиентской стороны – полноценную программно-браузерную архитектуру

Платформа Service as a Service



Платформа Service as a Service

Разработка сервисов

- На сегодняшний день практики разработки сервисов сводятся к docker-compose: настройка всей сборки может занять пару недель
 - Платформа Service as a Service позволяет поднять всю связку сервисов в 2 клика
 - Для разработчика нет разницы между монолитом и микросервисами
 - Все данные на сервере, разработка производится через 1 клиентскую программу
- 
- Эффективная разработка, гибкая оптимизация
 - Быстрая включаемость и высокая мобильность разработчика

Платформа Service as a Service

Сетевой протокол

- Кластерный сетевой протокол объединяющий преимущества UDP + gRPC + TLS + dTLS
- Схема API хранится в версионной БД, не позволяющей ломать совместимость



- Оптимальное время отклика
- Адекватный уровень шифрования
- Удобная кластерная модель данных в API
- Нет просадки SLA
- Нет явных и скрытых ошибок сломанной совместимости

Платформа Service as a Service

Анализируемость

- Сложность обработки запроса сервером анализируема
- Сложность работы клиентской части анализируема
- За счёт сетевого протокола дерево обращений к микросервисам анализируемо
- Таким образом, анализируема вся структура сервисов и клиентов
- Стэк ошибок записывается и однозначно идентифицируется



- Интеграция различных сетевых сервисов анализируема как целое
- Логика выполнения запроса анализируема
- Цельная отладочная информация на этапе получения ошибки

Платформа Service as a Service

Управление ресурсами

- Доступны все системные механизмы управления ресурсами (как в K8S)
- Детальная изоляция ресурсов существует на уровне языка
- Оптимизация ресурсов и кода реализуется на базе анализа и аналитического профилирования
- Факторизация микросервисов производится конфигурацией



- Высочайший уровень безопасности
- Отсутствуют оверхеды на контейнерную изоляцию и деплой
- Крайне эффективный процесс оптимизация кода
- Гибкое и удобное аналитическое управление ресурсами

Платформа Service as a Service

Эксплуатация сервисов

- Первичное разворачивание в облаке не требует параметризации, т. е., осуществляется в 2 клика
- Дополнительные настройки минималистичны, т. к. отсутствуют ненужные сущности
- Нет понятия деплоя контейнера
- Статистика и логи собираются без потерь



- Простая и надёжная эксплуатация сервисов: DevOps'ы не нужны!!
- Очень быстрая развёртка и откаты сервисов
- Детальная информация по инцидентам

Платформа Service as a Service

Сервис Service as a Service

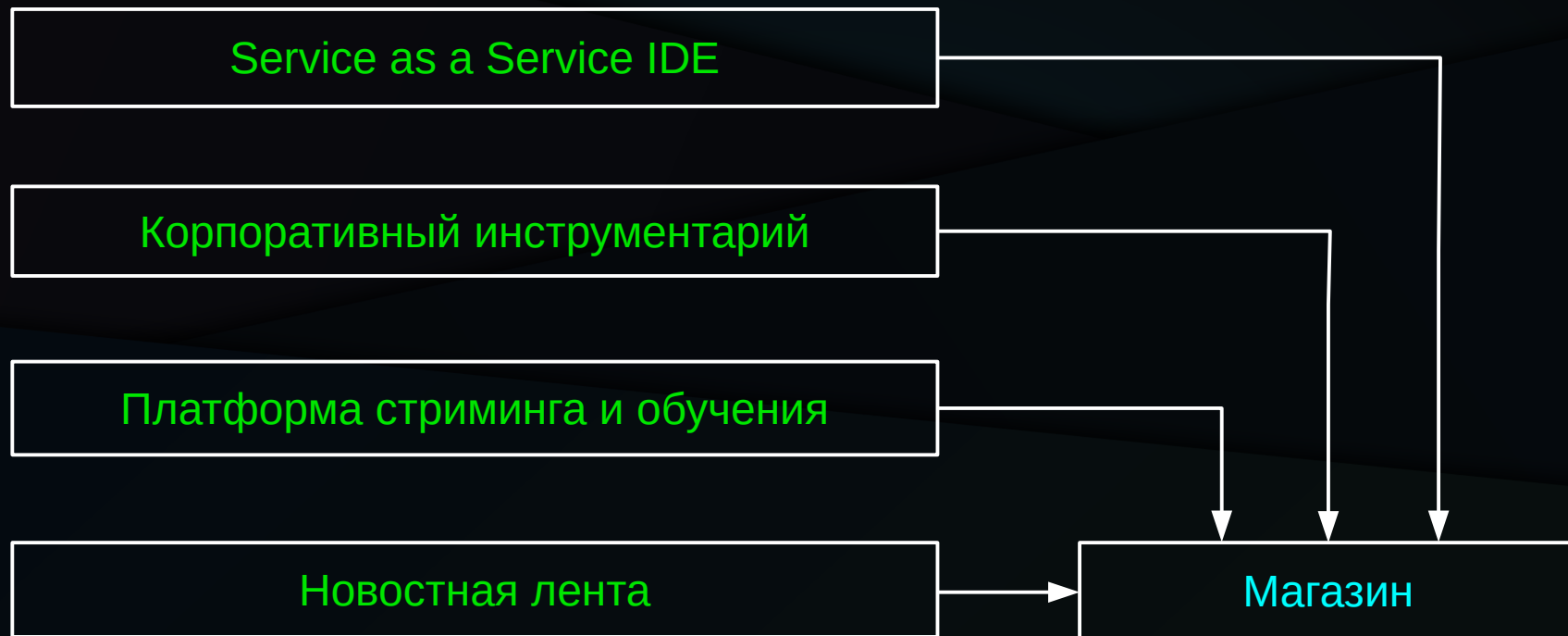
- Сервис сводится к проектированию API и реализации обработчиков запросов (как FaaS + Swagger, но без желудей и спичек)
- Сложные схемы деплоя доступны сразу для любого сервиса
- История событий деплоя и запросов собирается и хранится платформой
- Можем запускать произвольный не доверенный код «из коробки»



- Эффективная разработка сервисов
- Нулевой простой «из коробки» при штатной работе
- Настоящая реализация Function as a Service
- Настоящая реализация Data as a Code
- Работоспособный Аутсорсинг Вычислений

NextGen IDE

Платформа распространения контента IT инфраструктуры



NextGen IDE

Service as a Service IDE

- IDE для разработки сервисов
- Интеграция с системой контроля версий
- Строгий семантический анализ, тестируемость, рефакторинг и отладка
- Управление развёртыванием и настройка CI/CD
- IDE является основным и единственным инструментом разработчика



- Высокая надёжность и эффективность
разработки, отладки, развёртывания и решения инцидентов

Корпоративный инструментарий

- Глубоко интегрированный трэкер задач (аналог: **Jira**)
- HR портал (корпоративные решения)/социальная сеть (аналог: LinkedIn)
- Мессенджер (аналог: Slack)
- Система уведомлений (аналог: **почта**)



- Готовые средства для корпоративных задач
- Настройка рабочих процессов перекладывается с менеджеров на ПО
- Унификация управления временем
- Эффективное управление кадрами/карьерой


Платформа стриминга и обучения

- Стриминг видео и работы с кодом
- Коллективное редактирование кода
- Встраиваемая система обучения
- Геймификация обучения
- Встраиваемый вопросник с гарантированно рабочими сниппетами (аналог: **Stack Overflow**)



- Эффективное обучение кадров и обмен опытом
- Легкодоступный и управляемый найм кадров

Новостная лента

- Рекомендации проектов
 - Рекомендации расширений IDE
 - Рекомендации обучающих курсов
 - Рекомендации вакансий
 - Статьи о HardCode и инфраструктуре
 - Новости IT-индустрии
- 
- Актуальная релевантная информация для разработчика
 - Широкие рекламные возможности

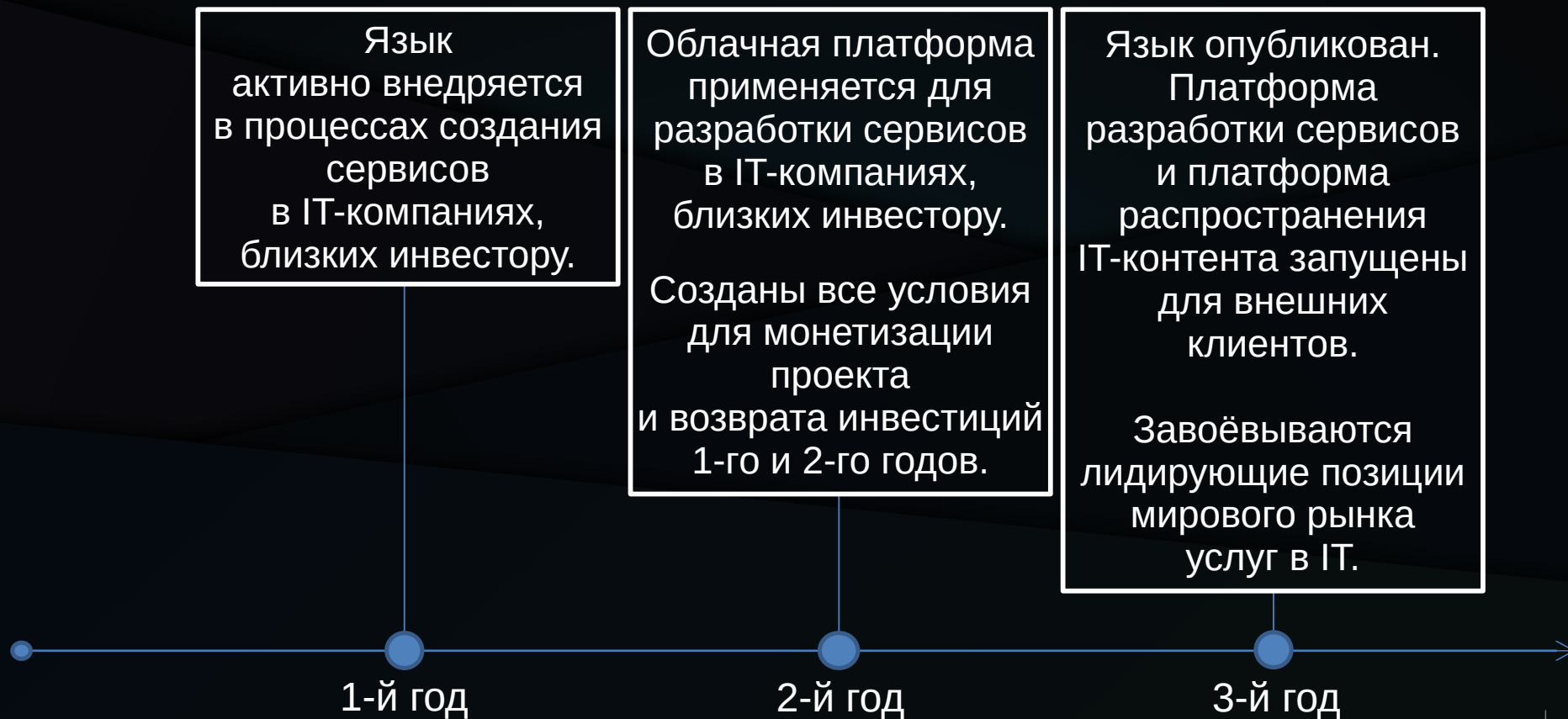
Магазин: собственные и партнёрские решения

- Ресурсы облачной платформы Service as a Service
- Расширения IDE: отладчики, анализаторы, редакторы, линтеры и т. д.
- Корпоративный инструментарий: обширное множество компонент
- Обучающие курсы, обзоры, хакатоны, подписки на стримы (близкие аналоги: YouTube, Twitch)
- Библиотеки и сервисы (аналог: GitHub)
- Клиентские приложения (аналог: Steam)
- Игровые движки (аналог: Unreal)
- Игры
- Браузерная платформа

Финансирование и сотрудничество

- **Разработки.**
Финансирование команды разработчиков порядка 10-20 человек.
- **Внедрение и получение обратной связи.**
Сотрудничество с IT компанией,
разрабатывающей сервисы для внутренней эксплуатации.
- **Продвижение и наращивание клиентской базы.**
Финансирование размещения и развития облачной платформы.
- **Развитие платформы распространения контента IT инфраструктуры.**
Финансирование развития платформы.
Партнёрские отношения для разработки и внедрения корпоративного ПО,
сопровождающего разработку.

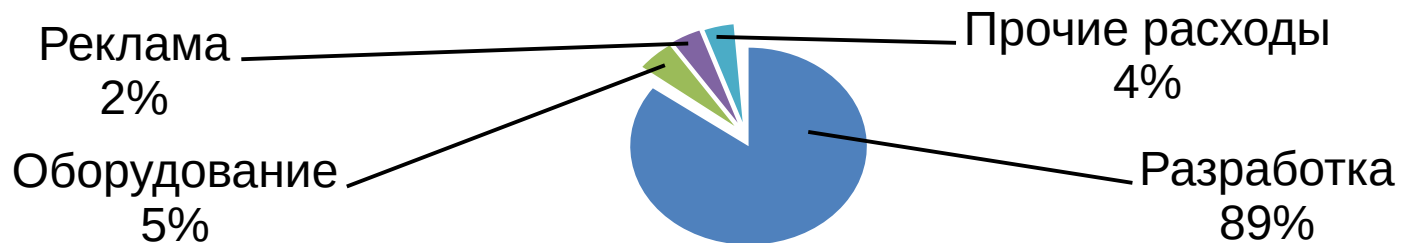
План внедрения (итоги по годам)



Финансовый план

	2022-2023	2023-2024	2024-2025	2025+
Расходы, млн. \$	3.0	5.0	8.0	20.0
Выручка, млн. \$	0	2.0	70.0	550.0

Примерная структура затрат (1-й год) – 3.0 млн. \$



Состояние на сегодняшний день

- Детально проработана концепция языка программирования HardCode
- Подготовлена детальная техническая презентация длительностью не менее 2 часов (не учитывая времени на вопросы и ответы)
- Имеется референтная реализация инновационного многопоточного неблокирующего Garbage Collector'a, составляющего определяющую роль в модели памяти
- Имеется референтная реализация (старой версии) системы версионного хранения кода и разработки на лету, проверенная в качестве средства разработки ПО
- Подробно прописана концепция сервисной платформы Service as a Service
- Разработана концепция платформы распространения контента IT инфраструктуры
- Сформулирована модель финансирования разработки и внедрения проекта, расписан поэтапный план и необходимый кадровый состав
- Определены принципы монетизации продукта с учётом опыта крупных успешных IT решений
- Проведён экспертный анализ технических деталей проекта

Зачем это инвесторам и софтверной компании

- Новая мощная технология разработки ПО.
Возможность создания бизнеса, лидирующего на глобальном рынке, используя новые технологические подходы (язык HardCode и платформа Service as Service)
 - в разы более эффективная экономика разработки сервисов ✓
 - снижение затрат на эксплуатацию сервисов на порядки ✓
- Надёжность сервисов, быстрая обработка обращений клиентов
 - лояльность клиентов ✓
- Оптимизация разработки, отладки и тестирования
 - быстрый старт продуктов ✓
 - эффективная поддержка и развитие продуктов ✓
- Продукт в виде облачной сервисной платформы
 - ROI по самым пессимистическим оценкам не менее 300-400 % ✓
 - новые партнёрские отношения (без боязни lock-in) ✓
- HR-бренд
 - интерес высококвалифицированных разработчиков ✓
 - пониженная текучка кадров ✓

Решаемые задачи

- Язык программирования общего назначения **HardCode**
 - Задаём новый уровень безопасности, выразительности и производительности кода
Программы не падают, выполняются быстрее, код в разы короче
 - Меняем парадигму разработки, отладки и оптимизации ПО
Избавляемся от рутины, ускоряем процесс разработки в разы
 - Реализуем sandboxing, не присутствующий в рабочем виде ни в одном современном языке программирования
- Платформа разработки и размещения сервисов в парадигме **Service as a Service**
 - Принципиально меняем процесс разработки и эксплуатации сервисов
Деплой в 2 клика
 - Выкидываем ненужную деятельность и профессии из обихода разработчиков сервисов
Админы и DevOps'ы не нужны
 - Делаем жизнеспособными привлекательнейшие концепции
Function as a Service, Data as a Code, Аутсорсинг Вычислений
 - Гарантируем консистентность сетевого API, снижаем сетевые ошибки до потенциального минимума
Обеспечиваем максимальный SLA «из коробки»

Решаемые задачи

- NextGen IDE: Платформа распространения контента IT инфраструктуры
 - IDE для разработки сервисов
Разрабатываем и эксплуатируем сервисы единым инструментарием
 - Версионное хранилище кода
Перетягиваем на себя клиентскую базу — разработчиков
 - Обширный корпоративный инструментарий
Обслуживаем процессы разработки и сопровождения ПО в корпоративной среде
 - Платформа стриминга и обучения
Формируем культуру разработки в нашей экосистеме