Тестирование

Что такое тестирование?

Тестирование - проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов

Зачем тестирование?

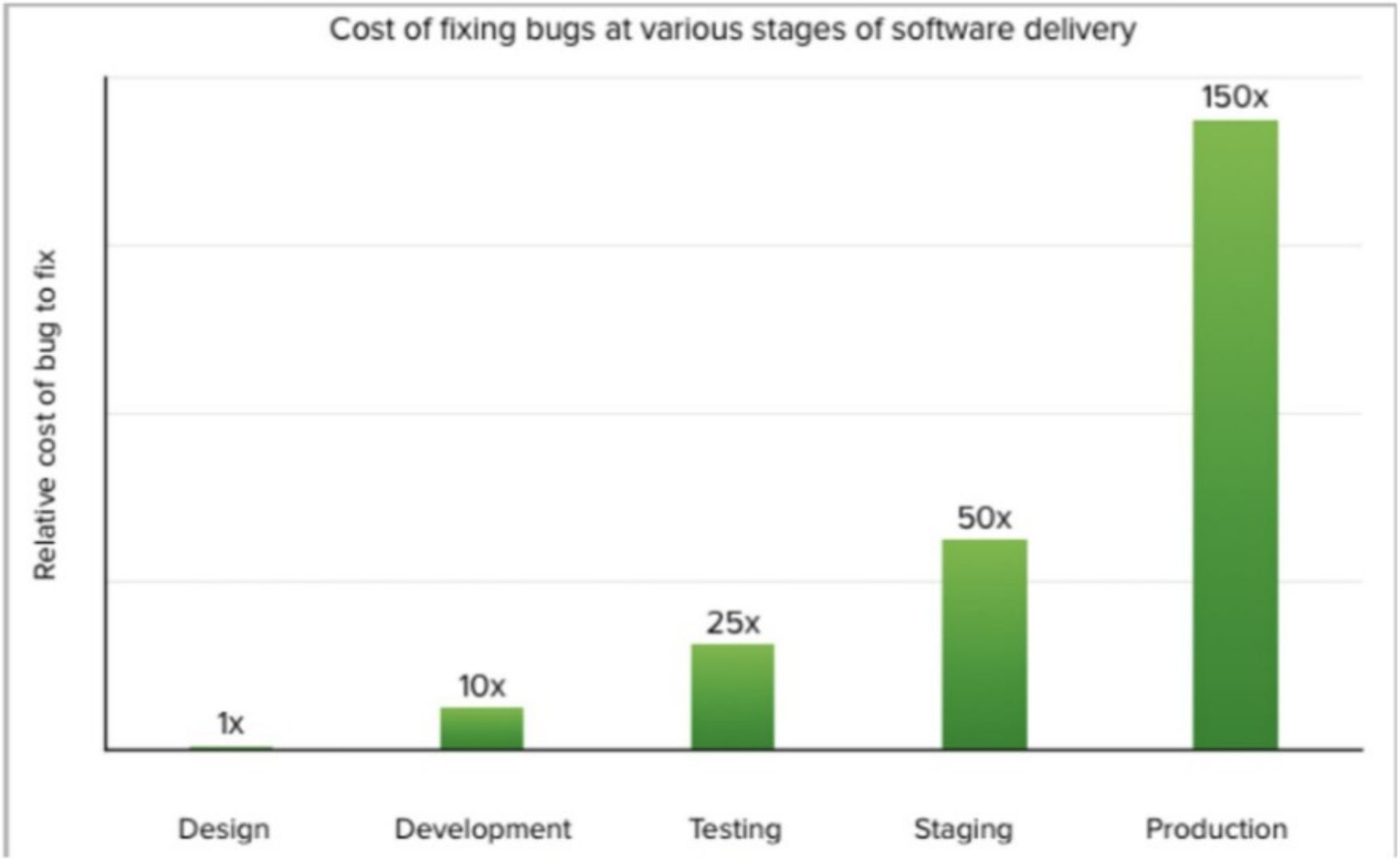


График Боэма

Место тестирования в процессе обеспечения качества

Виды тестирования в зависимости от доступа к коду

- White Box
- Black Box
- Gray box

По объекту тестирования

- Модульное/unit-тестирование (класс/функция)
- Интеграционное (взаимодействие модулей например model + view + template)
- Системное (end-to-end) frontend + backend; ручное/автоматическое

По типу тестируемых характеристик

- → Функциональное
- → Нагрузочное/Стресс/Производительности
- → Безопасности
- → Удобства использования (UI тесты)

О каком тестировании будем говорить?

White Box

Модульное/Интеграционое

Функциональное

Mentimeter

Что дает такое тестирование?

- Проверяем, что старое поведение не изменилось после добавления нового кода
- Легко проводить рефакторинг/миграцию библиотек
- Уверенность в поведении базового функционала
- При написании тестов больше задумываешься об отказоустойчивости
- Не дергаем зря тестировщиков

Принципы автоматизации тестов

- Атомарность
- Независимость
- Изолированность / герметичность

Mentimeter

```
1. def test_add_smth():
2.     user = create_new_user(email='some@ema.il')
3.     user.register()
4.     user.auth()
5.     smth = user.create_smth()
6.     smth.add()
7.     check(user.is_authorized())
8.     check(smth.is_added())
9.     check('some@ema.il' == user.email)
```

Атомарность отсутствует

Mentimeter

```
1. def test_add_smth():
2.     user = create_new_user(email='some@ema.il')
3.     user.register()
4.     user.auth()
5.     smth = user.create_smth()
6.     smth.add()
7.     check(user.is_authorized())
8.     check(smth.is_added())
9.     check('some@ema.il' == user.email)
```

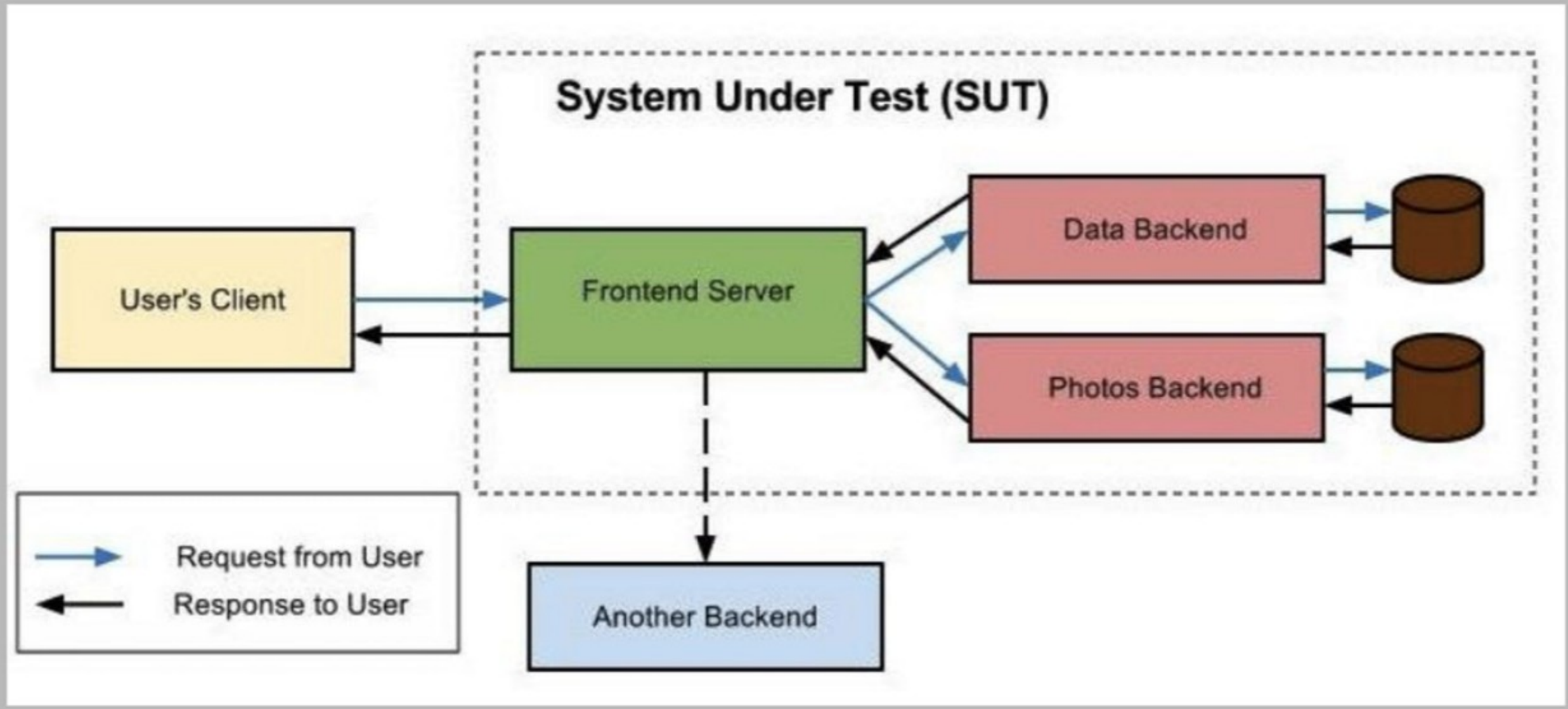
Что лишнее

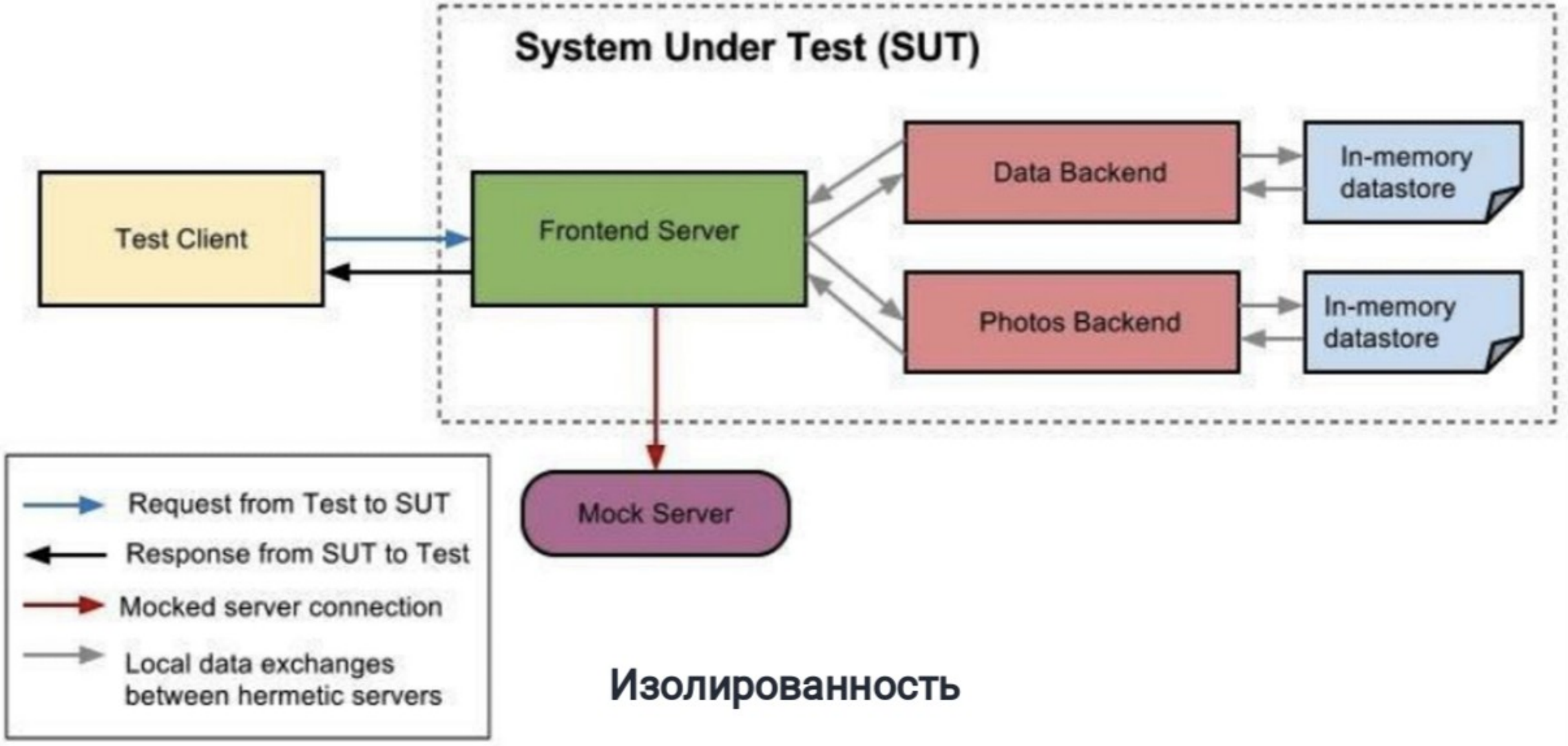
```
1. def test_add_smth():
2.     user = create_new_user()
3.     smth = user.create_smth()
4.     smth.add()
5.     check(smth.is_added())
```

Теперь тест проверяет только одну функциональную единицу

Независимость

- test1 test2 test3
- test2 test3 test1
- test3 test1 test2
- ...


System Under Test (SUT)
User's Client
Frontend Server
Data Backend
Photos Backend
Another Backend
Request from User
Response to User



Изолированность

Также в тестах стараются избегать

- → проверок в циклах
- → ветвления в тестах

Какие инструменты для тестирования есть в Python ?

- → unittests
- → nose/nose2
- → pytest

Mentimeter

Давайте посмотрим на unittests

Напишем тесты для

- Проверки статуса страницы индекса
- Проверка html страницы индекса
- Проверка обращения к существующей/несуществующей группе
- Проверить отправку формы/ошибка в форме

Дополнительно нам понадобятся

→ pip install factory-boy==2.12.0

→ pip install coverage==5.2

Демонтрация

Вопросы