



Платформа DBaaS: зачем и как



Владимир Алёшин

Руководитель разработки платформы DBaaS

- Тимлид в Авито.
- Работаю в Авито с 2018 года: Database Developer, DBA, Platform Engineer, Team Lead
- Разрабатываю и развиваю платформу DBaaS с 2019 года.
- Рад ответить на вопросы tg: @vovik0134



О чём это всё?

- Предпосылки создания платформы DBaaS.
- Выработка целевого решения для платформы DBaaS.
- Прекрасное настоящее.
- [следующий доклад] Особенности работы Kubernetes со stateful приложениями
- [доклад после него] Паттерны управления базами данных в multi-cluster (multi-dc) kubernetes среде
- [финальный доклад] Особенности адаптации классических СУБД к платформе DBaaS (на примере PostgreSQL)



DBaaS — это



DBaaS — это

- Базы данных как услуга.



DBaaS — это

- Базы данных как услуга.
- Полная автоматизация жизненного цикла базы данных: от развёртывания до удаления и отправки данных на архивное хранение.



DBaaS — это

- Базы данных как услуга.
- Полная автоматизация жизненного цикла базы данных: от развёртывания до удаления и отправки данных на архивное хранение.
- Снятие паразитной когнитивной нагрузки по работе с базами данных для продуктовых разработчиков.



Предпосылки создания платформы DBaaS

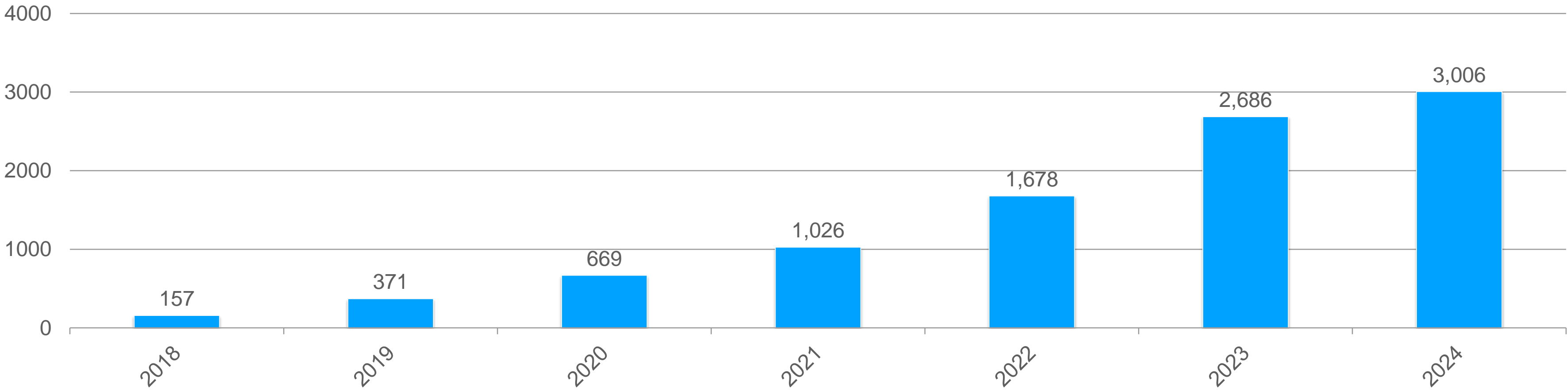


Переход Авито на микросервисную архитектуру.	09
Существовавший процесс по созданию баз данных.	14
Попытка номер раз: старая добрая классическая автоматизация.	15
Переход Авито в multi-dc.	20
Осознание.	24

Переход Авито на микросервисную архитектуру



Количество микросервисов в Авито

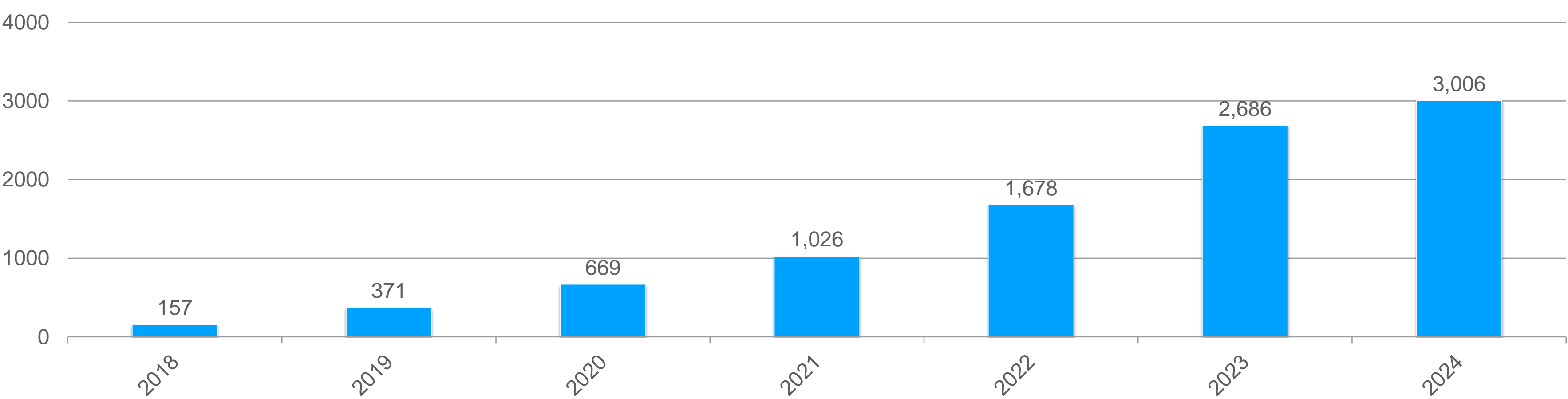


Переход Авито на микросервисную архитектуру

- В 2018 году Авито начала массовый переход от монолита к микросервисам: их количество стало быстро расти



Количество микросервисов в Авито

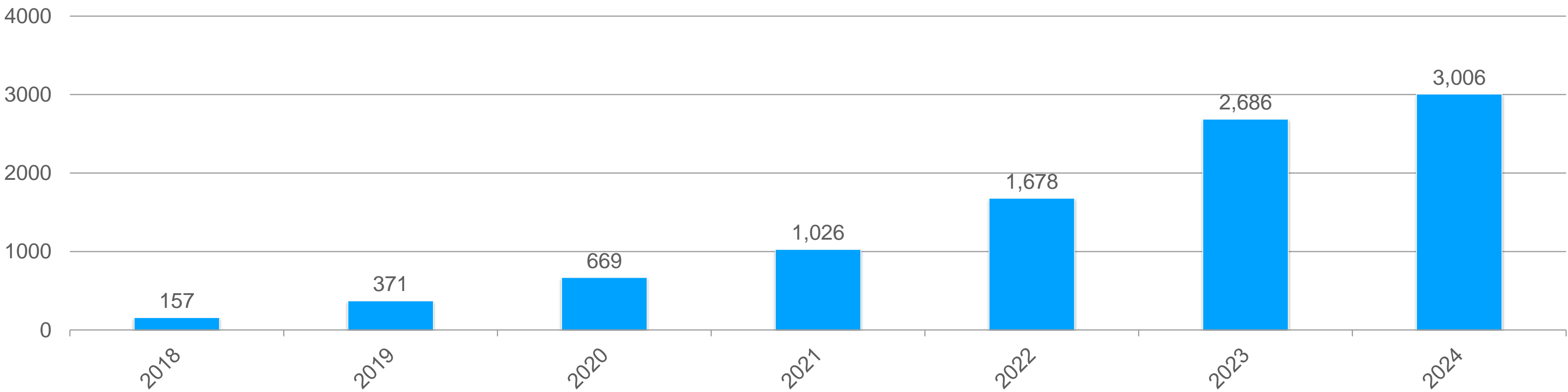


Переход Авито на микросервисную архитектуру

- В 2018 году Авито начала массовый переход от монолита к микросервисам: их количество стало быстро расти
- Согласно паттернам МА, микросервисы не могут использовать общую базу данных (database per service).



Количество микросервисов в Авито

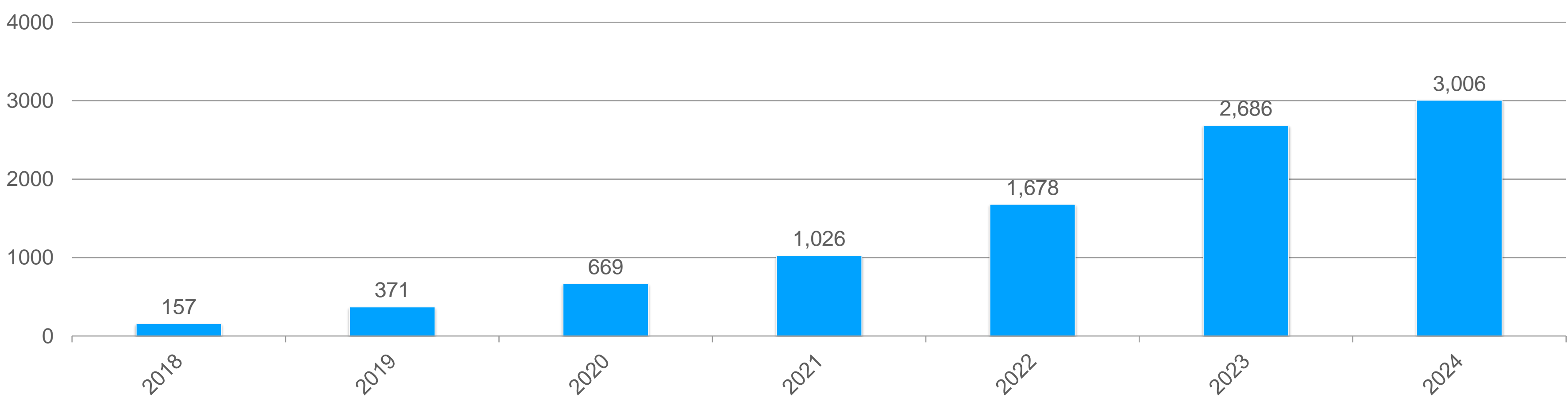


Переход Авито на микросервисную архитектуру

- В 2018 году Авито начала массовый переход от монолита к микросервисам: их количество стало быстро расти.
- Согласно паттернам МА, микросервисы не могут использовать общую базу данных (database per service).
- Микросервисы могут использовать разные технологии СУБД для решения разных задач (polyglot persistence).



Количество микросервисов в Авито

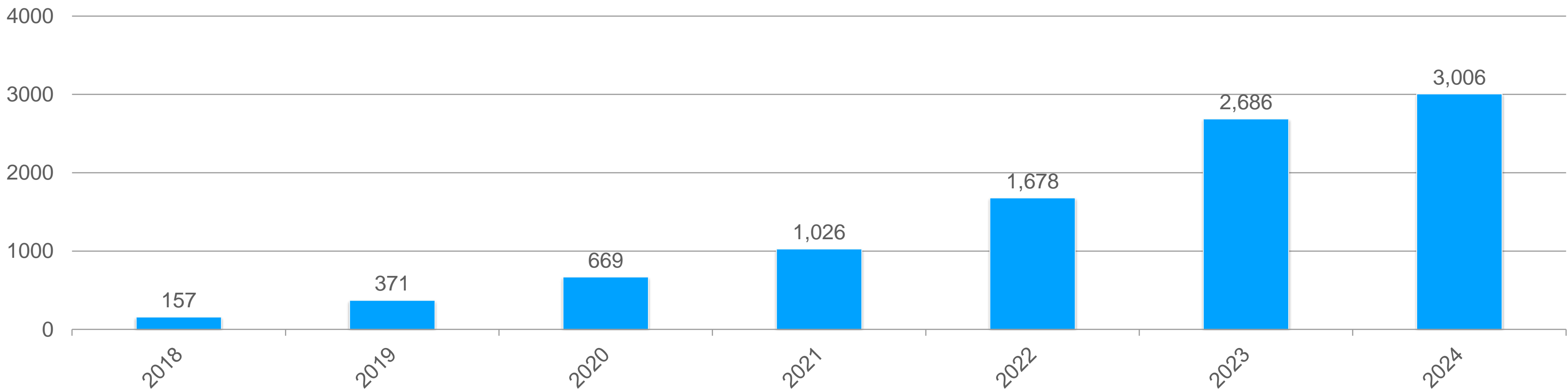


Переход Авито на микросервисную архитектуру

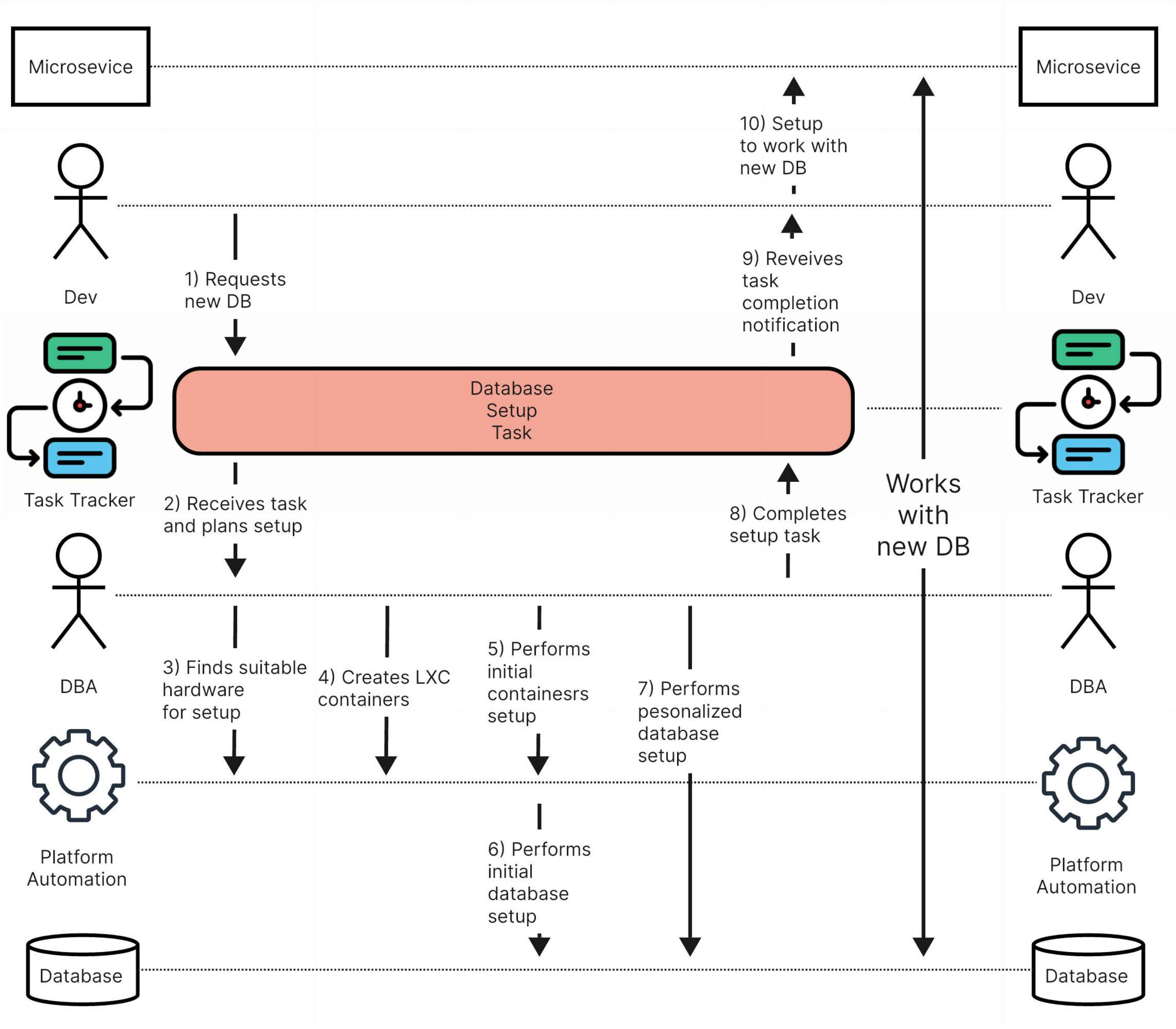
- В 2018 году Авито начала массовый переход от монолита к микросервисам: их количество стало быстро расти.
- Согласно паттернам МА, микросервисы не могут использовать общую базу данных (database per service).
- Микросервисы могут использовать разные технологии СУБД для решения разных задач (polyglot persistence).
- К 2021 году команда DBA перестала успевать обслуживать растущий спрос на развёртывание новых баз данных и эксплуатацию уже развёрнутых.



Количество микросервисов в Авито



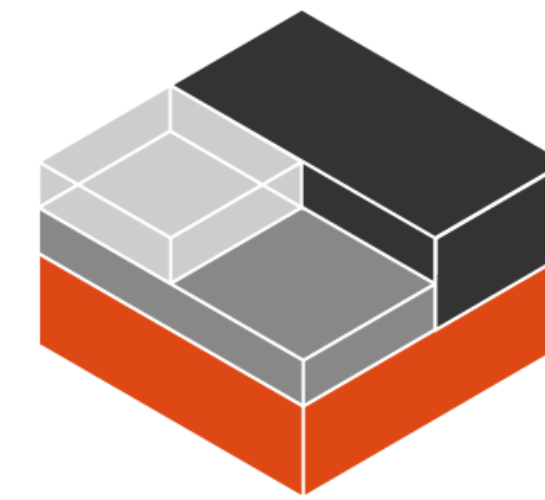
Существовавший процесс по созданию баз данных



Попытка номер раз: старая добрая классическая автоматизация

Мы автоматизировали из того, что было, и как умели, но этого оказалось недостаточно...

Мы столкнулись со следующими проблемами:



LXC

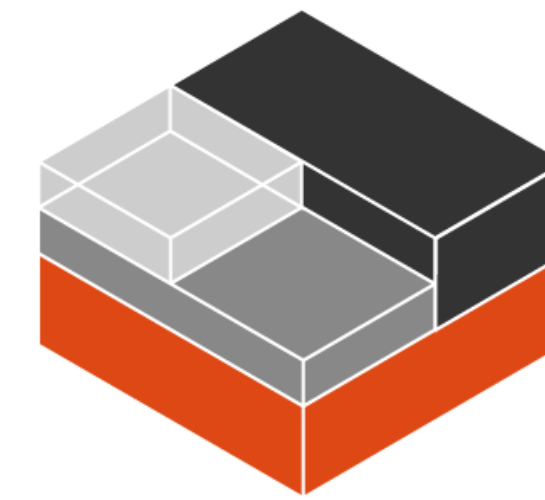


Попытка номер раз: старая добрая классическая автоматизация

Мы автоматизировали из того, что было, и как умели, но этого оказалось недостаточно...

Мы столкнулись со следующими проблемами:

- Отсутствие единого реестра баз данных.



LXC

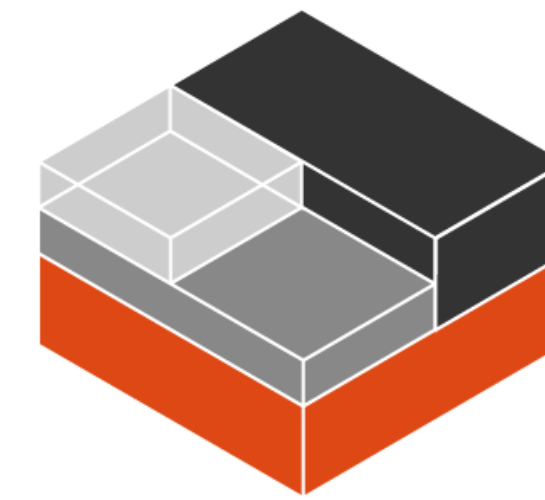


Попытка номер раз: старая добрая классическая автоматизация

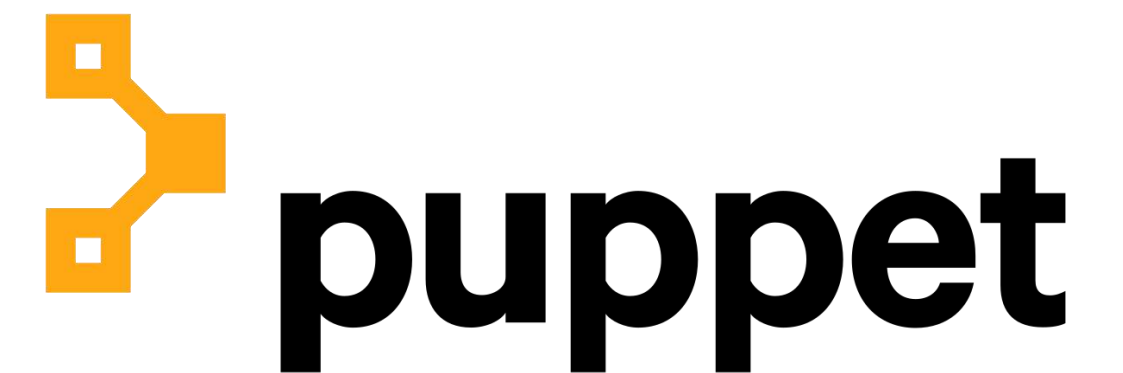
Мы автоматизировали из того, что было, и как умели, но этого оказалось недостаточно...

Мы столкнулись со следующими проблемами:

- Отсутствие единого реестра баз данных.
- Полуавтоматическое развёртывание новых баз данных и изменение конфигурации.



LXC



Попытка номер раз: старая добрая классическая автоматизация

Мы автоматизировали из того, что было, и как умели, но этого оказалось недостаточно...

Мы столкнулись со следующими проблемами:

- Отсутствие единого реестра баз данных.
- Полуавтоматическое развёртывание новых баз данных и изменение конфигурации.
- Сложное и дорогое управление и планирование на уровне bare-metal.



LXC



Попытка номер раз: старая добрая классическая автоматизация

Мы автоматизировали из того, что было, и как умели, но этого оказалось недостаточно...

Мы столкнулись со следующими проблемами:

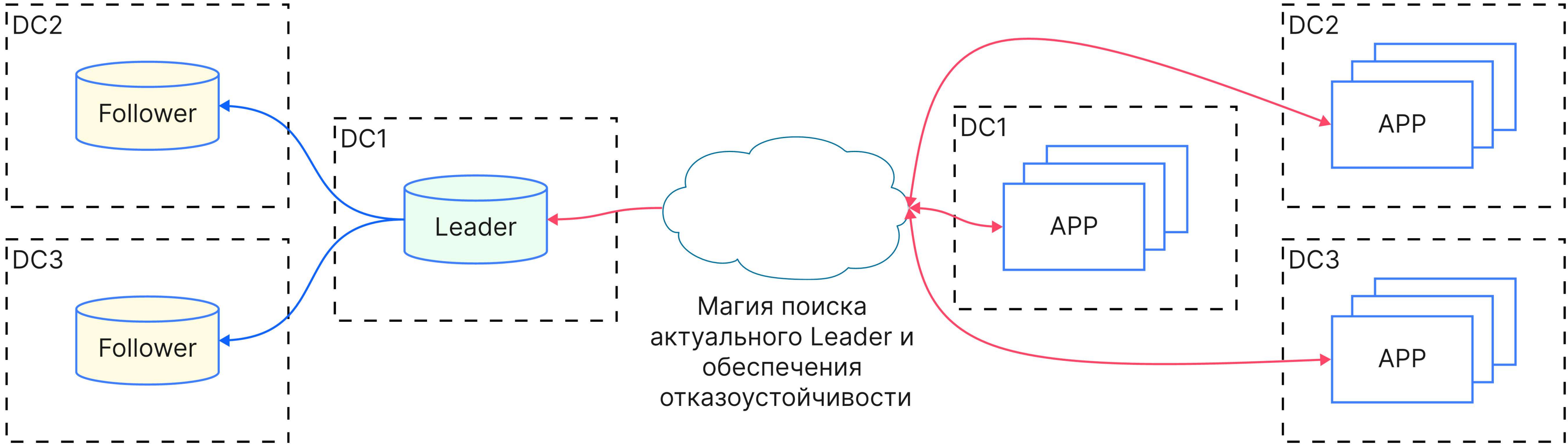
- Отсутствие единого реестра баз данных.
- Полуавтоматическое развёртывание новых баз данных и изменение конфигурации.
- Сложное и дорогое управление и планирование на уровне bare-metal.
- Гетерогенная инфраструктура в зависимости от окружения.



LXC

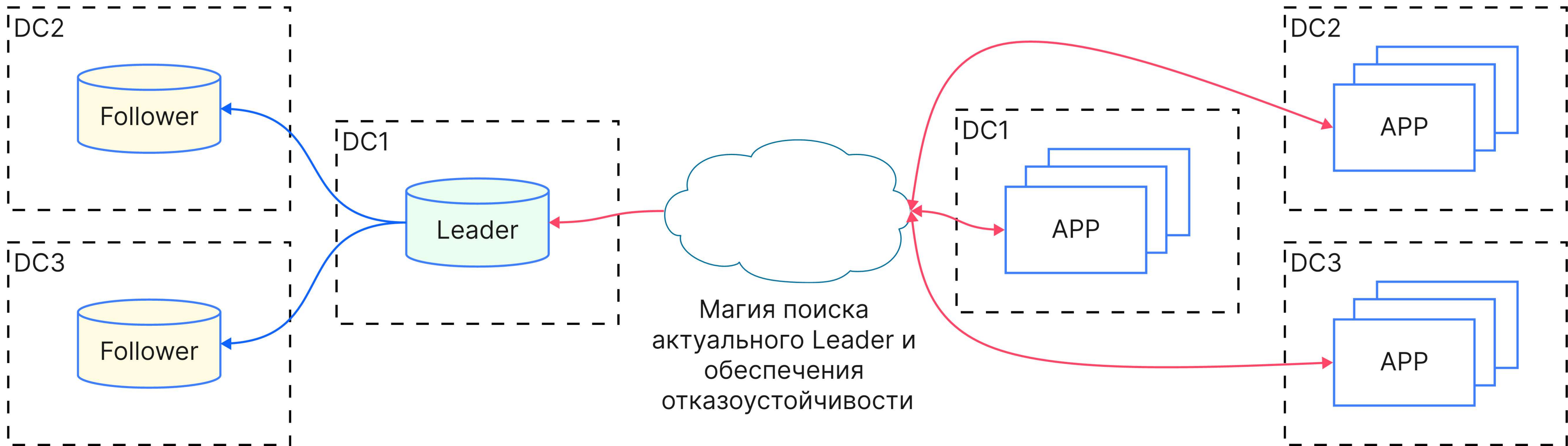


Переход Авито в multi-dc



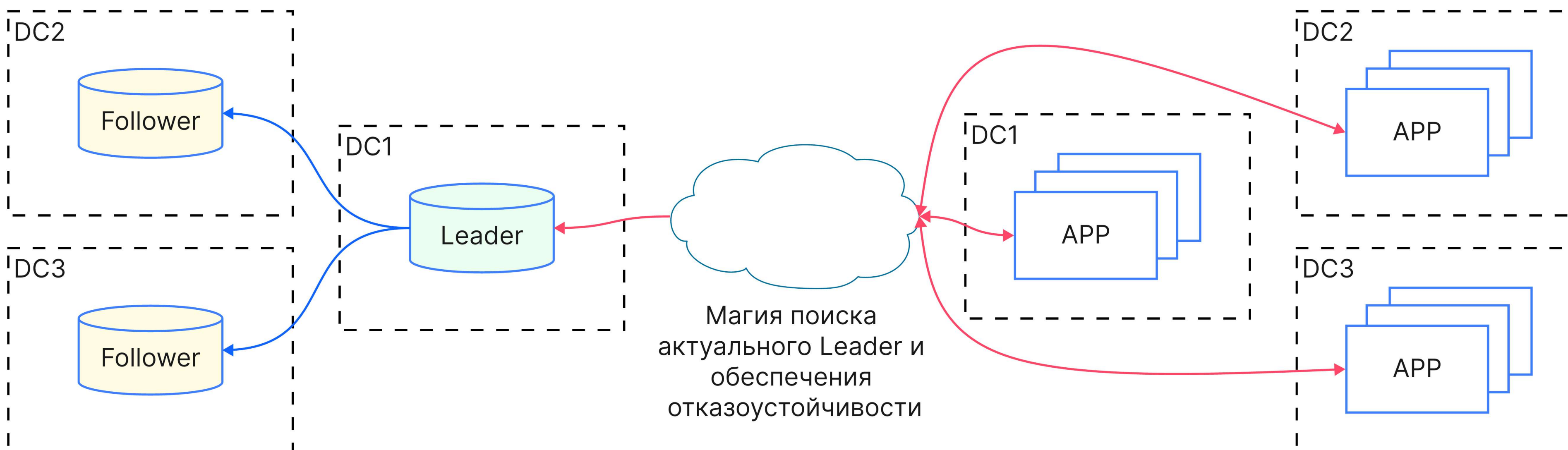
Переход Авито в multi-dc

- Управлять и планировать на уровне bare-metal стало ещё сложнее.



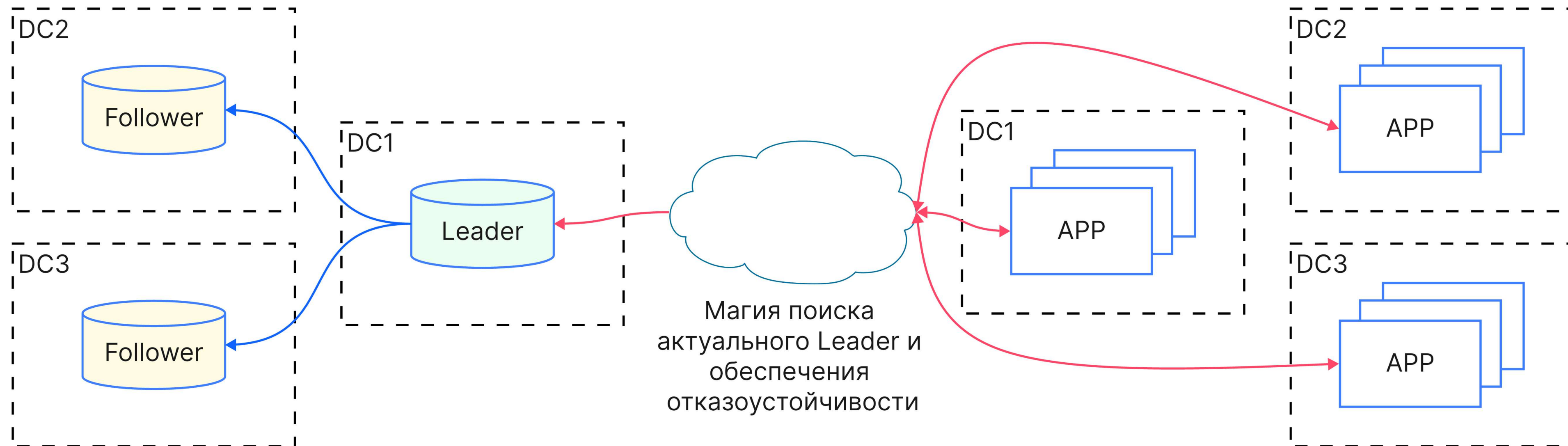
Переход Авито в multi-dc

- Управлять и планировать на уровне bare-metal стало ещё сложнее.
- Появилась необходимость обеспечивать отказоустойчивость баз данных в рамках multi-dc (основное количество баз данных пришлось на Redis, PostgreSQL и MongoDB).

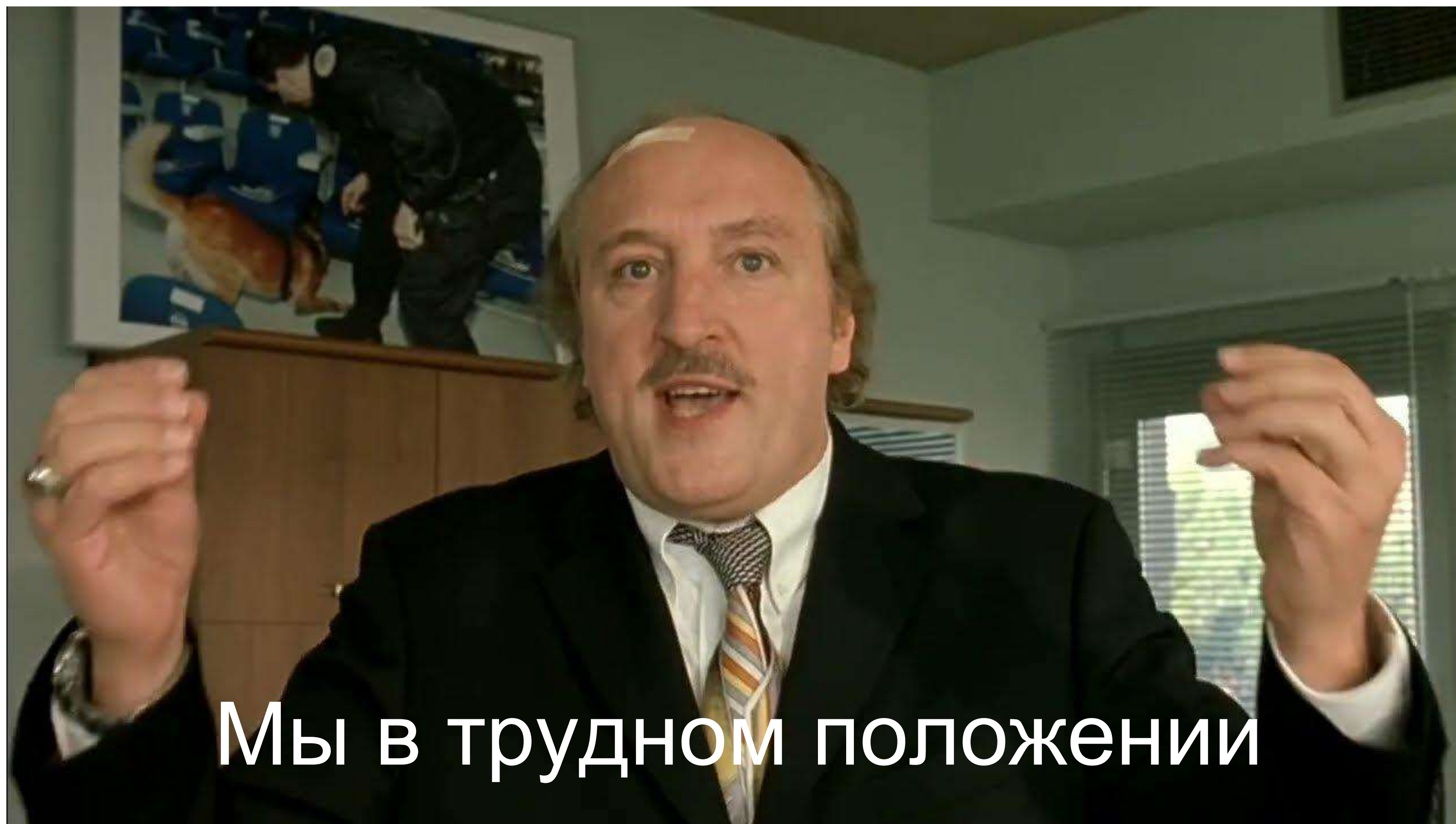


Переход Авито в multi-dc

- Управлять и планировать на уровне bare-metal стало ещё сложнее.
- Появилась необходимость обеспечивать отказоустойчивость баз данных в рамках multi-dc (основное количество баз данных пришлось на Redis, PostgreSQL и MongoDB).
- Появилась необходимость переносить базы данных между DC, а самое главное — делать это без привлечения конечных пользователей.



Осознание



Выработка целевого решения для платформы DBaaS



Требования.	26
А что там у ребят из соседних команд?	33
Какой вариант лучше?	39
Принятие решения.	40

Требования



Требования

- Единый реестр всех баз данных в Авито.



Требования

- Единый реестр всех баз данных в Авито.
- Полностью автоматическое управление базами данных: развертывание новых баз данных без участия DBA и изменение конфигурации без коммитов с автоматическим применением.



Требования

- Единый реестр всех баз данных в Авито.
- Полностью автоматическое управление базами данных: развертывание новых баз данных без участия DBA и изменение конфигурации без коммитов с автоматическим применением.
- Простое управление на уровне bare-metal, работа с железом не оказывает влияния на пользователей.



Требования

- Единый реестр всех баз данных в Авито.
- Полностью автоматическое управление базами данных: развертывание новых баз данных без участия DBA и изменение конфигурации без коммитов с автоматическим применением.
- Простое управление на уровне bare-metal, работа с железом не оказывает влияния на пользователей.
- Надёжная и производительная работа со stateful.



Требования

- Единый реестр всех баз данных в Авито.
- Полностью автоматическое управление базами данных: развертывание новых баз данных без участия DBA и изменение конфигурации без коммитов с автоматическим применением.
- Простое управление на уровне bare-metal, работа с железом не оказывает влияния на пользователей.
- Надёжная и производительная работа со stateful.
- Согласованное функциональное поведение СУБД вне зависимости от окружения, в котором она разворачивается.



Требования

- Единый реестр всех баз данных в Авито.
- Полностью автоматическое управление базами данных: развертывание новых баз данных без участия DBA и изменение конфигурации без коммитов с автоматическим применением
- Простое управление на уровне bare-metal, работа с железом не оказывает влияния на пользователей.
- Надёжная и производительная работа со stateful.
- Согласованное функциональное поведение СУБД вне зависимости от окружения, в котором она разворачивается.
- Работа в multi-dc.



А что там у ребят из соседних команд?



А что там у ребят из соседних команд?

- Коллеги из DevOps сидят ровно на таком же стеке, что и мы: CMDB, puppet, LXC.



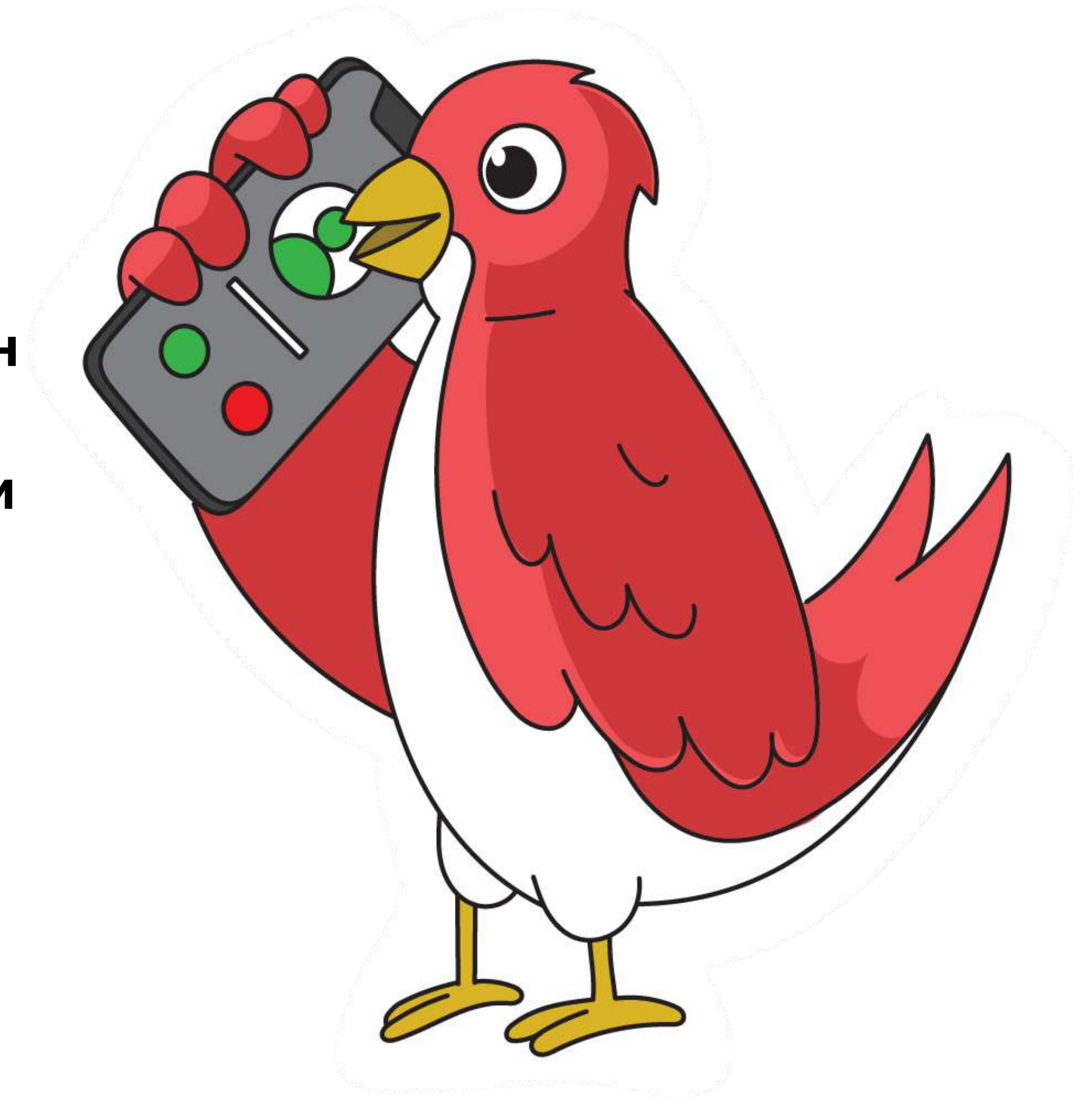
А что там у ребят из соседних команд?

- Коллеги из DevOps сидят ровно на таком же стеке, что и мы: CMDB, puppet, LXC.
- Коллеги из DevOps имеют сильно менее подвижную платформу: выдали контейнер пользователю, а дальше он сам.



А что там у ребят из соседних команд?

- Коллеги из DevOps сидят ровно на таком же стеке, что и мы: CMDB, puppet, LXC.
- Коллеги из DevOps имеют сильно менее подвижную платформу: выдали контейнер пользователю, а дальше он сам.
- Коллеги из PaaS во всю используют kubernetes и написали множество «обязок» для него.



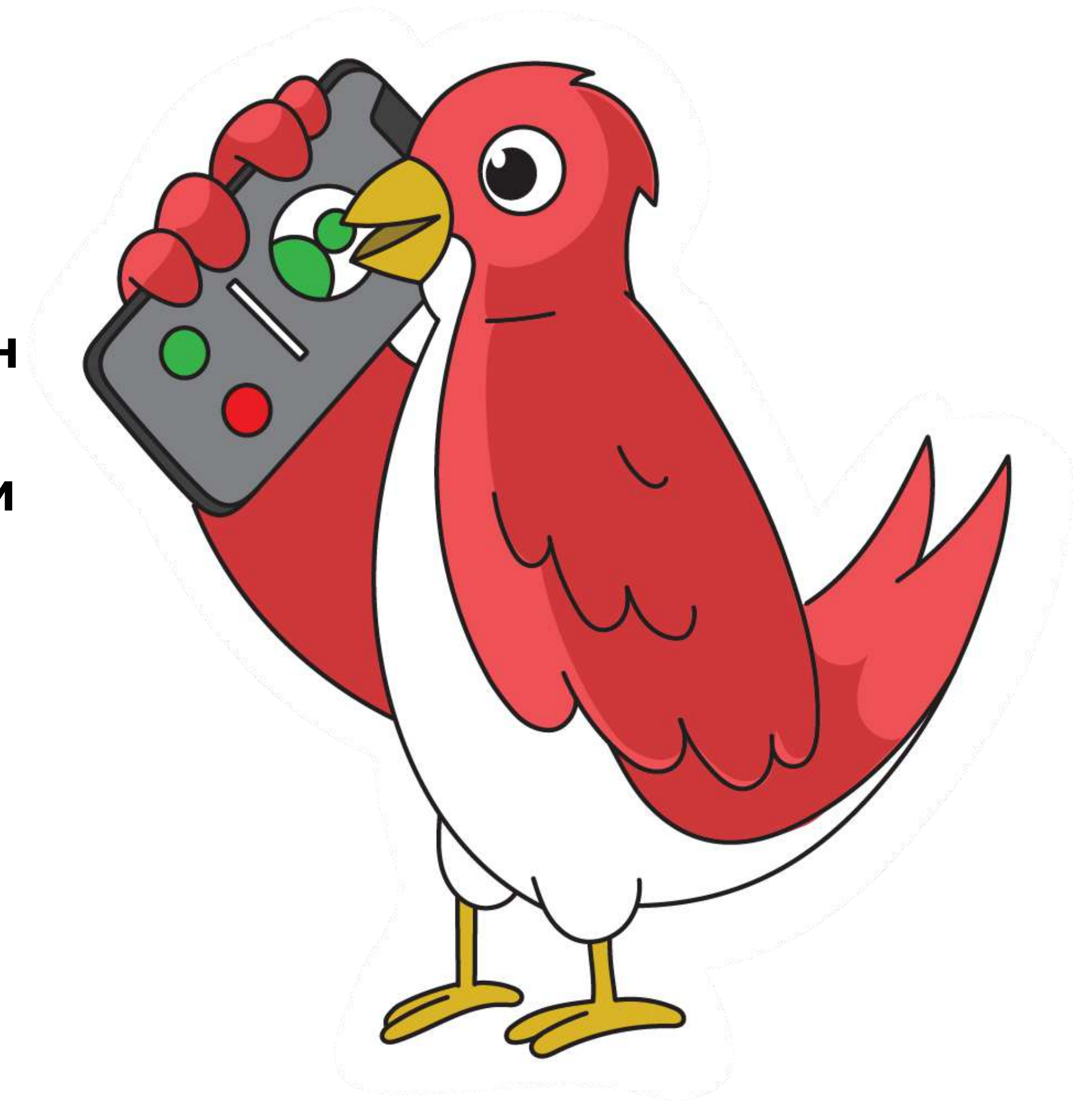
А что там у ребят из соседних команд?

- Коллеги из DevOps сидят ровно на таком же стеке, что и мы: CMDB, puppet, LXC.
- Коллеги из DevOps имеют сильно менее подвижную платформу: выдали контейнер пользователю, а дальше он сам.
- Коллеги из PaaS во всю используют kubernetes и написали множество «обвязок» для него.
- У коллег из PaaS схожий «паттерн нагрузки» и идеология с предоставлением платформы как услуги — «деплой по кнопке».



А что там у ребят из соседних команд?

- Коллеги из DevOps сидят ровно на таком же стеке, что и мы: CMDB, puppet, LXC.
- Коллеги из DevOps имеют сильно менее подвижную платформу: выдали контейнер пользователю, а дальше он сам.
- Коллеги из PaaS во всю используют kubernetes и написали множество «обвязок» для него
- У коллег из PaaS схожий «паттерн нагрузки» и идеология с предоставлением платформы как услуги — «деплой по кнопке».
- Только мы (DBA) работаем с базами данных по умолчанию, а это значит, что никто из коллег не сталкивался со всеми прелестями stateful на своих задачах.



Какой вариант лучше?

Требование	Доработка существующих решений на существующем стеке (LXC)	Разработка с использованием kubernetes (k8s)	Победитель
Единый реестр всех баз данных в Авито	Значительные доработки	Значительные доработки	—
Полностью автоматическое развёртывание новых баз данных без участия DBA по запросу от пользователя	Значительные доработки	Адаптация готовых решений	k8s
Возможность внесения изменений, в том числе в конфигурацию баз данных, без внесения изменений в систему контроля версий	Значительные доработки	Адаптация готовых решений	k8s
Полностью автоматическое применение любых изменений	Значительные доработки	Адаптация готовых решений	k8s
Согласованное функциональное поведение СУБД вне зависимости от окружения, в котором она разворачивается	Невозможно в принципе	Адаптация готовых решений	k8s
Простое управление на уровне bare-metal, работа с железом не оказывает влияния на пользователей	Значительные доработки	Адаптация готовых решений	k8s
Надежная и производительная работа со stateful	Работает из коробки	Адаптация готовых решений	LXC
Работа в multi-dc	Значительные доработки	Значительные доработки	—
Наличие готовых open-source решений	Большинство кода придётся написать самостоятельно	Адаптация готовых решений	k8s
Наличие необходимых компетенций или возможность быстро их приобрести	Уже эксперты	Есть эксперты внутри Авито	LXC

Принятие решения



Принятие решения



Прекрасное настоящее

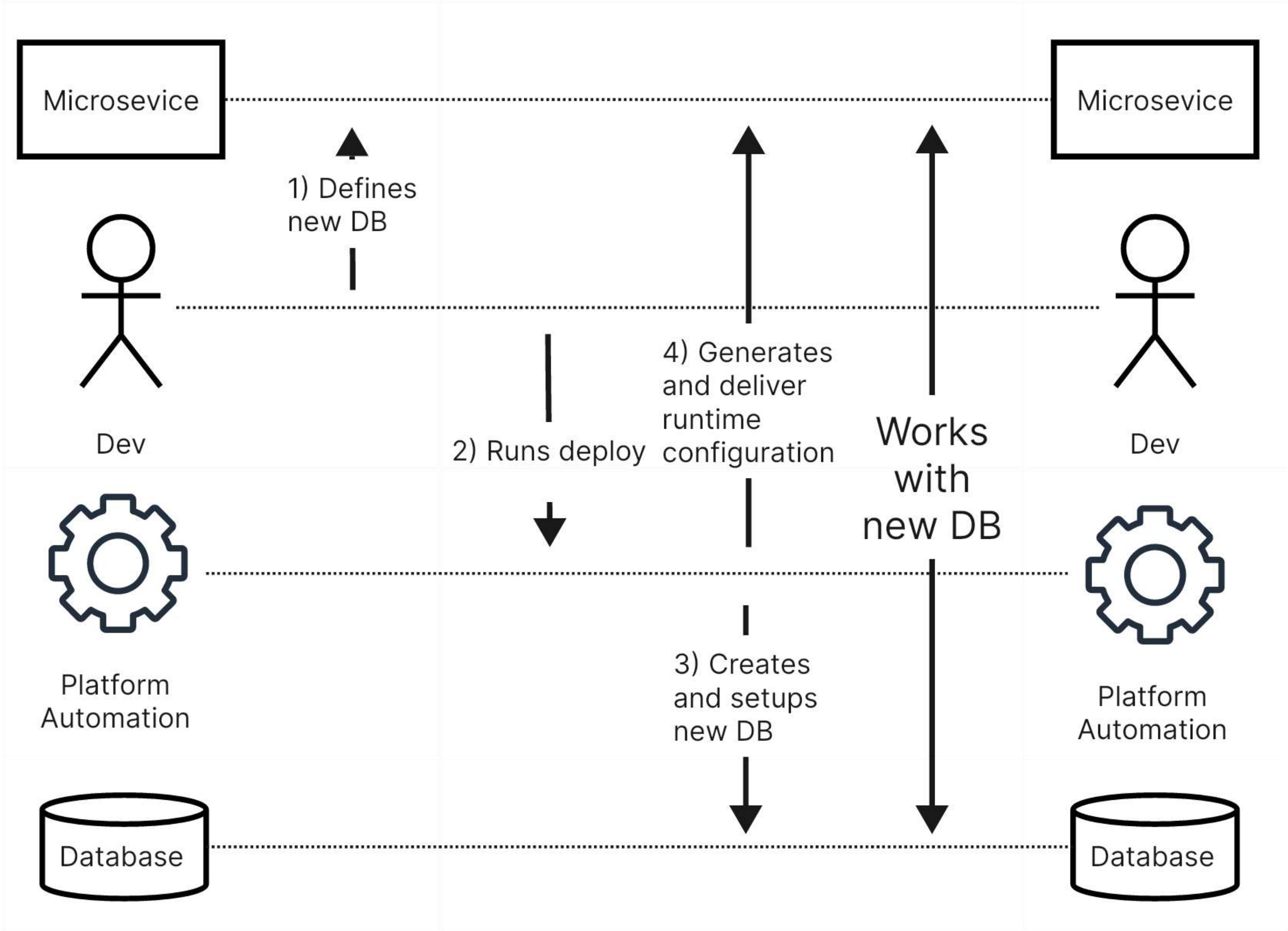


Функциональность.	43
Текущий процесс по созданию баз данных.	44
Количество баз данных, создаваемых в год, по технологиям.	45
Количество баз данных, создаваемых в год, по платформам.	46
Выводы.	47

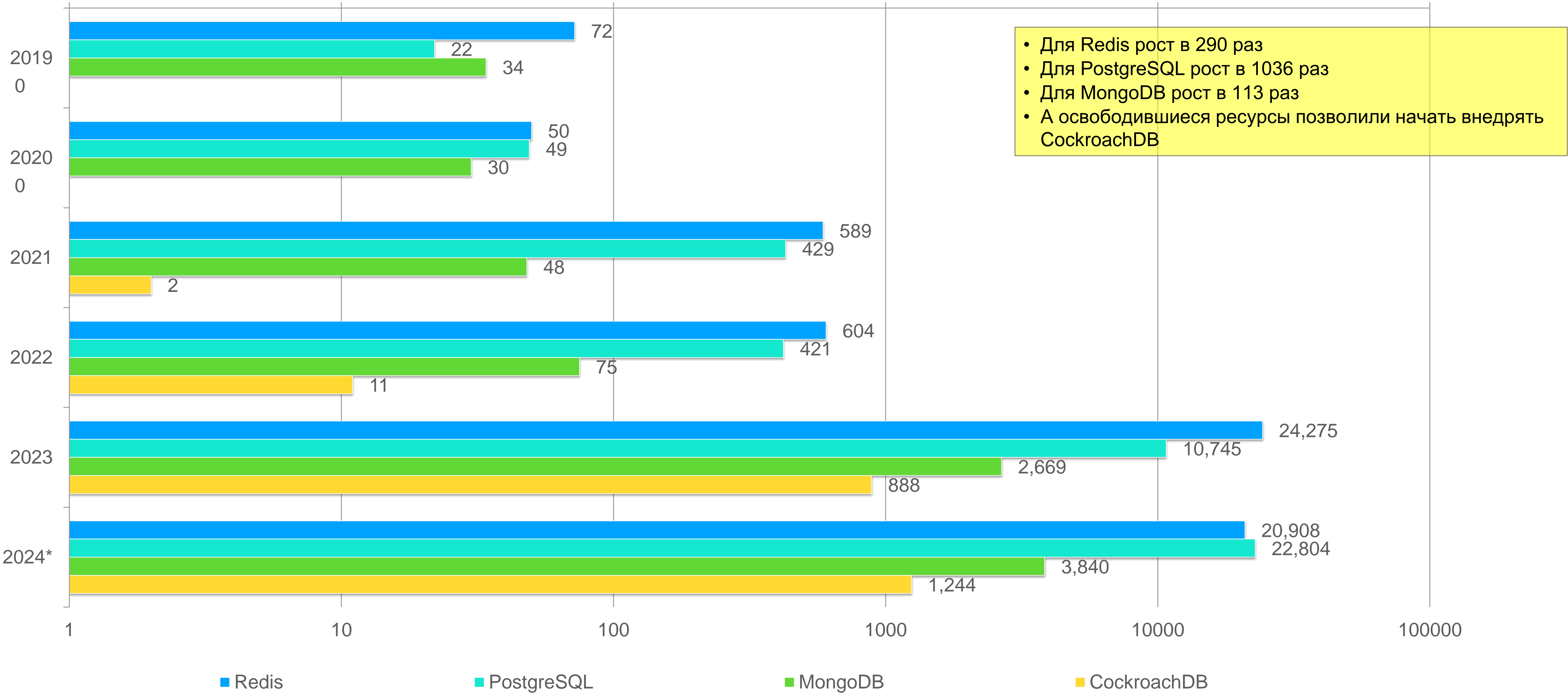
Функциональность

Критерий	LXC	k8s
Единый реестр всех баз данных в Авито	Есть и работает	Есть и работает
Работа в multi-dc	Есть и работает	Есть и работает
Надёжная и производительная работа со stateful	Есть и работает	Есть и работает
Полностью автоматическое применение любых изменений	—	Есть и работает
Простое управление на уровне bare-metal, работа с железом не оказывает влияния на пользователей	—	Есть и работает
Согласованное функциональное поведение СУБД вне зависимости от окружения, в котором она разворачивается	—	Осталось только влезть на ноутбук разработчику для некоторых типов СУБД
Ожидание пользователя для развертывания новой базы данных	3-7 рабочих дней	2-5 минут

Текущий процесс по созданию баз данных

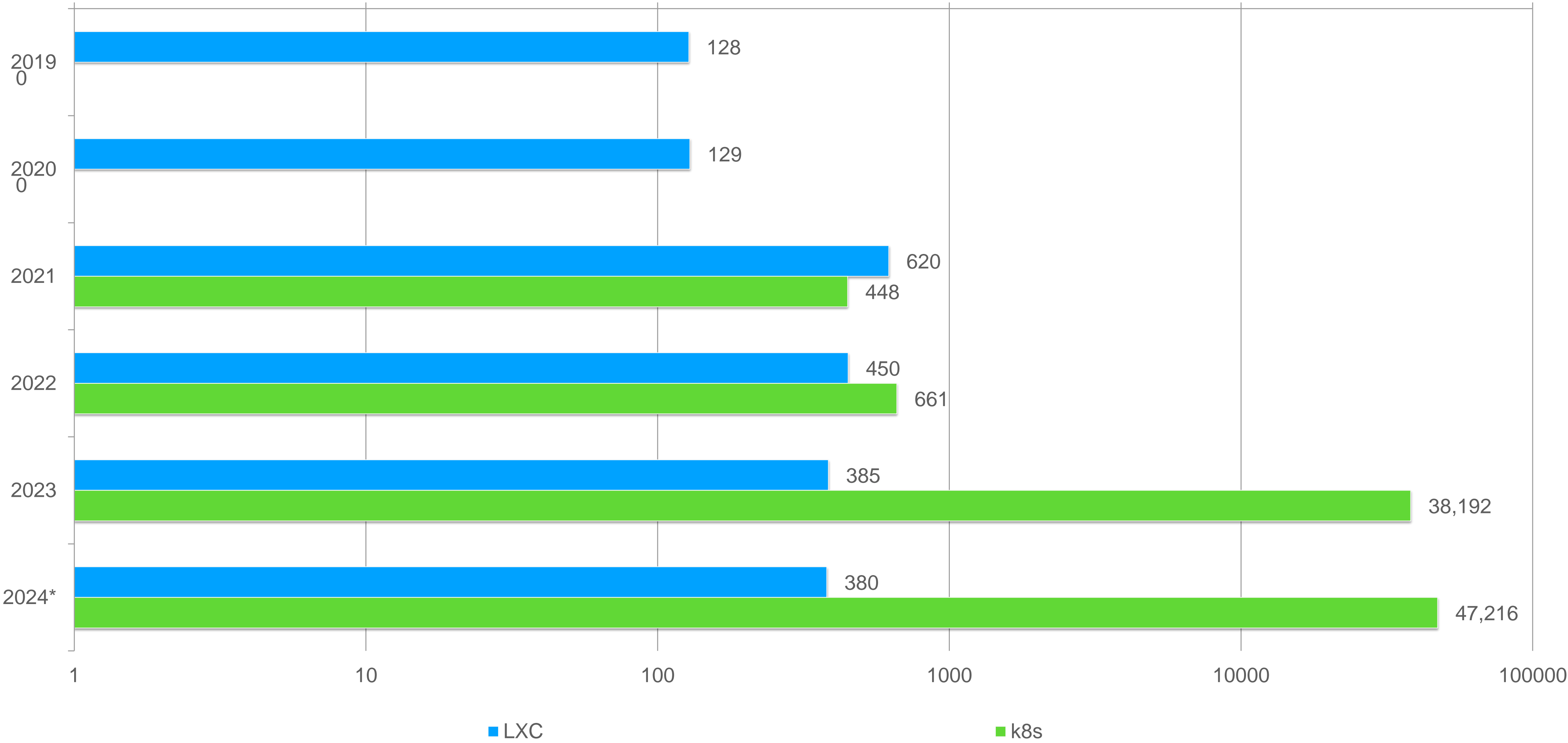


Количество баз данных, создаваемых в год, по технологиям



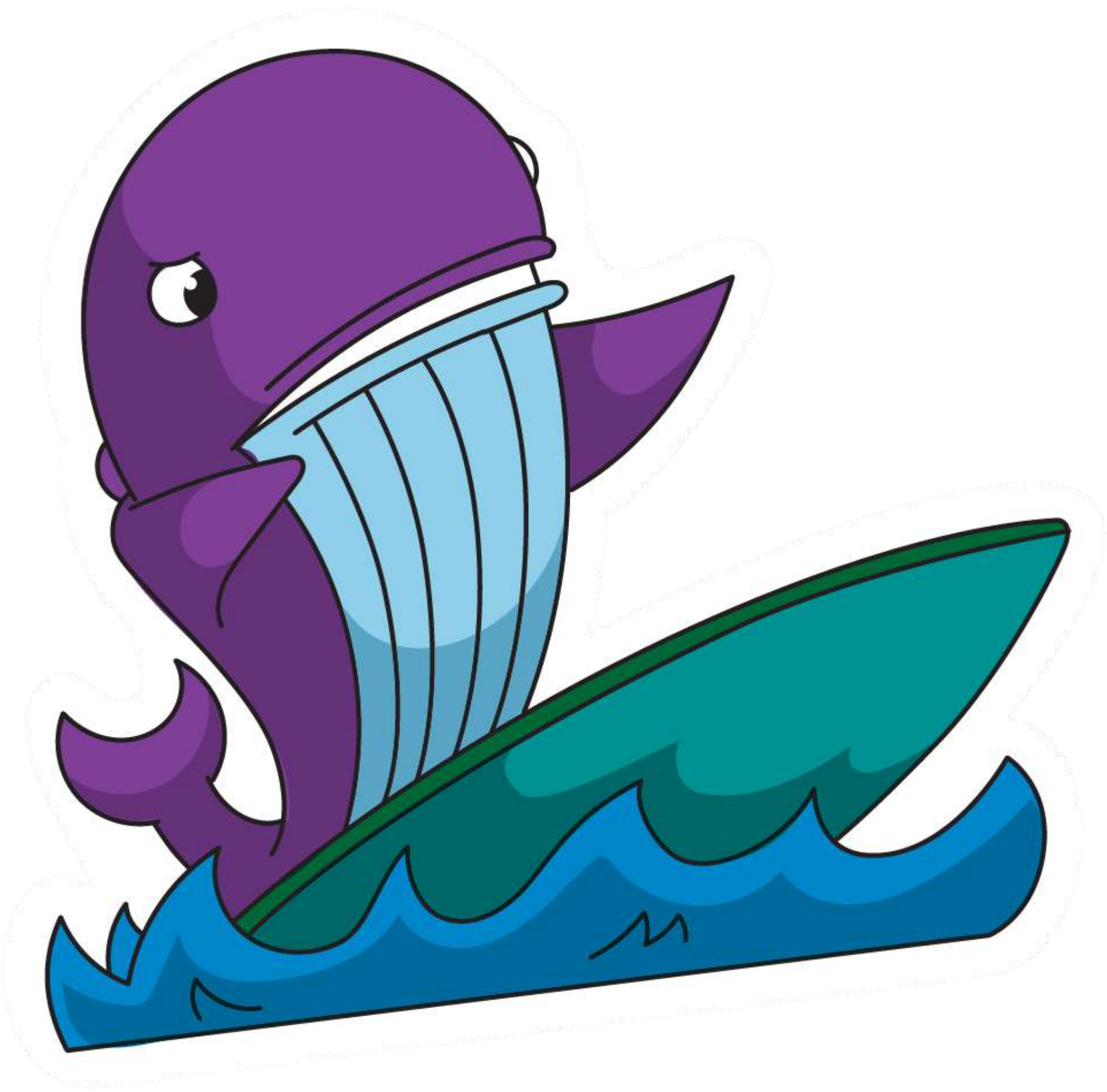
* использована линейная экстраполяция по результатам первого квартала

Количество баз данных, создаваемых в год, по платформам



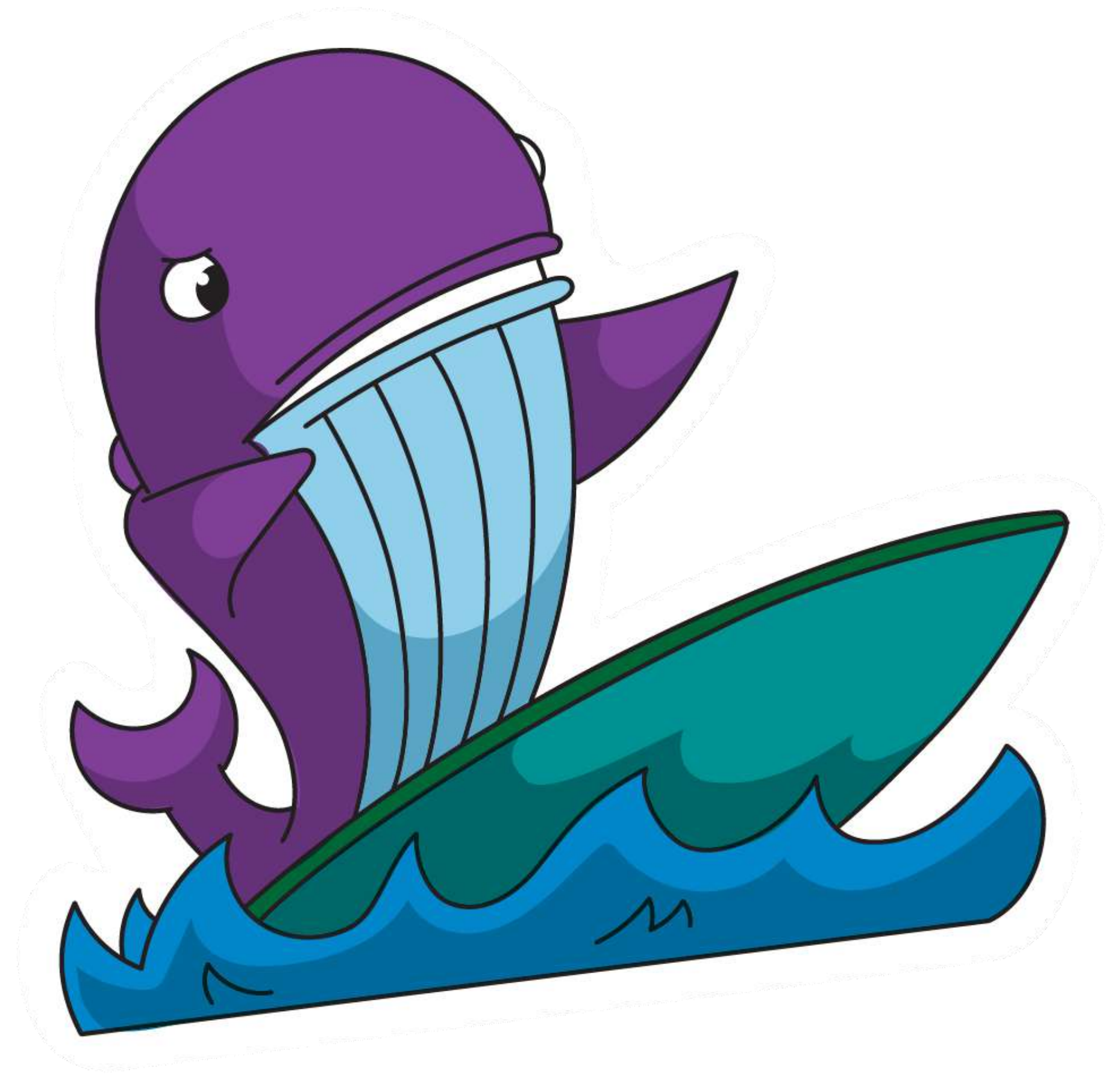
* использована линейная экстраполяция по результатам первого квартала

Выводы



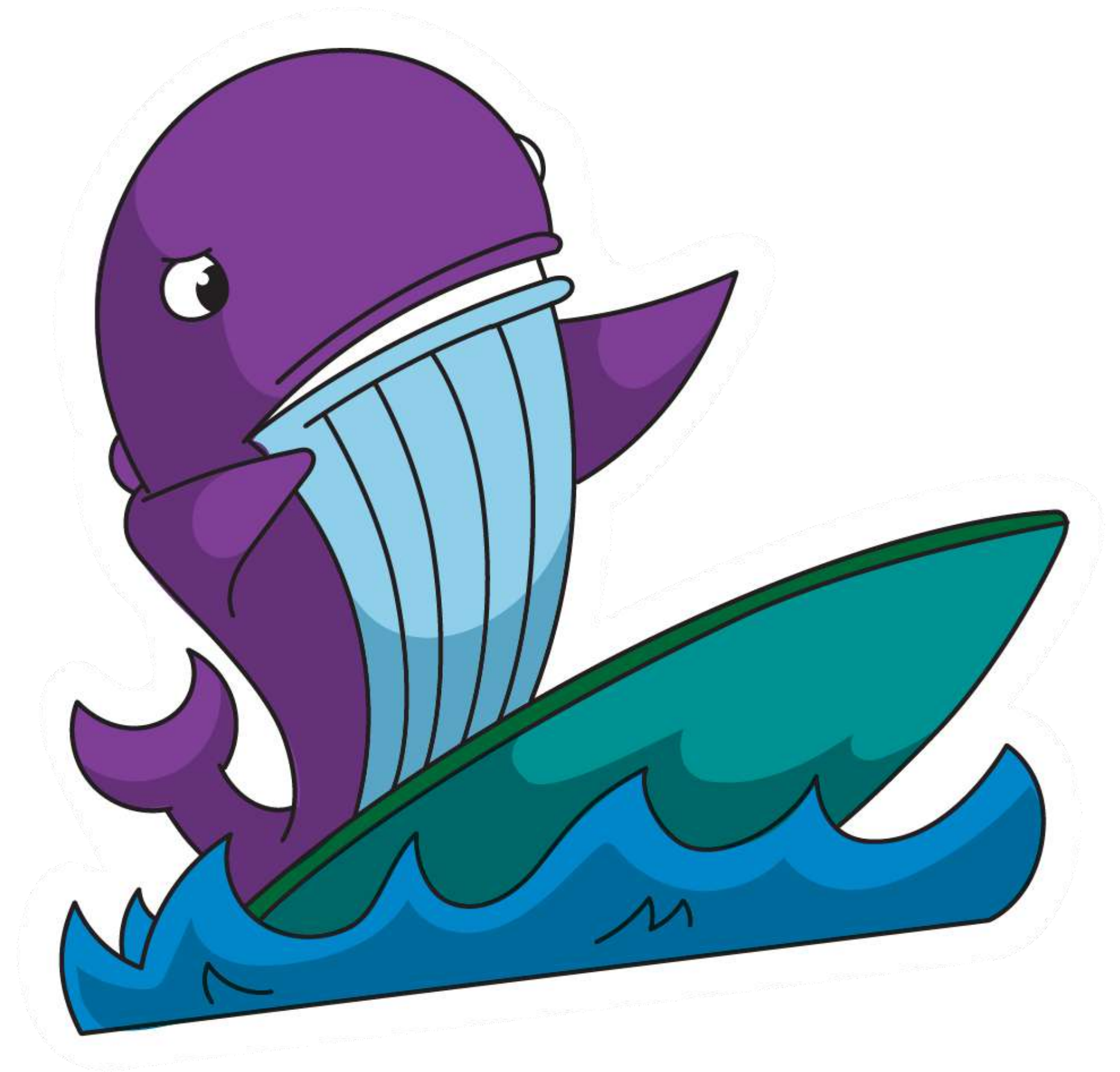
Выводы

- У МА есть неочевидные последствия.



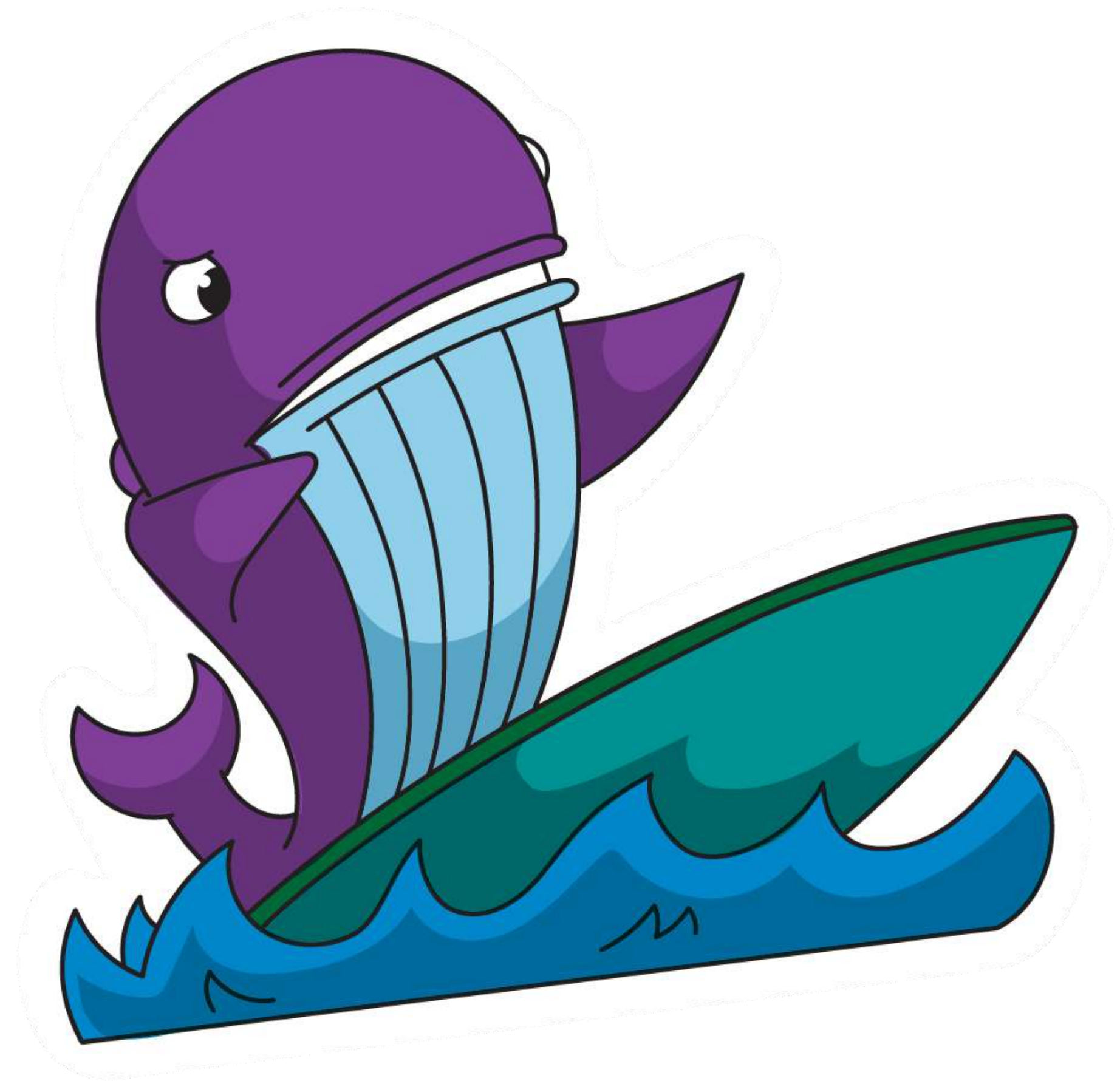
Выводы

- У МА есть неочевидные последствия.
- Инфраструктура должна не мешать бизнесу расти, а помогать трансформироваться в платформу. Автоматизация должна быть полной.



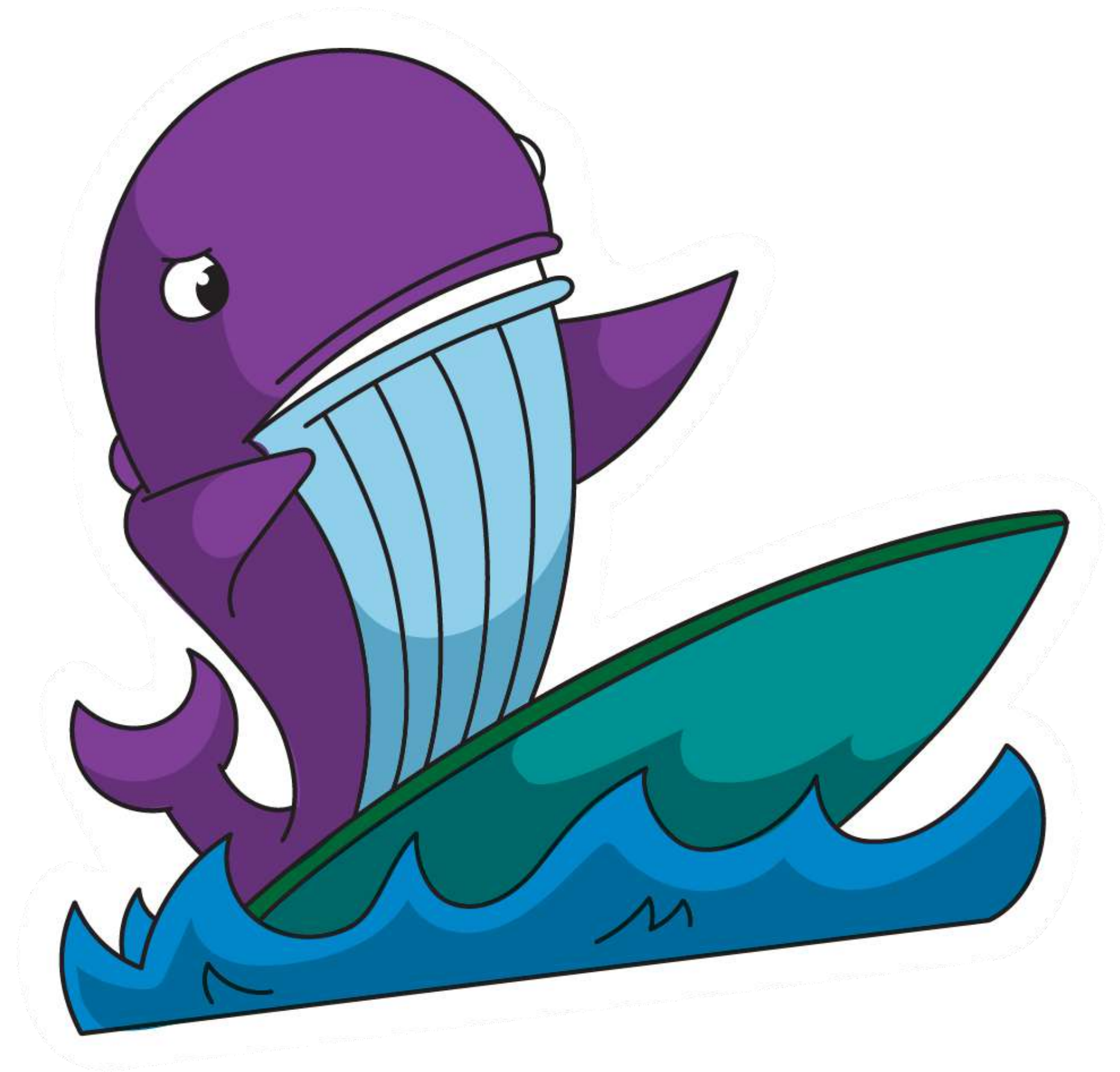
Выводы

- У МА есть неочевидные последствия.
- Инфраструктура должна не мешать бизнесу расти, а помогать трансформироваться в платформу. Автоматизация должна быть полной.
- Не всегда есть возможность начать жизнь с «чистого листа», поэтому может быть нужно какое-то время посидеть на двух стульях.



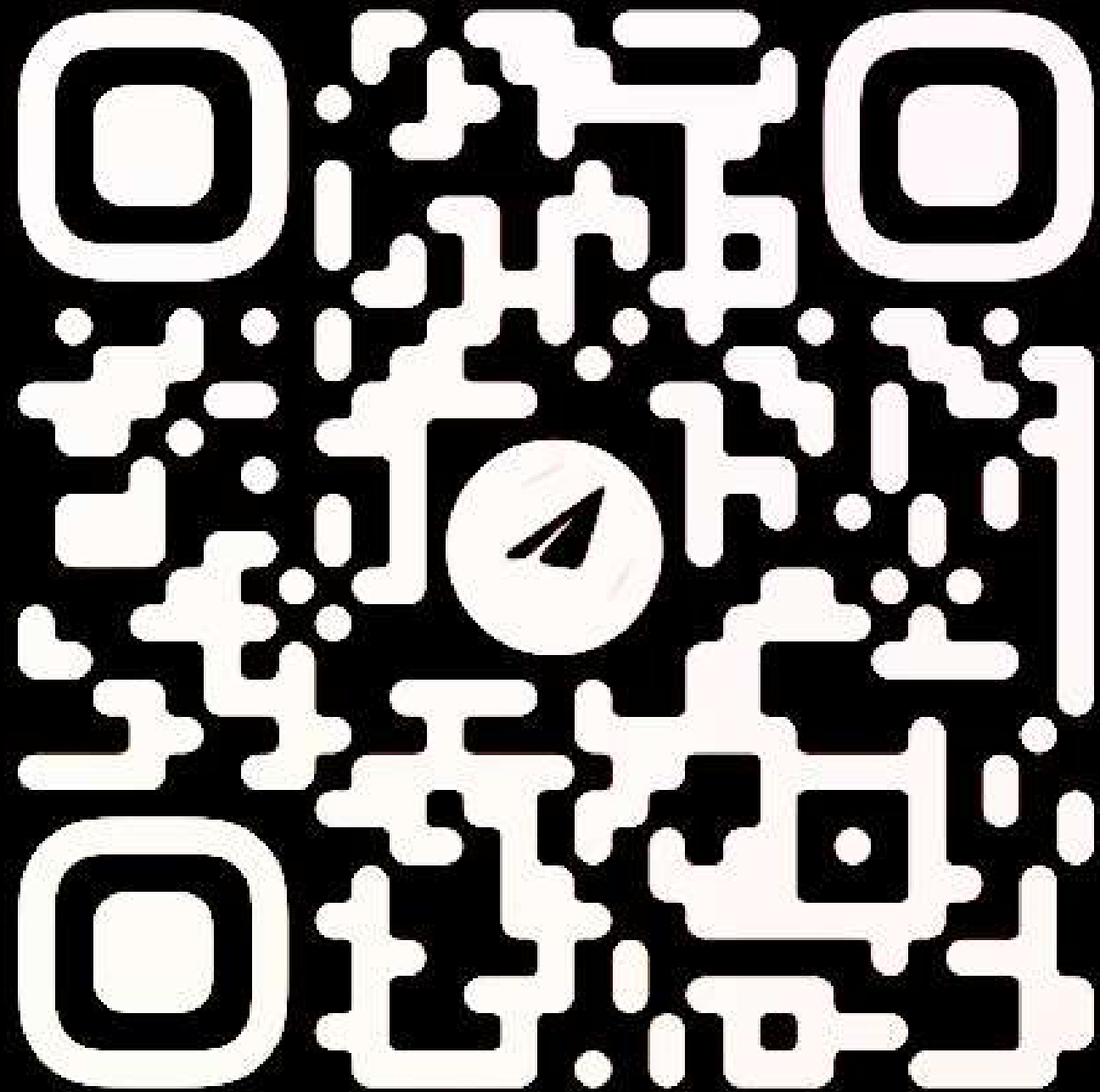
Выводы

- У МА есть неочевидные последствия.
- Инфраструктура должна не мешать бизнесу расти, а помогать трансформироваться в платформу. Автоматизация должна быть полной.
- Не всегда есть возможность начать жизнь с «чистого листа», поэтому может быть нужно какое-то время посидеть на двух стульях.
- Kubernetes — отличная и неплохо стандартизированная платформа, которая решает массу задач эксплуатации, в том числе и баз данных.



Владимир Алёшин

Руководитель разработки платформы DBaaS



“If a human operator needs to touch your system during normal operations, you have a bug. The definition of normal changes as your systems grow.”

Carla Geisser, Google SRE
Site Reliability Engineering: How Google Runs Production Systems,
Chapter 5 - Eliminating Toil