

avito.tech

2022

# Платформенный продукт

## Как это делать?

Лукьянченко Александр

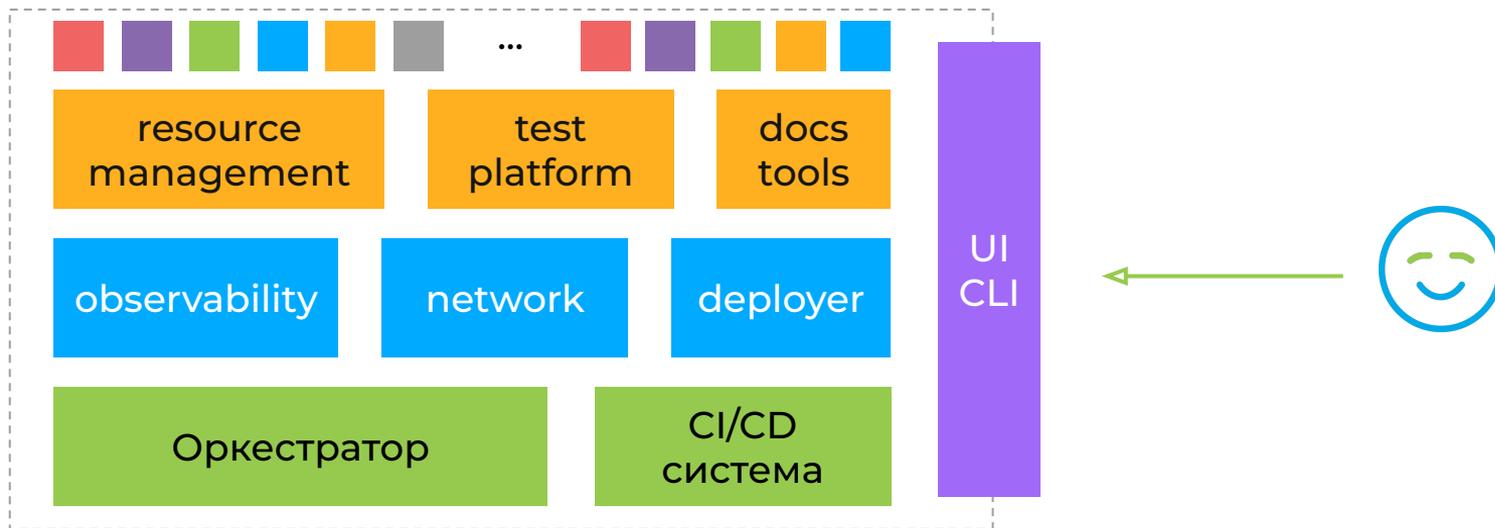
Руководитель PaaS, Avito

# План

1. Что такое платформенный продукт? (PaaS/IDP)
2. Уровень автоматизации
3. Фазы зрелости и внедрение
4. Продуктовый подход
5. Check list полноты платформы

# Что такое платформенный продукт?

# PaaS / Internal Developer Platform



# Что покрывает платформа?



# Продуктовые процессы



Support



Единое окно  
описания  
проблем и  
идей



Сбор метрик  
удовлетворенности  
и custdev



Discovery +  
проработка  
CJM

# Уровень автоматизации

# Поиск золотой середины

Гибкость



Простота

# Поиск золотой середины

Догоняющие метрики



Объём  
саппорта



Количество  
запросов на  
новые  
ВОЗМОЖНОСТИ

# Поиск золотой середины

## Догоняющие метрики

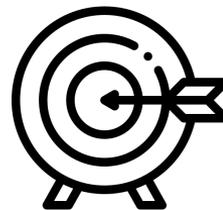


Объём  
саппорта



Количество  
запросов на  
новые  
возможности

## Метрики этапа внедрения



Эффективность  
автоматизаций

# Ценность платформы для бизнеса

07. **Time to market:** время на onboarding, старт новых сервисов / продуктов, ускорение релизов в production



# Ценность платформы для бизнеса

01. **Time to market:** время на onboarding, старт новых сервисов / продуктов, ускорение релизов в production



02. **Фокус на ценности:** self service без необходимости наличия экспертизы в продуктовых командах



# Ценность платформы для бизнеса

**01.** **Time to market:** время на onboarding, старт новых сервисов / продуктов, ускорение релизов в production



**02.** **Фокус на ценности:** self service без необходимости наличия экспертизы в продуктовых командах



**03.** **Стабильность системы и качество:** унифицированные средства observability и общие quality gates



# Ценность платформы для бизнеса

01.



**Time to market:** время на onboarding, старт новых сервисов / продуктов, ускорение релизов в production

02.



**Фокус на ценности:** self service без необходимости наличия экспертизы в продуктовых командах

03.



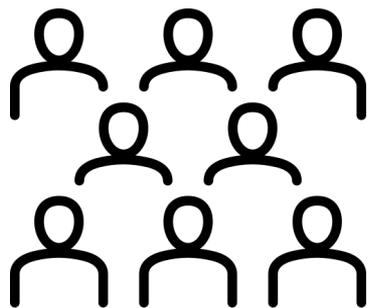
**Стабильность системы и качество:** унифицированные средства observability и общие quality gates

04.



**Уверенность в будущем:** централизованный контроль зоопарка технологий, подходов разработки

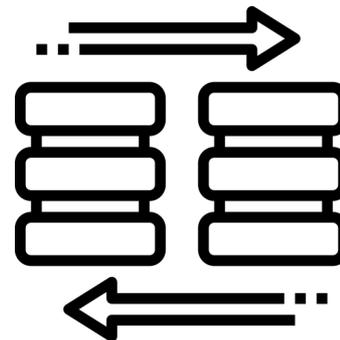
# Чего это стоит?



Разработка



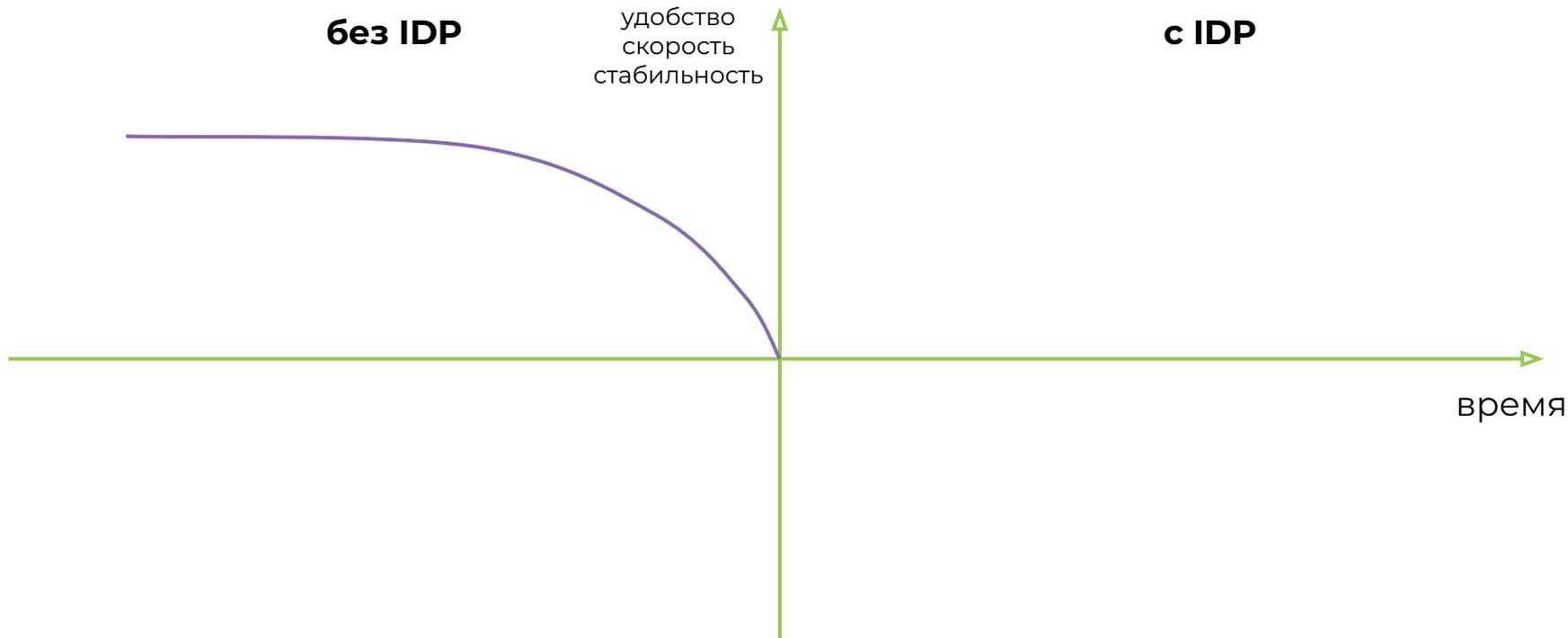
Возможный vendor lock



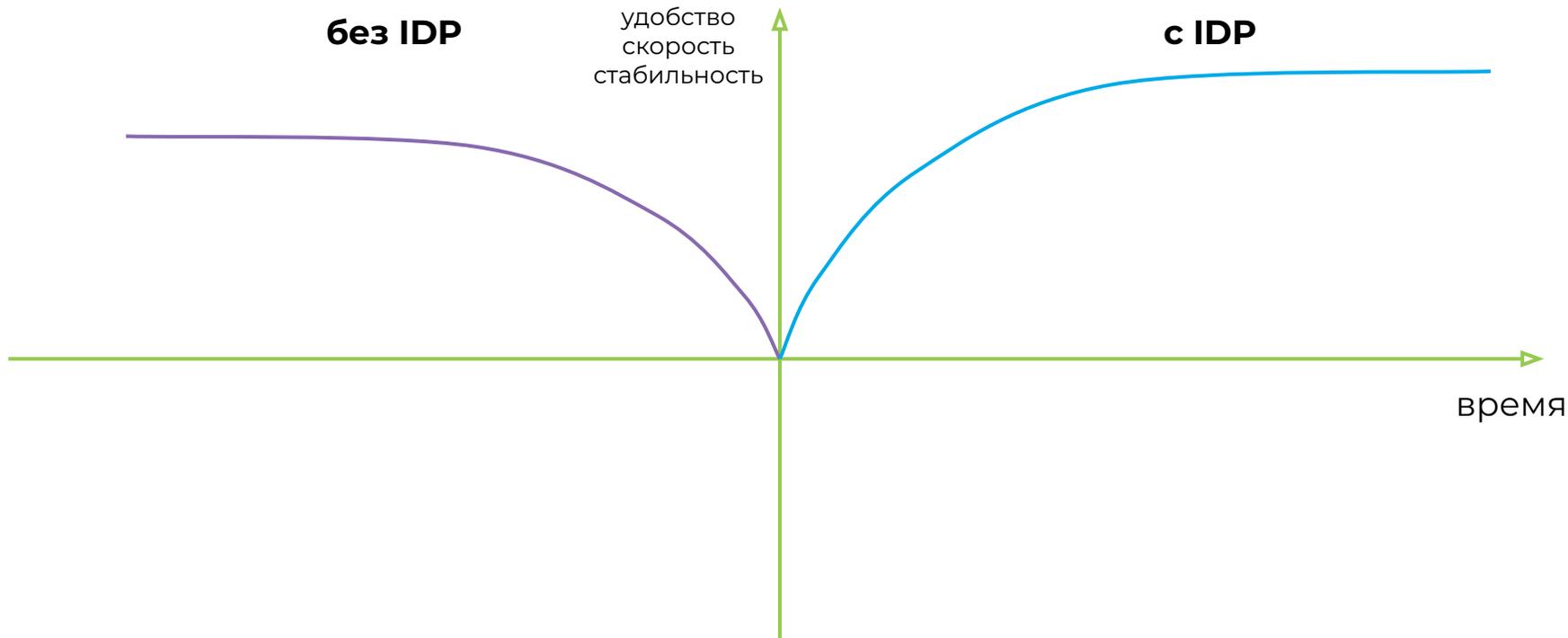
Миграция и внедрение подходов

# Фазы зрелости и внедрение

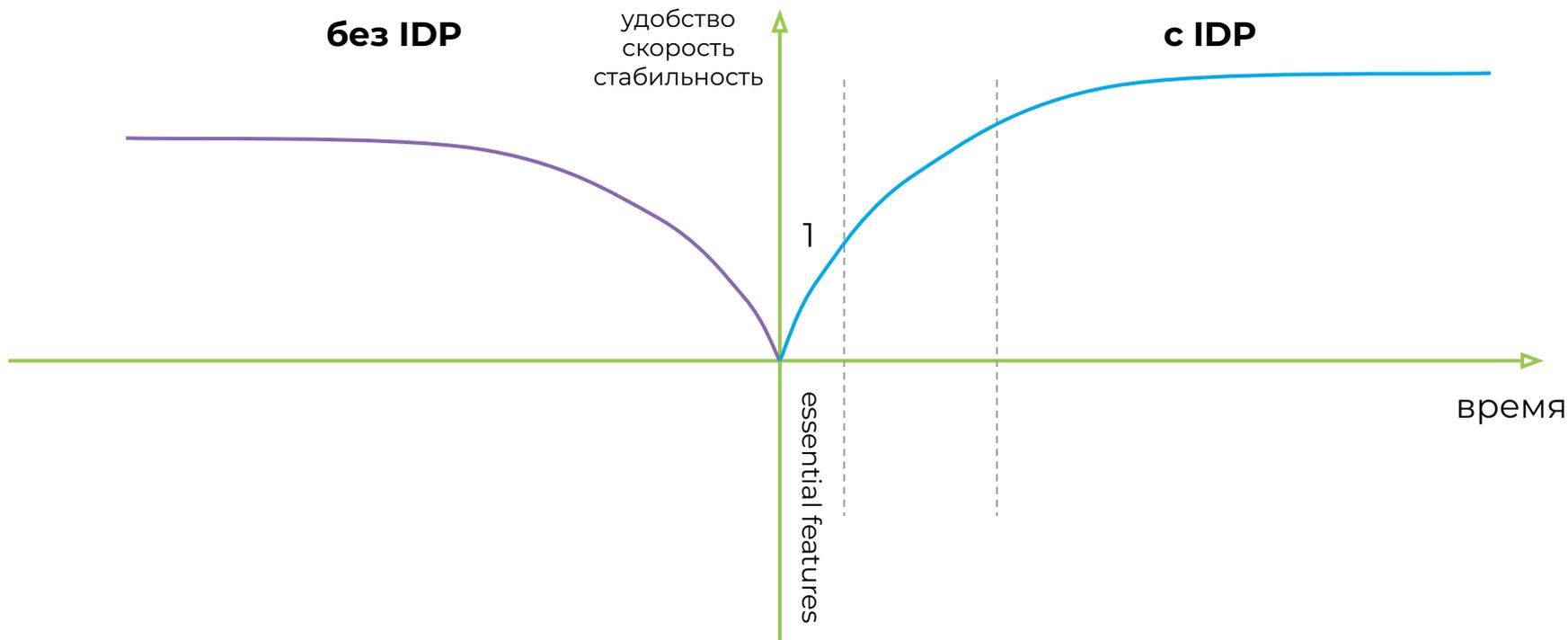
# Фазы зрелости



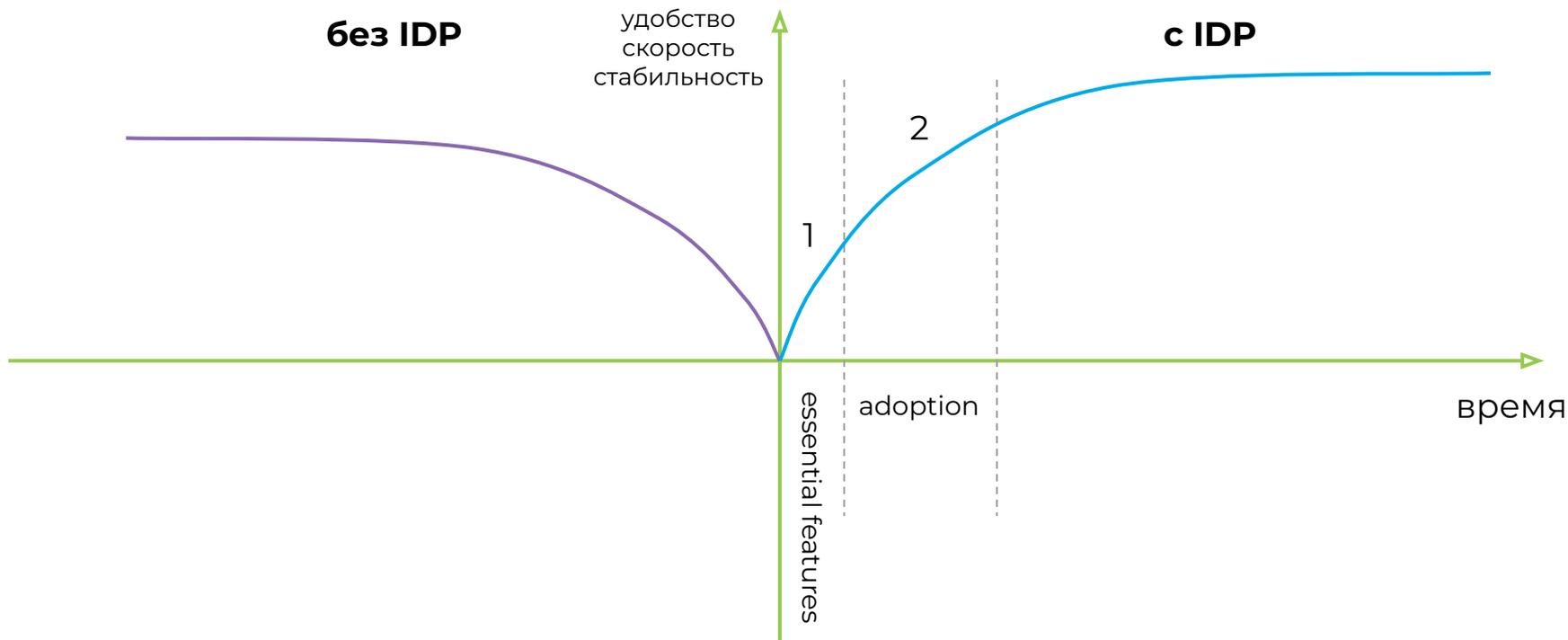
# Фазы зрелости



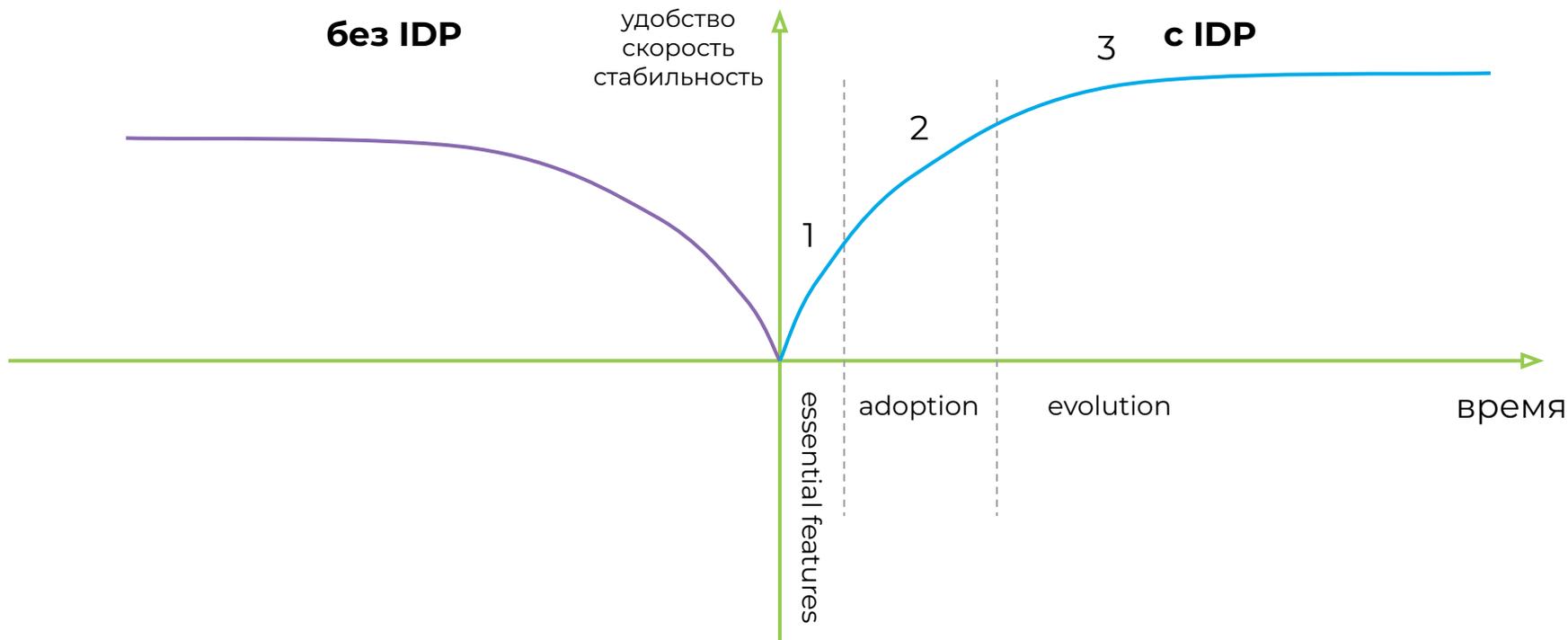
# Фазы зрелости



# Фазы зрелости



# Фазы зрелости



# Продуктовый ПОДХОД

# Метрики и их сбор

*Скорость*



Onboarding time

Локальное окружение (развертывание, разработка)

Время на интеграцию с инфраструктурой

Время решения локальных проблем

# Метрики и их сбор

*Фокус на ценности*



CJM coverage

Отношение ручных действий / автоматизации по направлениям

Необходимость обращения в platform/ops команду

# Метрики и их сбор

*Стабильность*



SLI ключевых компонентов

Quality gates эффективность (red -> green после изменения)

MTTU / MTTR

Self healed cases / manual

# Метрики и их сбор

*Уверенность в будущем*



Актуальность технического стека (версии языков, библиотек)  
Adoption общих практик

# Что важно пользователю продукта?

1. Удобство работы
2. Полнота функциональности
3. Минимум усилий для достижения целей
4. Оперативное решение проблем

# Продуктовые процессы



Support



Единое окно  
описания  
проблем и  
идей



Сбор метрик  
удовлетворенности  
и custdev



Discovery +  
проработка  
CJM

# Support



1. Выделенная единая линия (дежурства по направлениям)
2. Сбор аналитики по зонам
3. Время ответа, удовлетворенность
4. База знаний

# Requests механика



1. Сбор информации: описание проблемы, окружения, потенциального импакта, контактных лиц.
2. Обработка высокоуровневая.
3. Проработка потенциального решения и ответ.
4. Планирование и компоновка в Roadmap.

# Discovery



1. Ценность (+ press release).
2. Stakeholders.
3. Как будем измерять impact.
4. Сложность реализации.
5. Артефакты на выходе (CJM, метрики, выводы из custdev).

# Effort score / satisfaction



1. Выделяем ключевые зоны CJM.
2. На финальных шагах задаём контекстные вопросы:
  - a. Сложность операции?
  - b. Удовлетворенность?

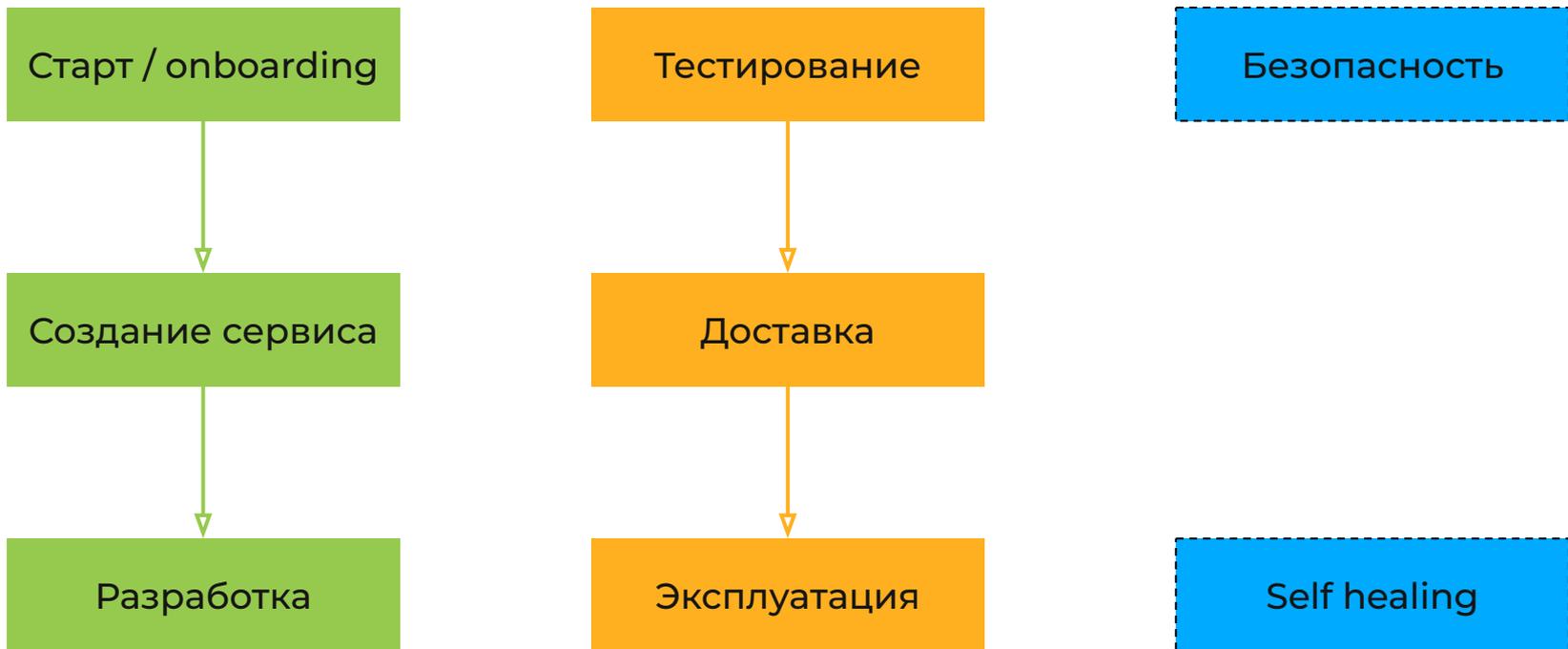
CJM



# Платформа и её ВОЗМОЖНОСТИ

+ checklist!

# Жизненный цикл разработки



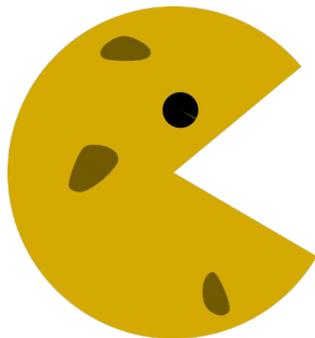
# Старт и onboarding

# Локальная среда

One button start



# Готовые компоненты и boilerplate



Cookiecutter



Boilerplate

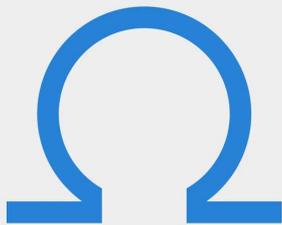
...



Monitoring

# Разработка

# Конфигурирование

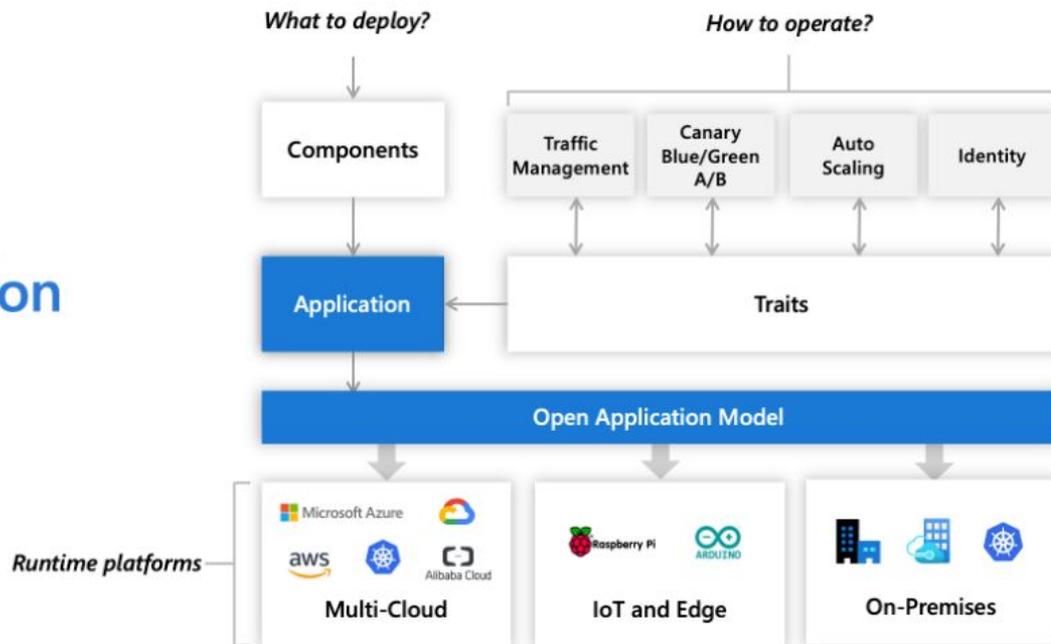


Open  
Application  
Model



# Конфигурирование

## Open Application Model

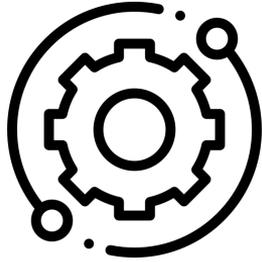


# Конфигурирование

```
apiVersion: core.oam.dev/v1beta1
kind: Application
metadata:
  name: first-app
spec:
  components:
  - name: helloworld
    type: webservice
    properties:
    image: oamdev/helloworld-python:v1
    env:
      - name: "TARGET"
        value: "KubeVela"
    port: 8080
  traits:
  - type: ingress
    properties:
      domain: localhost
      http:
        /: 8080
```



# Zero configuration init



```
sessionRedis := redis.New()
```



Env variables

# Кодогенерация и сервисное взаимодействие

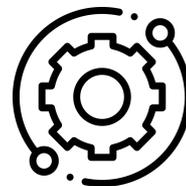
gRPC



Server



Client



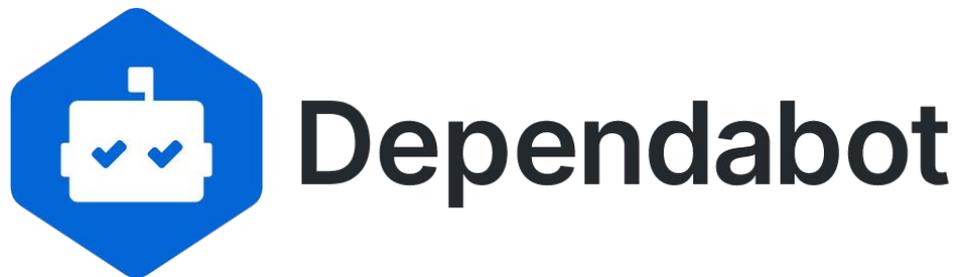
Сгенерированный код  
(circuit breakers, retries,  
timeout propagation,  
observability, zero conf)

# Хаас зависимости

1. Cron
2. Workers
3. Databases
4. Object storage
5. Queues, data-bus

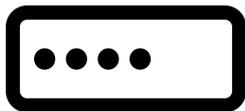


# Поддержка актуальности тех стека



# Безопасность

# Менеджмент секретов



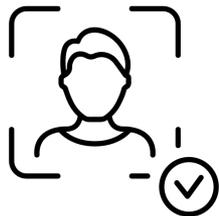
UI / CLI



Env/file  
Injection



# Межсервисное взаимодействие и безопасность



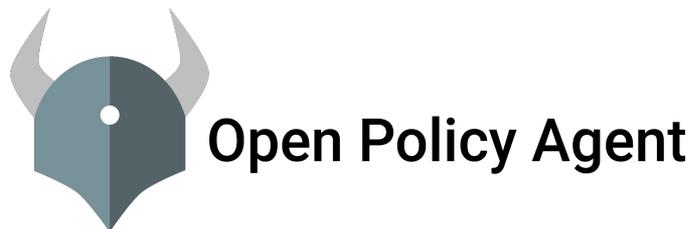
mTLS



Авторизация  
сервисов



# IDP авторизация



Role based access control

# Security crawlers

1. OS / docker images
2. Libraries
3. Secrets leaks
4. Security linters

# Тестирование

# Фазы тестирования



# Auto test platform



Линтинг, style guides

actions



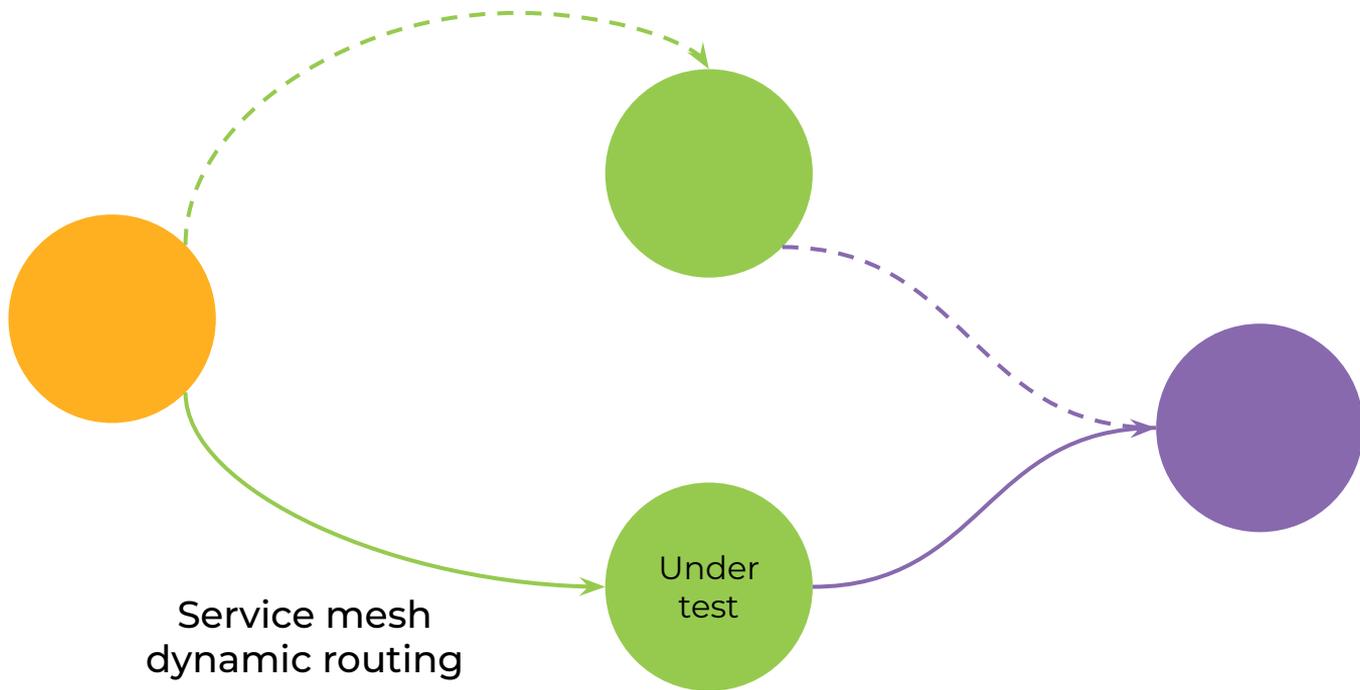
Unit / Integration tests  
mutation



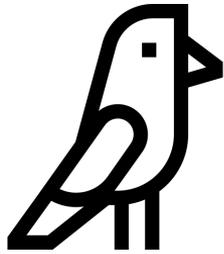
Контрактное  
тестирование

РАСТ 

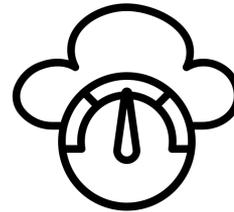
# E2E



# Shift right



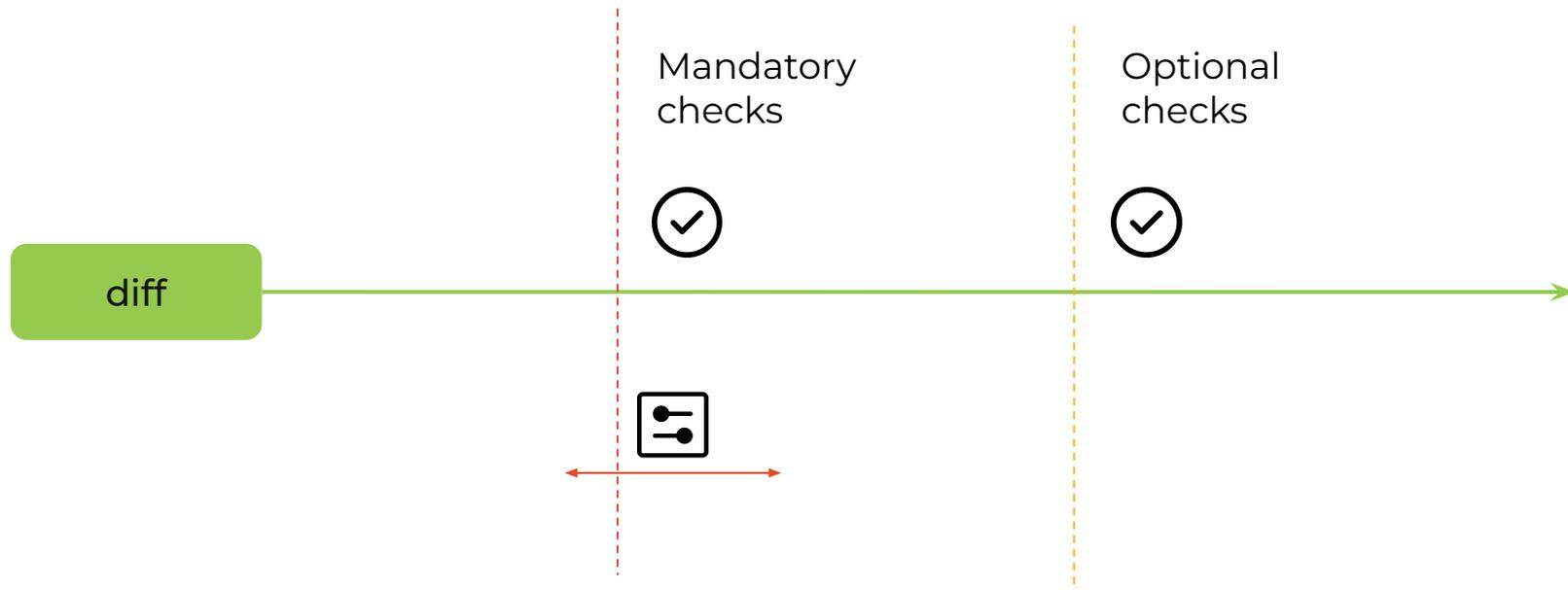
Canary



Chaos

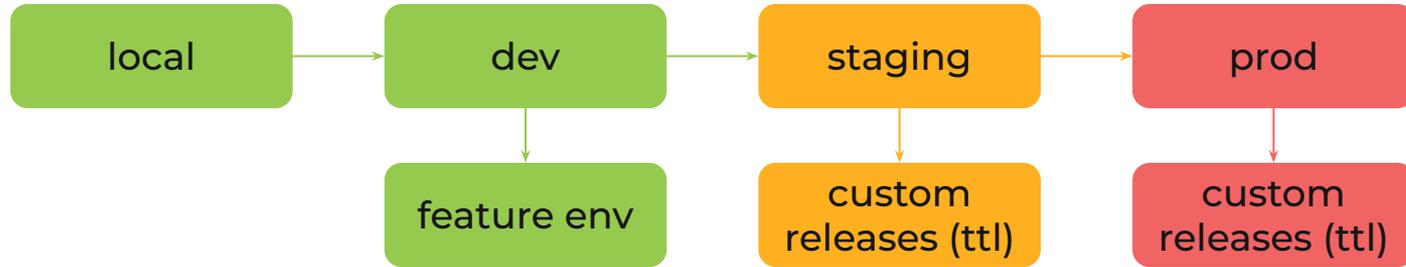


# Контролируемые quality gates



# Доставка

# Environment management



# Pipelines

1. Готовые унифицированные сборки
2. One button deploy
3. Поддержка различных стратегий: canary, blue/green, shadow.
4. Continuous deployment
5. Возможность расширения без ручного внедрения в базовые пайплайны
6. Переиспользование артефактов

# Problem detector

1. Root cause analyzer
2. Детектор типичных проблем: OOM, throttling, падения контейнеров, проблемы шедулинга, деградации воркер нод и прочие.



# Эксплуатация

# Реестр сервисов и инцидент менеджмент

Home | Services | Backstage x +

demo.backstage.io/catalog

Good afternoon, Guest!  
Backstage Service Catalog

SERVICES WEBSITES LIBRARIES DOCUMENTATION OTHER

## Services

PERSONAL

- Owned 7
- Starred 0

BACKSTAGE

All 10

Refine Results CLEAR

TAGS

- data
- go
- java
- playback
- website

Owned (7)

Search

NAME	SYSTEM	OWNER	LIFECYCLE	DESCRIPTION	TAGS	ACTIONS
<a href="#">artist-lookup</a>	<a href="#">artist-engagement-portal</a>	team-a	experimental	Artist Lookup	java, data	<a href="#">🔗</a> <a href="#">✎</a> <a href="#">★</a>
<a href="#">playback-order</a>	audio-playback	user:guest	production	Playback Order	java, playback	<a href="#">🔗</a> <a href="#">✎</a> <a href="#">★</a>
<a href="#">searcher</a>		user:guest	production	Searcher	go	<a href="#">🔗</a> <a href="#">✎</a> <a href="#">★</a>
<a href="#">shuffle-api</a>	audio-playback	user:guest	production	Shuffle API	go	<a href="#">🔗</a> <a href="#">✎</a> <a href="#">★</a>
<a href="#">wayback-archive</a>		team-a	production	Archive of the...		<a href="#">🔗</a> <a href="#">✎</a> <a href="#">★</a>
<a href="#">wayback-archive-storage</a>		team-a	production	Storage...		<a href="#">🔗</a> <a href="#">✎</a> <a href="#">★</a>
<a href="#">wayback-search</a>		team-a	production	Search of the...		<a href="#">🔗</a> <a href="#">✎</a> <a href="#">★</a>



Backstage

# Реестр сервисов и инцидент менеджмент

Setup & Config

names or tags...

[Performance](#)  
Monitor App Performance

ols | Showing 193 matching services

SERVICE	TEAM
web-store-mongo	dba
auth-dotnet-postgres	dba
send-email-mysql-db	dba
email-mysqldefaultdb	
send-email-redis-queue	transactions
web-store	shopist
auth-dotnet	shopist
email-api-py	shopist
orders-sqlserver	orders
product-recommendation-db	dba
shipping-queue-redis	transactions
ad-server	ads
requests	
...	...

Configuration | Dashboards | Definition | Dependencies

Finish setting up your service

**Recommended setup for this service**

- Monitors (121) **ENABLED** Get alerted on anomalous behavior [View Monitors](#)
- SLOs (46) **ENABLED** Define targets for application performance [View SLOs](#)
- Source Code Integration **ENABLED** Connect traces directly to the source code [View Repository](#)
- Version Tagging **ENABLED** View service metrics by version
- Runtime Metrics **ENABLED** Gain additional insights into an application's ...
- Error Tracking **ENABLED** Group myriad of errors into manageable nu... [View Issues](#)
- Logs Correlation **ENABLED** Investigate traces with the relevant logs in c... [View Logs](#)

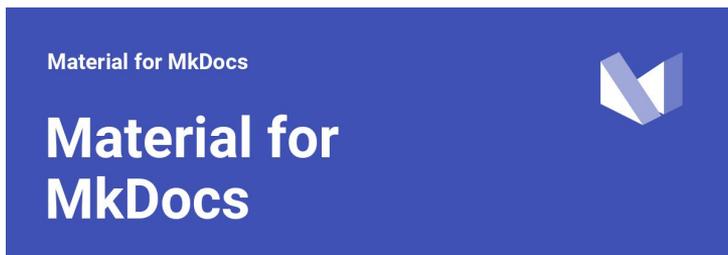
**Recommended telemetry**

- Distributed Tracing **ENABLED** Track application requests, errors, and latency [View Traces](#)
- Universal Service Monitoring **NOT ENABLED** Capture service health metrics without touch... [See Instructions](#)
- Infrastructure Monitoring **ENABLED** View host and container metrics [View Infra Metrics](#)
- Log Management **ENABLED** Collect, process, archive, explore, and monito... [View Logs](#)
- Synthetic Tests (36) **ENABLED** Run automated API tests and browser tests [View API Tests](#)
- Continuous Profiler **ENABLED** Analyze code performance in production [View Profiles](#)
- Application Security Monitoring **ENABLED** Detect threats in minutes [View Suspicious Requests](#)

Datadog  
service catalog

# Documentation as a service

1. Документация в кодовой базе
2. Простая инициализация
3. Markdown
4. Code review для документации



# Capacity management

1. Vertical / Horizontal Pod Autoscaler
2. Квотирование по орг структуре
3. Автоматический поиск ненужных сервисов, неэффективного использования железа

# Унифицированный observability



App centric



envoy

service mesh

tracing



JAEGER

metrics



Prometheus

continuous profiling

logs

exceptions  
tracker



kibana



SENTRY

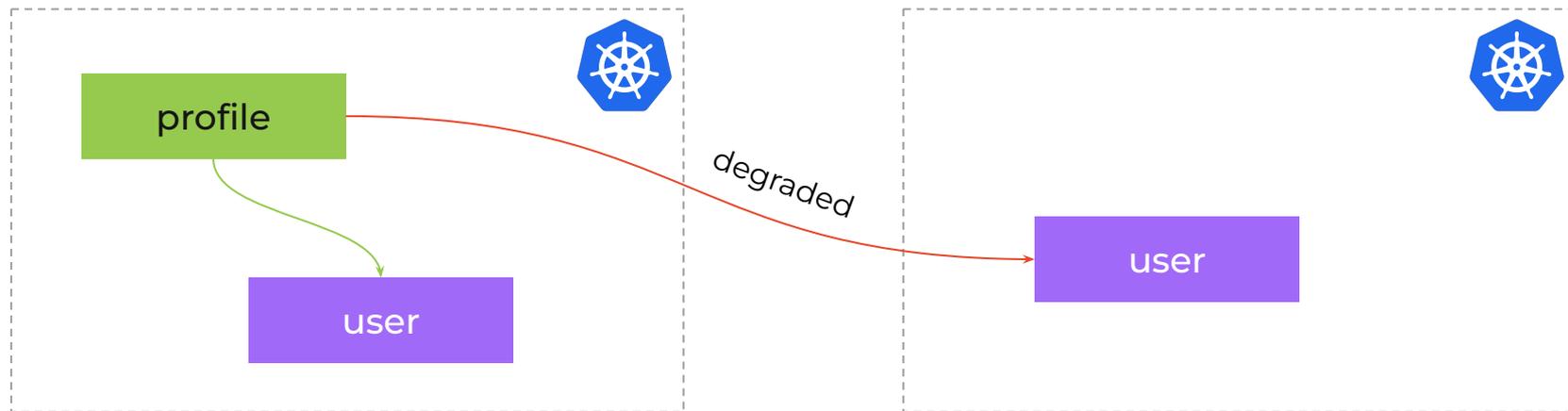
# Non Functional Requirements

1. Описание требований по API: latency, нагрузка.
2. Автоматическое вычисление.
3. Отображение статистики, SLI.
4. Alerting на  $SLI < SLO$ .

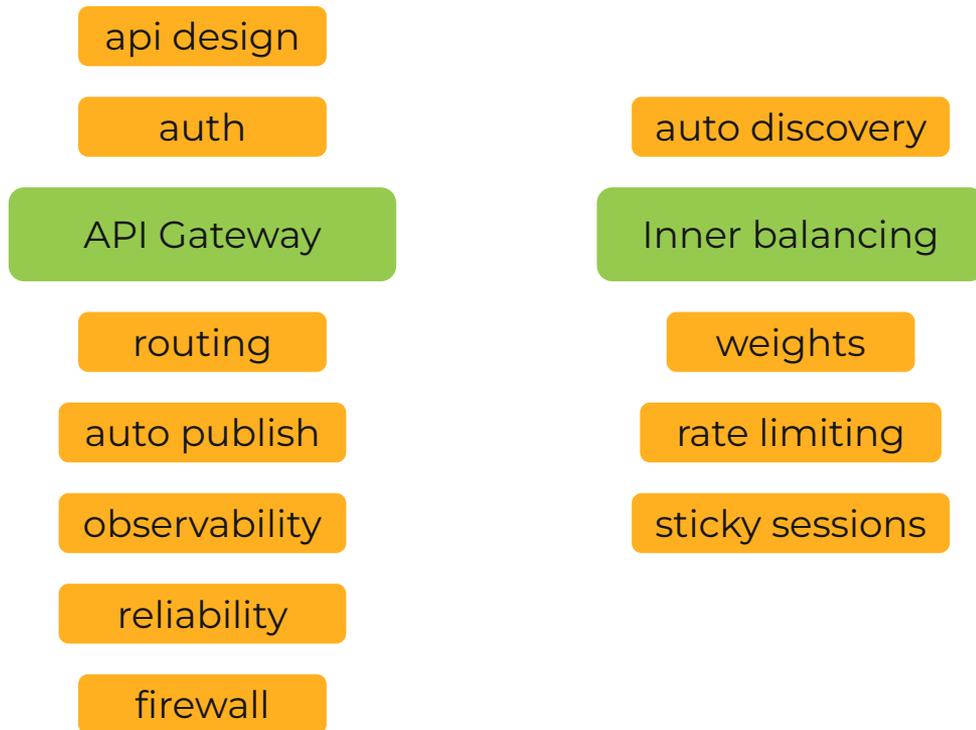
# Self healing infrastructure (runtime)

1. Descheduling.
2. Network retries.
3. Outlier detection.
4. Unhealthy nodes detection.

# MultiDC multi region поддержка



# Балансировка и API gateway



## PaaS / IDP Checklist



<https://forms.gle/JcKMENfgCuYYqRGg6>

Ваши вопросы!

**avito.tech**

Москва — 2022

# Александр Лукьянченко

Руководитель PaaS

 [@lookyan](https://t.me/lookyan)