

Deckhouse Development Portal – новый стандарт управления разработкой

Эффективные команды и прозрачные процессы

deckhouse

Спикер

Максим Пшеничников

Менеджер продукта
Deckhouse Development Portal

- 20 лет опыта в разработке
- Solution-архитектор
- Platform-engineering-энтузиаст



Оглавление

- 01 О «Фланте» и экосистеме Deckhouse
- 02 Ценность платформ разработки
- 03 Как устроена платформа разработки Deckhouse
- 04 Возможности контроля для руководителя
- 05 Возможности для разработчиков
- 06 Кастомизируемость под специфику организации

deckhouse

СФЛАНТ

Синергия опыта вендора, интегратора,
сервисной и консалтинговой компании

deckhouse

Deckhouse — продуктивное подразделение, разработчик продуктов для построения надёжной enterprise-инфраструктуры

DaaS

DaaS — комплексное DevOps-сопровождение инфраструктуры в режиме 24/7 силами выделенной DevOps-команды

Express 42

«Экспресс 42» — DevOps-консалтинг. От анализа узких мест в ИТ-процессах до создания роадмапа изменения ИТ для реализации цифровой трансформации

О вендоре Deckhouse

deckhouse

18+

лет опыта
в Open Source

С 2017

года используем
Kubernetes в production

№1

контрибьютор в проекты
CNCF из России

550+

сотрудников

>260

компаний-пользователей

№1

в рейтинге вендоров систем
мониторинга и управления ИТ-
инфраструктурой*



Реестр
российского ПО



Лицензии и сертификаты
ФСТЭК России



АРПП «Отечественный
софт»

* Рейтинг «Российский рынок систем мониторинга и управления ИТ-инфраструктурой», TAdviser, 2026

**Что такое
платформа
разработки?**

Платформа разработки – это больше чем Git + пайплайны

Не заменяет Git + пайплайны,
а включает как один из строительных кубиков

Живёт вокруг исходного кода

Git + CI/CD-платформа

Репозитории

Merge Requests

CI/CD-пайплайны

Container registry

Quality gates

Тесты

Живёт вокруг продукта и эксплуатации

Внутренняя платформа разработки

Сервисный каталог

Зависимости

Команды

Golden paths: типовой путь создания и релиза сервиса

Сборка и доставка

Временные окружения

DORA-метрики

SDLC-метрики

Унификация стека разработки

Общие политики ИБ

Самообслуживание

Заказ ресурсов инфраструктуры

Детальная стоимость эксплуатации

Наблюдаемость

Здоровье приложений

Принципиально другая ценность

Ускорение целостного процесса продуктовой разработки

	Живёт вокруг исходного кода		Живёт вокруг продукта и эксплуатации
	 GitLab Community	 GitLab Enterprise	 Development Portal
Шаблоны приложений и пайплайнов	✓	✓	✓
Временные окружения	✗	✓	✓
DORA- и SDLC-метрики	✗	✓	✓
Общие политики ИБ	✗	✓	✓
Сервисный каталог, зависимости микросервисов	✗	✗	✓
Самообслуживание, заказ ресурсов инфраструктуры	✗	✗	✓
Унификация стека разработки	✗	✗	✓
Наблюдаемость, здоровье приложений	✗	✗	✓
Подсчет стоимости эксплуатации	✗	✗	✓
Единое окно, снижение когнитивной нагрузки	✗	✗	✓

Международный опыт

На рынке десятки
продуктов и фреймворков

Не только on-prem,
но и облачные managed-решения

Платформы разработки — это уже
не эксперимент, а подтверждённый
способ ускорить и удешевить разработку

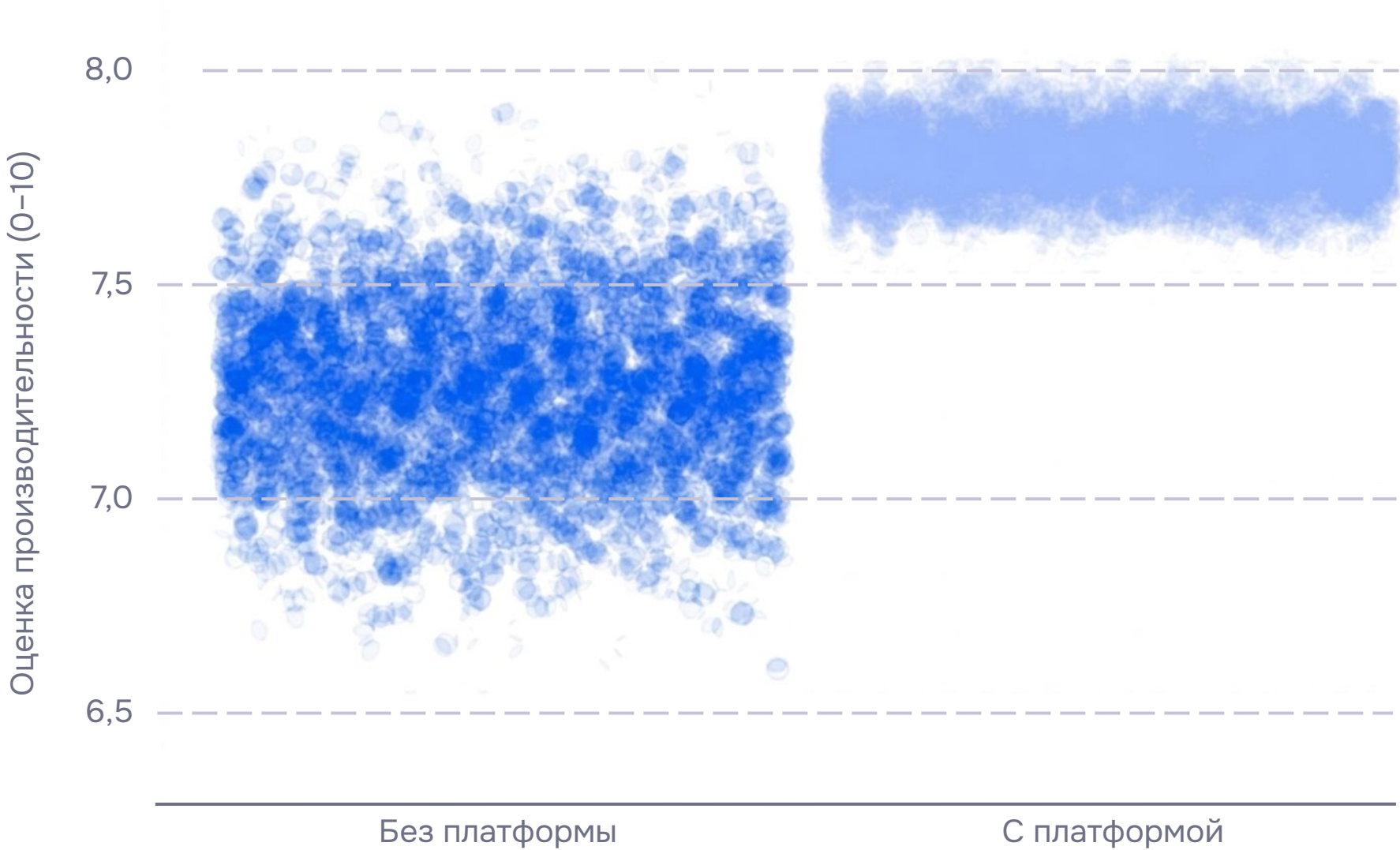


и многие другие...

Внутренние платформы разработки

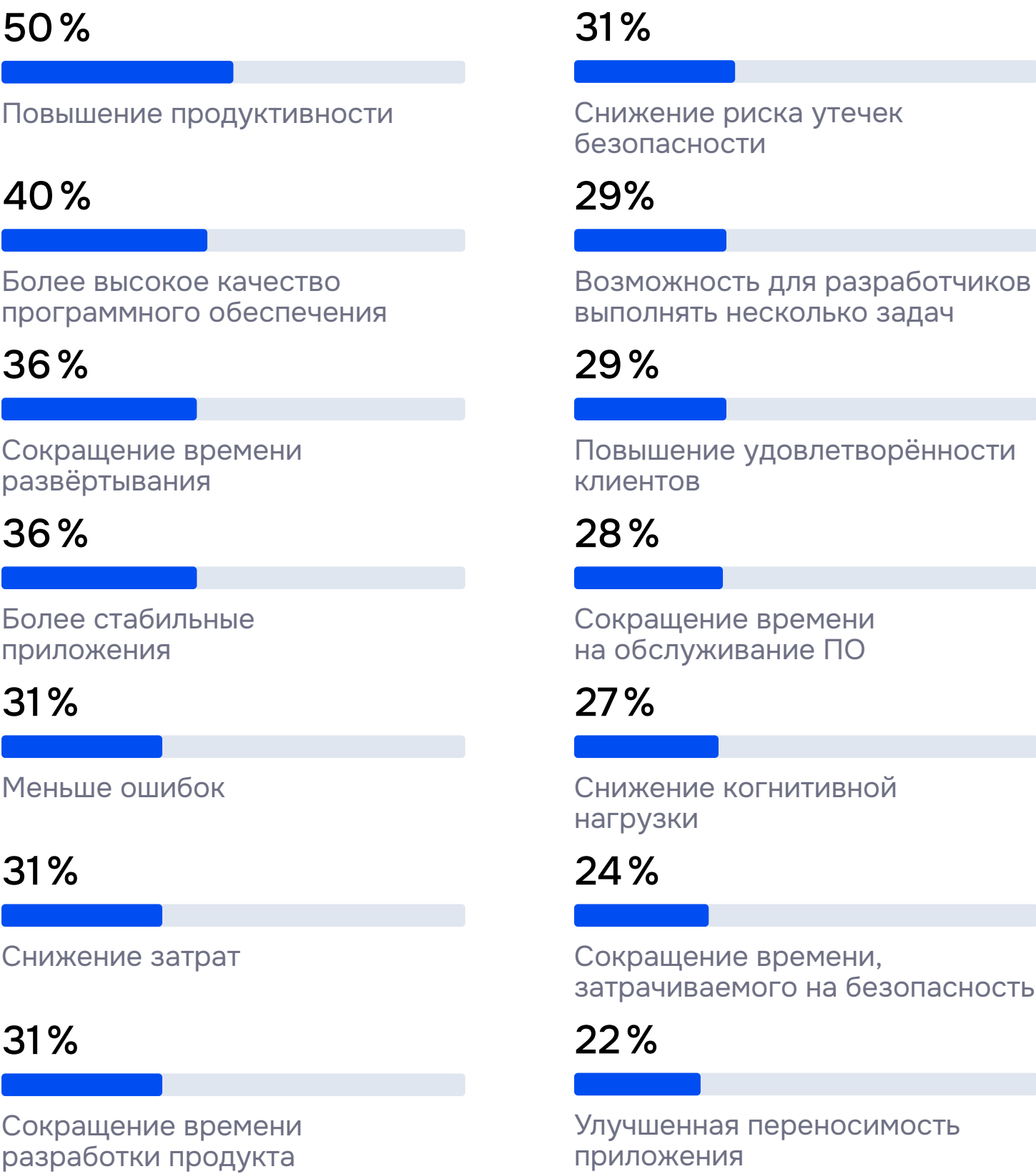
Организации с платформой автоматически вырываются в топ по всем показателям качества разработки и эксплуатации

DORA Accelerate State of DevOps 2024



Ключевые преимущества платформенной инженерии для разработчиков

[State of DevOps Russia 2025](#)

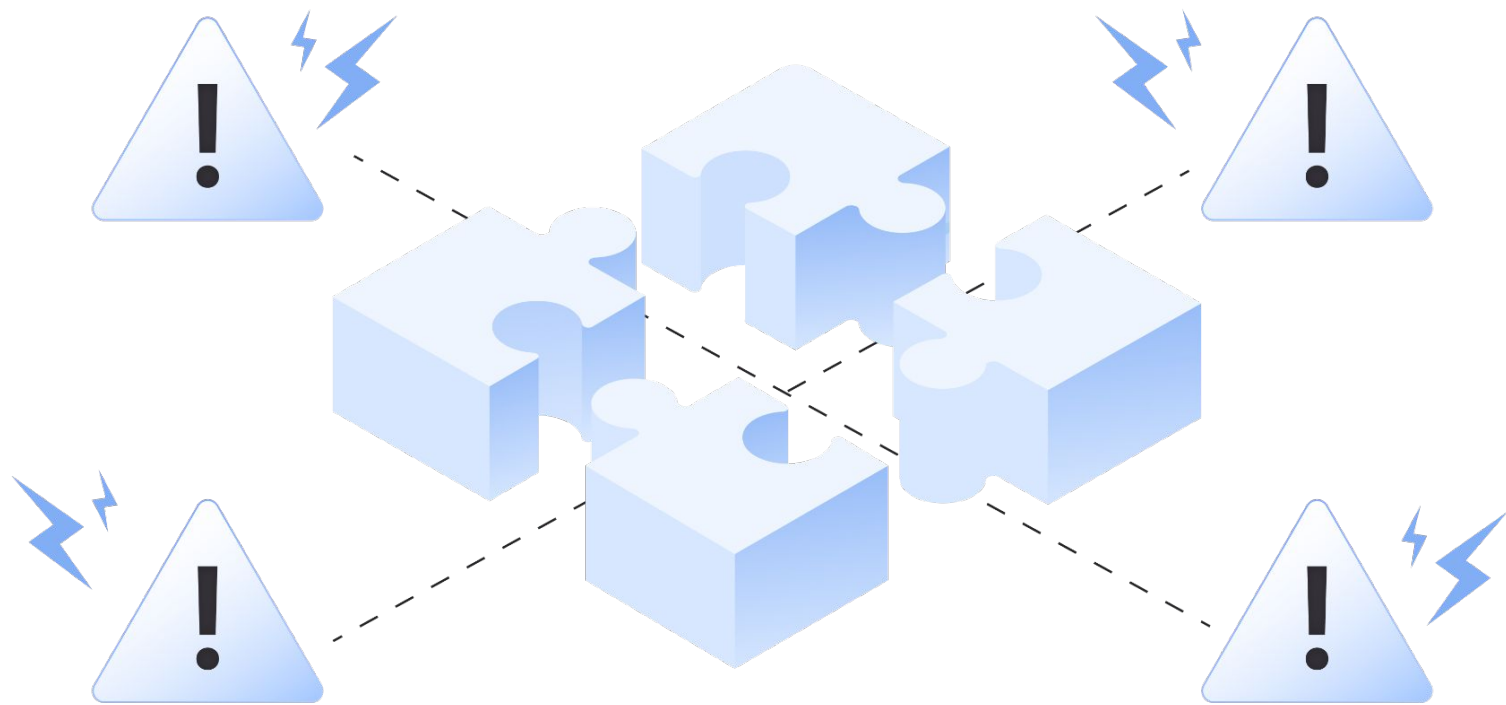


Результат создают не инструменты, а люди и процессы



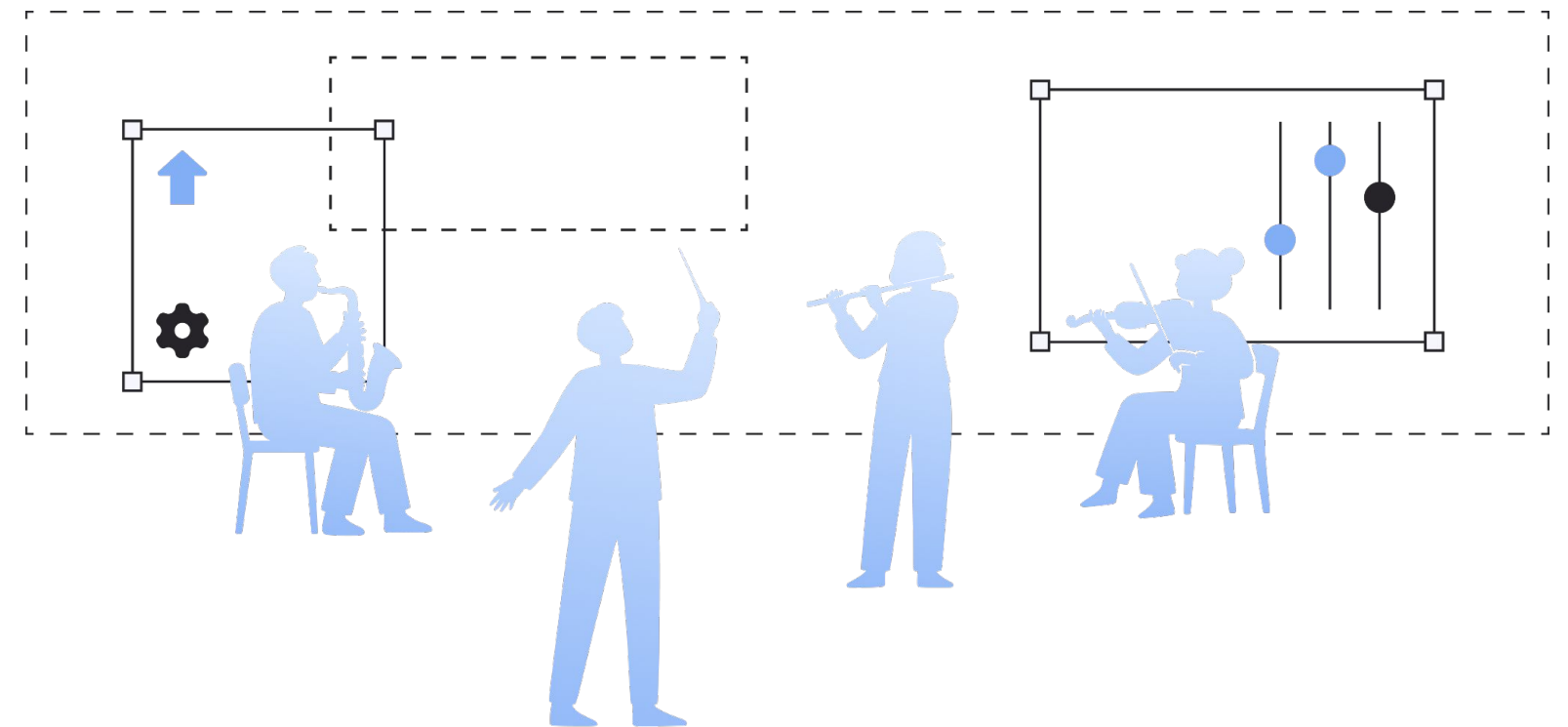
Продуктивность организации теряется на стыках инструментов

Проблемы возникают при передаче данных,
переключении контекста и ручной синхронизации



Непрерывный опыт в реальных сценариях важнее списка функций

Enterprise-инфраструктура – это не просто набор
«инструментов», а слаженно играющий «оркестр»



**Комплексное решение, где разработчики
получают готовые пути доставки, а организация
сохраняет контроль над бизнес-показателями,
безопасностью и эксплуатацией**



**Deckhouse
Development Platform**



***deckhouse*
Development Portal**

Преднастроенные интеграции с инструментами разработки

Быстрый старт с множеством доступных «из коробки» адаптеров синхронизации.
Автоматическое обнаружение сущностей в вашем ландшафте.

Код и Доставка

GitLab

GitHub

Bitbucket

Jenkins

Harbor

Sonatype Nexus

Docker Registry

Deckhouse Galleon

Ресурсы и Данные

Любой Kubernetes

Любое Облако

Deckhouse Platform

Эксплуатация

Grafana

Deckhouse Stronghold

Prometheus

HashiCorp Vault

Безопасность и Качество

CodeScoring

SonarQube

Svacer

DefectDojo

Инфраструктурные и Учётные системы

KeyCloak

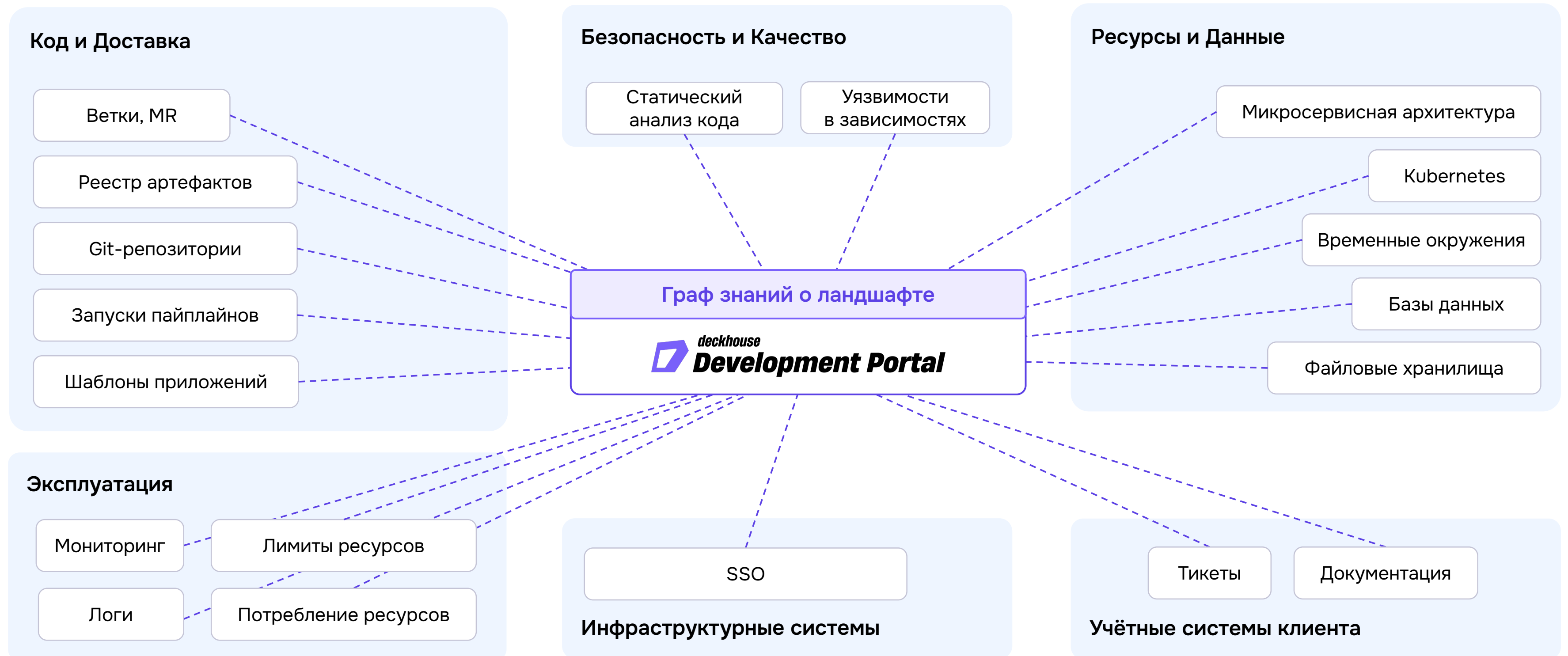
Dex

Jira

[Любая система с API](#)

Цифровая копия вашего ландшафта

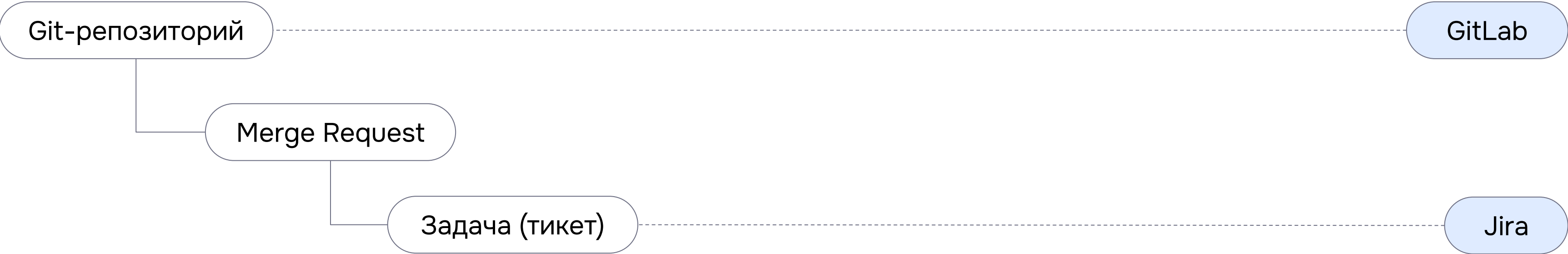
Все данные про разработку и эксплуатацию собираются в одну постоянно синхронизируемую подробную базу данных

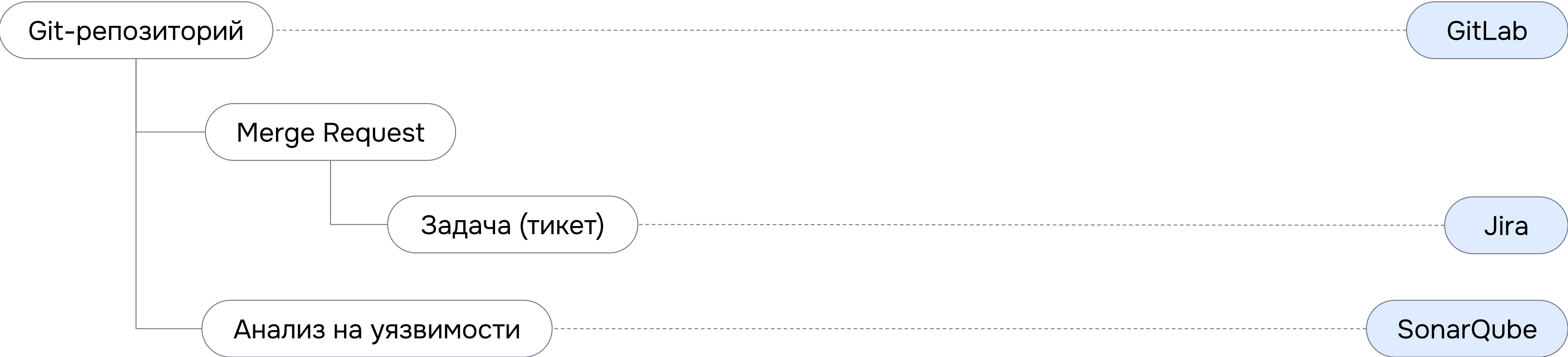


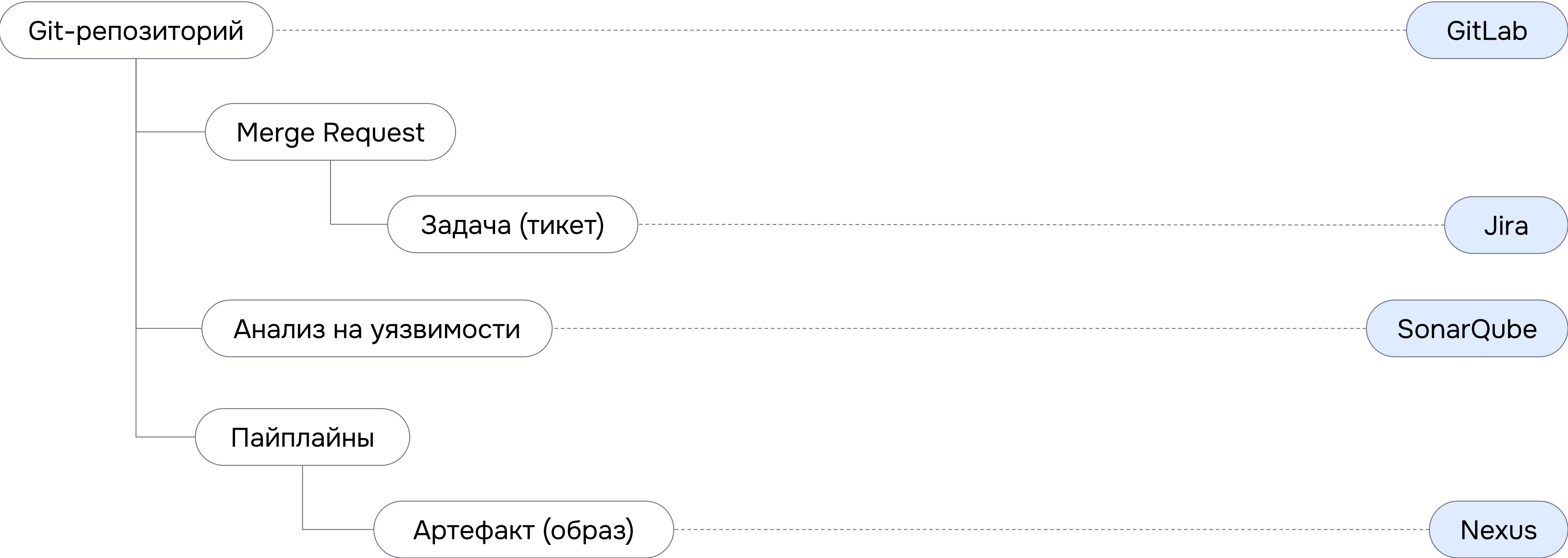
Git-репозиторий

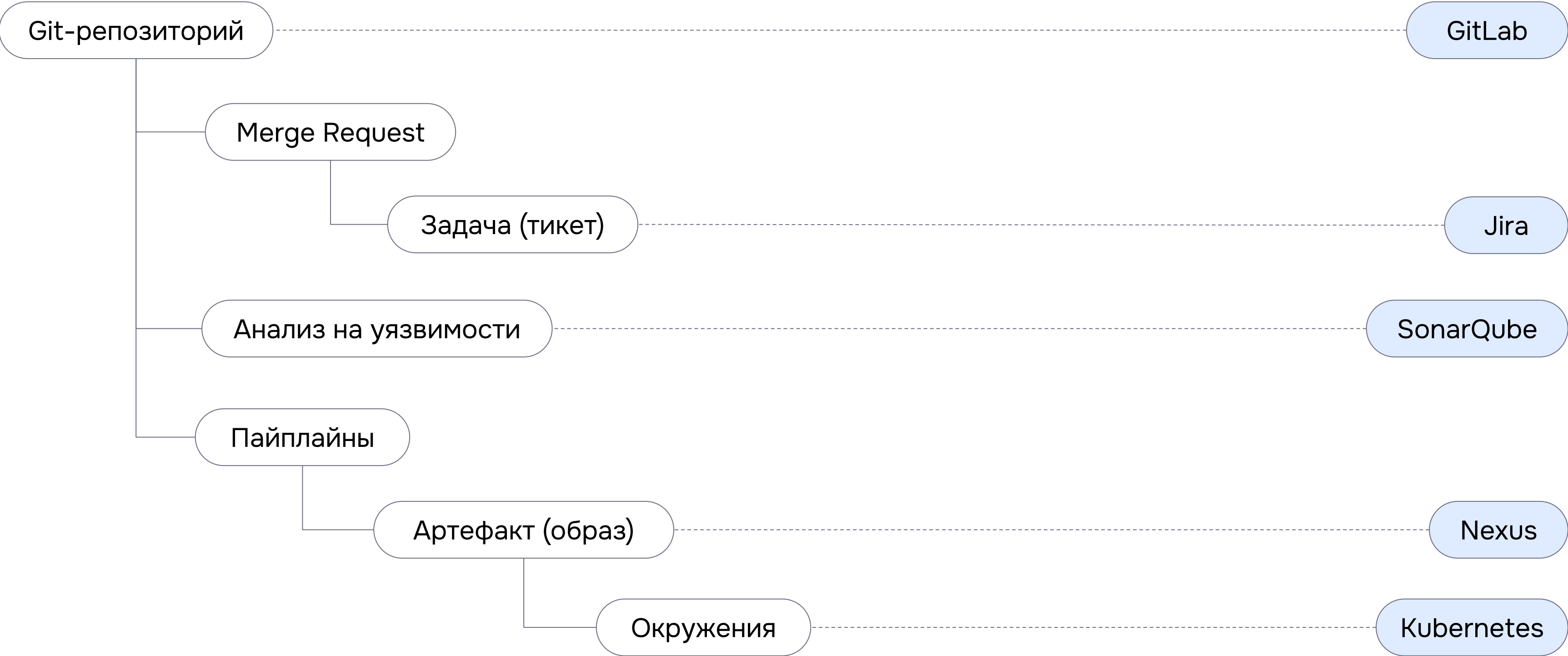


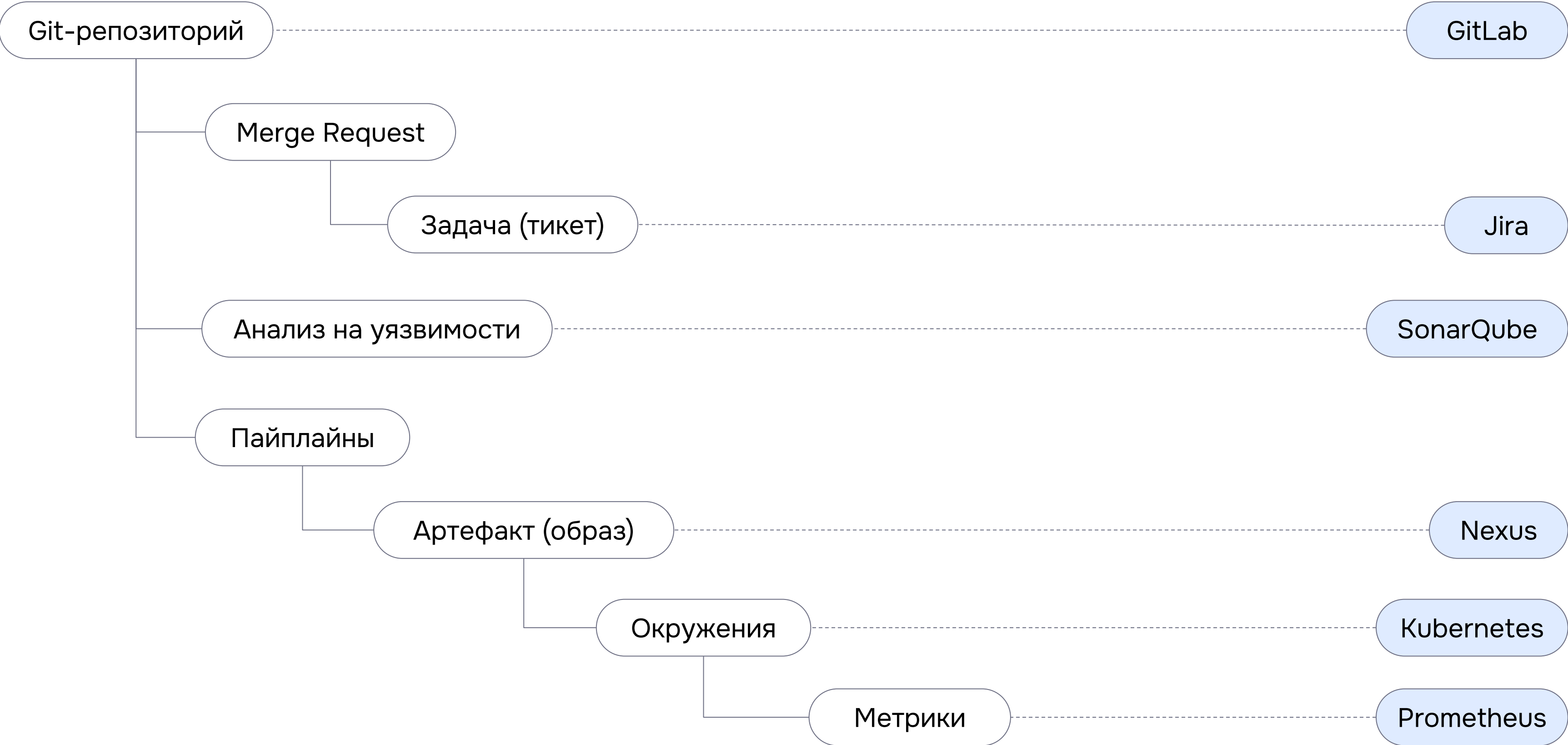
GitLab











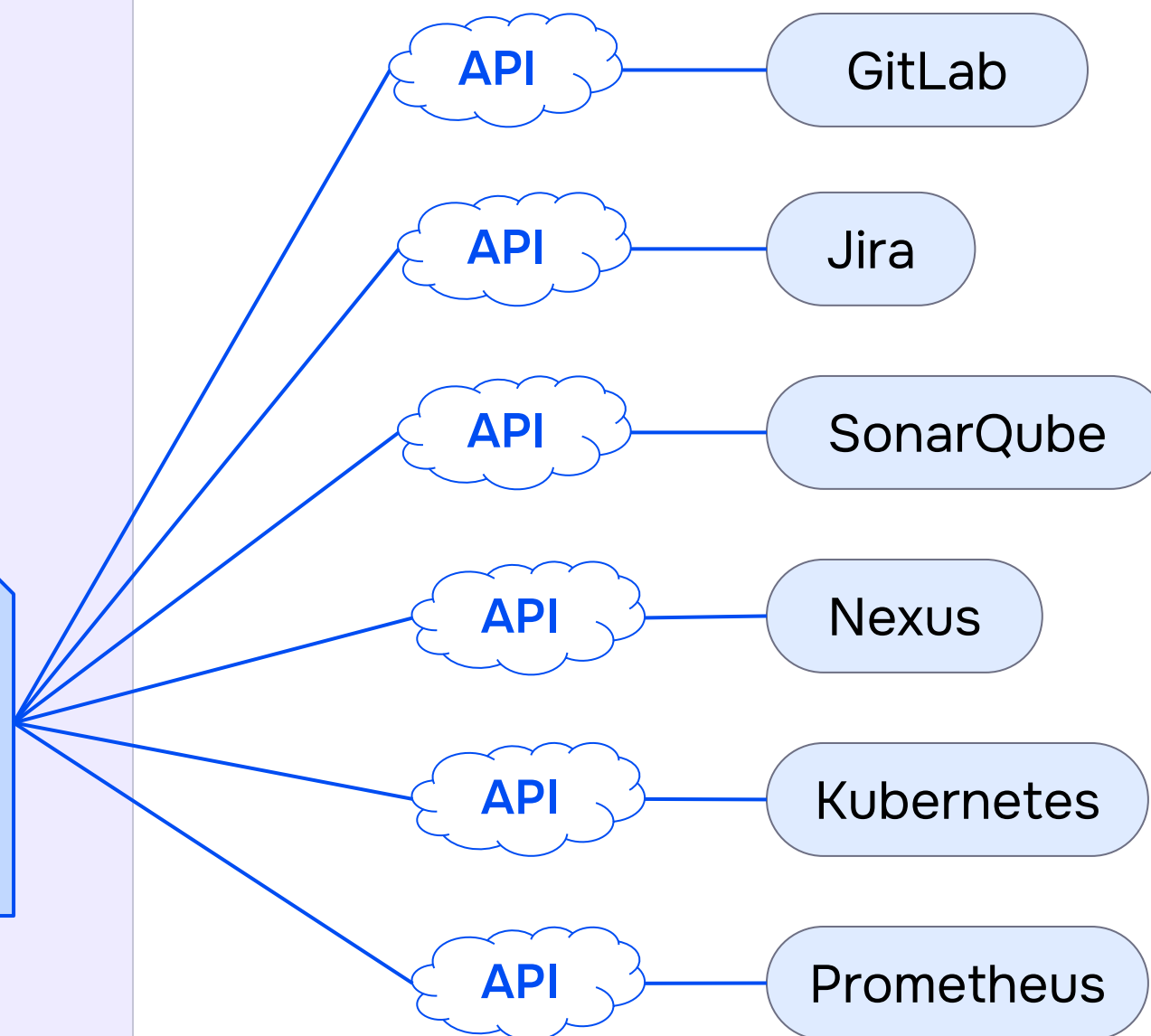
Граф знаний о ландшафте



Граф знаний о ландшафте



No-code-
интеграции



Граф знаний о ландшафте

Git-репозиторий

Merge Request

Задача (тикет)

Анализ на уязвимости

Пайплайны

Артефакт (образ)

Окружение

Метрики

No-code-
интеграции

API

GitLab

BitBucket

API

Jira

Kaiten

API

SonarQube

DefectDojo

API

Nexus

Docker Registry

API

Kubernetes

VM

API

Prometheus

New Relic

API

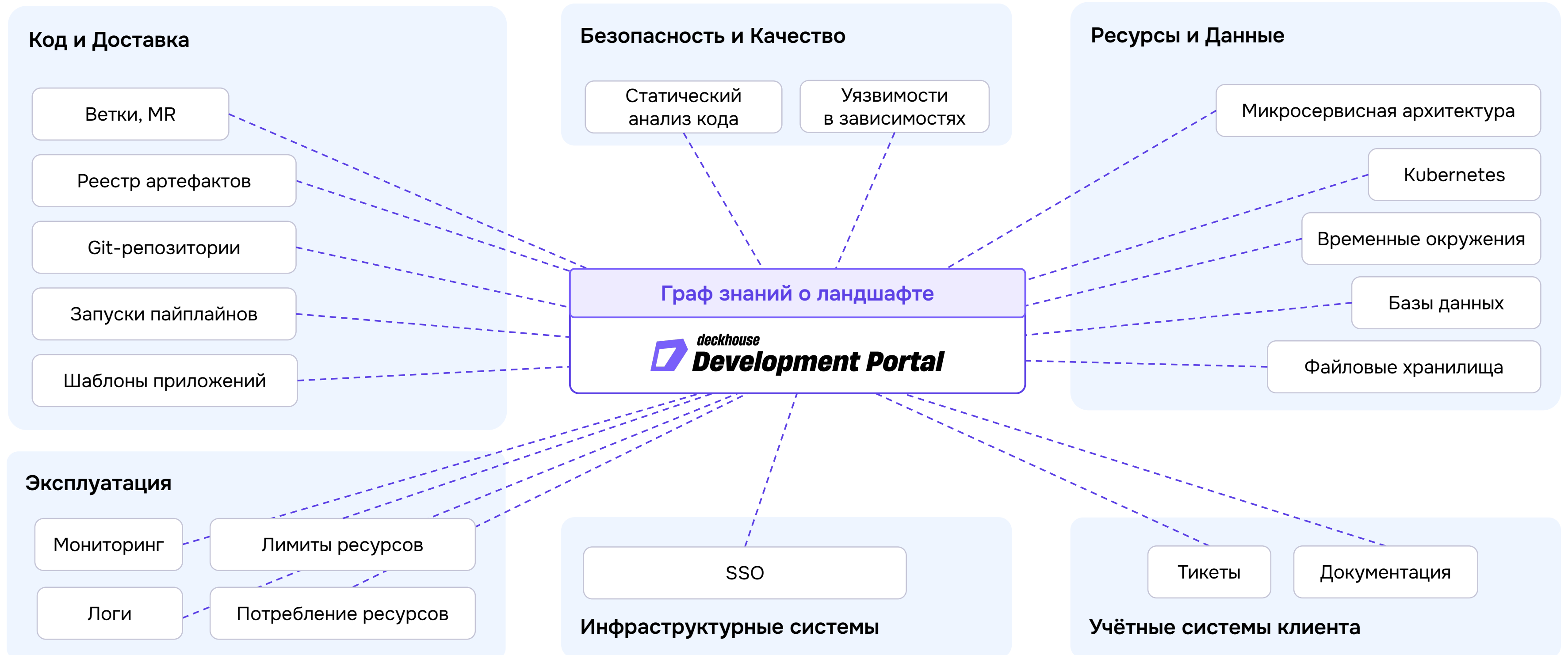
Инфраструктурная система

API

Любая учётная система

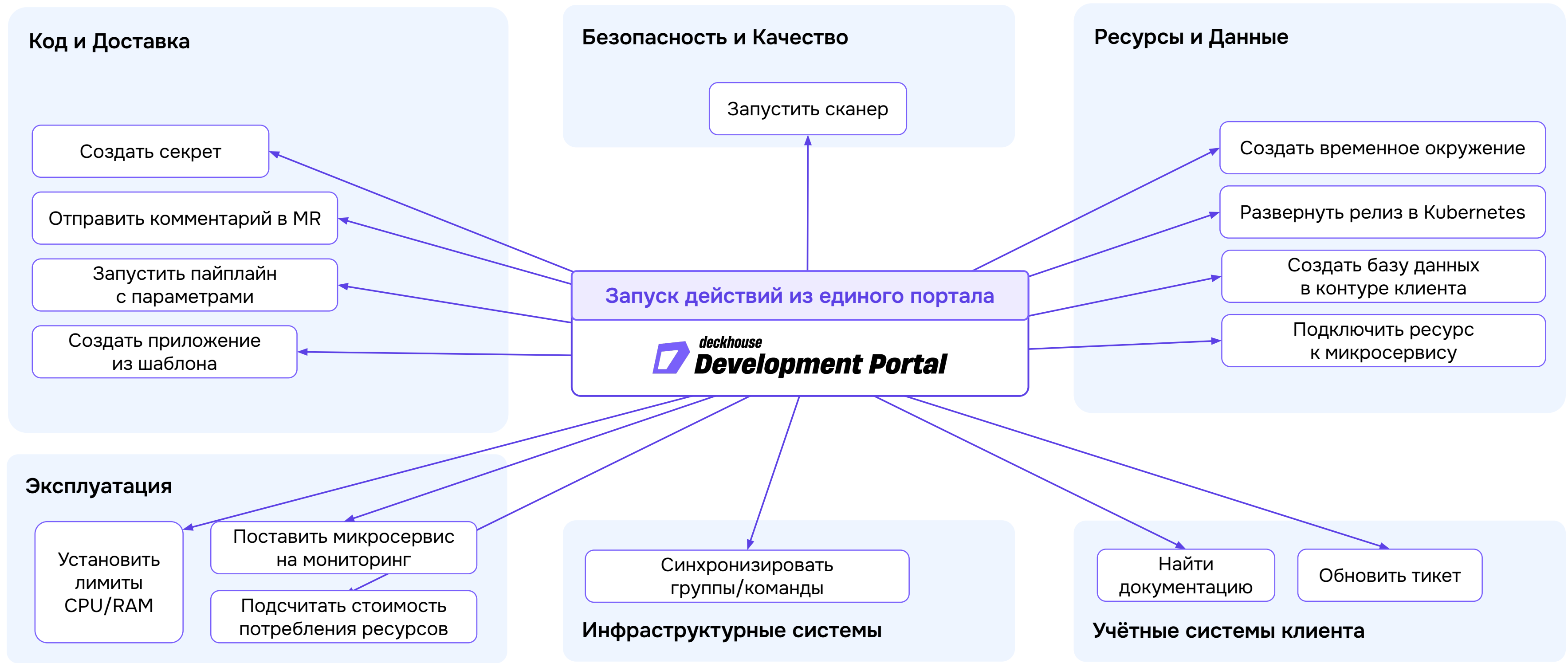
Цифровая копия вашего ландшафта

Все данные про разработку и эксплуатацию собираются в одну постоянно синхронизируемую подробную базу данных



Действия самообслуживания

Возможность разработчикам самостоятельно выполнять повторяющиеся действия без привлечения команды эксплуатации



Запустить действие



Запустить действие



Запустить действие

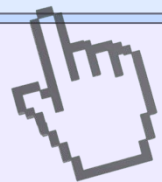


Параметр:

Опция:

Далее

Запустить действие



Параметр:

Опция:

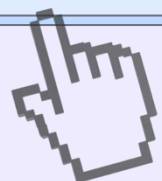
Далее

No-code-интеграция

API

Любая
инфраструктурная
система

Запустить действие



Контроль
доступа

Параметр:

Опция:

Далее

Многошаговый
процесс

No-code-
интеграция

No-code-
интеграция

No-code-
интеграция

API

Инфраструктурная
система А

API

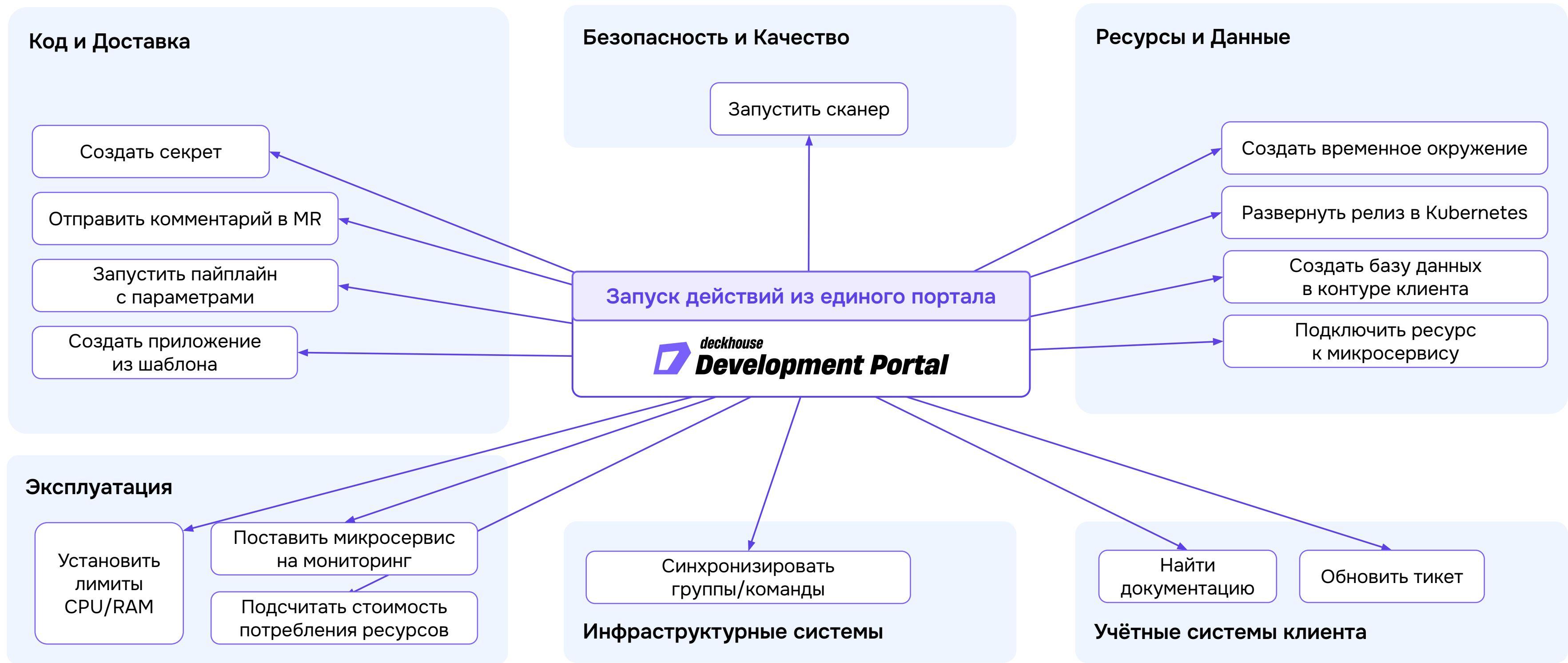
Инфраструктурная
система В

API

Инфраструктурная
система С

Действия самообслуживания

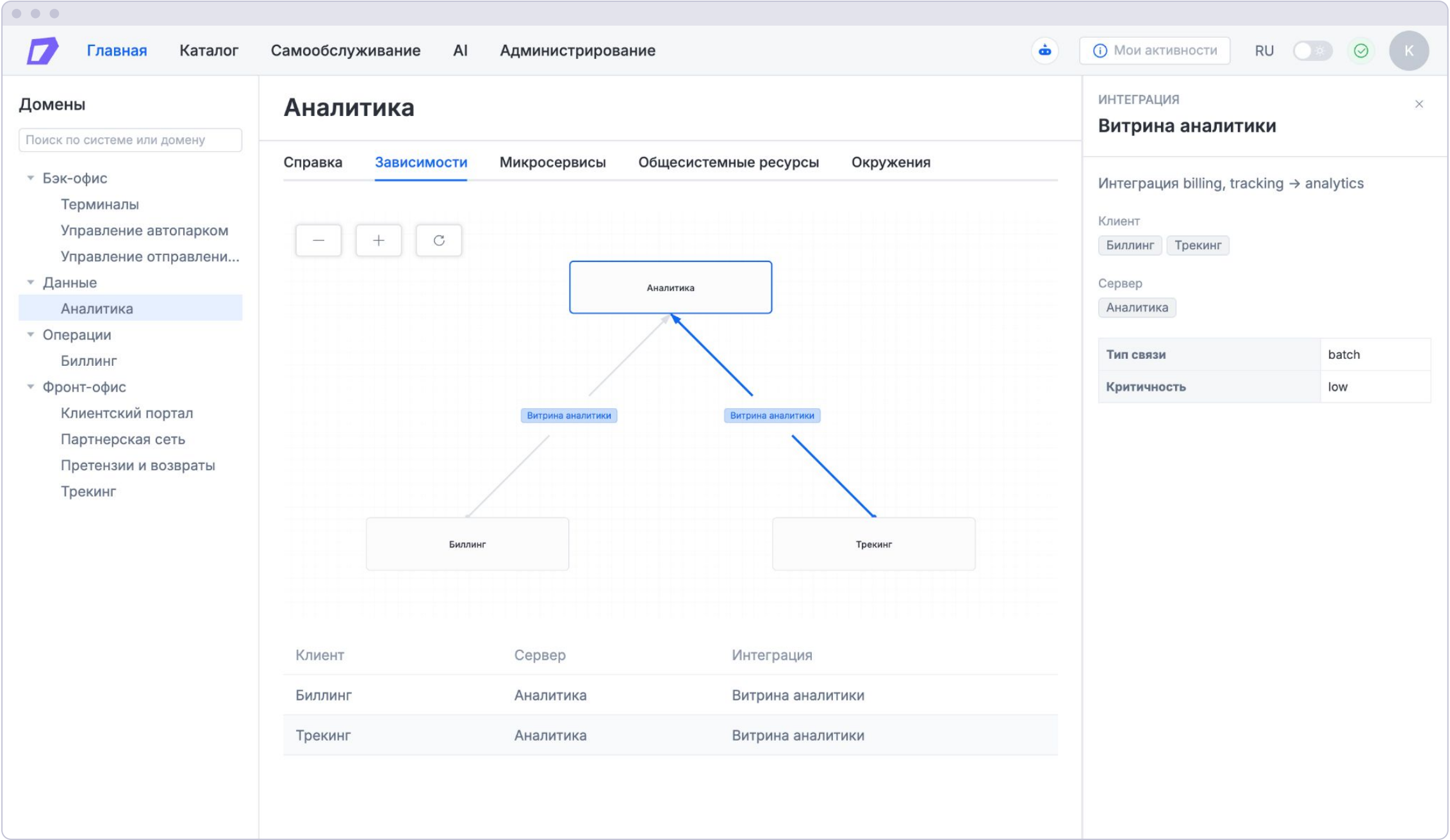
Возможность разработчикам самостоятельно выполнять повторяющиеся действия без привлечения команды эксплуатации



Продуктивный онбординг с первого дня

- Вижу все свои материалы
- Вижу реальную инфраструктуру

- ✓ Как выглядит архитектура
- ✓ Зависимости систем
- ✓ Команда (к кому обратиться)
- ✓ Тикеты
- ✓ Документация платформы



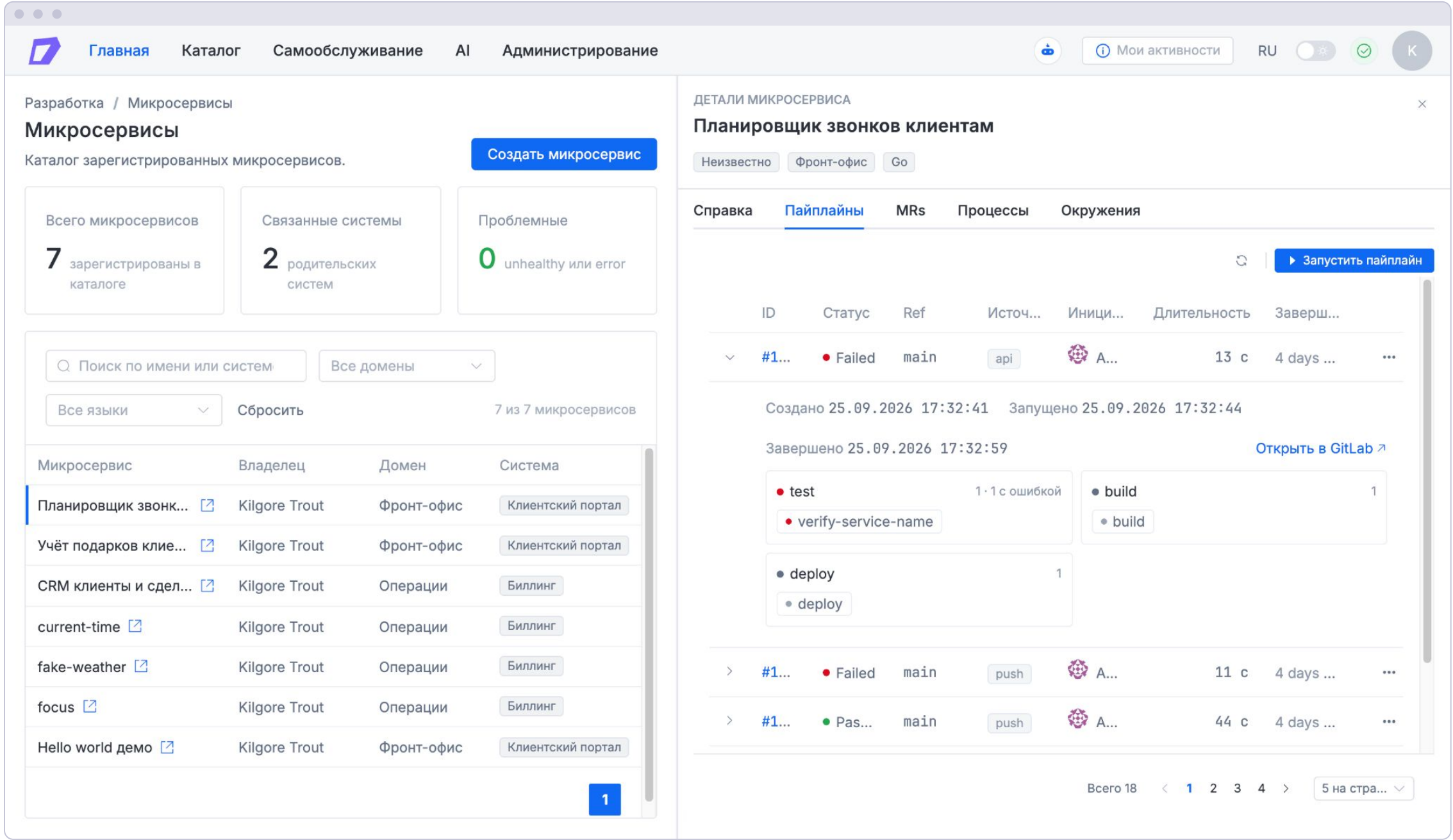
Продуктивный онбординг с первого дня

Вижу все свои материалы

Вижу реальную инфраструктуру

- ✓ Репозитории
- ✓ Композиции микросервисов
- ✓ Базы данных
- ✓ Сборка и доставка
- ✓ Заказ ресурсов (самообслуживание)
- ✓ Временные окружения
- ✓ Здоровье сервисов
- ✓ Скейлинг и стоимость инфраструктуры

С первого дня могу деплоить и чинить



Сокращение Time to Market до 10 раз



Быстрый старт новых продуктов и команд

Шаблоны, готовые интеграции и базовая инфраструктура предоставляются платформой как продукт



Меньше ручных шагов в релизном цикле

Окружения, конфигурации, доступы и деплой — через самообслуживание и автоматизацию



Снижение рисков и длительности инцидентов

Стандарты и наблюдаемость «по умолчанию» ускоряют поиск контекста и владельцев, упрощают обновления



Меньше возвратов на доработку

Встроенные проверки, политики и стандартные пайплайны снижают количество ошибок и повторной работы



Снижение стоимости изменений через «эталонные пути»

Типовые решения вместо уникальных сборок в каждой команде — меньше переделок и быстрее поставка

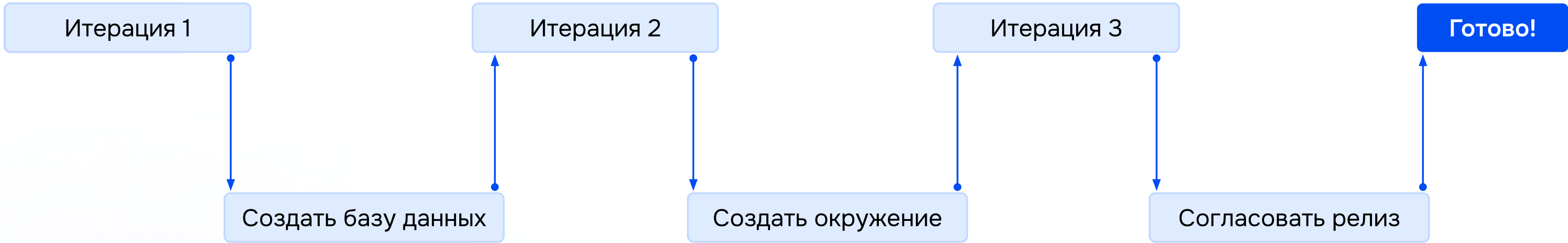


Меньше ожидания платформенной команды

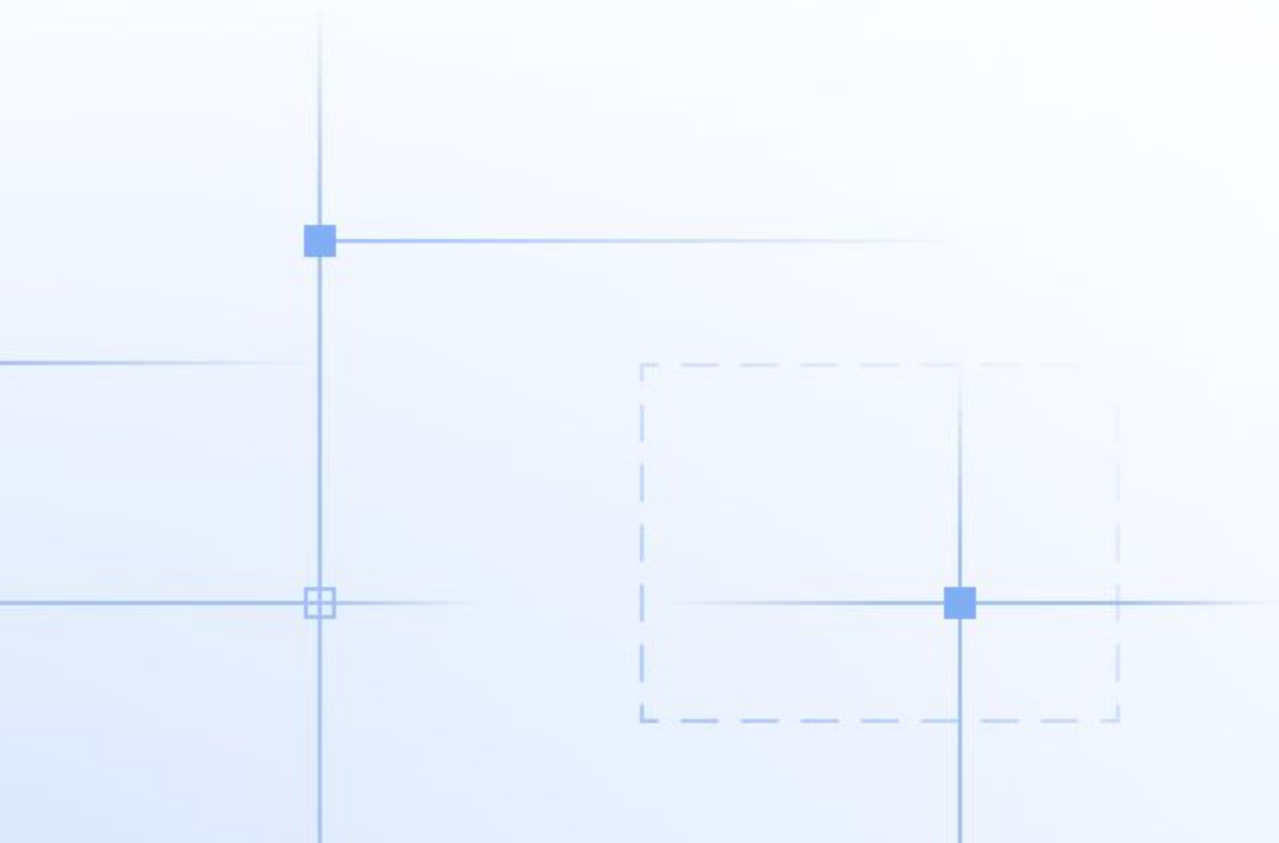
Стандарты и типовые практики доступны командам разработки «по кнопке», без очередей и ручной поддержки

Потери продуктивности при переключении контекста

Команда разработки



Команда эксплуатации

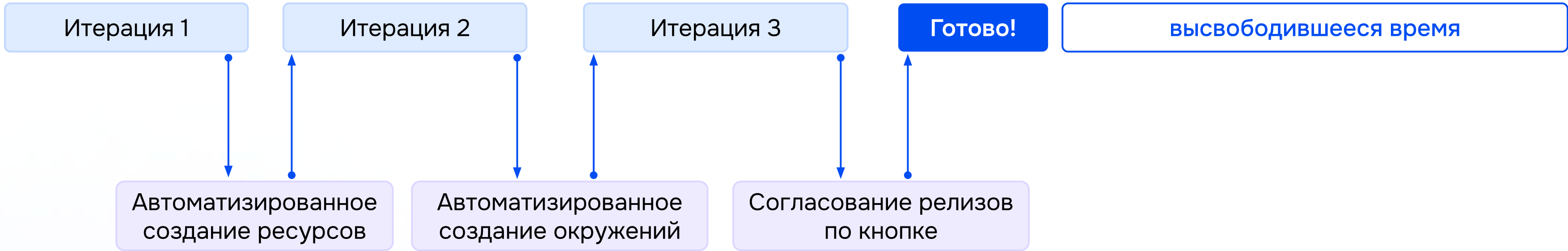


Получите чекап SDLC

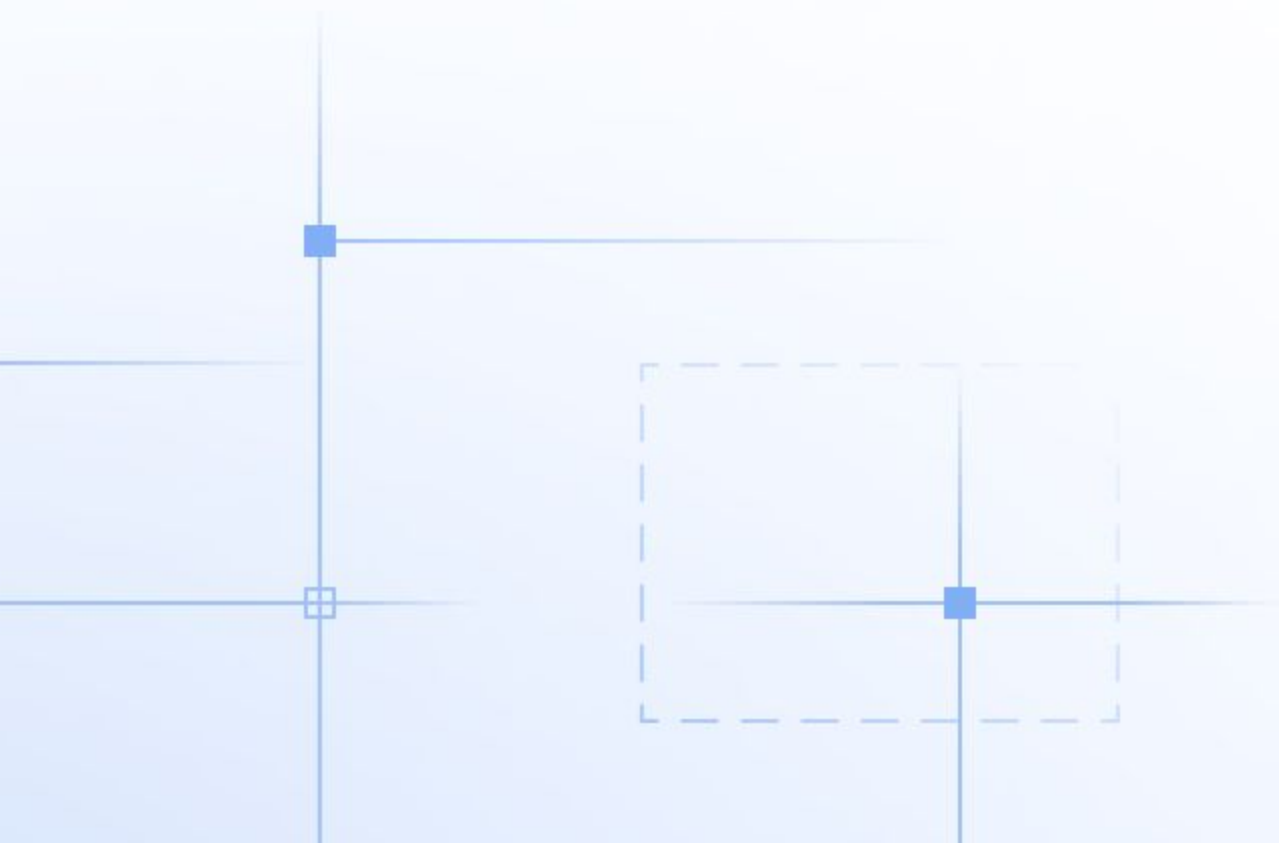


Продуктивность без разрыва контекста

Команда разработки



Платформа разработки



Получите чекап SDLC



Продуктивность всех процессов разработки



Автономность команд

Разработчики меньше зависят от загруженности команд эксплуатации или длительных согласований заявок вручную.

Типовые операции доступны «по кнопке» без привлечения профильных специалистов.



Ускорение разработки

Разработчик может самостоятельно создавать микросервисы, развёртывать их в окружениях и заказывать необходимые инфраструктурные ресурсы.

Портал предоставляет единую точку входа к инженерным инструментам, документации и информации о сервисах.



Снижение риска ошибок

Автоматизация типовых операций снижает количество человеческих ошибок и риск некорректных изменений.

Встроенные шаблоны и проверенные инженерные практики задают единый безопасный путь выполнения операций.

Контроль над развитием инженерной организации



Стандартизация лучших практик

Экспертные знания и лучшие практики закрепляются в платформе, снижая операционную зависимость от отдельных специалистов.

Политики безопасности автоматически применяются в процессе разработки и поставки, а не проверяются вручную перед выходом в продуктив.



Масштабирование организации

Рост количества разработчиков и микросервисов не требует пропорционального расширения команды эксплуатации.

Уникальные требования к шаблонам и процессам отдельных команд достигаются механизмами конфигурации портала без излишнего дублирования.



Контроль расходов

Агрегация стоимости ресурсов по сервисам и окружениям показывает, какие бизнес-инициативы потребляют инфраструктуру и в какую сумму они обходятся организации.

Бюджеты, квоты и уведомления помогают своевременно выявлять неэффективные расходы.

Контроль бюджетов на инфраструктуру разработки



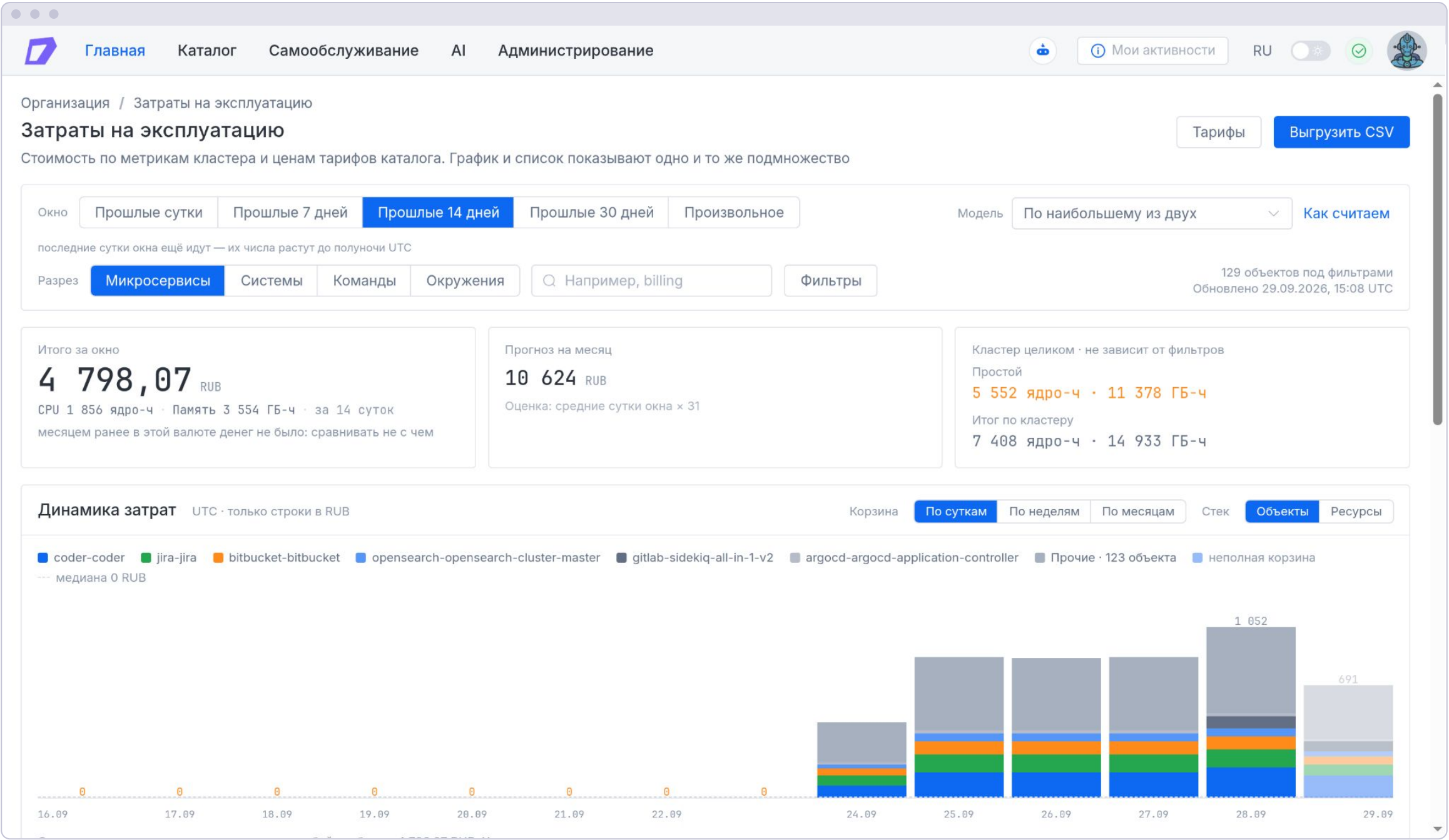
Агрегация стоимости ресурсов по сервисам и окружениям.
Структура затрат в разрезе команды или бизнес-инициативы



Общесистемная аллокация — часть ИТ-ландшафта, чья стоимость соразмерно распределяется по всем бизнес-инициативам



Бюджетирование. Идентификация превышений пороговых значений и динамики роста.



DORA- и SDLC-метрики продуктивности



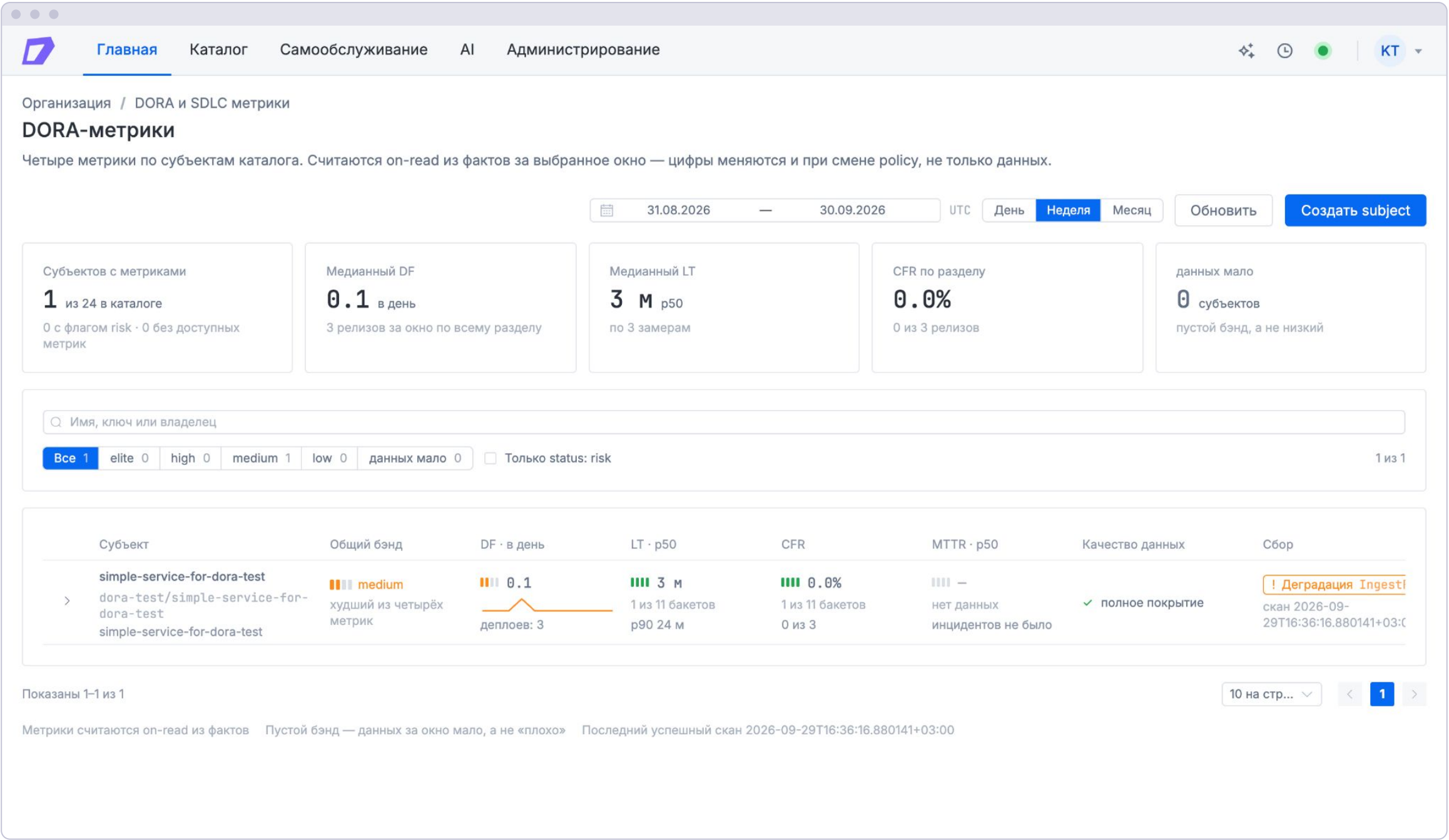
Эффективность DevOps

- Частота развёртываний
- Время от изменений до релиза
- Доля изменений с инцидентами
- Восстановление после сбоя



Жизненный цикл разработки ПО

- Контроль качества кода
- Покрытие тестами
- Тренды метрик безопасности



Контроль качества, встроенный в Secure SDLC

Портал не заменяет отдельные сканеры, а интегрируется с ними и синхронизирует данные через API. Возможность интегрировать любой сканер безопасности в контуре организации.

Доступны «из коробки»:

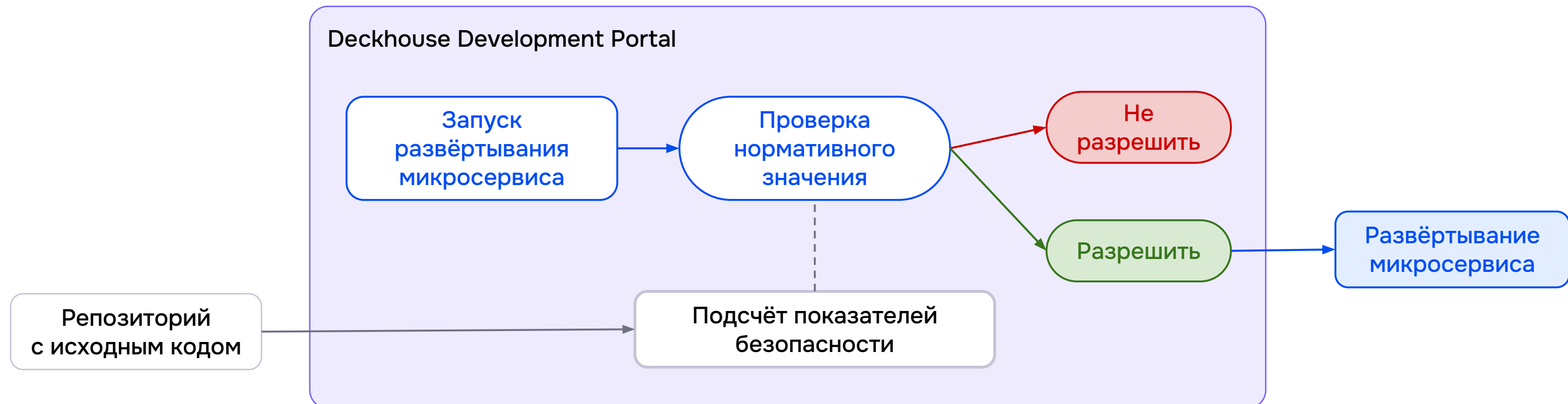
CodeScoring

SonarQube

Svacer

DefectDojo

Quality Gate позволяет встроить контроль безопасности непосредственно в процесс доставки ПО.



Портал самообслуживания разработчика

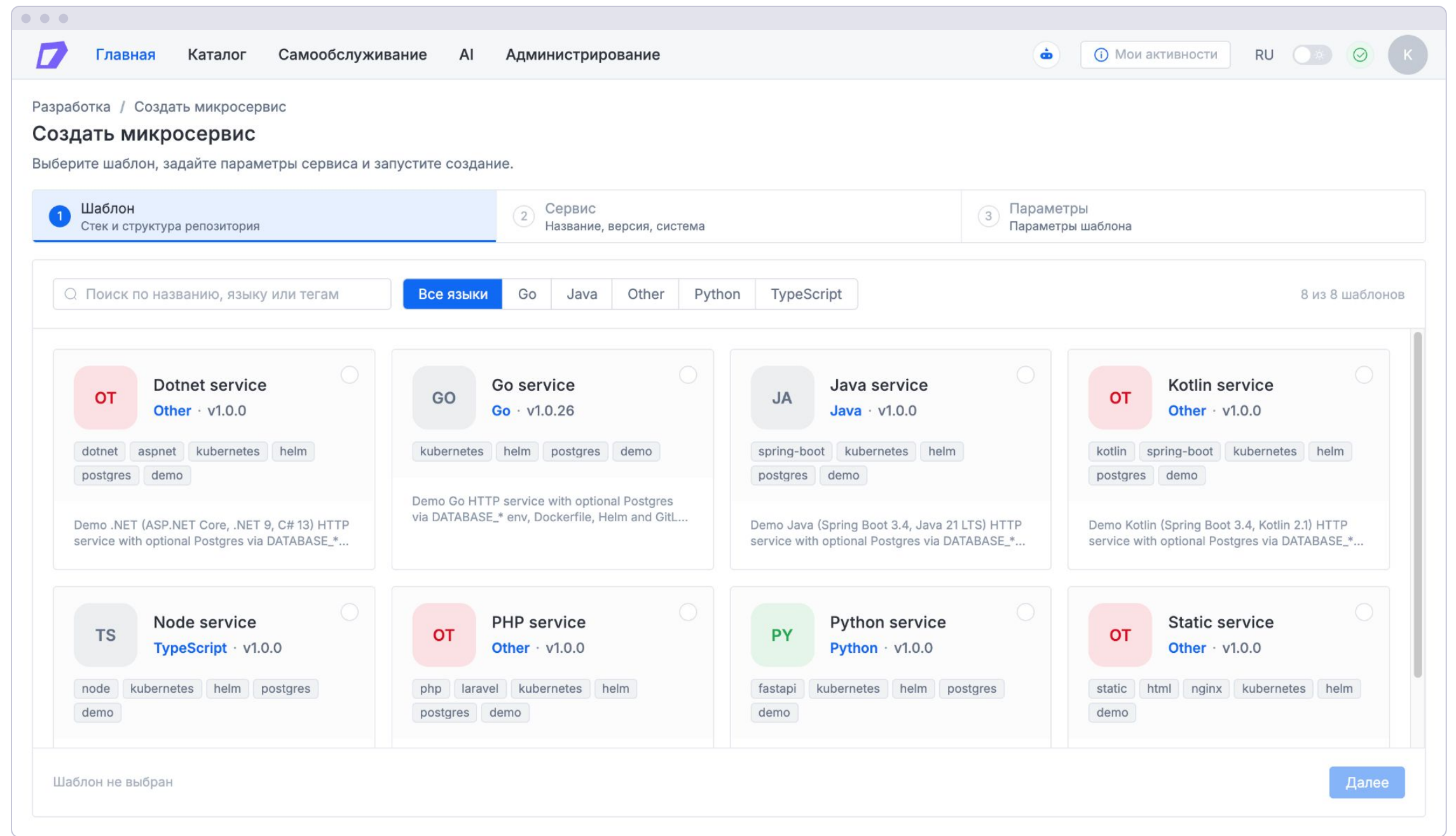
Быстрое создание и развёртывание микросервисов

Новый микросервис

Собирается в GitLab

Развёртывается в Kubernetes

- Шаблоны репозиторий на всех популярных языках программирования
- Встроенный пайплайн сразу работает и собирает артефакт
- Импорт собственных шаблонов с нужным стеком и библиотеками
- Сервисы получают обновления, когда шаблон изменится



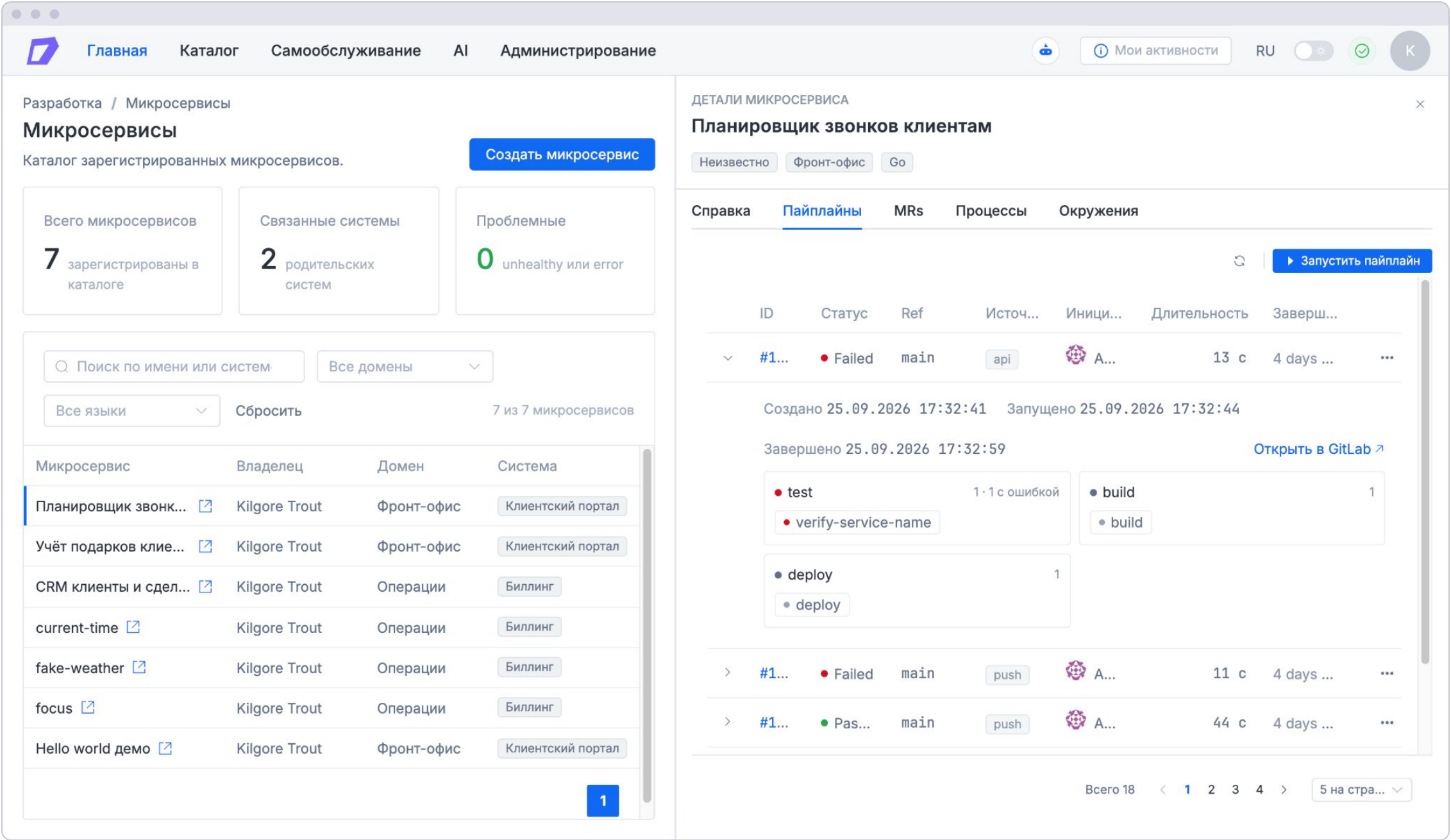
Быстрое создание и развёртывание микросервисов

Новый микросервис

Собирается в GitLab

Развёртывается в Kubernetes

- Единое окно в CI/CD
- Собранные артефакты
- Ошибки сборки
- Запуски авто- и юнит-тестов
- Анализ на уязвимости
- Провалившийся контроль качества (quality gates)
- Успешные развёртывания
- Ошибки доставки



Быстрое создание и развёртывание микросервисов

Новый микросервис

Собирается в GitLab

Развёртывается в Kubernetes

- Где развёрнут каждый микросервис
- Временные окружения на отдельном URL
- Автоматические окружения для Merge Request
- Подсчёт стоимости инфраструктуры
- Удаление временных окружений при простое

ГлавнаяКаталогСамообслуживаниеAИАдминистрирование

Разработка / Окружения

Окружения

5 окружений · постоянных: 4 · временных: 1

Создать окружение

Все 5Постоянные 4Временные 1

Поиск по названию или к

Название	Тип	Кластер	Домен
coder сервисов: 0 · систем: 0	—	—	—
dev сервисов: 0 · систем: 0	Non-production	Кластер разработки	dev.acmelogistics.ex
prod сервисов: 3 · систем: 2	Production	Промышленный кластер	demo.ddp.flant.dev
stage сервисов: 0 · систем: 0	Pre-production	Кластер разработки	stage.acmelogistics.
temp-review сервисов: 0 · систем: 0	Non-production Помечено на удалени	Кластер разработки	temp-review.acmelogi

Временными считаются окружения с заданным TTL.

ДЕТАЛИ ОКРУЖЕНИЯ

temp-review

НеизвестноNon-productionКластер разработкиПомечено на удаление

Название	temp-review
Тип	Non-production
Кластер	Кластер разработки
Домен	temp-review.acmelogistics.example.com
Класс	Временные
TTL	7d

ПРОИСХОЖДЕНИЕ

Merge Request	https://gitlab.com/development-platform/-/merge_requests/872
Ветка	main
Репозиторий	Fox
Ответственный	Александр Миловидов

БАЗОВОЕ ОКРУЖЕНИЕ

Не выбрано

Опасная зона

Композиция связанных микросервисов и ресурсов



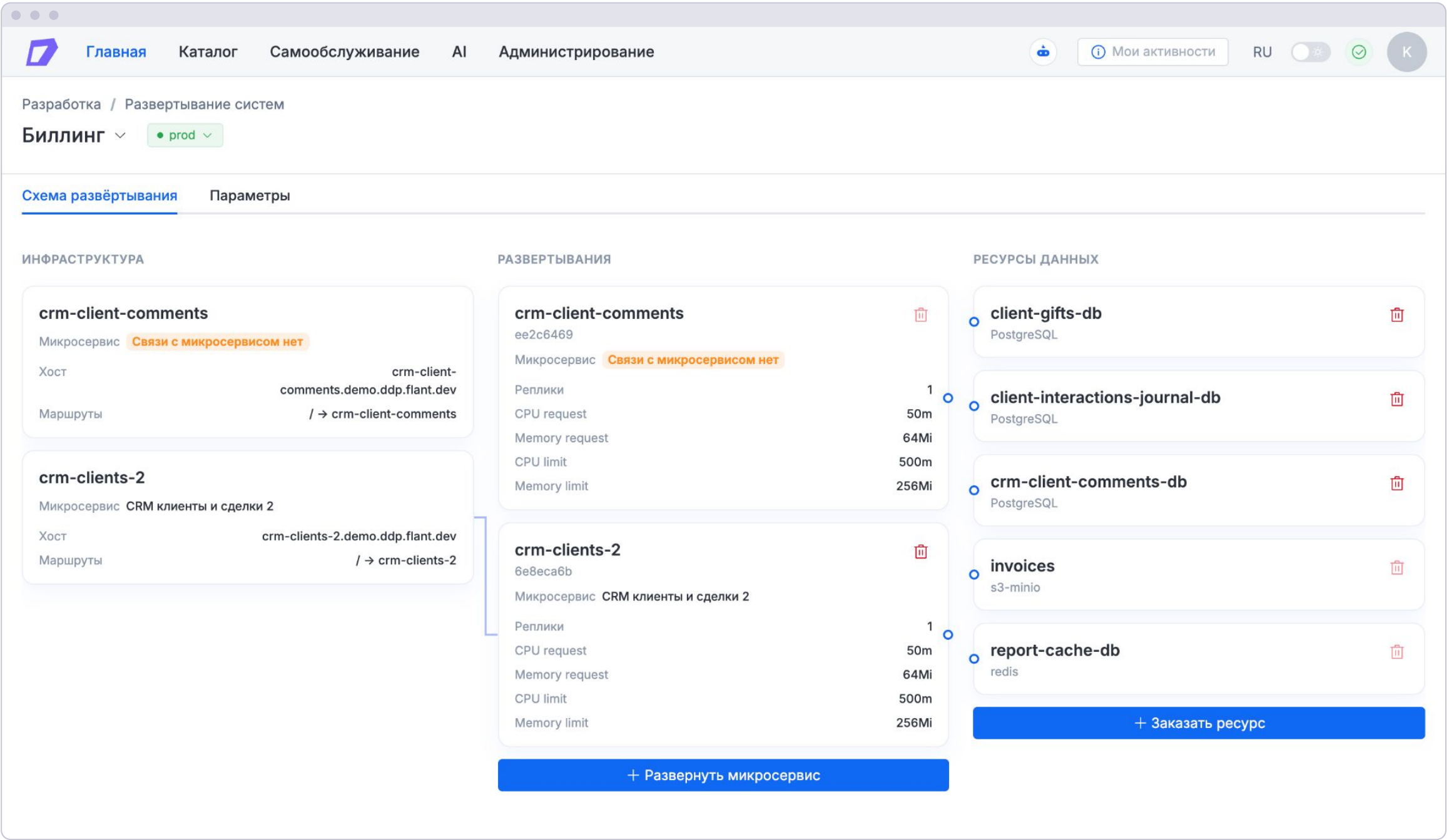
Визуальная карта развертывания
нескольких связанных микросервисов
внутри одной Системы



Перечень всех подключенных
баз данных и хранилищ



Компактная форма
представления, понятная
разработчикам



Автоматизированный заказ баз данных и хранилищ



Заказать ресурсы:

Реляционные базы данных

Key-Value

NoSQL

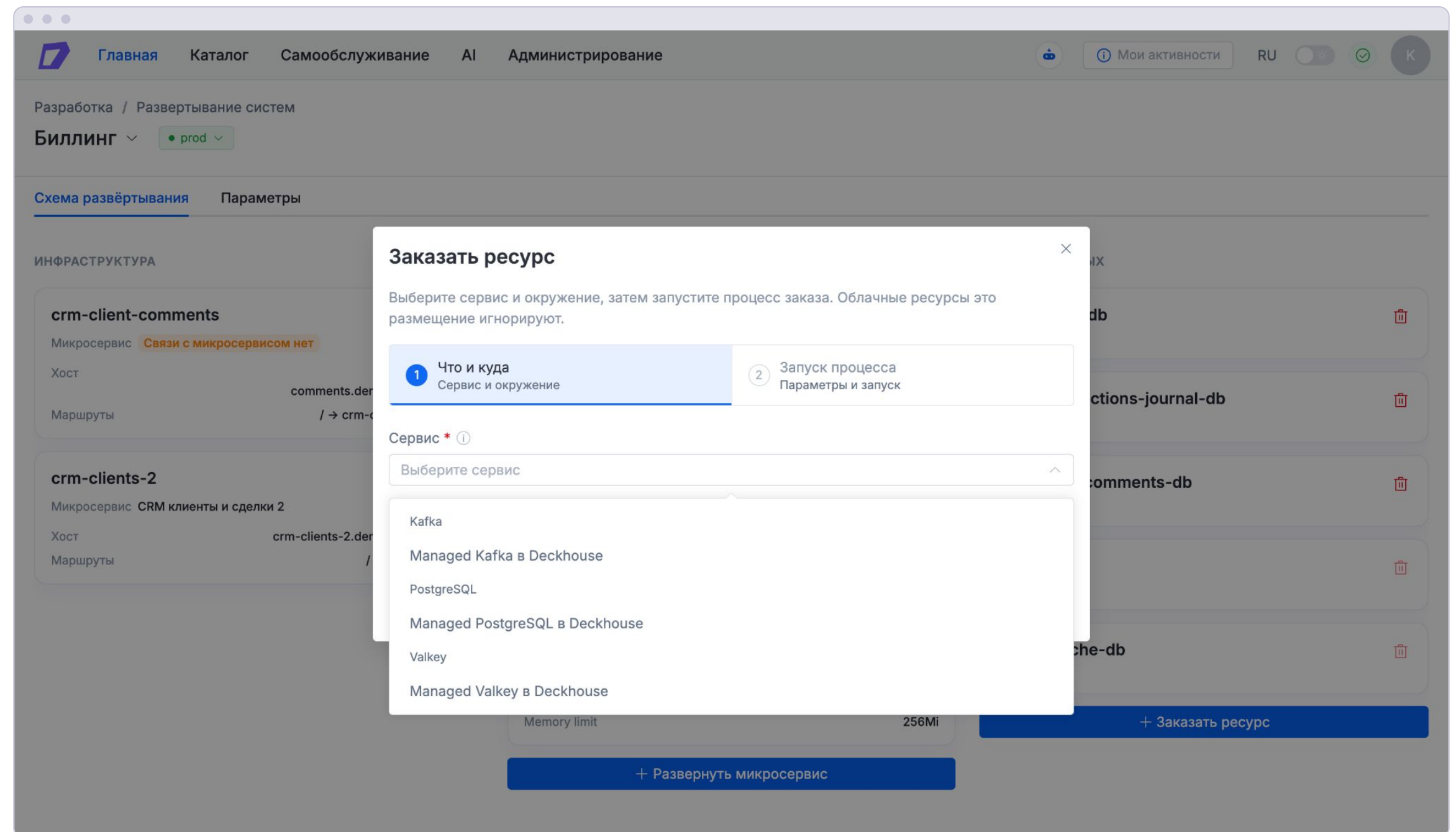
Файловое хранилище S3



Интегрируется с любым provisioner'ом ресурсов в контуре организации или облаке



Реквизиты доступа (secrets) безопасно пробрасываются в поды



Получите чекап SDLC бесплатно

- ✓ Проведем чекап вашего процесса SDLC
- ✓ Подготовим индивидуальный отчёт с конкретным планом действий и оценкой эффекта от внедрения:
 - Как снизить финансовые издержки
 - Как повысить эффективность команд



Кастомизируемость под специфику организации

Приземляется в любой ландшафт

- Любые Kubernetes-кластеры и инфраструктурные системы
- Множество готовых адаптеров
- Легко интегрироваться по API
- Не обязательно приобретать другие продукты Deckhouse

Разработчик



Доставка



Argo CD

Nexus

GitLab

Vault

Эксплуатация

Мониторинг

Логирование

Безопасность

CodeScoring

SonarQube

DefectDojo

Ресурсы и данные

Kubernetes

Базы данных,
хранилища

Инфраструктурные
системы клиента

Учётные системы
клиента

Отлично встраивается в экосистему Deckhouse

Разработчик



Доставка



Эксплуатация



Безопасность

CodeScoring

SonarQube

DefectDojo

Ресурсы и данные



Инфраструктурные
системы клиента

Учётные системы
клиента

Простые интеграции с вашим ландшафтом без кодирования

Подключите ваши внутренние инфраструктурные и учётные системы. Данные из этих систем-источников будут доступны внутри портала для использования в сценариях.



Интеграция через REST API без кодирования

Сконфигурируйте отображение стороннего REST API в сущности портала без написания кода с помощью визуального интерфейса.



Синхронизация по расписанию или по событиям

Получайте изменения из систем-источников с заданной периодичностью или через Webhook и обновляйте данные на портале без задержек.



Запуск действий в интегрированных системах

Запускайте последовательности действий в ваших внутренних инфраструктурных и учётных системах напрямую из интерфейса портала.



Пользовательская аутентификация или техническая учётная запись

Вызов стороннего REST API может выполняться либо с токеном аутентификации текущего пользователя или из-под общей технической учётной записи.

Сконфигурируйте согласно вашим стандартам



Ваши собственные шаблоны сервисов с одобренным стеком технологий

Выберите нужные фреймворки и библиотеки и опубликуйте на портале как доступный шаблон. Созданные из шаблона сервисы получают обновления, когда шаблон изменится в будущем.



Ваши собственные пайплайны

Простой механизм связывания используемых в вашей организации GitLab-CI-шаблонов с логикой портала.



Настраиваемые действия самообслуживания

Ограничьте доступ с помощью RBAC-like-механизма. Сконфигурируйте формы и виджеты для отображения. Настройте обязательное подтверждение чувствительных действий инженером эксплуатации.

Заказать виртуальную машину

Создать базу данных

Создать окружение

и любые другие...

Фреймворк для конструирования SDLC-процессов



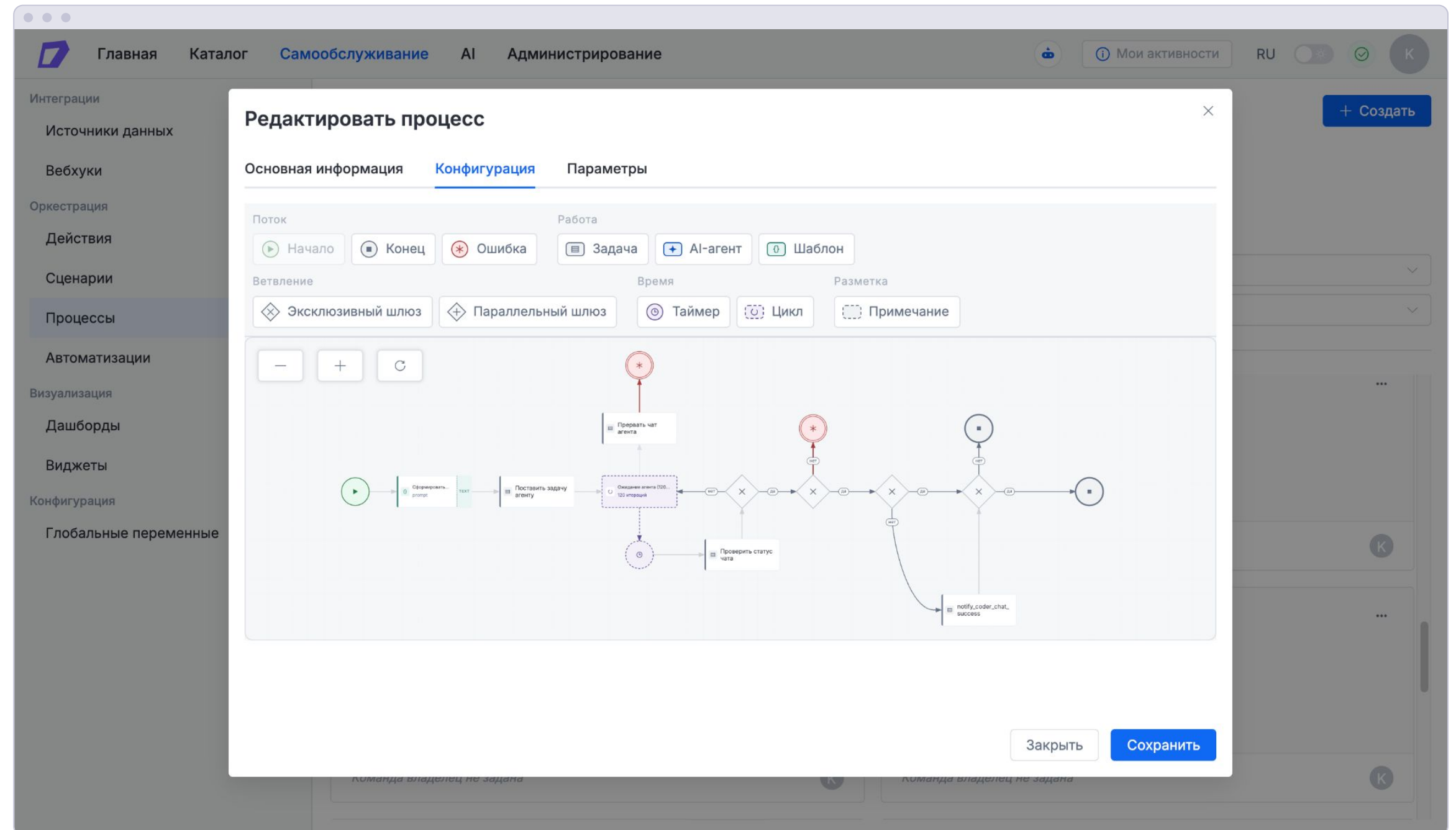
БPMN-подобный движок для автоматизации SDLC процессов и моделирования любых внутренних правил и согласований



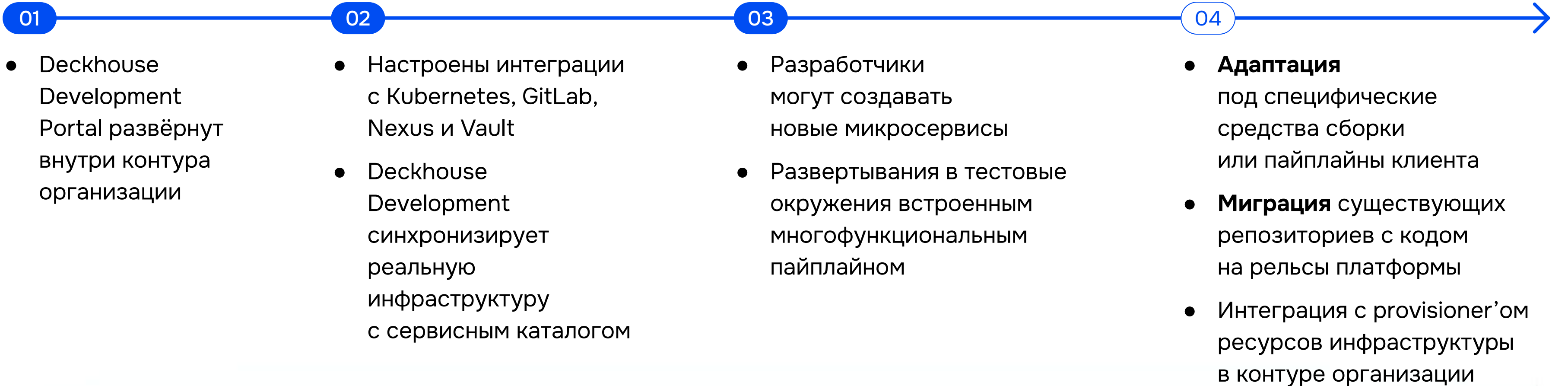
Визуальный редактор с условиями, ветвлениями и параллельным выполнением действий



RBAC-подобный механизм для ограничения доступа



Типовой пилотный проект



Deckhouse Professional Services – консалтинг и внедрение

Эксперты Deckhouse
берут на себя полный цикл работ –
от проектирования архитектуры
до построения готовой платформы
под ваши задачи

На 60 % быстрее

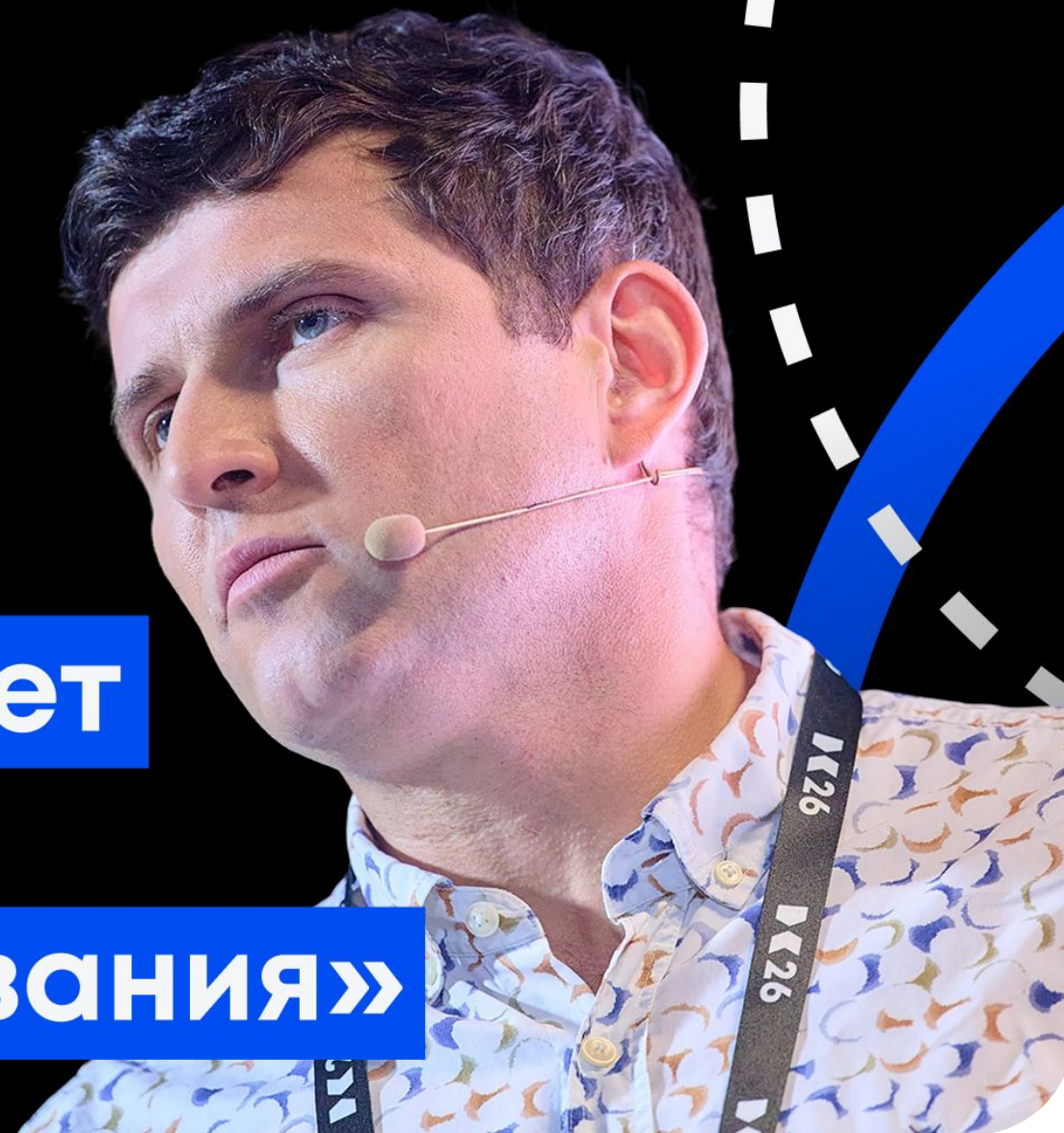
достигается Time to Value

-  Быстрый и предсказуемый старт
-  Целостный подход
-  Передача знаний
-  Адаптация работ под клиента

DECKHOUSE
CONF 2026

● Антон Исанин

Как **DDP** ускоряет 50+ команд «АльфаСтрахования»



АЛЬФА
СТРАХОВАНИЕ

 [YouTube](#)

 [RuTube](#)

 [VK Video](#)

Спасибо за внимание!

Максим Пшеничников
Менеджер продукта
Deckhouse Development Portal

✉ maksim.pshenichnikov@flant.com

☎ +7 (926) 786-51-71



Получите чекап SDLC бесплатно

Проведем чекап вашего процесса SDLC
и подготовим индивидуальный отчёт
по снижению финансовых издержек
и повышению эффективности команд за счёт
внедрения внутренней платформы разработки