

Петр Коробейников

Founding Engineer & Architect - Go, Kotlin, Postgres, Cloud, DBaaS, Kubernetes

Желаемая позиция: Principal Engineer

Желаемый стек: Kotlin, Kubernetes, Postgres, Kafka

LinkedIn: [linkedin.com/in/petr-korobeinikov](https://www.linkedin.com/in/petr-korobeinikov)

ПРОФИЛЬ

Principal-инженер с опытом проектирования и запуска отказоустойчивых облачных сервисов: DBaaS для Postgres и Redis, Managed RabbitMQ, voice/real-time-системы на Go и gRPC, событийные интеграции на Kafka, оркестрация бизнес-процессов на Temporal. Веду архитектурные ревью, формирую инженерные стандарты и DevEx-практики на уровне направления, выступаю ментором во внутренней менторской программе. Адаптирую AI-инструменты к процессу разработки команды. Веду публичный блог principal-engineering.ru, YouTube- и Telegram-каналы об инженерных практиках.

ОПЫТ РАБОТЫ

MTC, FinTech Founding Engineer & Architect

октябрь 2024 - июль 2026

Стек: Kotlin, Spring Boot, Go, GraphQL, gRPC, protobuf, Kafka, PostgreSQL, Kubernetes, Kustomize, OpenTelemetry, Terraform / OpenTofu, k6

За 2 месяца спроектировал и реализовал бэкэнд-часть раздела мобильного приложения «Мой МТС».

За полгода работы превзошли все самые смелые финансовые ожидания. Конкретные цифры под строжайшим NDA.

- Заложил архитектуру GraphQL-бэкенда для BDUI (Backend-Driven UI).
- Адаптировал AI-инструменты к процессу разработки: настроил кастомные skills и subagents в AI-агенте для разработки, подключил MCP-серверы к внутренним системам команды, сформулировал гайдлайны использования AI и шаблоны промптов под домен - типовые задачи (генерация тестов, скаффолдинг сервисов, первичное code review) ушли в AI-цикл.
- Разработал шаблон сервиса: GraphQL, OpenAPI, кодогенерация по спецификациям.
- Реализовал интеграцию с финансовым партнёром через Kafka - строгая очерёдность сообщений, protobuf, идемпотентный consumer, dead-letter queue.
- Адаптировал Spring Boot + Kotlin в нескольких микросервисах: за счёт хорошо отлаженных компонентов и лаконичности Kotlin время разработки и отладки новых сервисов сократилось с 1-2 недель до нескольких дней.
- Настроил observability на OpenTelemetry: распределённые трейсы, структурированные логи, метрики, алерты.
- Развернул инфраструктуру через Terraform / OpenTofu.
- Построил воспроизводимое локальное окружение разработчика на colima / lima.
- Подготовил технический онбординг - время адаптации новых разработчиков сократилось в 5 раз.
- Провёл менторинг команды по инфраструктуре (Terraform) и нагрузочному тестированию (k6).
- Вёл 8 менти в рамках внутренней менторской программы.

MTC, VoiceTech Founding Engineer & Architect

январь 2024 - сентябрь 2024

Стек: Go, gRPC, WebRTC, Opus (аудио), Temporal, Kubernetes, Helm, Argo CD, OpenTelemetry, D2

Команда продукта голосовой экосистемы: real-time обработка телефонных звонков, AI-ассистент прямо в звонке, последующая обработка записей.

- Спроектировал и построил кору голосовой экосистемы и адаптер петли: Go-каналы, gRPC-стримы, неблокирующая работа с каналами, try-send-семантика для real-time обработки.
- Запустил голосового AI-помощника «Марвин» внутри активного телефонного звонка - пользователь обращается по имени, получает ответ от модели голосом; обработка на стороне оператора (работает с любого телефона, включая кнопочный и бабушкин дисковый). Показан широкой аудитории на конференции platforma.mts.ru.
- Построил браузерные инструменты разработчика голосовой модели на Opus + WebRTC - новые члены команды вошли в разработку и отладку навыков без выдачи реальных телефонных аппаратов и служебных тарифов.

- Реализовал сбор саммари телефонных разговоров: обработка записей, prompt-engineering для модели суммаризации, Temporal для отказоустойчивости, Telegram-бот как канал коммуникации с пользователем.
- Настроил observability на OpenTelemetry: распределённые трейсы, структурированные логи, метрики, алерты.
- Внедрил «диаграммы как код» с d2lang (после оценки PlantUML, C4 as code, Structurizr).
- Построил процесс технического онбординга - время адаптации разработчиков сократилось в 20 раз.
- Закрепил единый DoD (output / outcome).
- Запустил структурированный найм Go-разработчиков, проводил собеседования по выработанному процессу.
- Свёл командное общение в один канал вместо нескольких чатов.
- Настроил воспроизводимое рабочее окружение разработчика.
- Поднял на уровне направления темы инцидент-менеджмента и Security Championship.
- Принят ряд патчей в Open Source: localmodule в [gci](#) - вошёл в [golangci-lint v1.58.0](#); давний PR в [wreulicke/http-timeout](#); поддержка x86 в [asdf-eza](#).
- Вёл двух менти в рамках внутренней менторской программы.

МТС, #CloudMTS Technical Lead

август 2021 - июль 2023, Москва

Стек: Go, Kubernetes, Helm, PostgreSQL, Redis, RabbitMQ, Ansible, Packer, OpenCensus

- Перепроектировал и запустил в продакшн DBaaS for PostgreSQL (Managed Postgres) с отказоустойчивой конфигурацией.
- Спроектировал и реализовал control-plane для запуска новых сервисов DBaaS - сократил время вывода новых продуктов в эксплуатацию.
- На единой платформе реализовал DBaaS for Redis (Managed Redis).
- На той же платформе построил Managed RabbitMQ.
- Внедрил трейсинг (OpenCensus), логгирование и сборку ошибок в проекты команды.
- Настроил воспроизводимое окружение разработчика: 20 минут онбординга - и человек готов к работе.
- Настроил провижининг виртуальных машин в два этапа: сборка шаблонов через HashiCorp Packer и финальная конфигурация VM из шаблона через Ansible.
- Проводил архитектурное ревью внутри команды - влияло на качество решений на этапе проектирования.
- Продвигал единые инженерные подходы и стандарты архитектуры на уровне компании.
- Развивал и внедрял практики Security Championship и Developer Experience внутри команды и направления.

Авито Senior Software Development Engineer

сентябрь 2016 - декабрь 2020, Москва

Стек: Go, Kubernetes, Helm, GraphQL, Consul, Vault, Python, React, Tornado

- Исследовал перевод функционала на FaaS-подход (kubernetesless, fission, knative, nuclio, openfaas). На уровне компании было решено не двигаться в этом направлении.
- Принял непосредственное участие в разработке внутреннего DBaaS: написал GraphQL API Gateway к внутренним сервисам (Consul, Atlas), разработал CLI для администраторов и пользователей. Записал скринкаст по работе с GraphQL для коллег. Участвовал в исследовании использования PV в Kubernetes.
- Автоматизировал сборки сэмплов баз для внутреннего PaaS: работа с Vault, выполнение требований по безопасности, dind для сборки образов, соответствующие service account и role binding в Kubernetes. Механизм сокрытия персональных данных получил высокую оценку PKN в ходе проверки.
- Принял участие в улучшении внутреннего PaaS на основе обратной связи пользователей (React, Go, Python).
- Входил в дежурную ротацию Kubernetes-администрирования - наращивал экспертизу.
- Сервис «саг» - координация распределённых транзакций между микросервисами по паттерну SAGA. Участвовал во всём цикле - от деплоя в Kubernetes до тестов, кода и алертинга. 100 % покрытие тестами (Tornado).
- Разработал ряд инфраструктурных микросервисов на Go с высоким покрытием тестами - работа с очередями и шиной данных, поисковые механизмы (наиболее критичный - доставка объявлений в поиск).

- Разработал новый инструмент сэмплирования БД с YAML-конфигом по строгой схеме, тестами и примерами; работоспособность подтверждается регулярными запусками в CI.

РДВ-медиа Senior Software Development Engineer

август 2010 - октябрь 2014, Москва

- Запуск платной модели для соискателей (май 2014, командный результат, нёс персональную ответственность).
- Заложил сильную основу для проекта «Профессиональные сообщества» pro.rabota.ru (Symfony2).
- Внедрил Composer для управления зависимостями и поднял внутреннее зеркало используемых пакетов.
- Запустил с нуля iOS- и Android-приложения Rabota.ru, передал разработку команде.
- Воспроизводимое окружение разработчика через SaltStack и Vagrant; перенёс часть данных с NFS на Riak; ввёл RabbitMQ для тяжёлых задач.

ПУБЛИЧНАЯ АКТИВНОСТЬ

- Блог об инженерных практиках: principal-engineering.ru
- YouTube-канал: [@principal-engineering](https://www.youtube.com/@principal-engineering)
- Telegram-канал: [@principalengineering](https://www.telegram.com/@principalengineering)