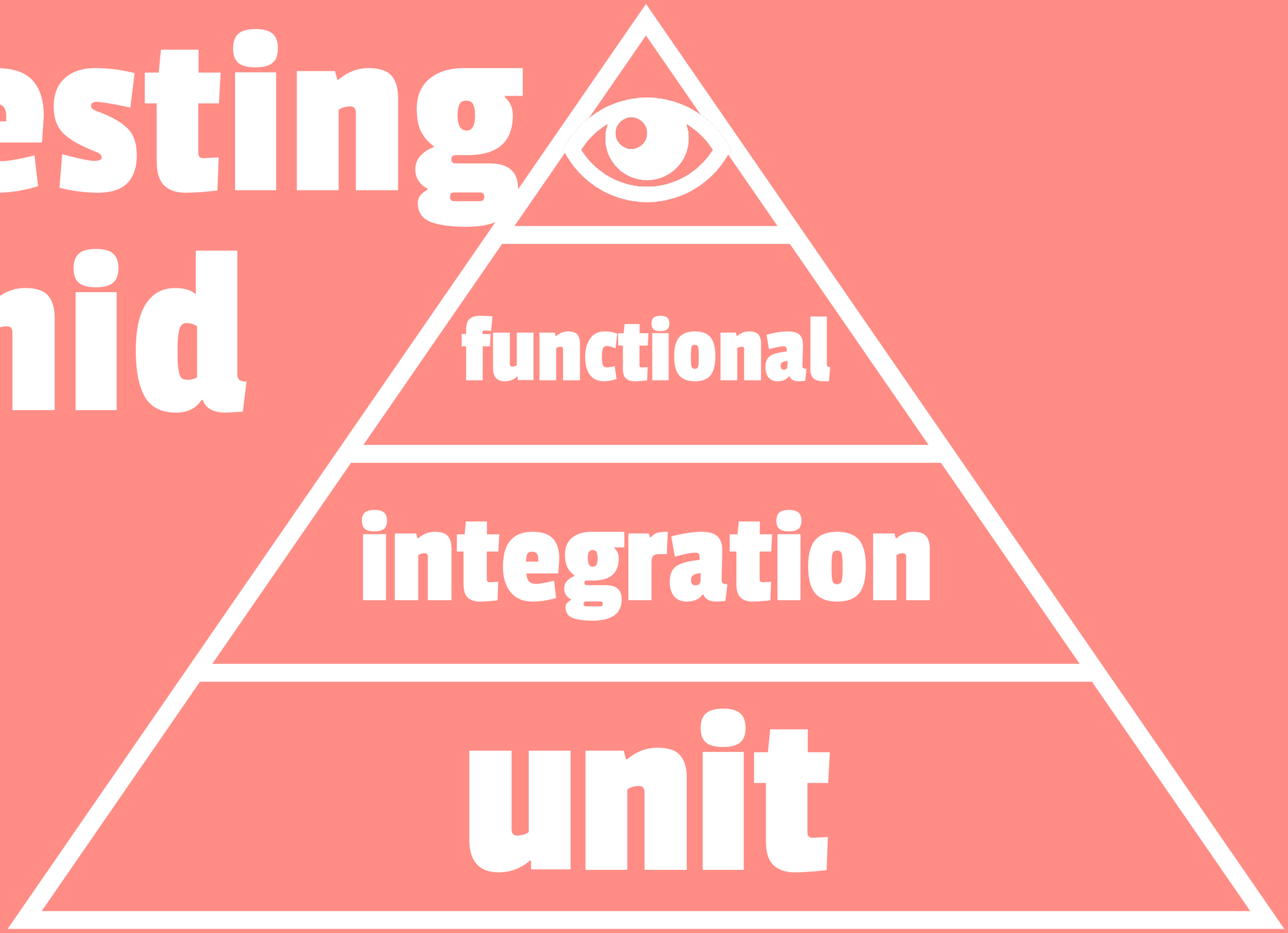
Django and the testing pyramid

@aaronbassett

Energy Pyramid

Food Pyramid

The testing pyramid



Unit Tests

test *1* thing in isolation


SUPER
FAST

Integration Tests

test things work together

Functional Tests

end-to-end testing

Manual Testing

people cycles not processor cycles

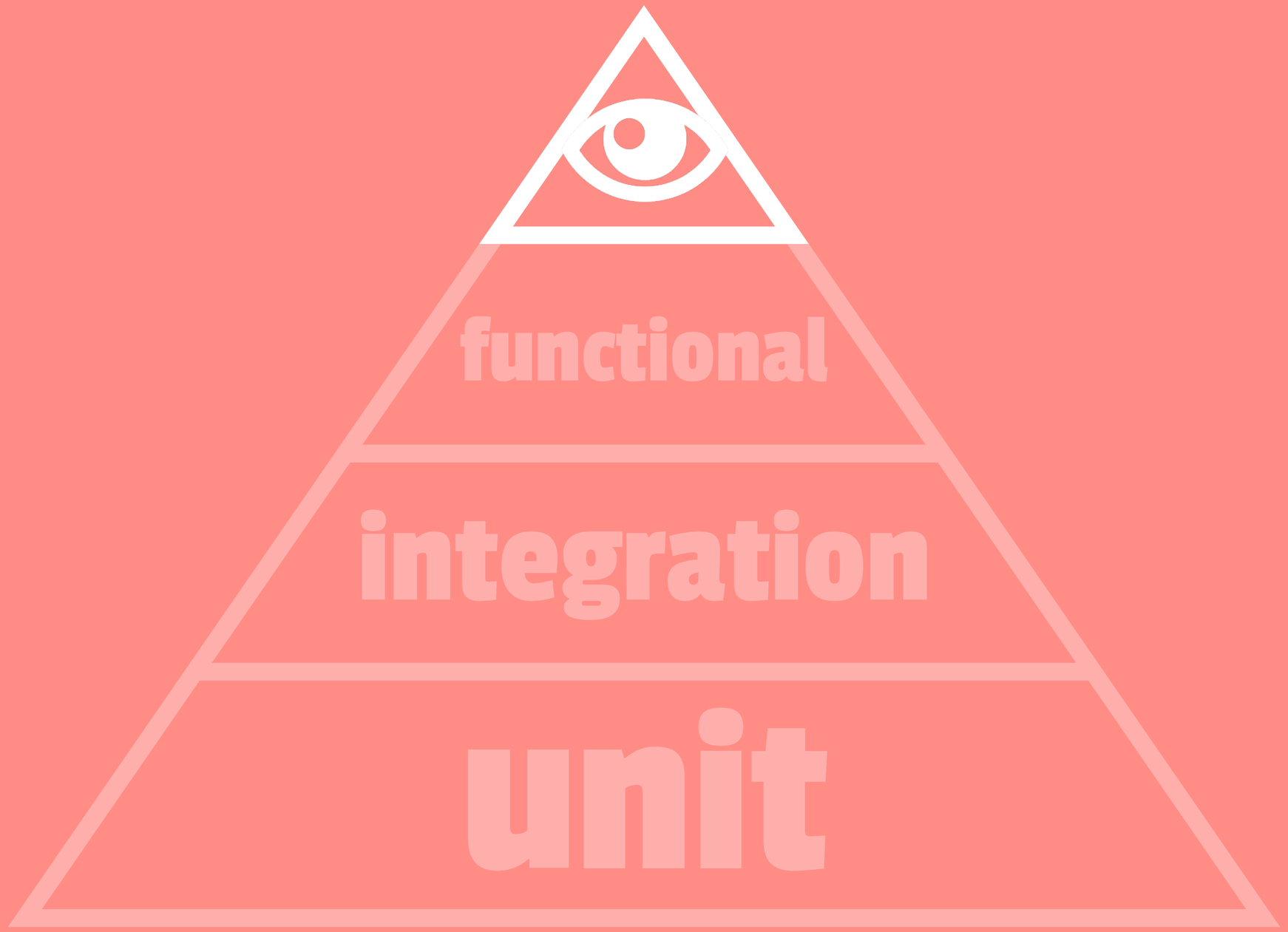
```
Feature: Number verification
    Users can add verified mobile numbers to their profile

Scenario: Adding a valid phone number
    Given I'm an authenticated user
    And I have a valid phone number
    When I go to the phone number page
    And I press the verify button
    Then I should not see the error message
```

```
Feature: Number verification
    Users can add verified mobile numbers to their profile

Scenario: Adding a valid phone number
    Given I'm an authenticated user
    And I have a valid phone number
    When I go to the phone number page
    And I press the verify button
    Then I should not see the error message
```

```
Feature: Number verification
    Users can add verified mobile numbers to their profile

Scenario: User enters an invalid phone number
    Given I'm an authenticated user
     When I go to the phone number page
    And I enter ‘not a number’
    And I press the verify button
    Then I should see the error message
```

```
Feature: Number verification
    Users can add verified mobile numbers to their profile

Scenario: User enters an invalid phone number
    Given I'm an authenticated user
     When I go to the phone number page
    And I enter <invalid_number>
    And I press the verify button
    Then I should see the error message

    Examples:
    | invalid_number  |
    | foo@example.com |
    | 0               |
    | +441411111111   |
    | +44712345678    |
```

```
Feature: Number verification
    Users can add verified mobile numbers to their profile

Scenario: User enters an invalid phone number
    Given I'm an authenticated user
     When I go to the phone number page
    And I enter <invalid_number>
    And I press the verify button
    Then I should see the error message

    Examples:
    | invalid_number  |
    | foo@example.com |
    | 0               |
    | +441411111111   |
    | +44712345678    |
```


functional
integration
unit

```
@given("I'm an authenticated user")
def authenticat_user(browser):
    browser.visit(urljoin(browser.url, '/login/')
    browser.fill('username', 'test_user')
    browser.fill('password', 'test_password')
    button = browser.find_by_css('button[name=submit]').first.click()
```

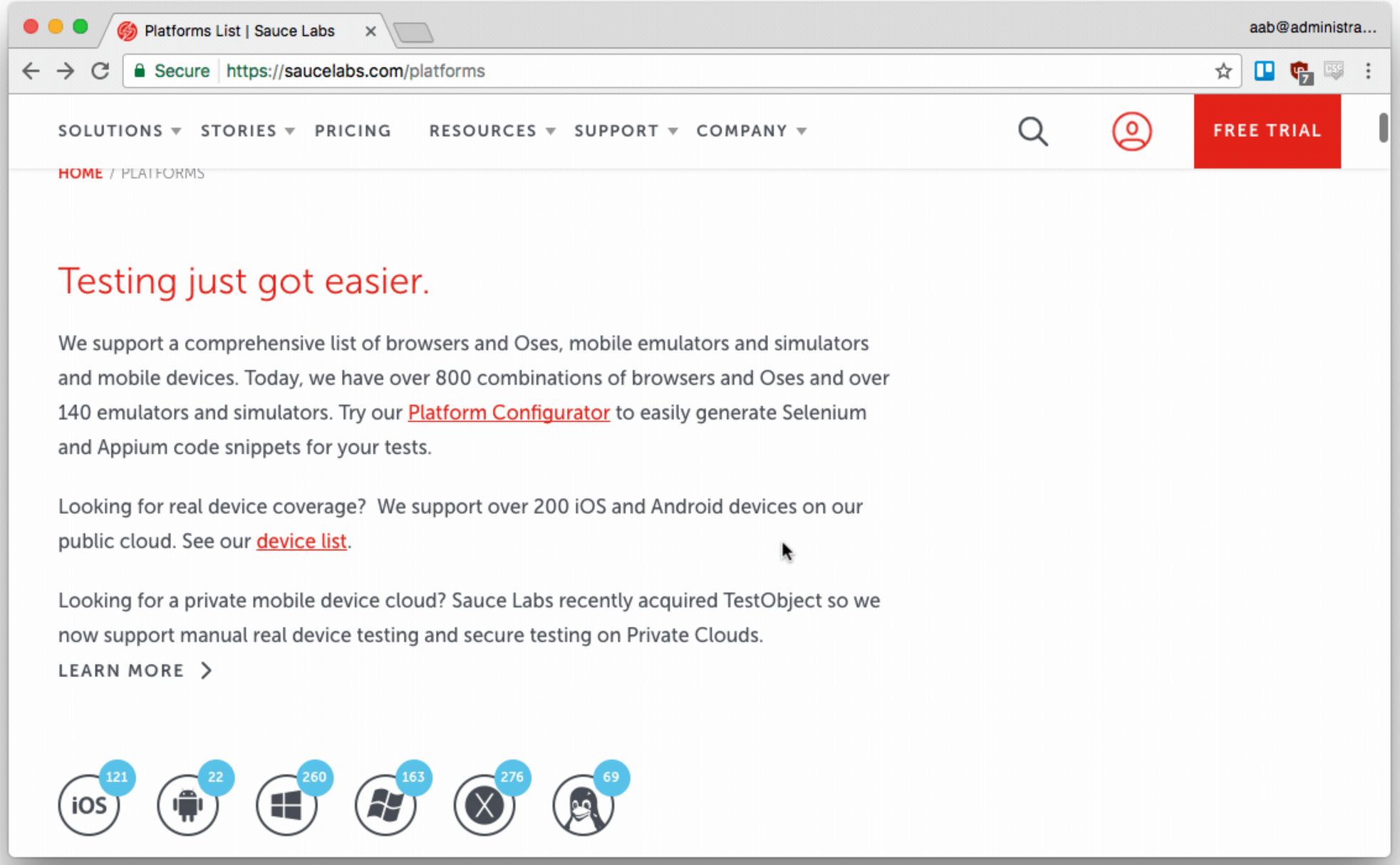
```
@given("I'm an authenticated user")
def authenticat_user(browser):
    browser.visit(urljoin(browser.url, '/login/')
    browser.fill('username', 'test_user')
    browser.fill('password', 'test_password')
    button = browser.find_by_css('button[name=submit]').first.click()
```

```
@given("I'm an authenticated user")
def authenticat_user(browser):
    browser.visit(urljoin(browser.url, '/login/')
    browser.fill('username', 'test_user')
    browser.fill('password', 'test_password')
    button = browser.find_by_css('button[name=submit]').first.click()
```

```
@given("I'm an authenticated user")
def authenticat_user(browser):
    browser.visit(urljoin(browser.url, '/login/')
    browser.fill('username', 'test_user')
    browser.fill('password', 'test_password')
    button = browser.find_by_css('button[name=submit]').first.click()
```

splinter & pytest-splinter




Platforms List | Sauce Labs
aab@administra...
Secure
https://saucelabs.com/platforms
SOLUTIONS
STORIES
PRICING
RESOURCES
SUPPORT
COMPANY
FREE TRIAL
HOME / PLATFORMS
Testing just got easier.
We support a comprehensive list of browsers and Oses, mobile emulators and simulators and mobile devices. Today, we have over 800 combinations of browsers and Oses and over 140 emulators and simulators. Try our Platform Configurator to easily generate Selenium and Appium code snippets for your tests.
Looking for real device coverage? We support over 200 iOS and Android devices on our public cloud. See our device list.
Looking for a private mobile device cloud? Sauce Labs recently acquired TestObject so we now support manual real device testing and secure testing on Private Clouds.
LEARN MORE
iOS
121
22
260
163
276
69

```python
@when('I go to the phone number page')
def go_to_number_submission_page(browser):
    browser.visit(urljoin(browser.url, '/number/')


@when('I enter <invalid_number>')
def enter_invalid_number(browser, invalid_number):
    browser.fill('number', invalid_number)


@when('I press the verify button')
def submit_number(browser):
    browser.find_by_css('button[name=verify]').first.click()
```

```
@when('I go to the phone number page')
def go_to_number_submission_page(browser):
    browser.visit(urljoin(browser.url, '/number/')


@when('I enter <invalid_number>')
def enter_invalid_number(browser, invalid_number):
    browser.fill('number', invalid_number)


@when('I press the verify button')
def submit_number(browser):
    browser.find_by_css('button[name=verify]').first.click()
```

```
@when('I go to the phone number page')
def go_to_number_submission_page(browser):
    browser.visit(urljoin(browser.url, '/number/')


@when('I enter <invalid_number>')
def enter_invalid_number(browser, invalid_number):
    browser.fill('number', invalid_number)


@when('I press the verify button')
def submit_number(browser):
    browser.find_by_css('button[name=verify]').first.click()
```

```
@then('I should see the error message')
def has_error_message(browser):
    browser.find_by_css('.message.error').first
```

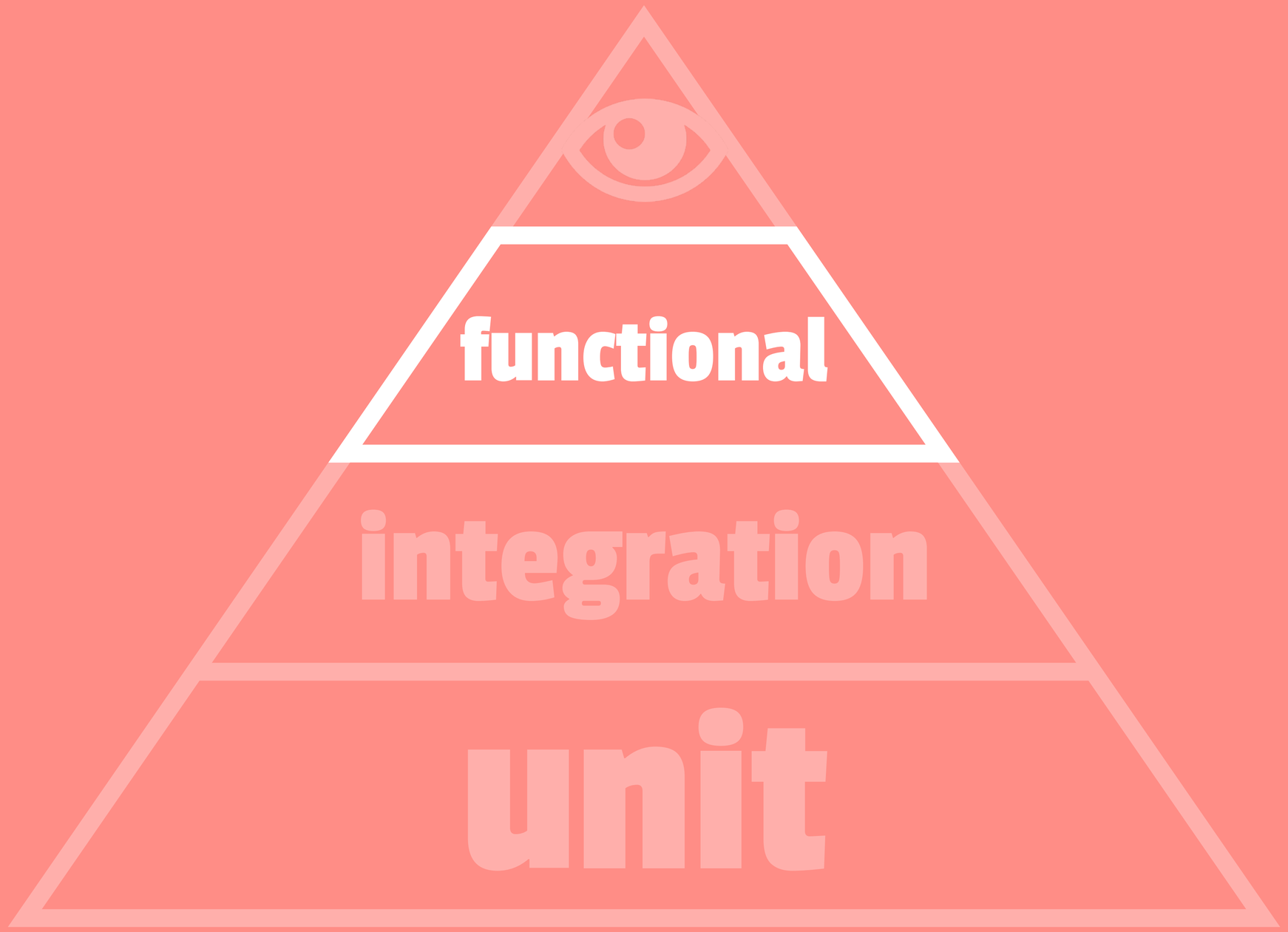
```
@then('I should see the error message')
def has_error_message(browser):
    browser.find_by_css('.message.error').first
```

```
@then('I should see the error message')
def has_error_message(browser):
    browser.find_by_css('.message.error').first
```

```
@given("I'm an authenticated user")
def authenticat_user(browser):
    browser.visit(urljoin(browser.url, '/login/')
    browser.fill('username', 'test_user')
    browser.fill('password', 'test_password')
    button = browser.find_by_css('button[name=submit]').first.click()
```

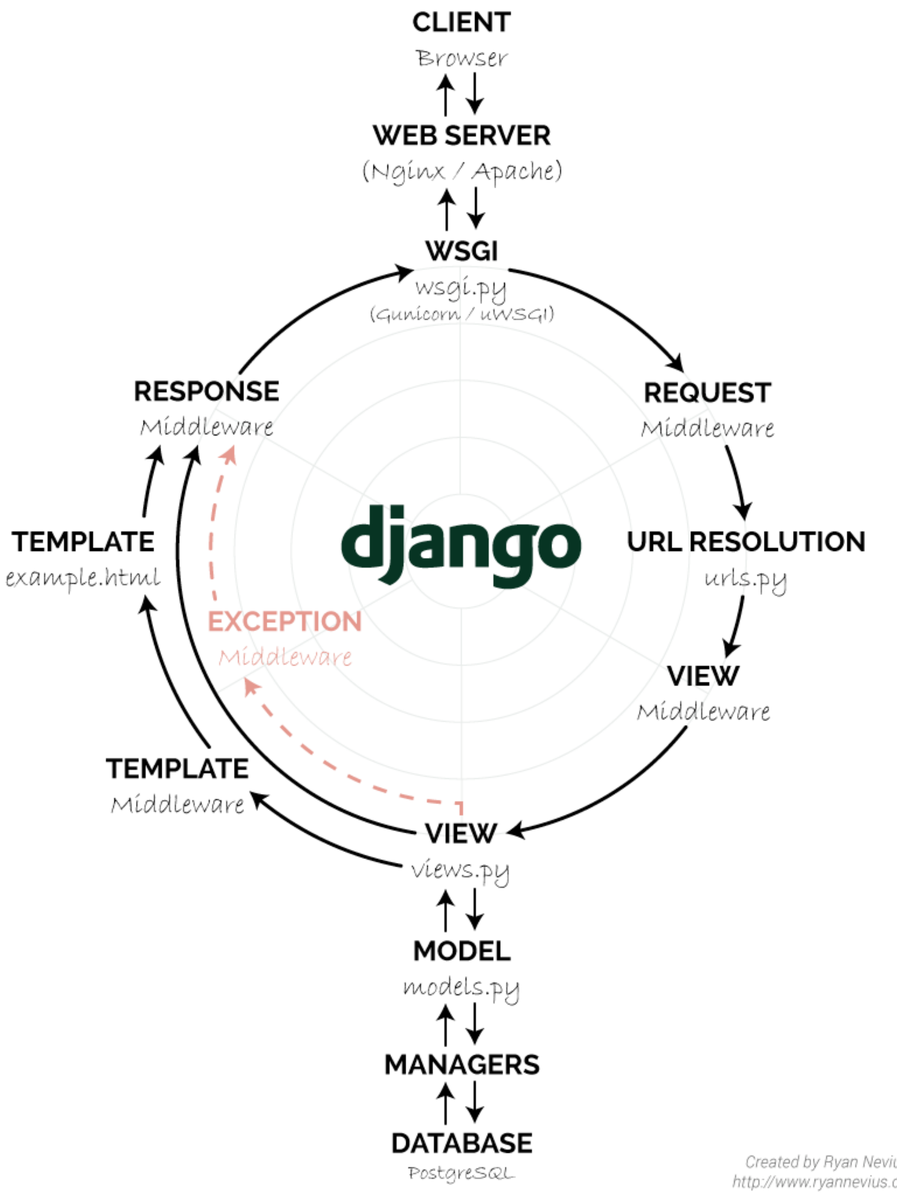
```
@given('I\'m logged in')
@given("I'm an authenticated user")
@given('I log in')
def authenticat_user(browser):
    browser.visit(urljoin(browser.url, '/login/')
    browser.fill('username', 'test_user')
    browser.fill('password', 'test_password')
    button = browser.find_by_css('button[name=submit]').first.click()
```

```
@given('I\'m logged in')
@given("I'm an authenticated user")
@given('I log in')
def authenticat_user(browser):
    browser.visit(urljoin(browser.url, '/login/')
    browser.fill('username', 'test_user')
    browser.fill('password', 'test_password')
    button = browser.find_by_css('button[name=submit]').first.click()
```


functional
integration
unit

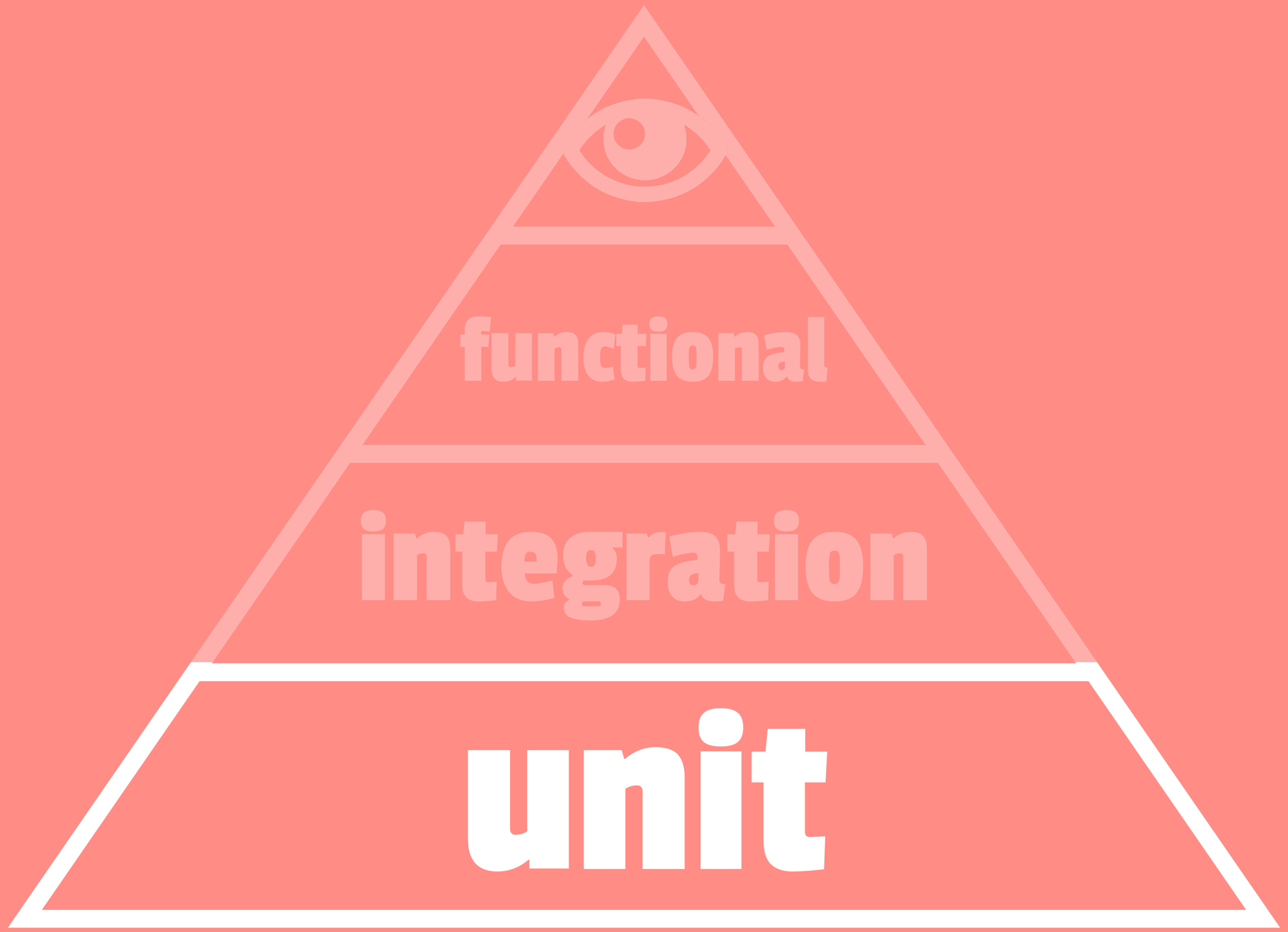
```
def start_number_verification(request):
    if request.method == "POST":
        if request.user.is_authenticated:
            number = request.POST.get("number", None)

            if re.search("[^0-9+-\\s]+", number):
                raise Exception

            existing_validation_requests = ValidationRequest.objects.filter(
                number=number,
                active=True
            )

            if existing_validation_requests.exists():
                raise Exception

            # AND SO ON...
```

django
Request-Response Cycle



Created by Ryan Nevius
http://www.ryannevius.com

ryannevius.com

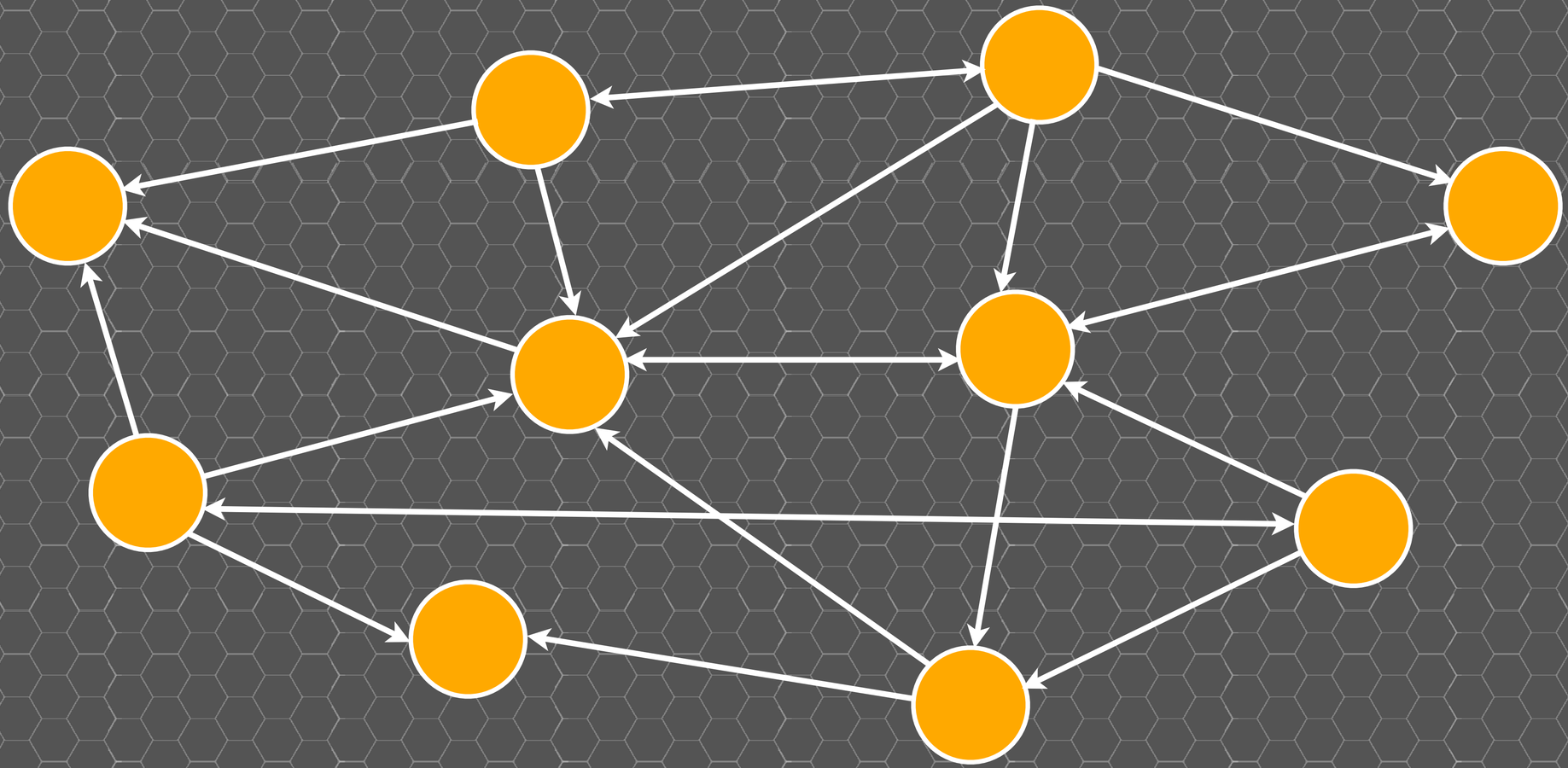

functional
integration
unit

```
def start_number_verification(request):
    if request.method == "POST":
        if request.user.is_authenticated:
            number = request.POST.get("number", None)

            if re.search("[^0-9+-\\s]+", number):
                raise Exception

            existing_validation_requests = ValidationRequest.objects.filter(
                number=number,
                active=True
            )

            if existing_validation_requests.exists():
                raise Exception

            # AND SO ON...
```

```
def start_number_verification(request):
    if request.method == "POST":
        if request.user.is_authenticated:
            number = request.POST.get("number", None)

            if re.search("[^0-9+-\\s]+", number):
                raise Exception

            existing_validation_requests = ValidationRequest.objects.filter(
                number=number,
                active=True
            )

            if existing_validation_requests.exists():
                raise Exception

            # AND SO ON...
```

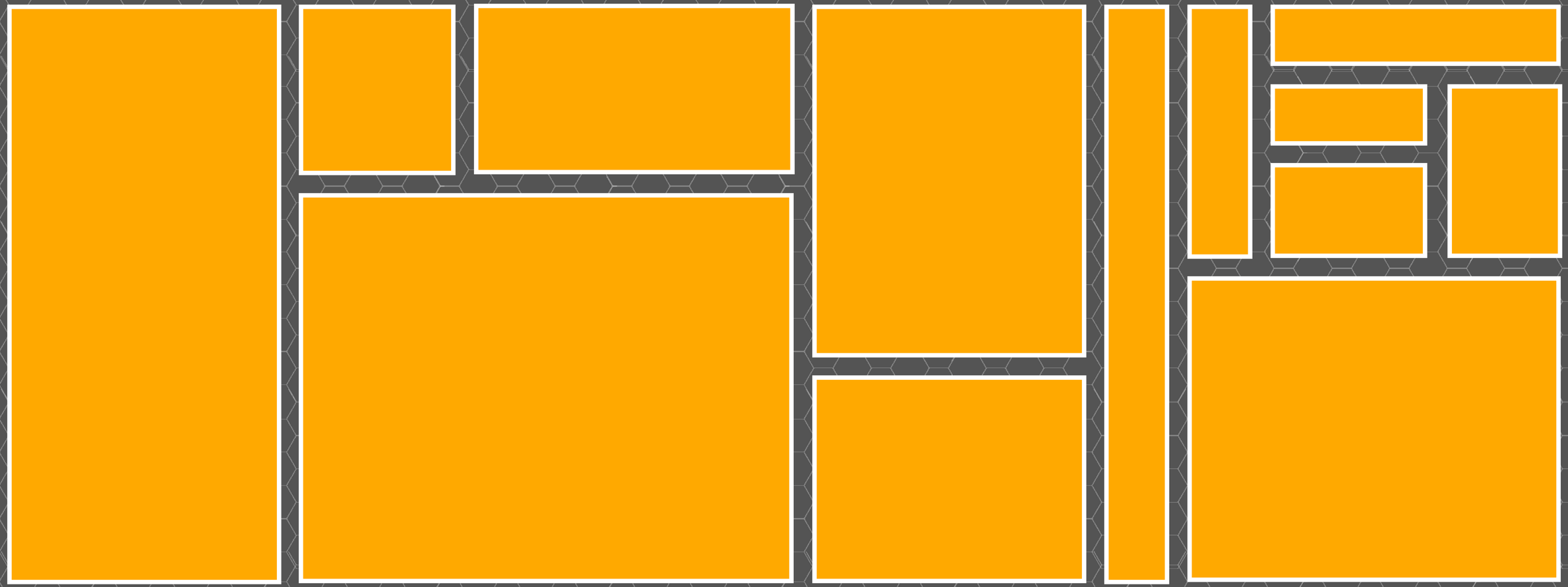

mock.patch

Connected




KALABAW

Modular



⌥⌘M

```
class NumberVerificationView(View):

    def start_number_verification(request):
        if request.user.is_authenticated:
            number = request.POST.get("number", None)

            if re.search("[^0-9+-\\s]+", number):
                raise Exception

            existing_validation_requests = ValidationRequest.objects.filter(
                number=number,
                active=True
            )

            if existing_validation_requests.exists():
                raise Exception

            # AND SO ON...
```

```
class NumberVerificationView(LoginRequiredMixin, View):
    login_url = '/login/'
    redirect_field_name = 'redirect_to'

    def start_number_verification(request):
        number = request.POST.get("number", None)

        if re.search("[^0-9+-\\s]+", number):
            raise Exception

        existing_validation_requests = ValidationRequest.objects.filter(
            number=number,
            active=True
        )

        if existing_validation_requests.exists():
            raise Exception

        # AND SO ON...
```

```
class NumberVerificationView(LoginRequiredMixin, FormView):
    ...

    def form_valid(self, form):
        number = request.POST.get("number", None)

        if re.search("[^0-9+-\\s]+", number):
            raise Exception

        existing_validation_requests = ValidationRequest.objects.filter(
            number=number,
            active=True
        )

        if existing_validation_requests.exists():
            raise Exception

        # AND SO ON...
```

```
class NumberVerificationView(LoginRequiredMixin, FormView):
    ...

    def form_valid(self, form):
        # AND SO ON...
        return super(ContactView, self).form_valid(form)
```

```
def validate_phone_number_characters(value):
    if re.search("[^0-9+-\\s]+", value):
        raise ValidationError(
            _('%(value) contains invalid characters'),
            params={'value': value},
        )
```

```
def validate_no_active_verification_requests(value):
    existing_validation_requests = ValidationRequest.objects.filter(
        number=value,
        active=True
    )

    if existing_validation_requests.exists():
        raise ValidationError(
            _('There is already a pending request for %(value)'),
            params={'value': value},
        )
```

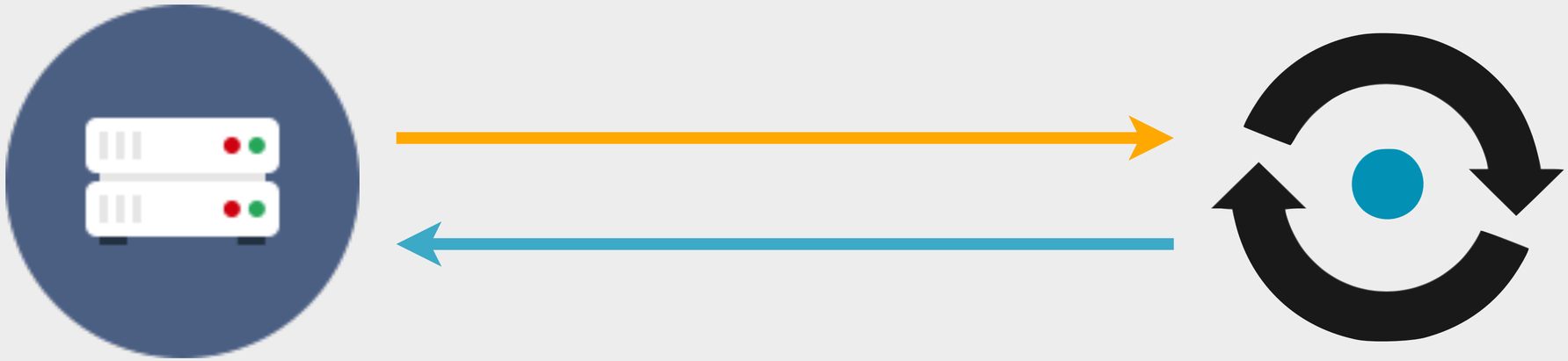

SUPER
FAST

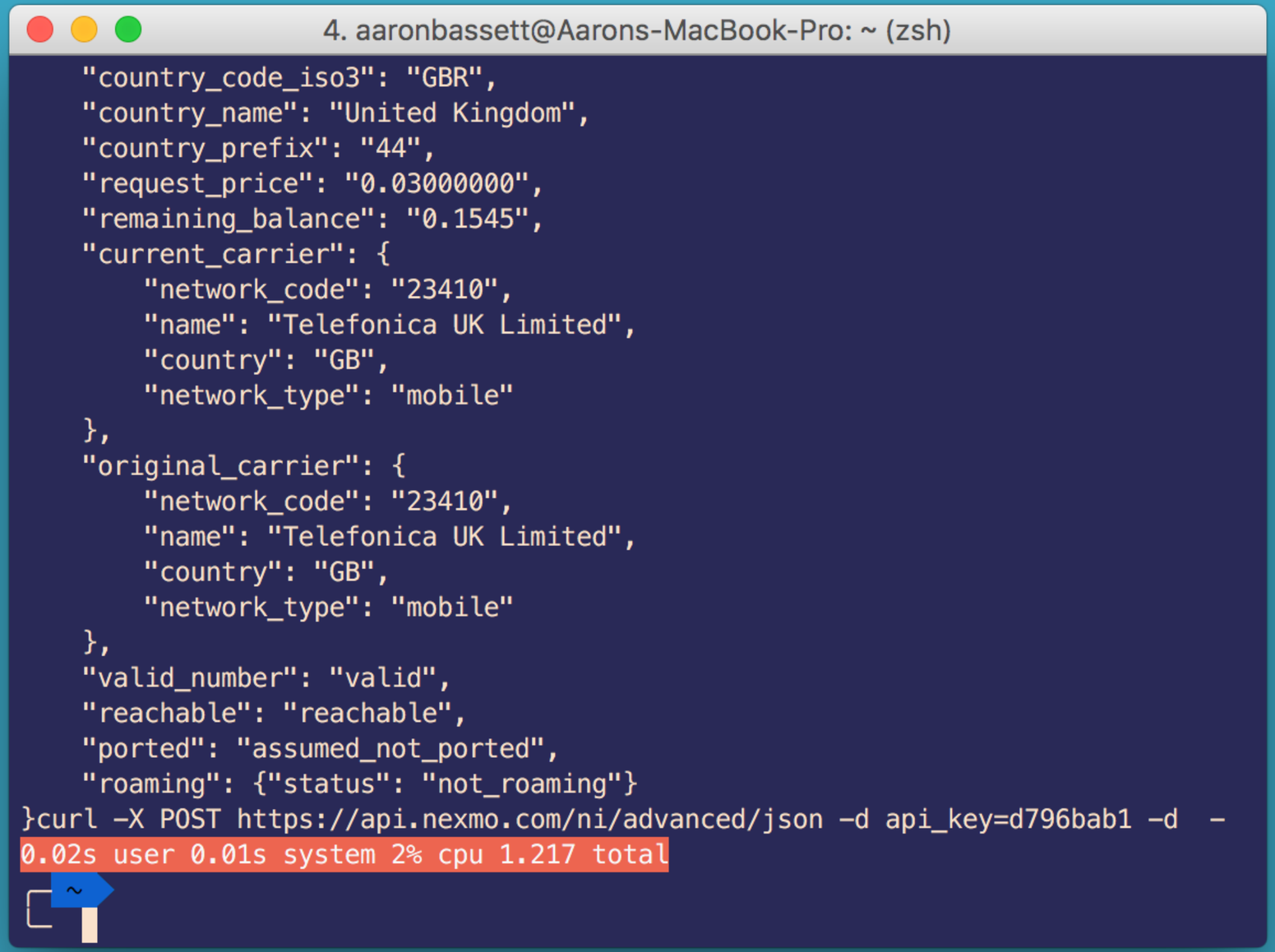
property based testing

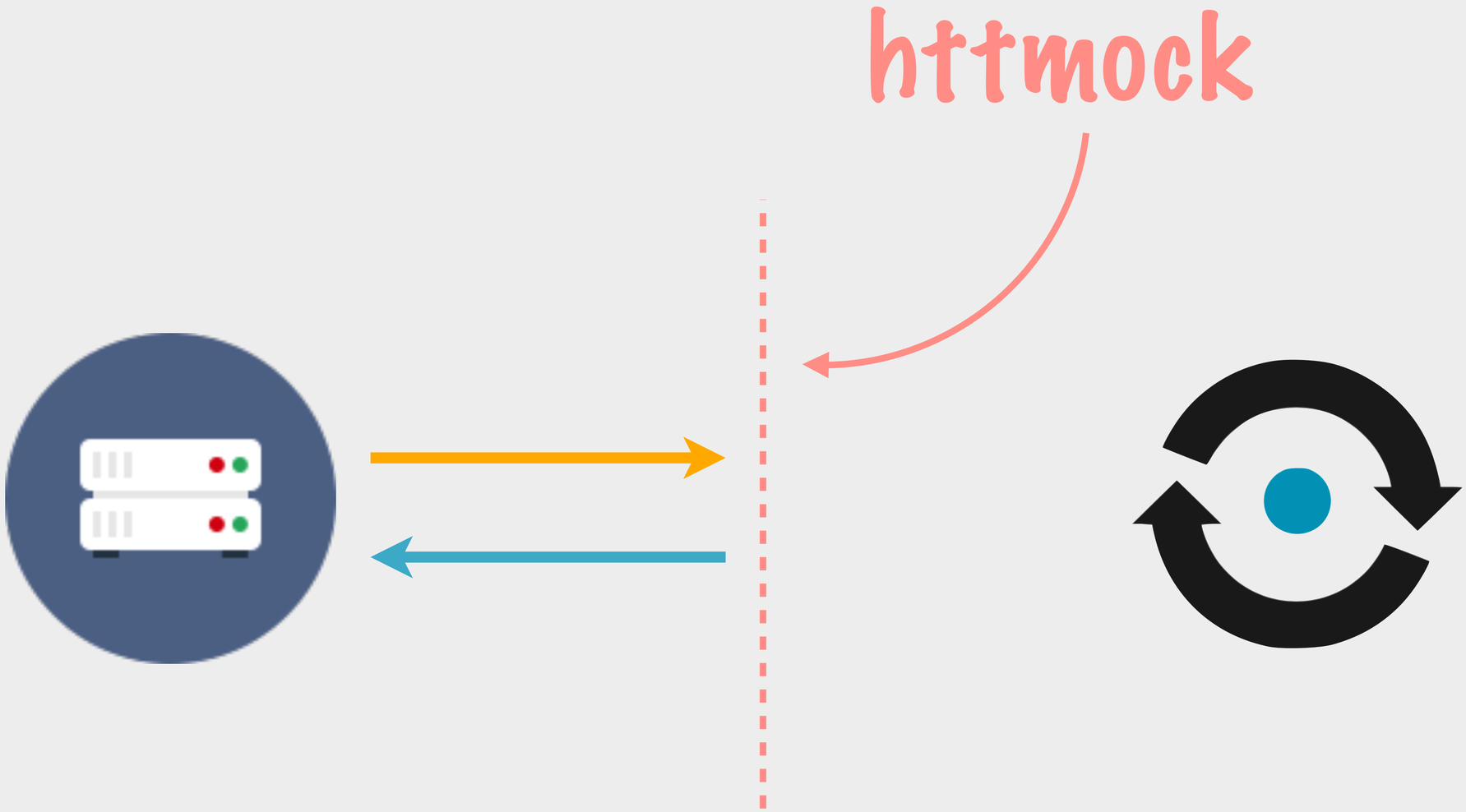
http://hypothesis.works/

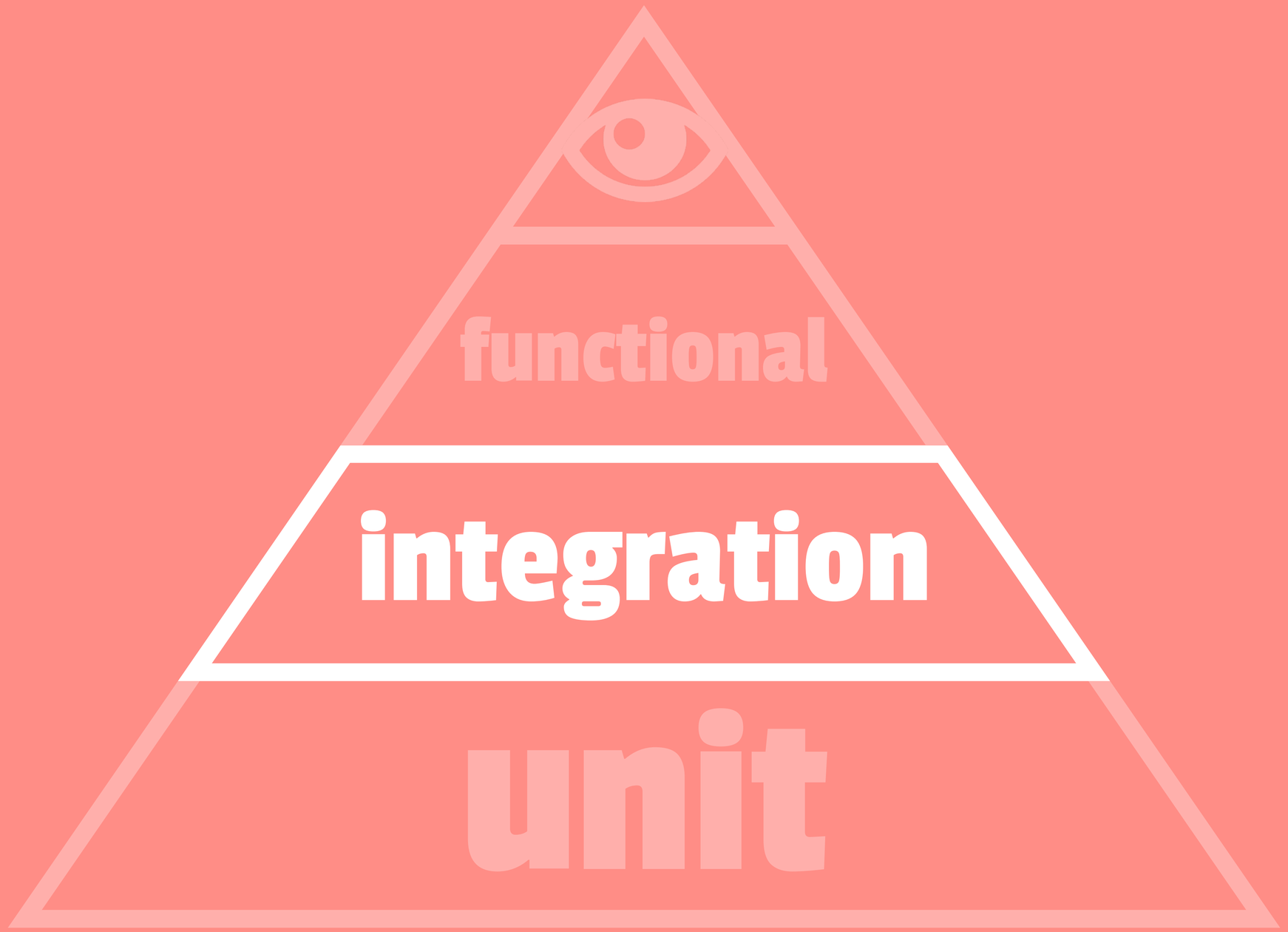
```python
@given(invalid_phone_number)
@settings(max_examples=1000)
def test_names_match_our_requirements(number):
    with pytest.raises(ValidationError, message="Expecting ValidationError"):
        validate_phone_number_characters(number)
```

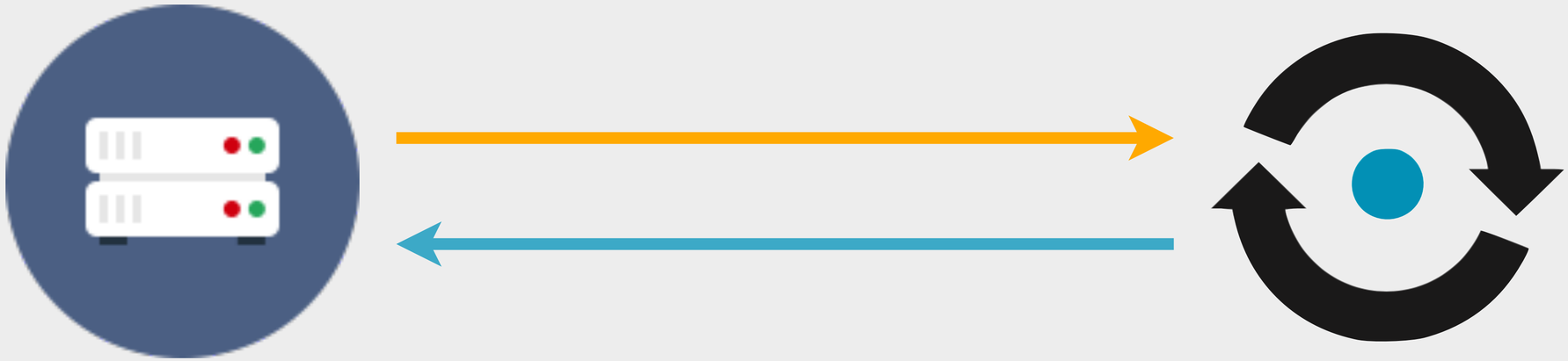
```
@given(invalid_phone_number)
@settings(max_examples=1000)
def test_names_match_our_requirements(number):
    with pytest.raises(ValidationError, message="Expecting ValidationError"):
        validate_phone_number_characters(number)
```




4. aaronbassett@Aarons-MacBook-Pro: ~ (zsh)
"country_code_iso3": "GBR",
"country_name": "United Kingdom",
"country_prefix": "44",
"request_price": "0.03000000",
"remaining_balance": "0.1545",
"current_carrier": {
"network_code": "23410",
"name": "Telefonica UK Limited",
"country": "GB",
"network_type": "mobile"
},
"original_carrier": {
"network_code": "23410",
"name": "Telefonica UK Limited",
"country": "GB",
"network_type": "mobile"
},
"valid_number": "valid",
"reachable": "reachable",
"ported": "assumed_not_ported",
"roaming": {"status": "not_roaming"}
}curl -X POST https://api.nexmo.com/ni/advanced/json -d api_key=d796bab1 -d -
0.02s user 0.01s system 2% cpu 1.217 total
~

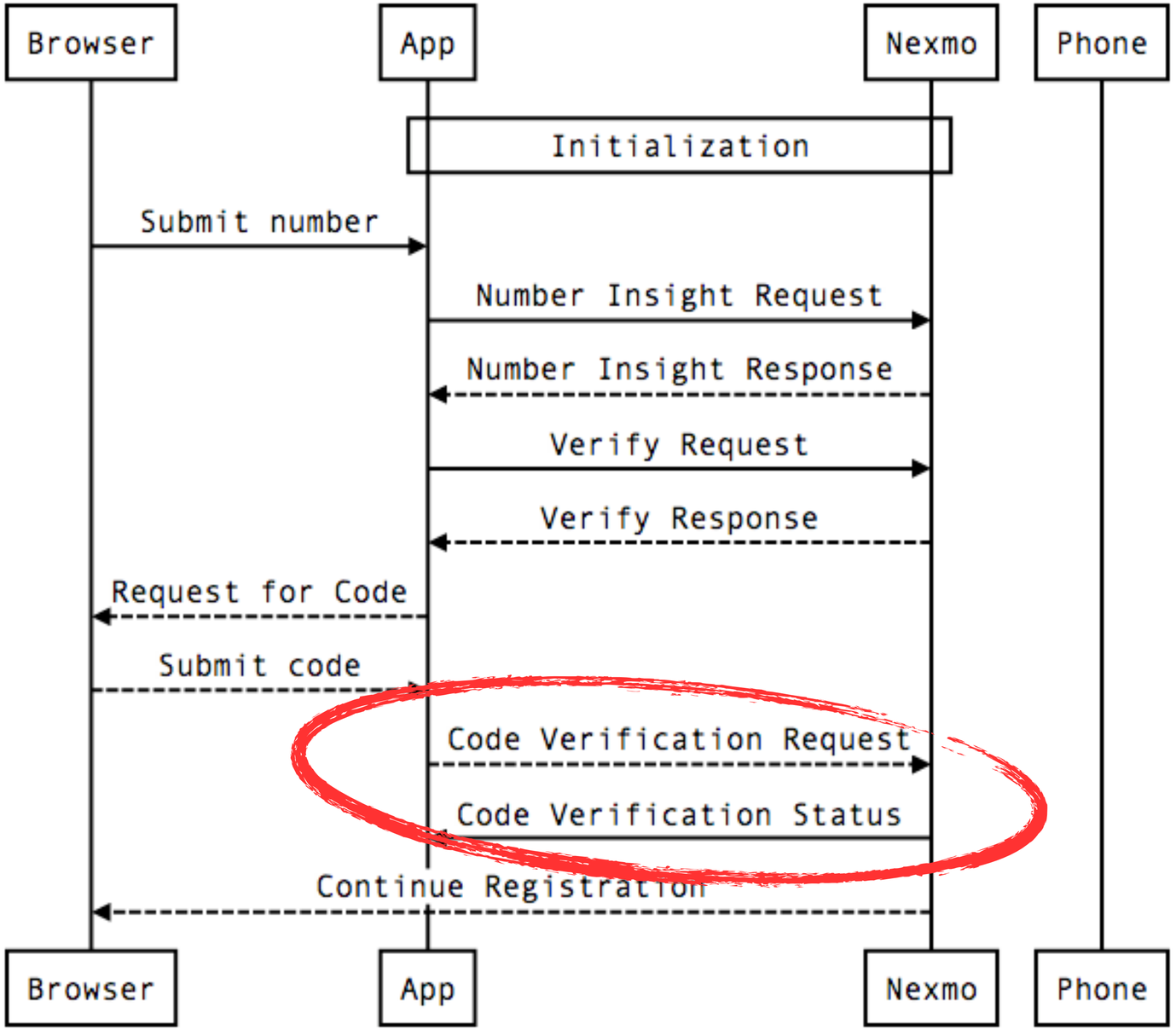

httmock

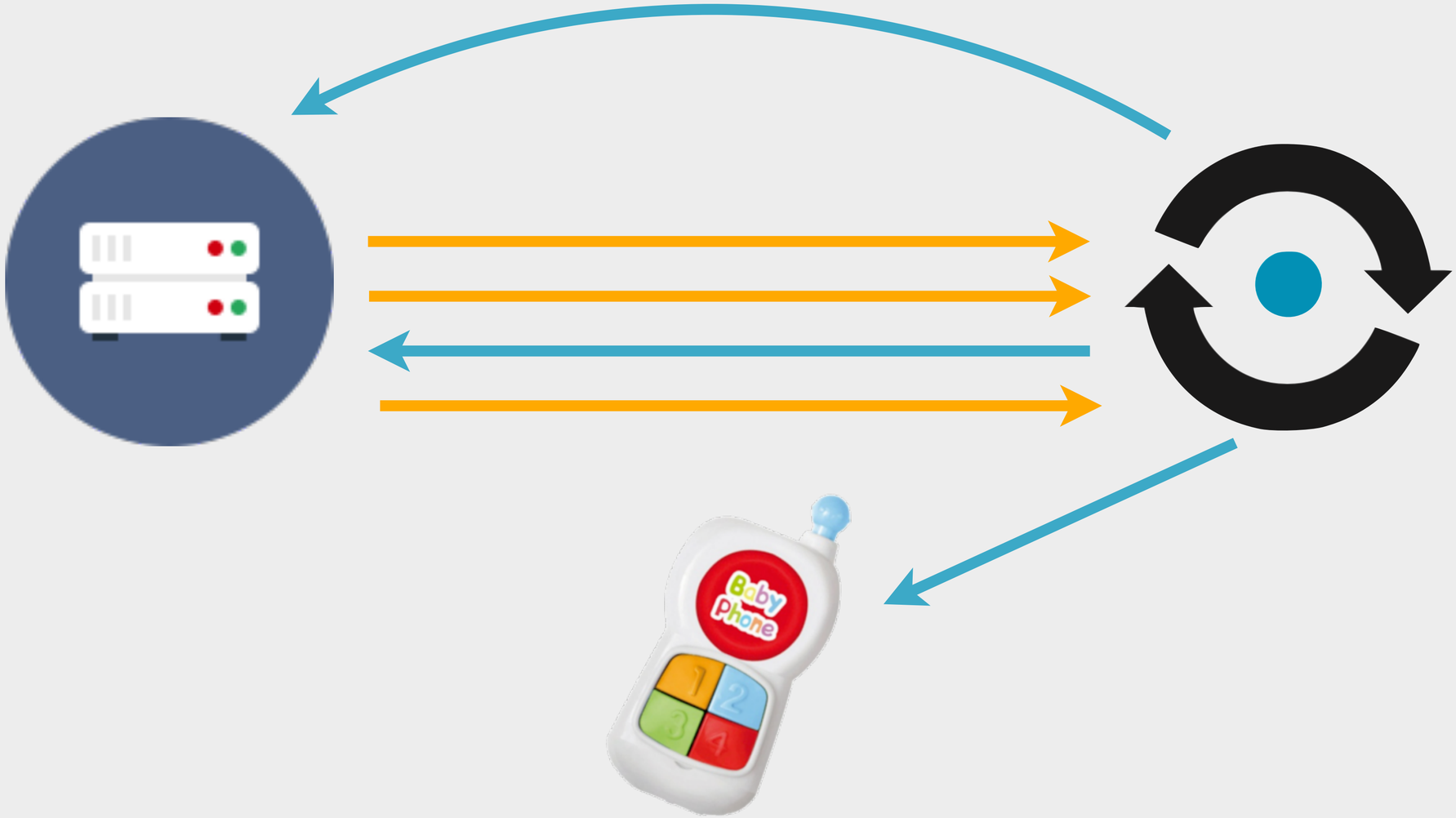

functional
integration
unit

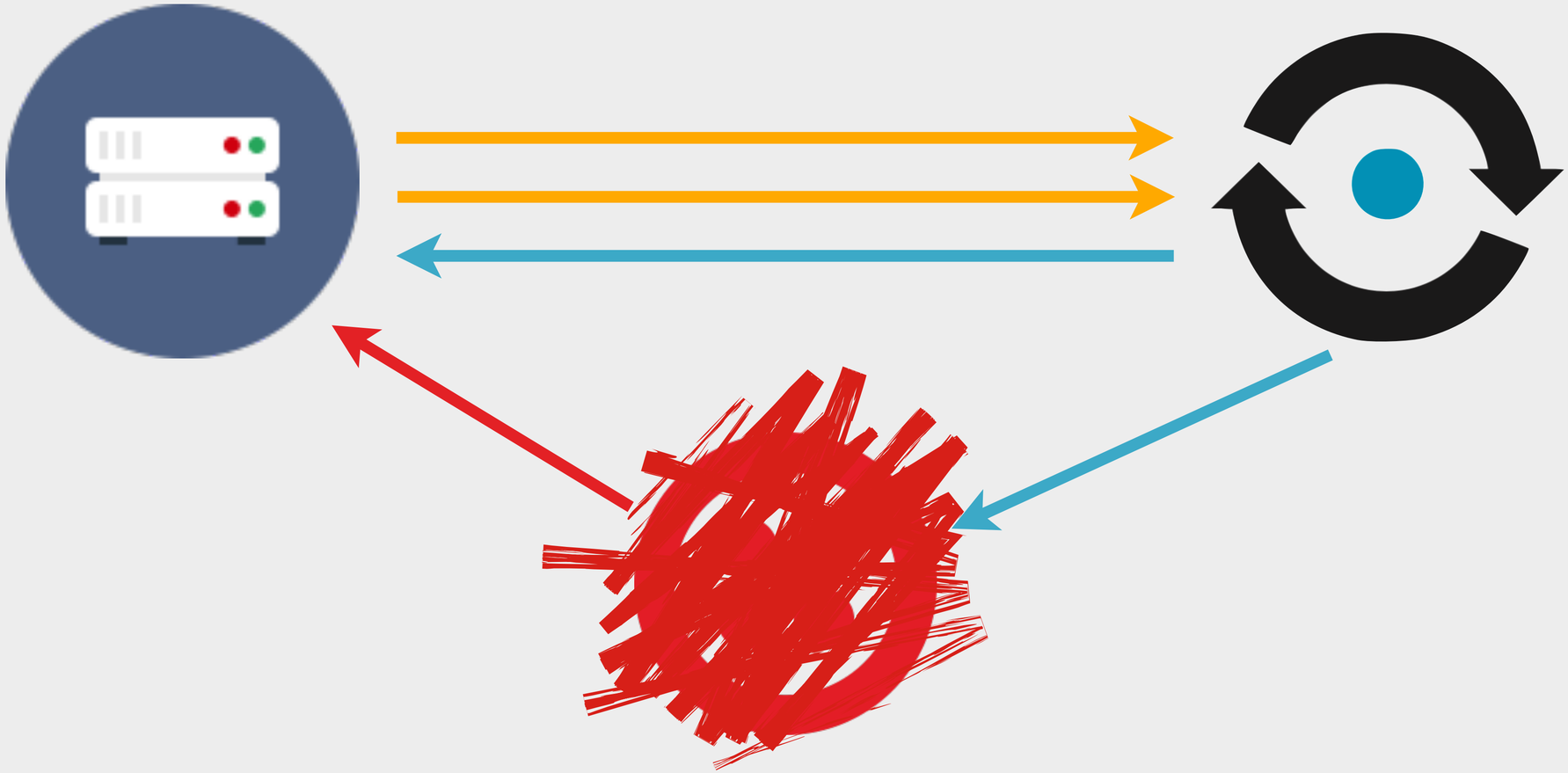


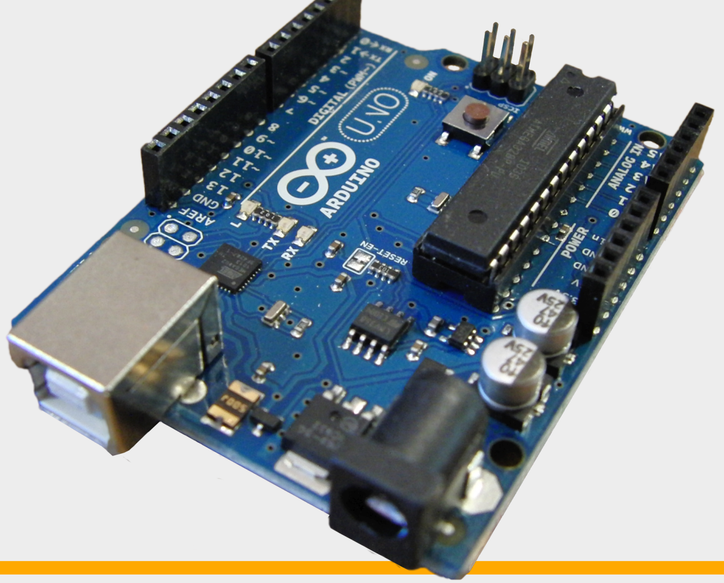
```
@pytest.mark.slowtest
def test_function():
    pass
```

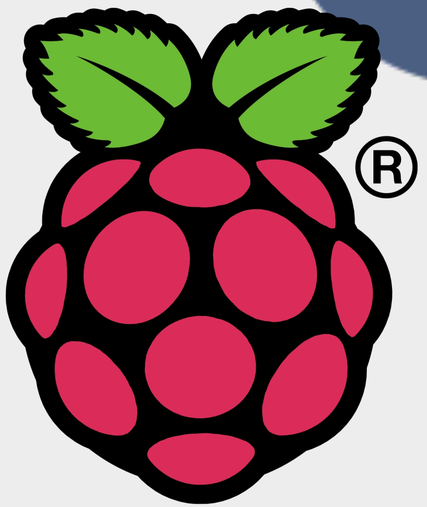
```
@pytest.mark.slowtest
def test_function():
    pass
```


Browser
App
Nexmo
Phone
Initialization
Submit number
Number Insight Request
Number Insight Response
Verify Request
Verify Response
Request for Code
Submit code
Code Verification Request
Code Verification Status
Continue Registration
Browser
App
Nexmo
Phone


Baby
Phone

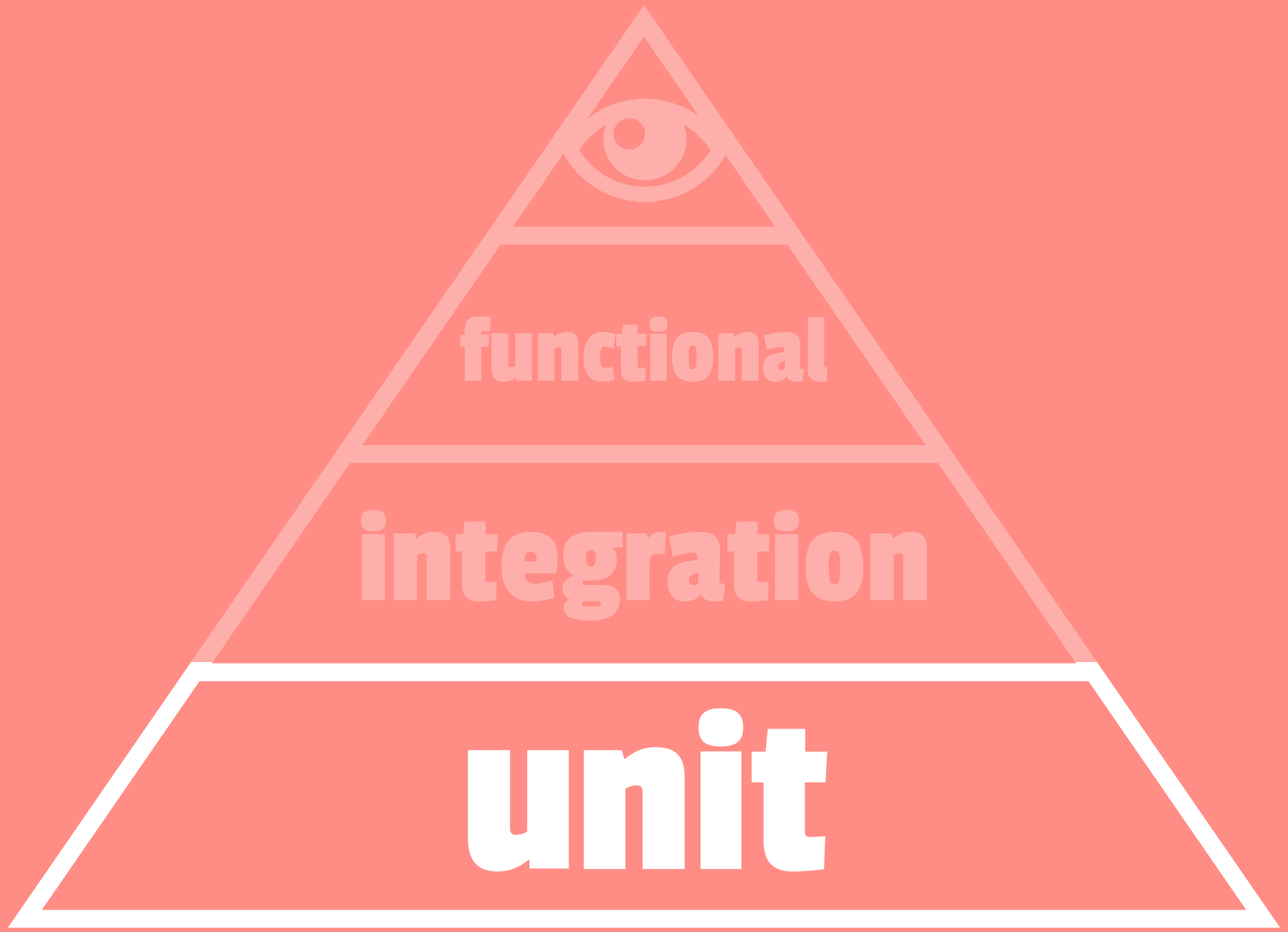


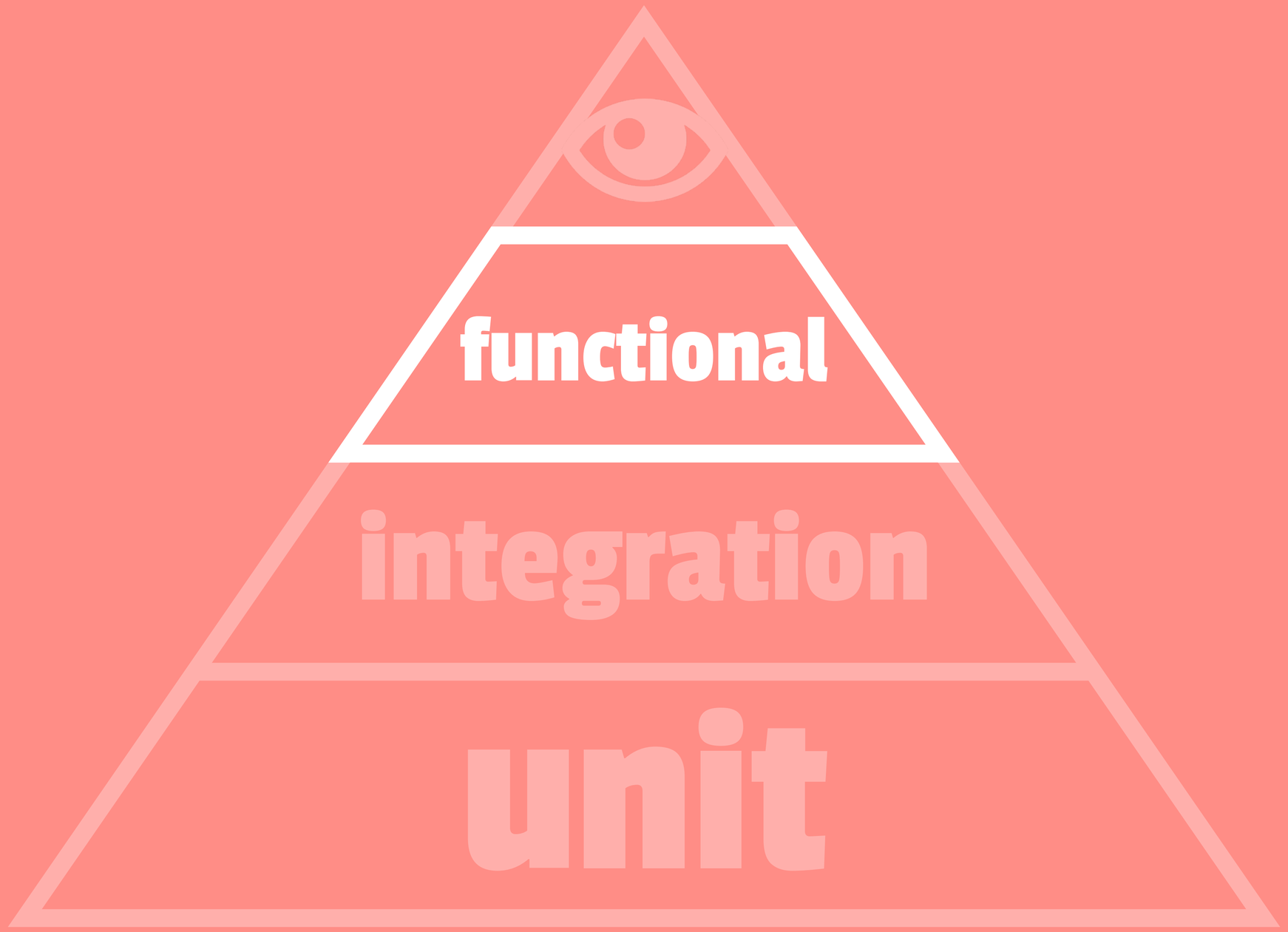


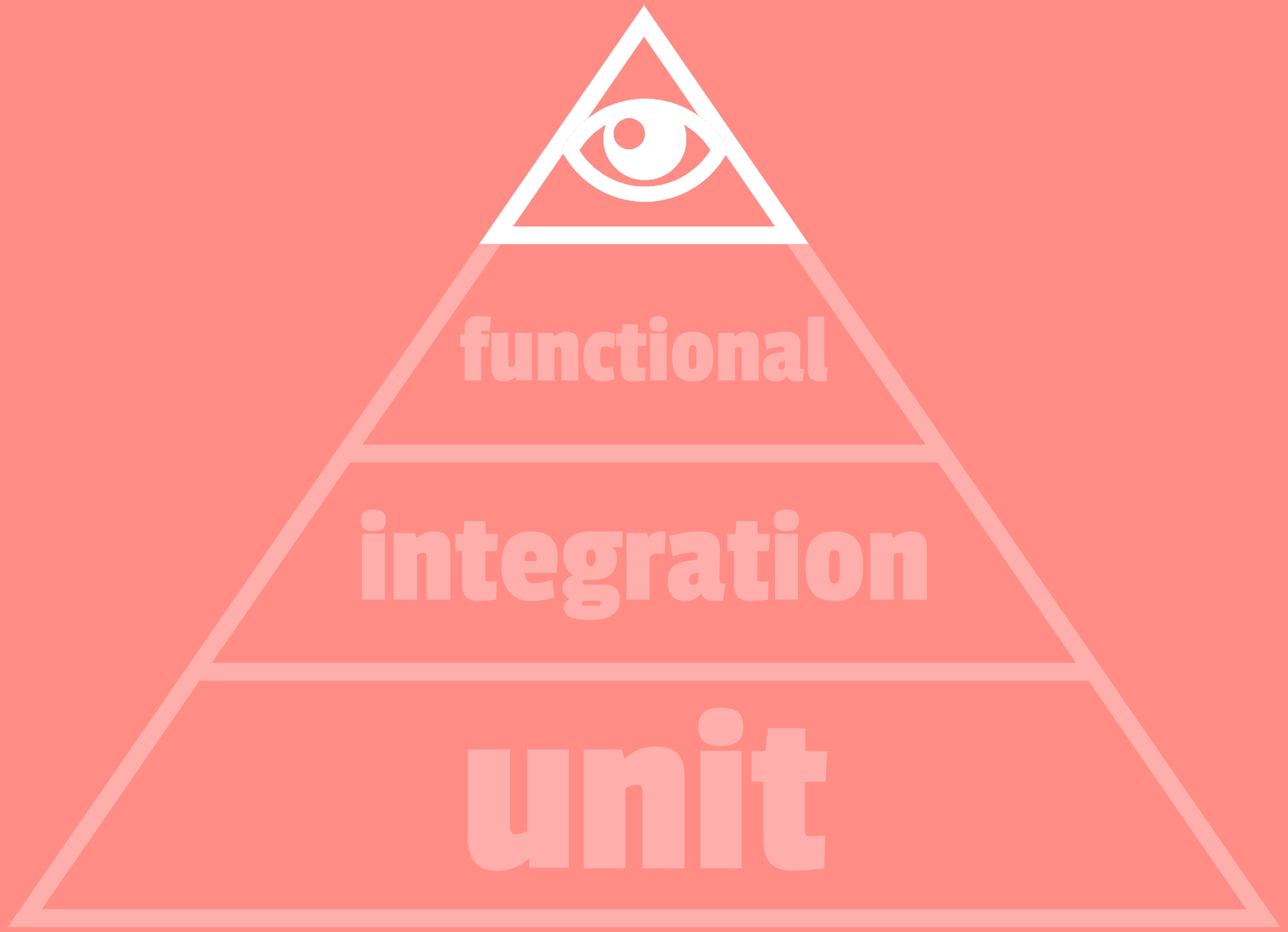



WARNING

STOP!


functional
integration
unit


functional
integration
unit


functional
integration
unit

Grazie

Django and the testing pyramid

@aaronbassett