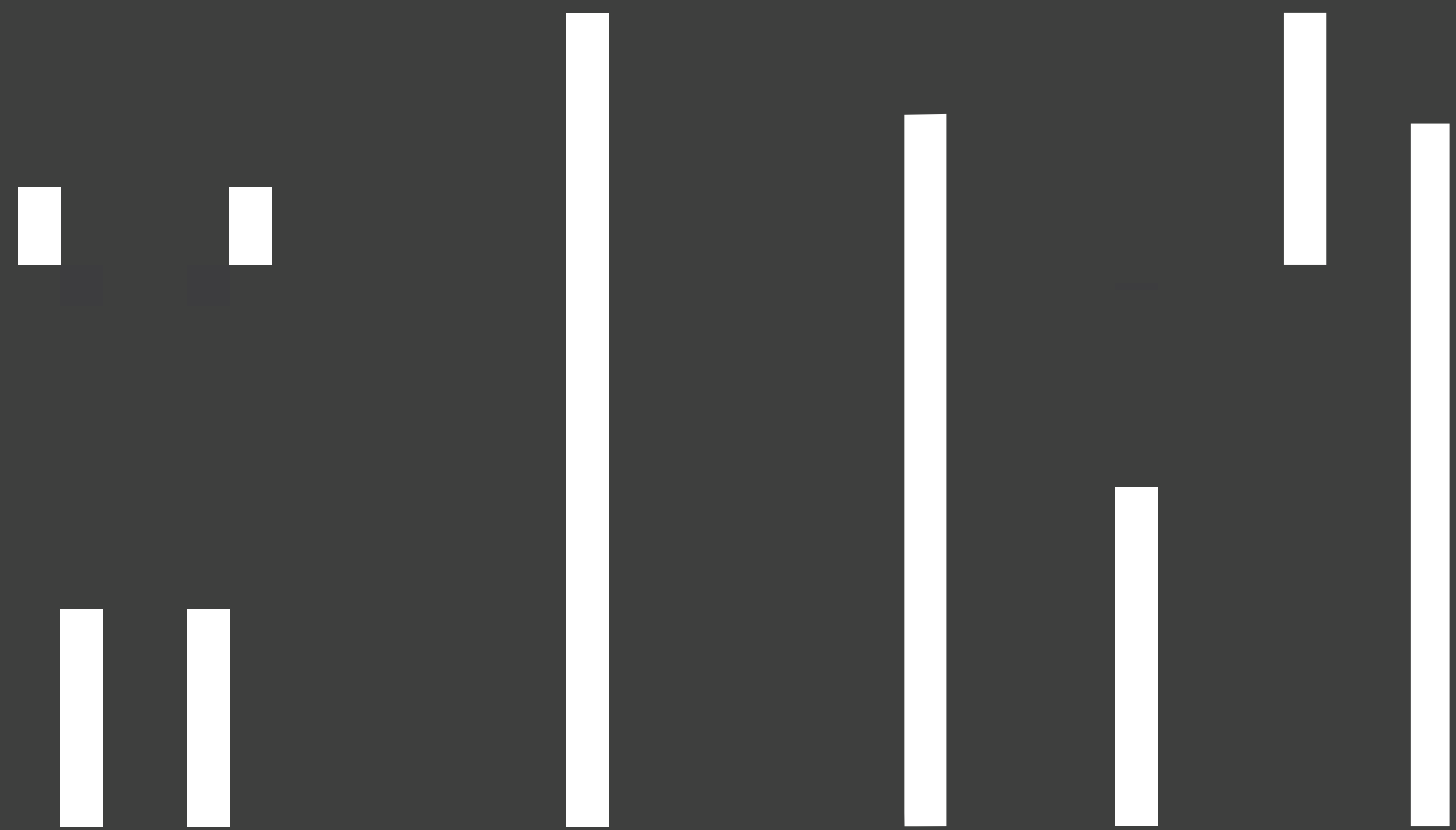


# Выбор архитектуры

## Monolith vs Microservices

Константин Труфанов  
технический директор  
Sparklingtide



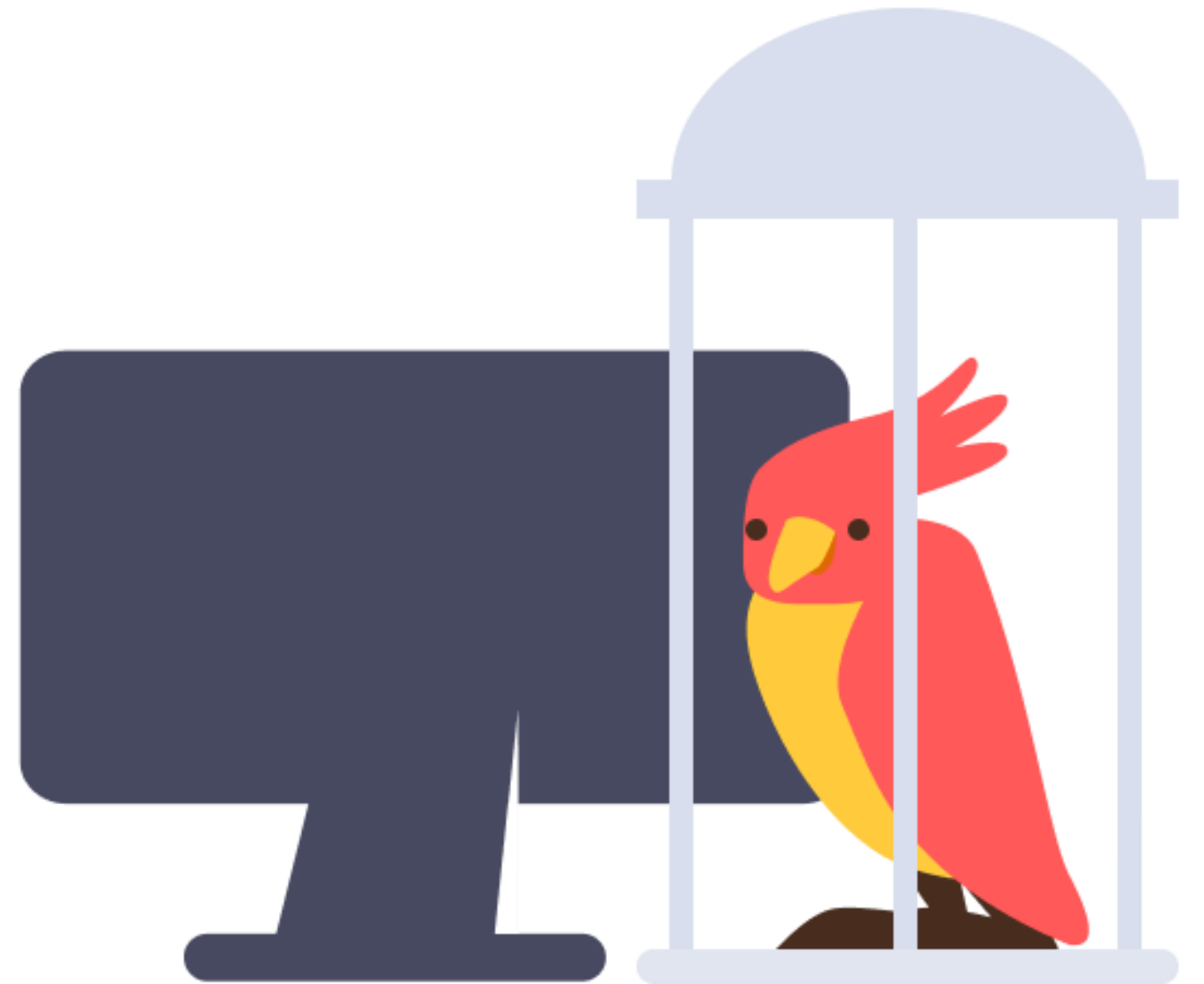
# Кто мы?

Профессионально занимаемся:

- Трансформацией процессов разработки
- Трансформацией legacy информационных систем
- Внедрением DevOps-практик в крупные российские и зарубежные компании



Шок! Попугай научился говорить  
«Зависит от контекста» и получил  
повышение до архитектора



# Метрики результата

Стоимость разработки и развертывания

Монолит

Скорость разработки и развертывания

Монолит

Стоимость поддержки

Монолит

Гетерогенный подход

Микросервисы

Горизонтальное масштабирование

Микросервисы

Предсказуемость затрат на будущие изменения

Микросервисы

# Метрики результата

Стоимость разработки и развертывания

Монолит

Скорость разработки и развертывания

Монолит

Стоимость поддержки

Монолит

Гетерогенный подход

Микросервисы

Горизонтальное масштабирование

Микросервисы

Предсказуемость затрат на будущие изменения

Микросервисы

# Предпосылки для миграции

- Большая команда разработки
- Потребность в масштабировании
- Разнообразии технических ограничений
- Высокая сложность
- Запрос на рефакторинг
- Высокие риски при изменениях в системе



# Стратегия миграции: **Strangler Fig**

01

Выделить часть монолита для миграции

02

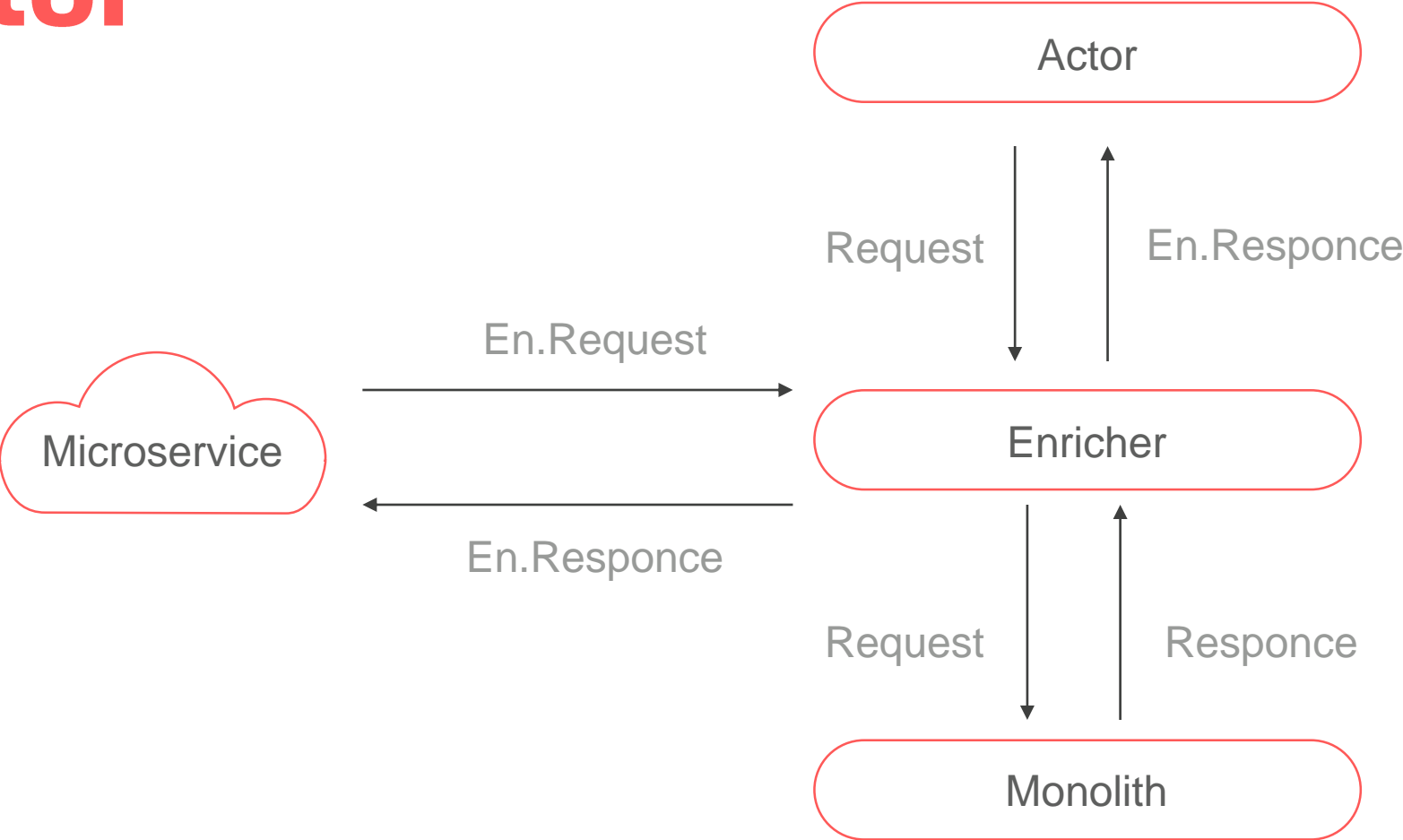
Разработать микросервис, реализующий нужную функциональность

03

Перенаправить вызовы замененных частей монолита на новый микросервис

# Стратегия миграции: Decorating collaborator

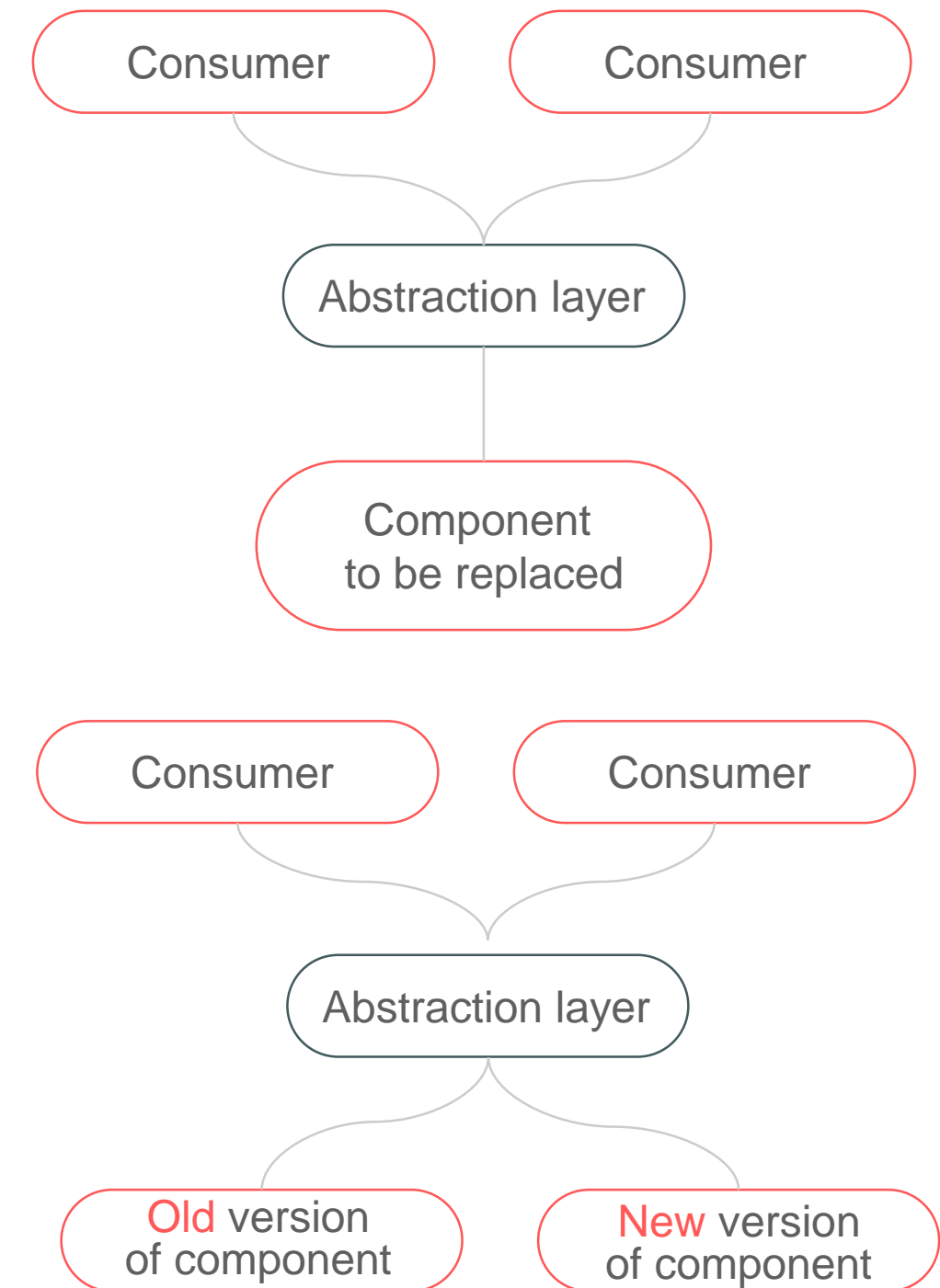
- 01 Разработать микросервис, выполняющий новую обработку
- 02 Выполнять перехват запросов в монолит и ответов из него
- 03 Извлекать нужные данные из запроса и ответа
- 04 Обогащать ответ от монолита обработкой в новом микросервисе





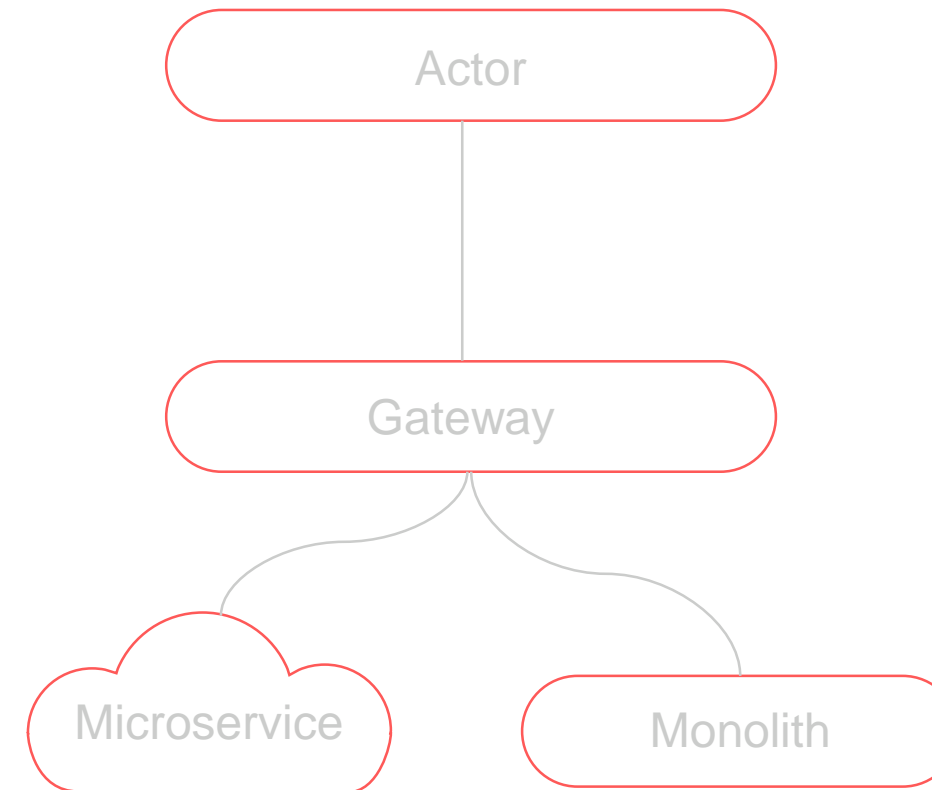
# Стратегия миграции: Branch by abstraction

- 01 Создать абстракцию для заменяемой функциональности
- 02 Изменить клиентов существующей функциональности так, чтобы они использовали новую абстракцию
- 03 Создать новую реализацию абстракции, которая будет вызывать новый микросервис вместо старой части монолита
- 04 Переключиться на новую реализацию
- 05 Удалить старую реализацию и, возможно, устранить абстракцию



# Стратегия миграции: Parallel processing/development

- 01 Разработать новую реализацию существующей функциональности
- 02 При поступлении вызова направлять его на обе реализации — старую и новую
- 03 Возвращать в ответе только один результат — от одной из реализаций
- 04 Верифицировать результаты новой реализации
- 05 Вывести старую реализацию из эксплуатации



# Что учитывать при оценке

- Требования бизнеса
- Готовность инфраструктуры
- Готовность команды
- Доступные ресурсы на миграцию



# Что почитать?

<https://martinfowler.com/microservices/>

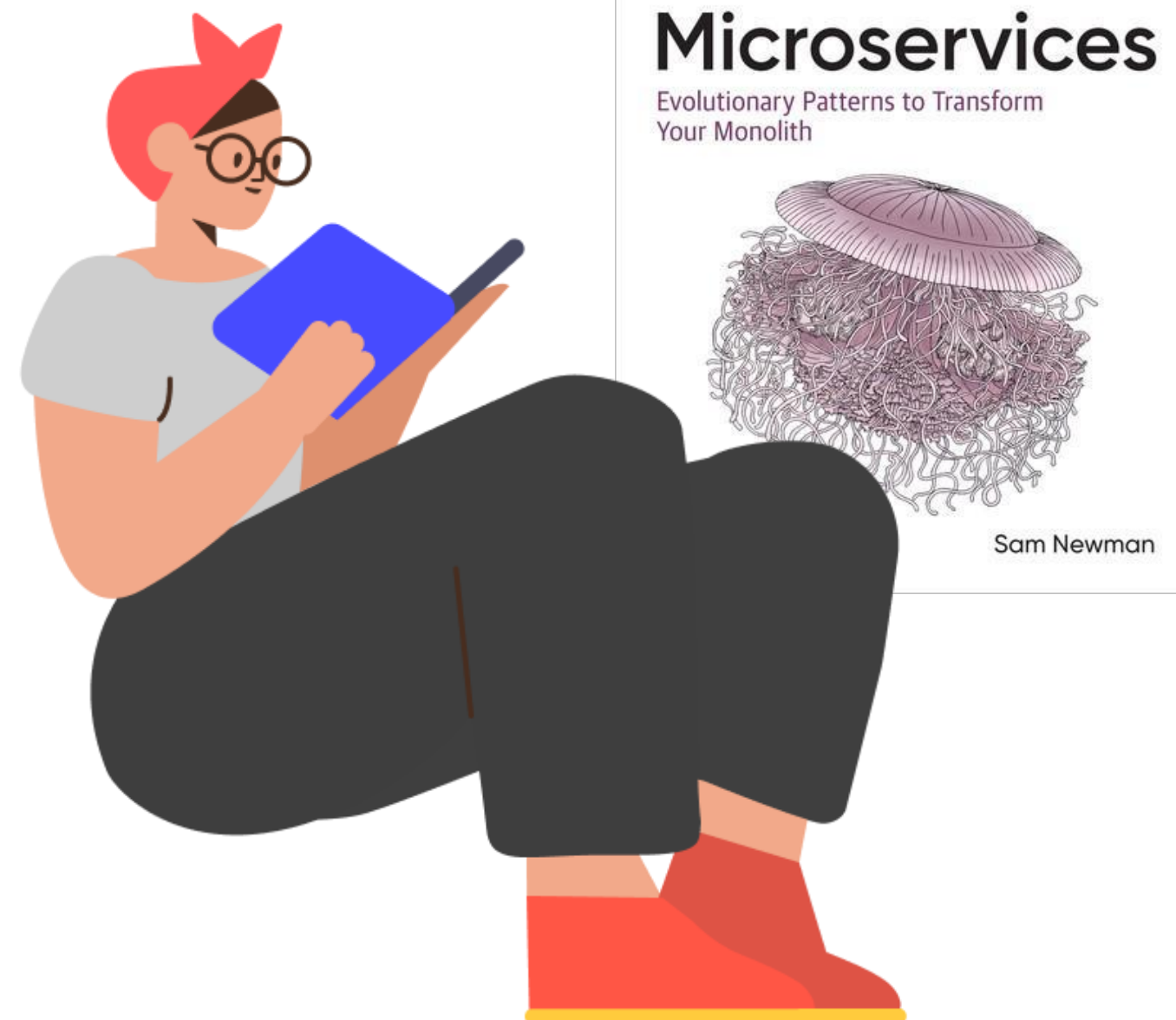
Много полезной информации, которую собрал в одном месте  
Мартин Фаулер

<https://samnewman.io/books/monolith-to-microservices/>

Sam Newman. Monolith to Microservices

<https://learning.oreilly.com/>

Платформа для обучения с подпиской. Она открывает доступ не только ко всем книгам издательства, но и к очень полезным лекциям и подкастам от таких звезд индустрии, как Мартин Фаулер и Роберт Мартин



**Спасибо!**

