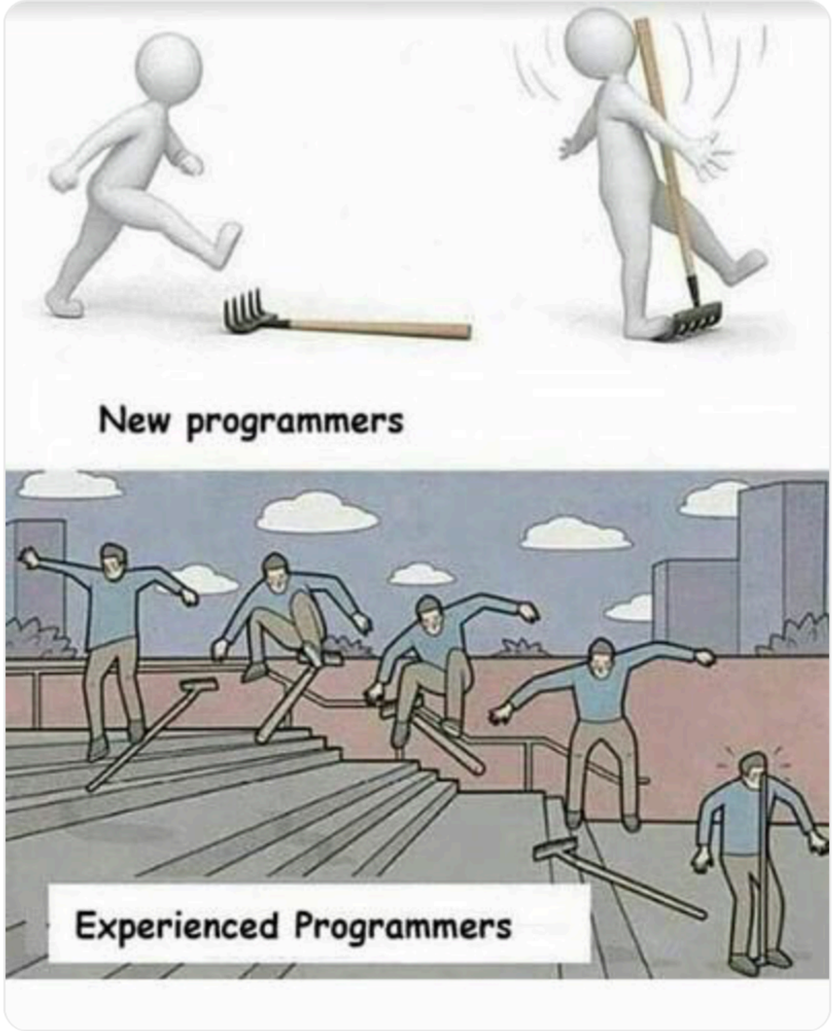
Chris Holland
@chrisholland

This made me laugh ... then cry a little.



10:47 AM - 25 Oct 2018 from Austin, TX

4,664 Retweets **12,631** Likes

Chris Holland - https://twitter.com/chrisholland/

Real-World TDD Application Development w/ Symfony and its Friends

Bootstrapping Symfony w/ REST, OAuth, User Management, Unit & Acceptance Tests

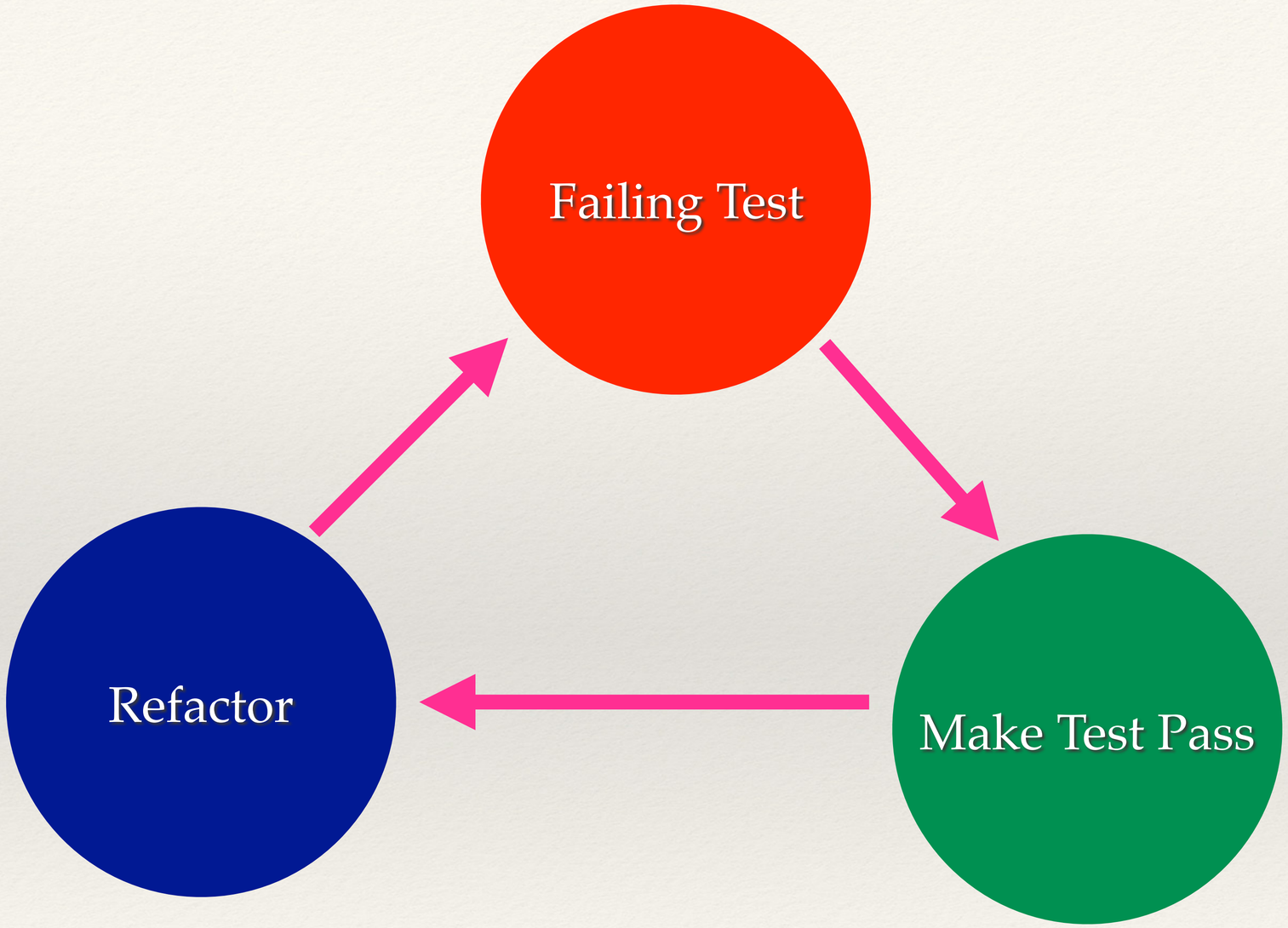
Slides:

http://bit.ly/real-world-tdd

What is TDD?

What is TDD

- Write a failing test
- Write just enough code to make that test pass
- Refactor

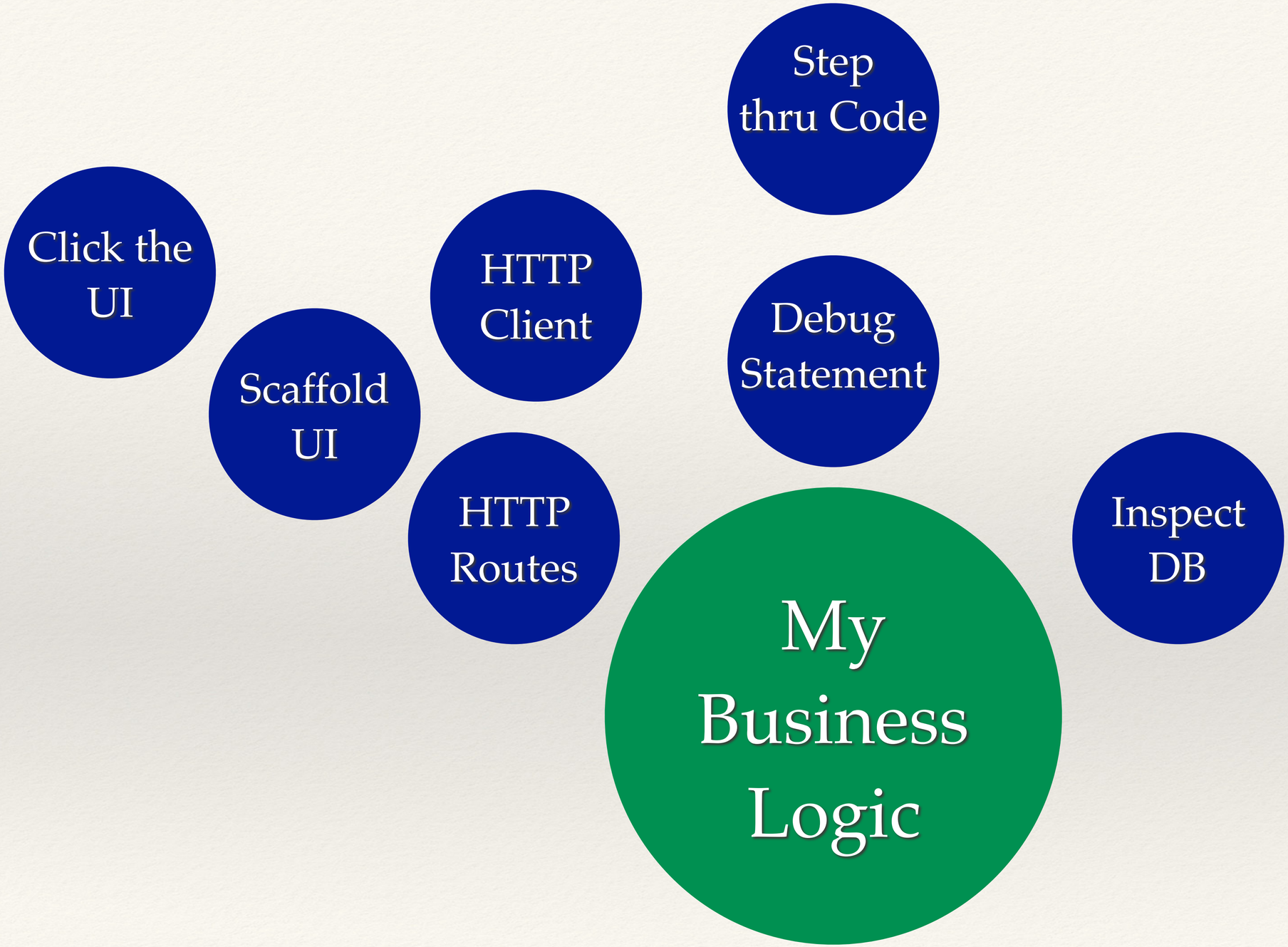

Failing Test
Make Test Pass
Refactor

Q: Why TDD?

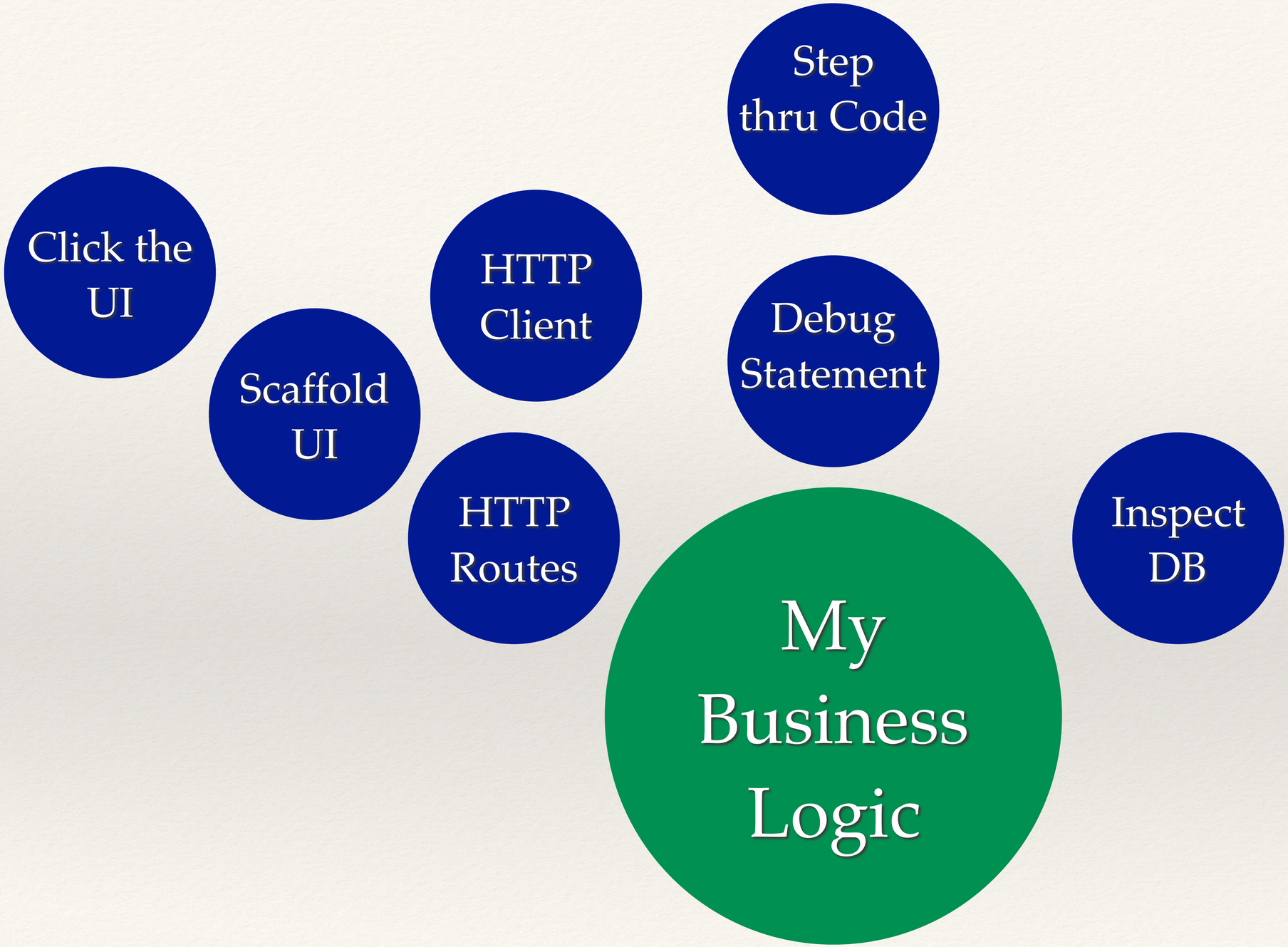
A: Faster Workflow

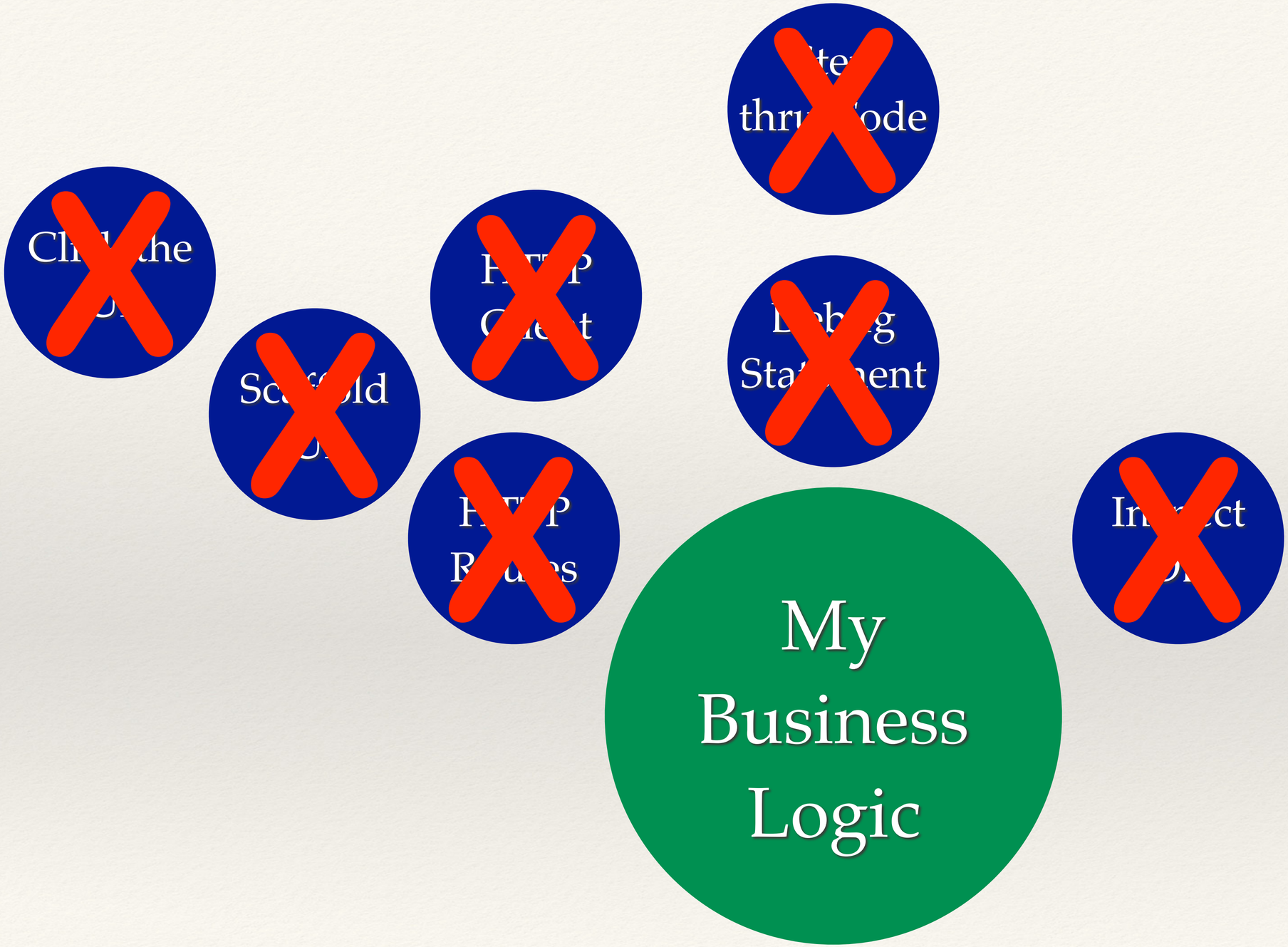
A: Faster Workflow (here's why)

Stimulating Code: Traditional Workflow


Step thru Code
Click the UI
HTTP Client
Debug Statement
Scaffold UI
HTTP Routes
Inspect DB
My Business Logic

Stimulating Code: TDD Workflow


Step thru Code
Click the UI
HTTP Client
Debug Statement
Scaffold UI
HTTP Routes
Inspect DB
My Business Logic


My
Business
Logic

Run Tests!

Real World TDD?

Real World TDD

- Test-Drive an Application with:
  - Data Operations
  - Business Logic
- Lean on Frameworks for:
  - RESTful Endpoints
  - User Registration / Authentication
  - OAuth-Based Authentication
- Don’t let Frameworks harm Tests

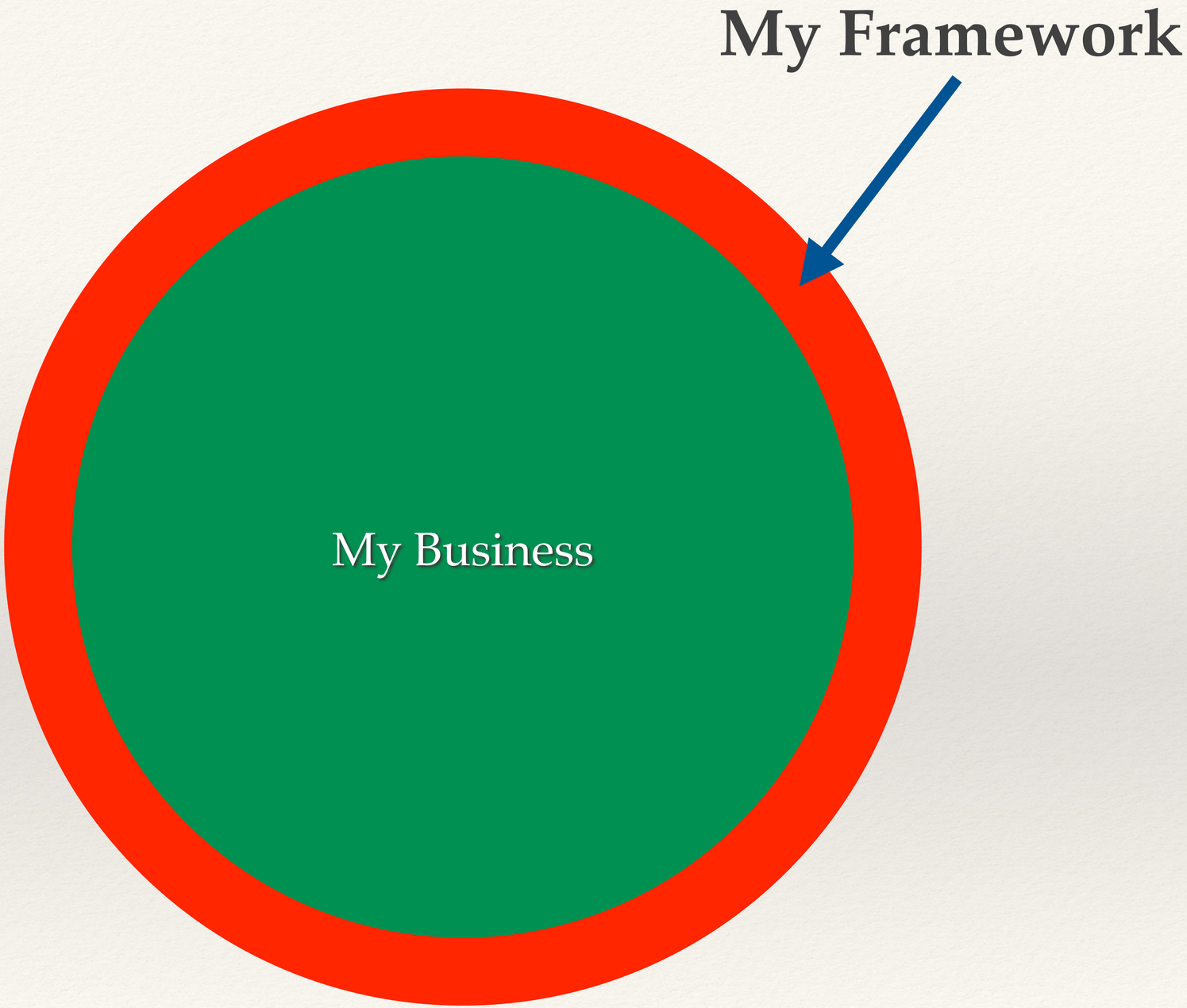
What I strive to avoid …

HEINEKEN

My Business

My Framework
My Business

What I strive to **achieve** …


My Framework
My Business

Outcome

Outcome

- JSON POST /api/register-user (with password)
- HTML GET /oauth/v2/auth?client_id=xxx&redirect=
  - user submits html form w/ username/password
  - redirect has **token** in the URL
- JSON POST/GET/PUT/PATCH /api/anything
  - with **token**

High-Level Tooling

Tooling: FOS

- FOSRestBundle
- FOSUserBundle
- FOSOAuthServerBundle

TDD Tooling

- PHPUnit
- Codeception

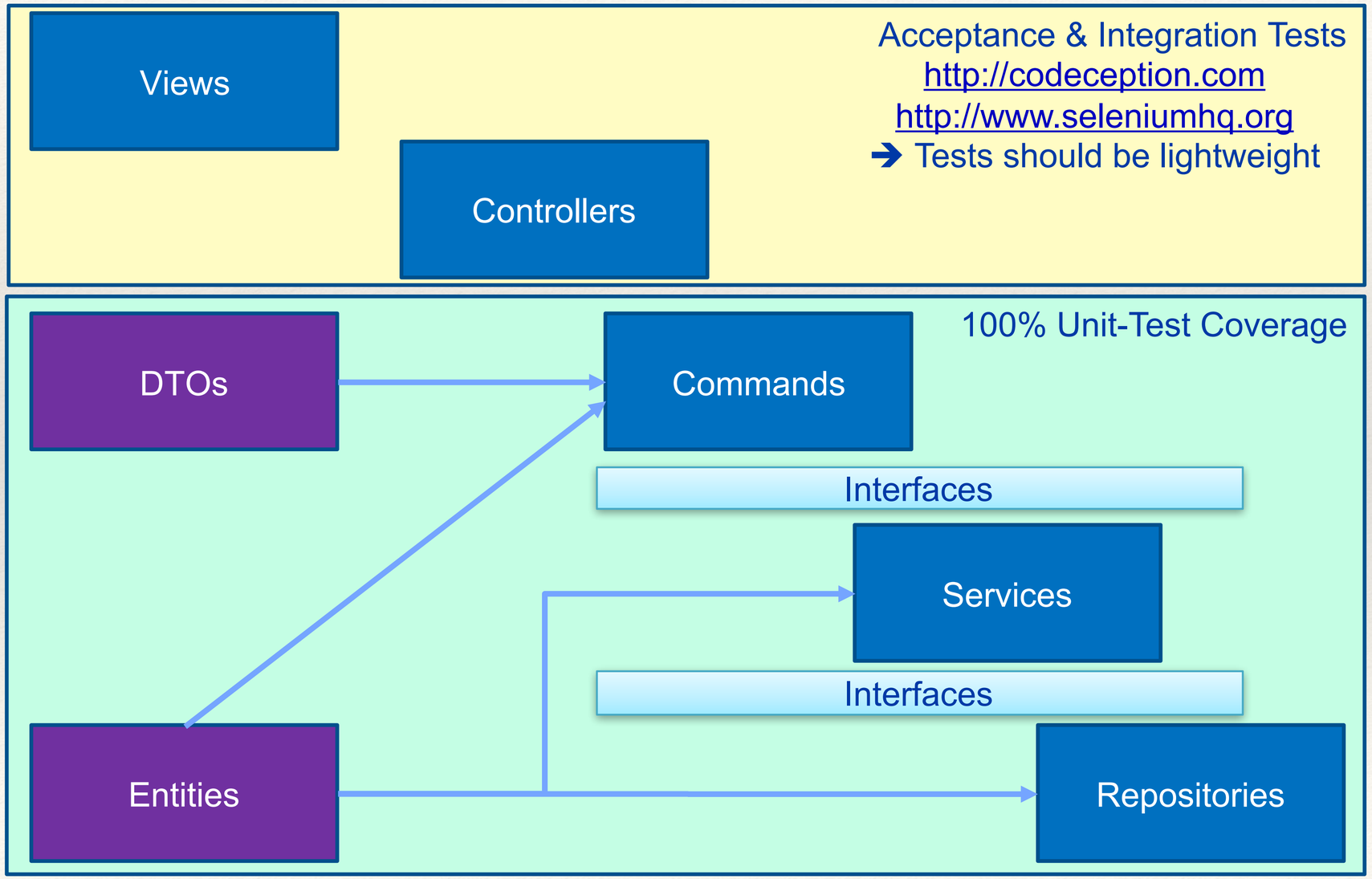
Scope: Symfony 3.4 & Beyond

Scope: Symfony 3.4 & Beyond

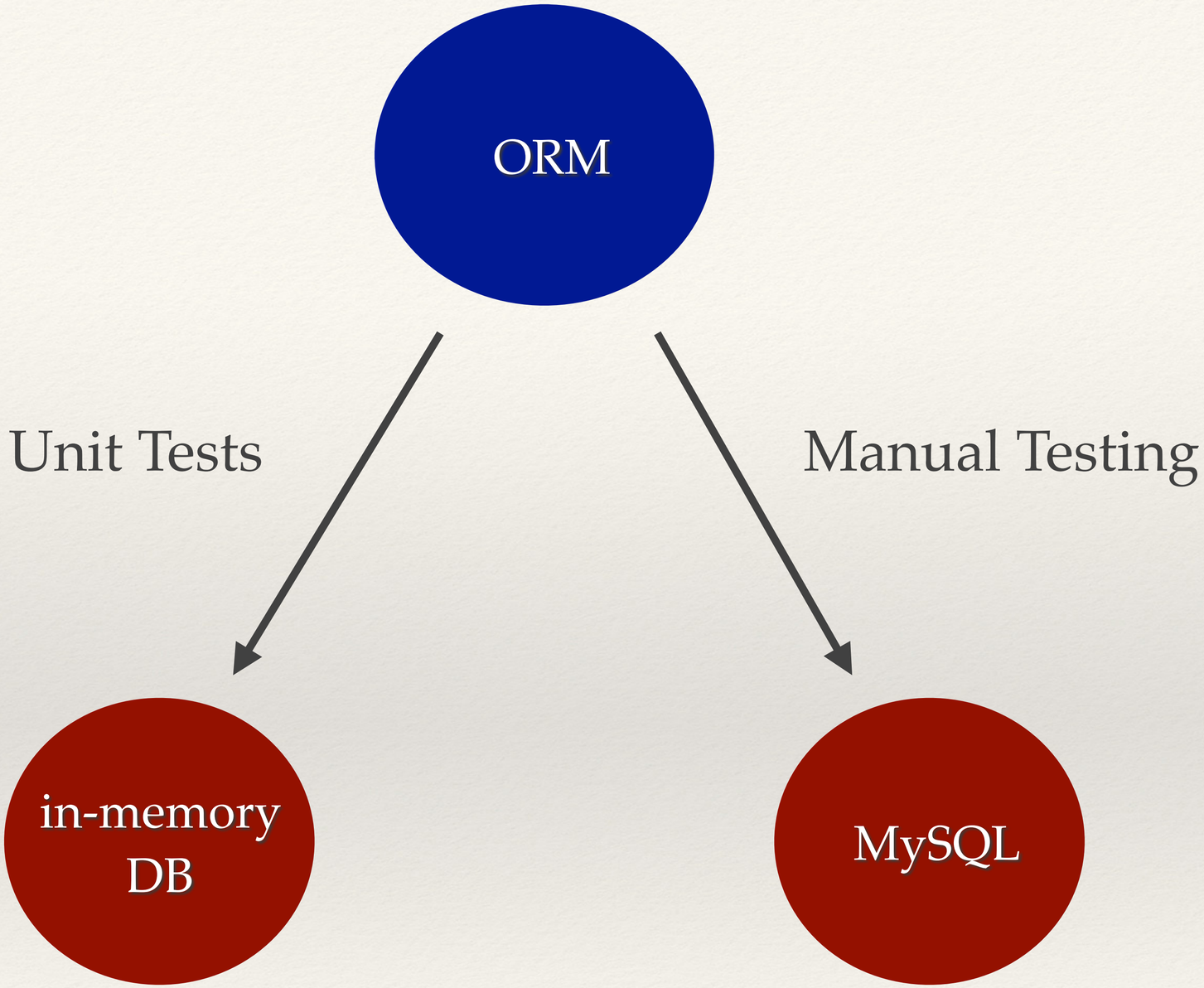
- Concepts are applicable to any framework
- Symfony makes it easy
  - everything works as of latest Symfony 3.4
    - I've not yet ported it to 4.x
      - I don't think it'll be hard

Architecture

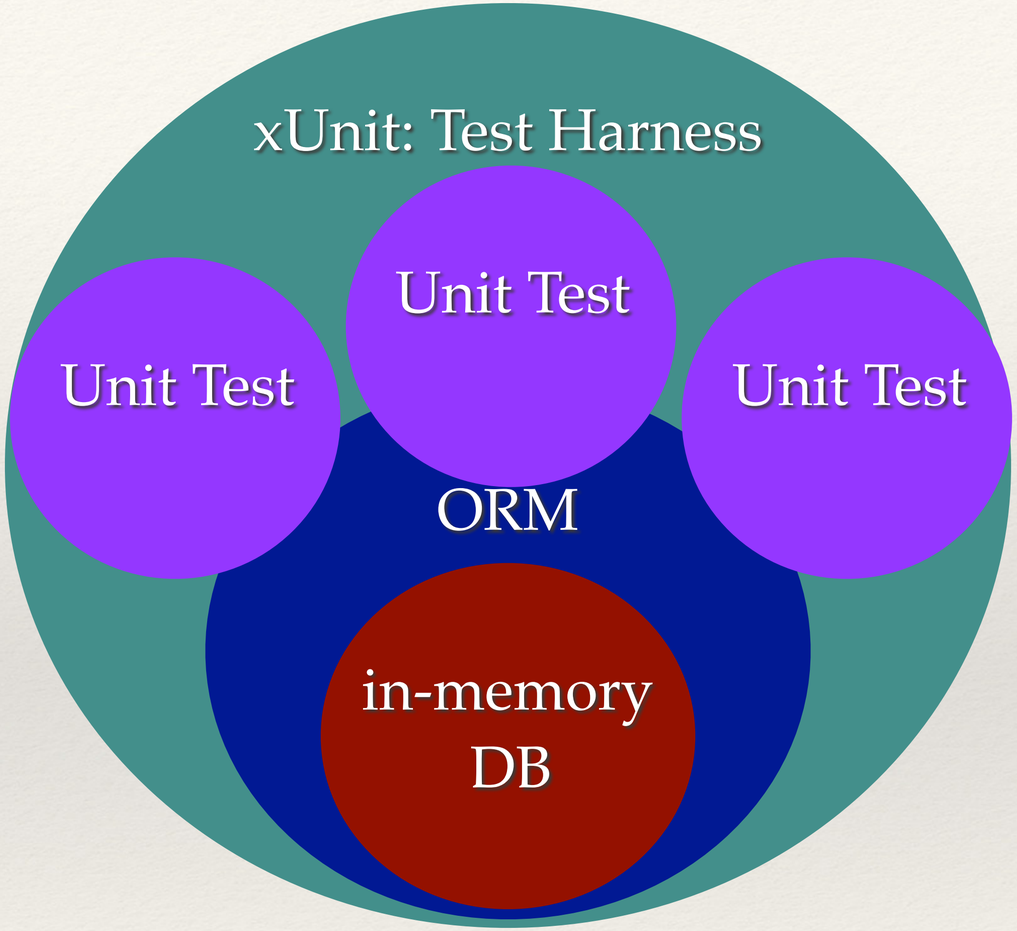
TDD-Friendly Architecture



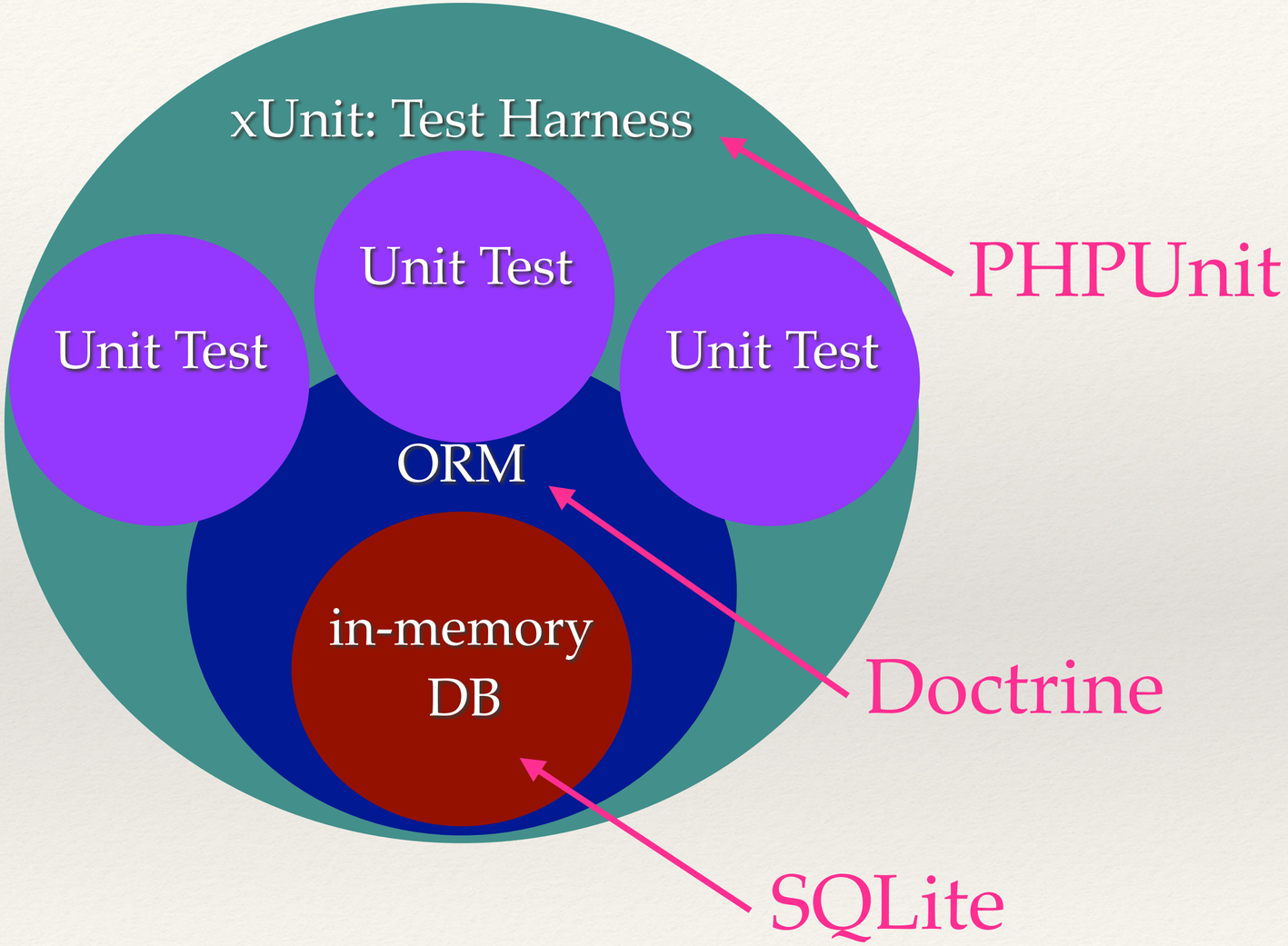
Test-Driving Repositories


ORM
Unit Tests
Manual Testing
in-memory DB
MySQL

TDD'ing Repositories: PHP



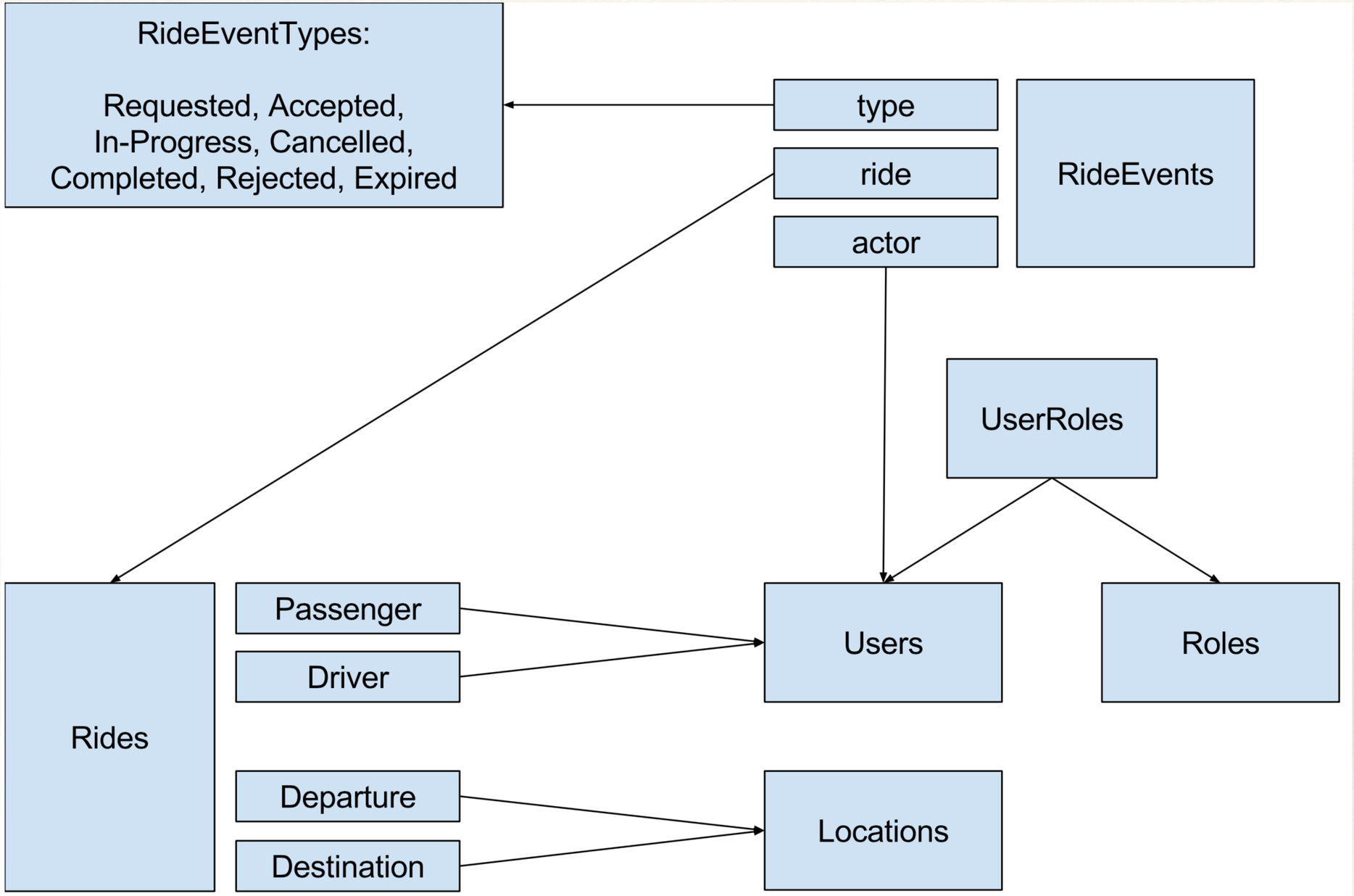
TDD'ing Repositories: PHP



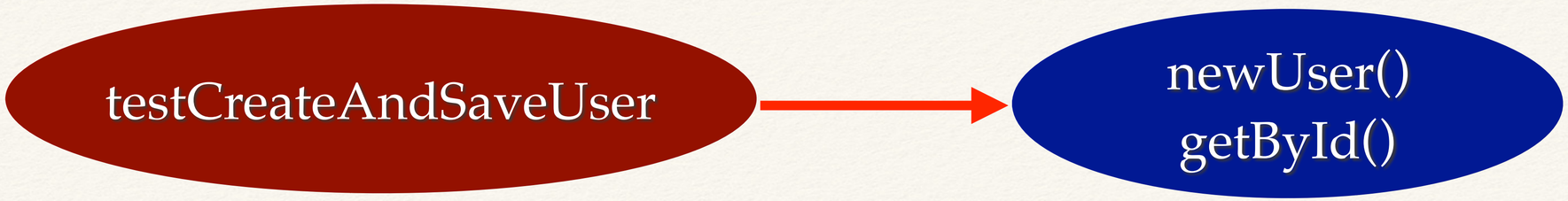
Test-Driving Repositories

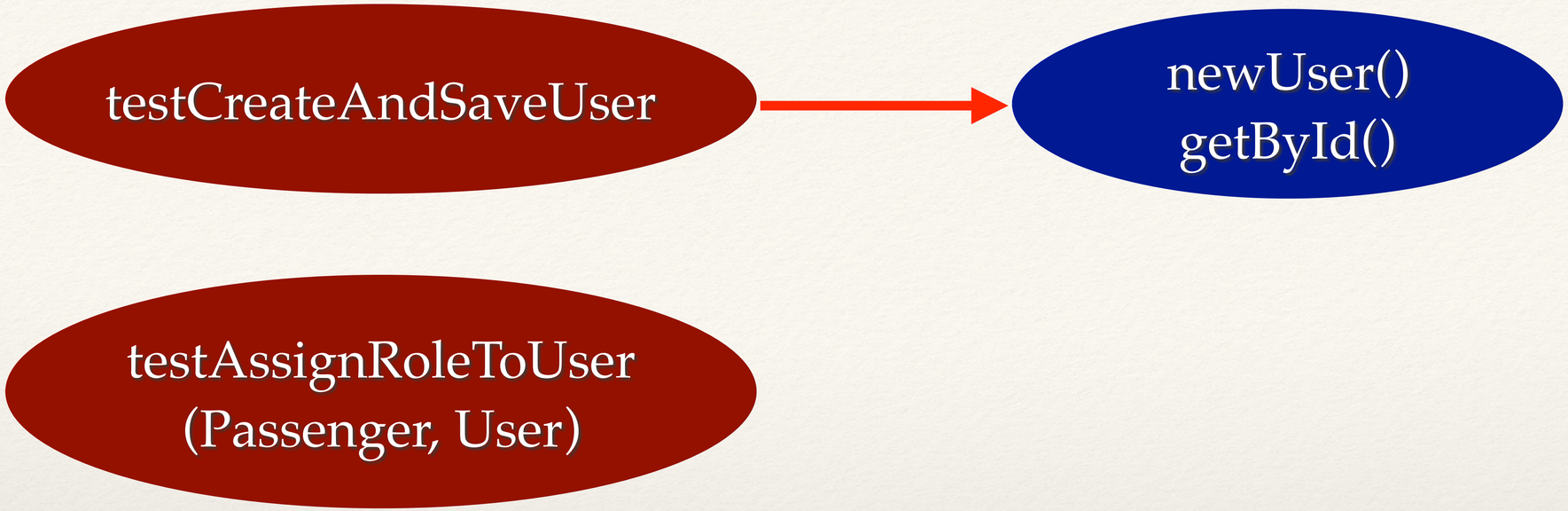
- Being able to Test-Drive Repositories …
  - … changes **Everything**

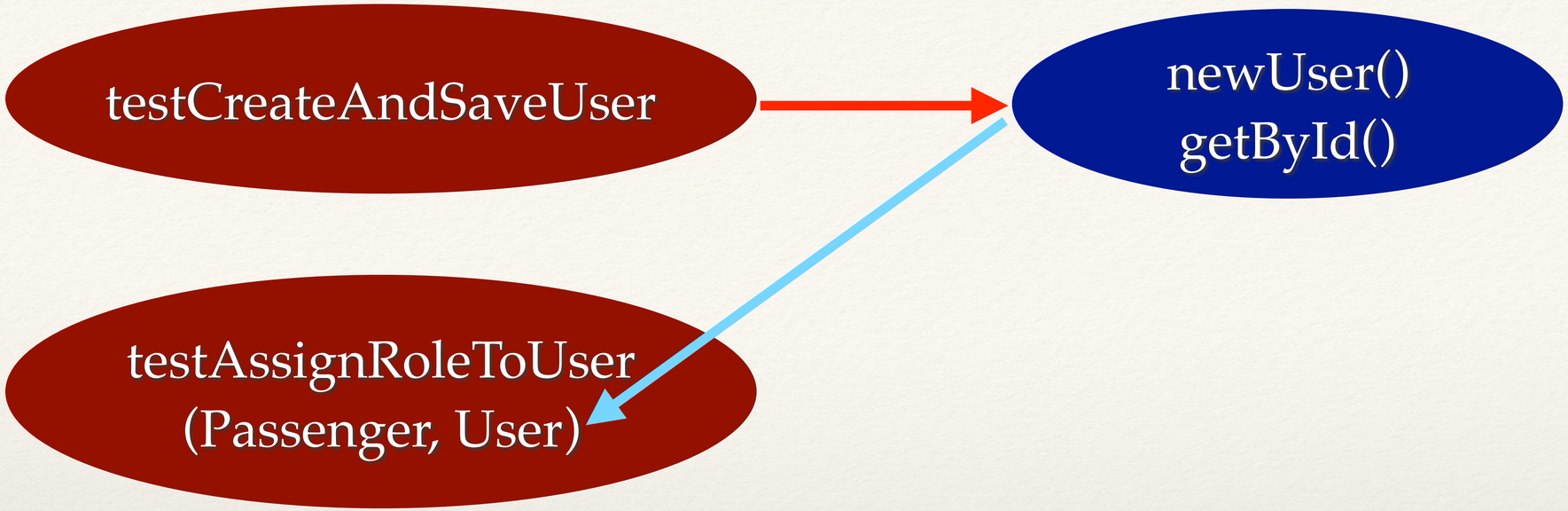
Test-Driving Repositories
Example: Ride-Hailing

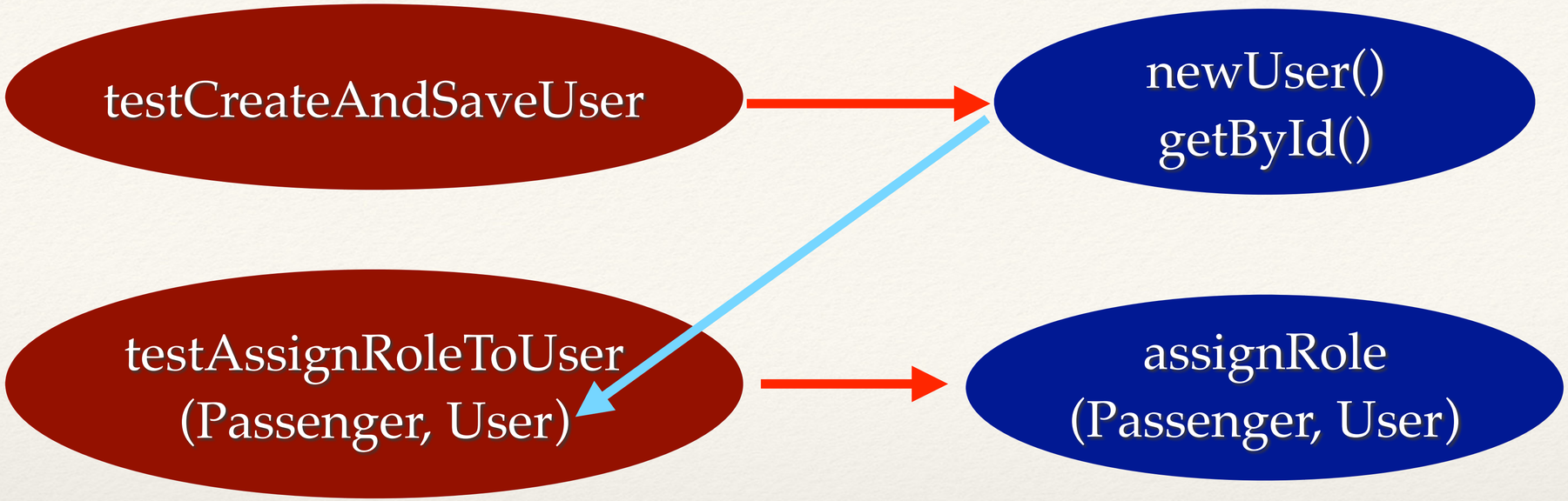

RideEventTypes:
Requested, Accepted, In-Progress, Cancelled, Completed, Rejected, Expired
type
ride
actor
RideEvents
UserRoles
Rides
Passenger
Driver
Users
Roles
Departure
Destination
Locations

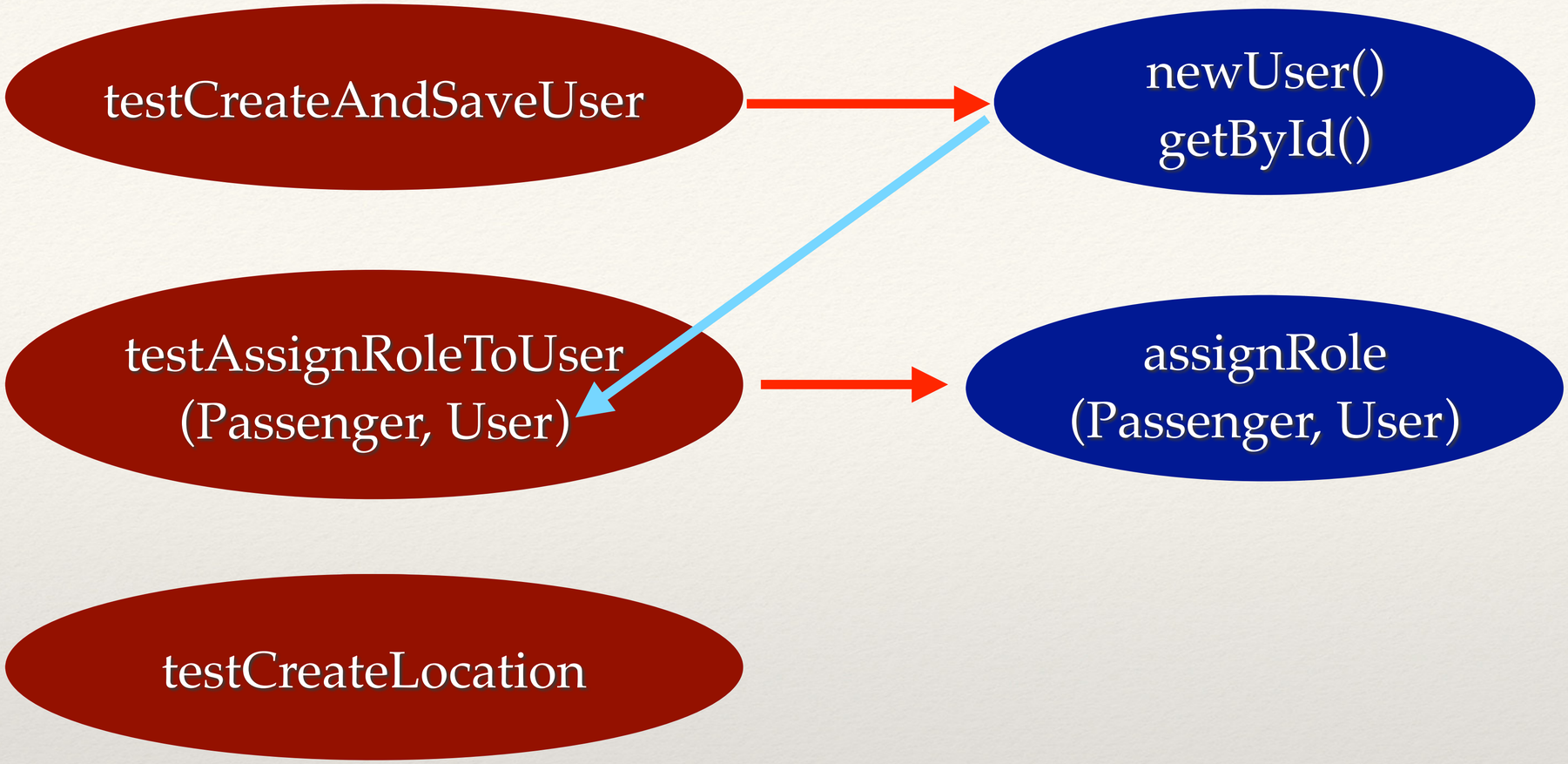
testCreateAndSaveUser

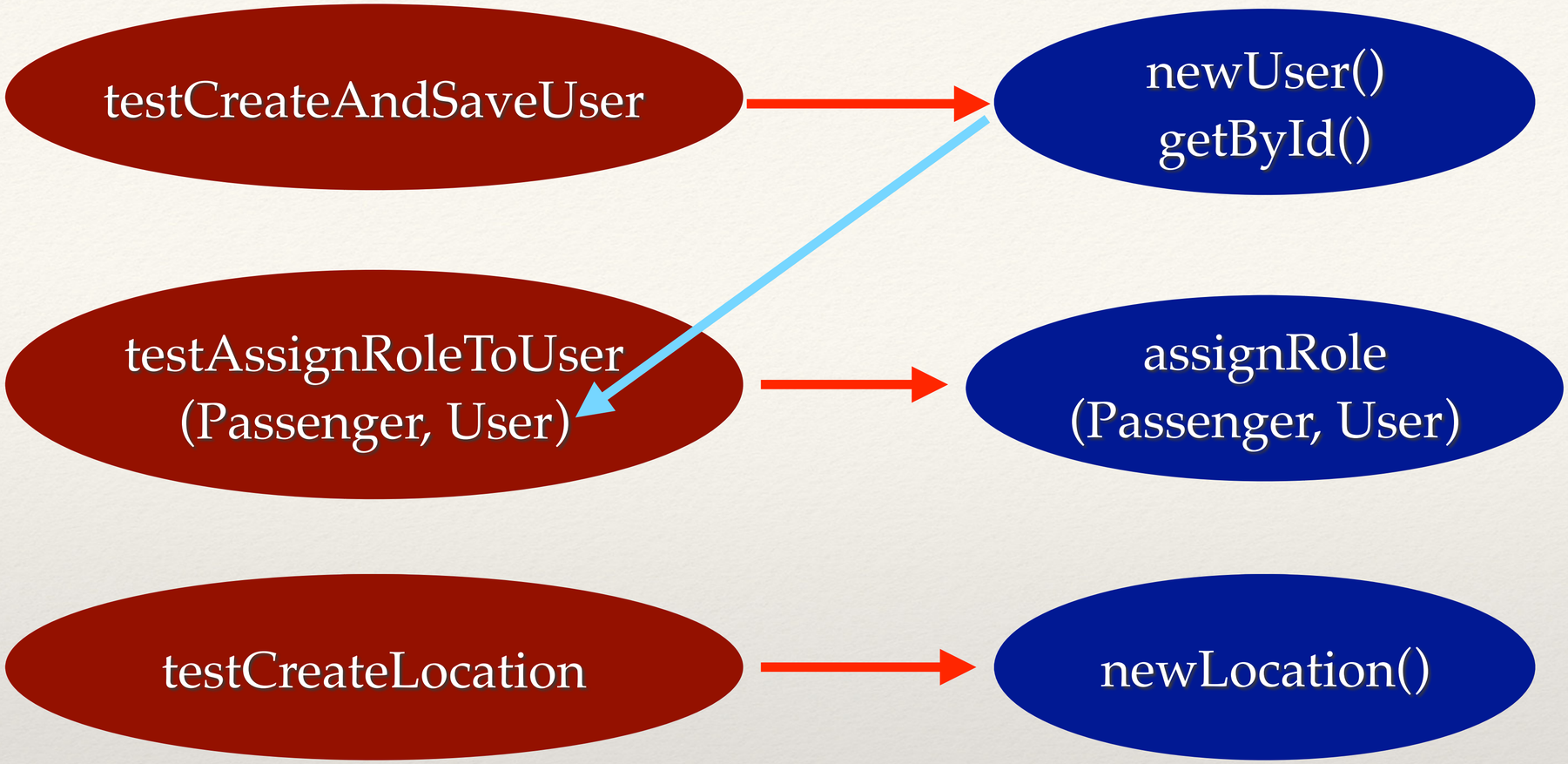

testCreateAndSaveUser
newUser()
getById()

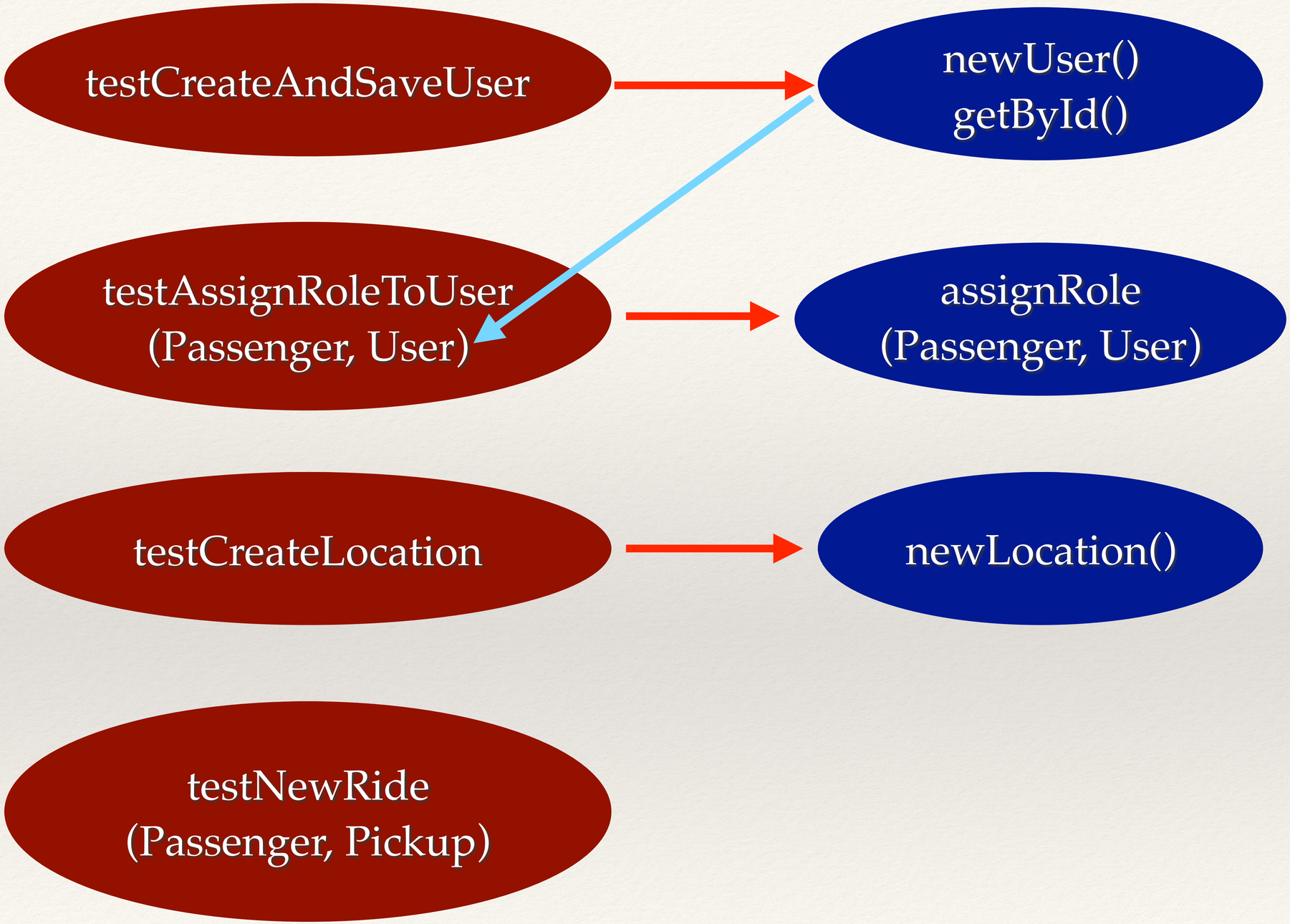

testCreateAndSaveUser
newUser()
getById()
testAssignRoleToUser
(Passenger, User)

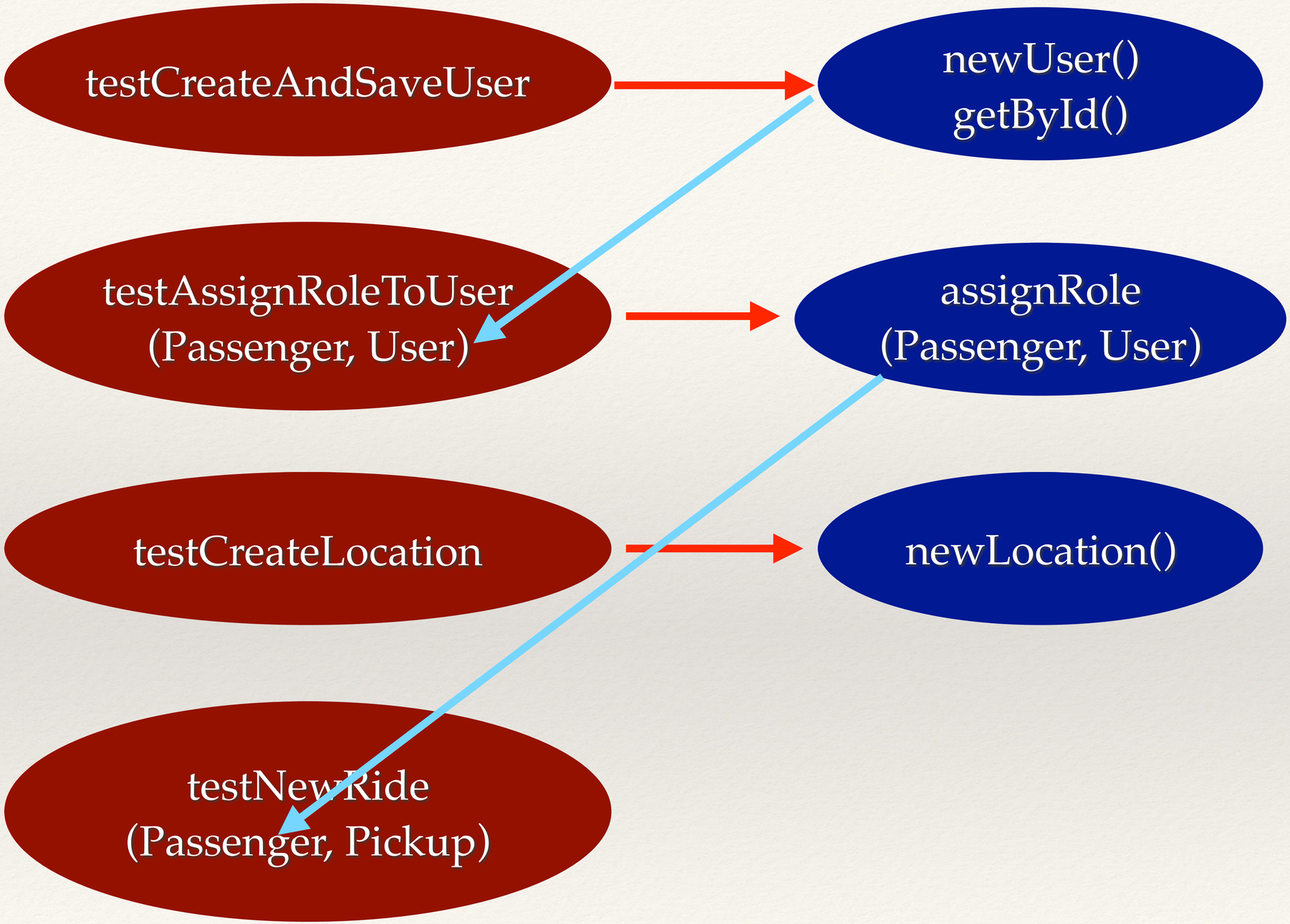

testCreateAndSaveUser
newUser()
getById()
testAssignRoleToUser
(Passenger, User)

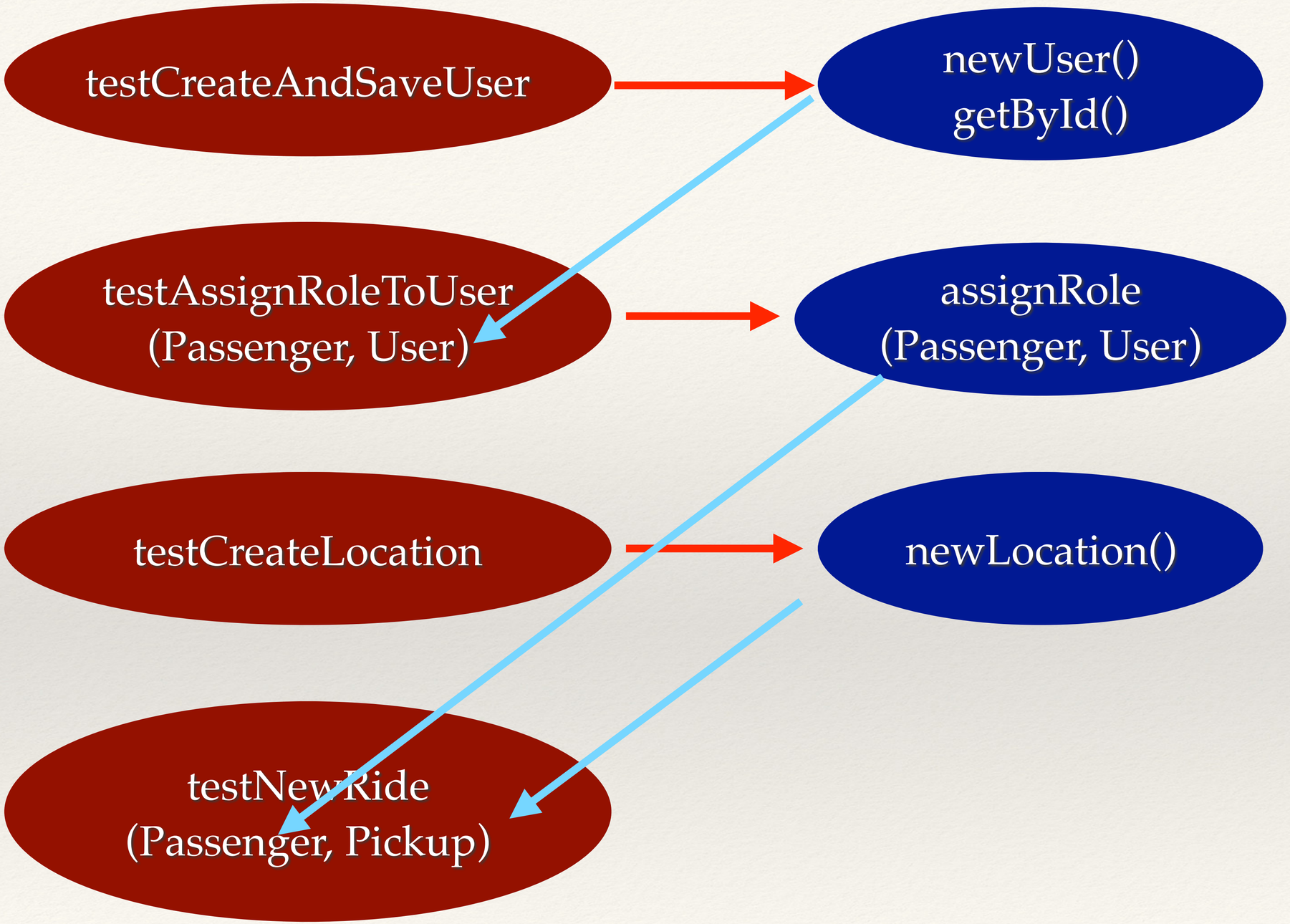

testCreateAndSaveUser
newUser()
getById()
testAssignRoleToUser
(Passenger, User)
assignRole
(Passenger, User)

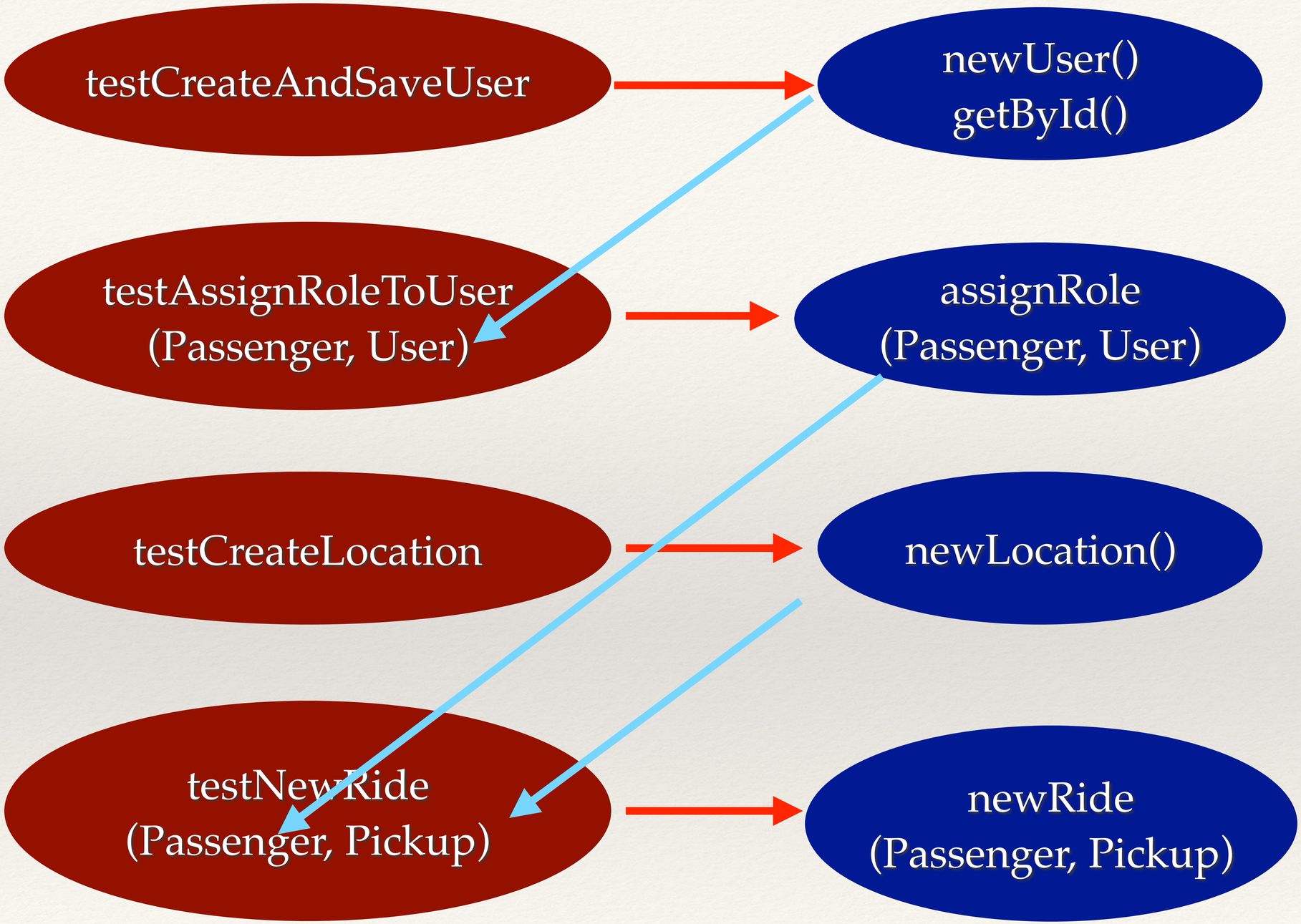

testCreateAndSaveUser
newUser()
getById()
testAssignRoleToUser
(Passenger, User)
assignRole
(Passenger, User)
testCreateLocation


testCreateAndSaveUser
newUser()
getById()
testAssignRoleToUser
(Passenger, User)
assignRole
(Passenger, User)
testCreateLocation
newLocation()


testCreateAndSaveUser
newUser()
getById()
testAssignRoleToUser
(Passenger, User)
assignRole
(Passenger, User)
testCreateLocation
newLocation()
testNewRide
(Passenger, Pickup)

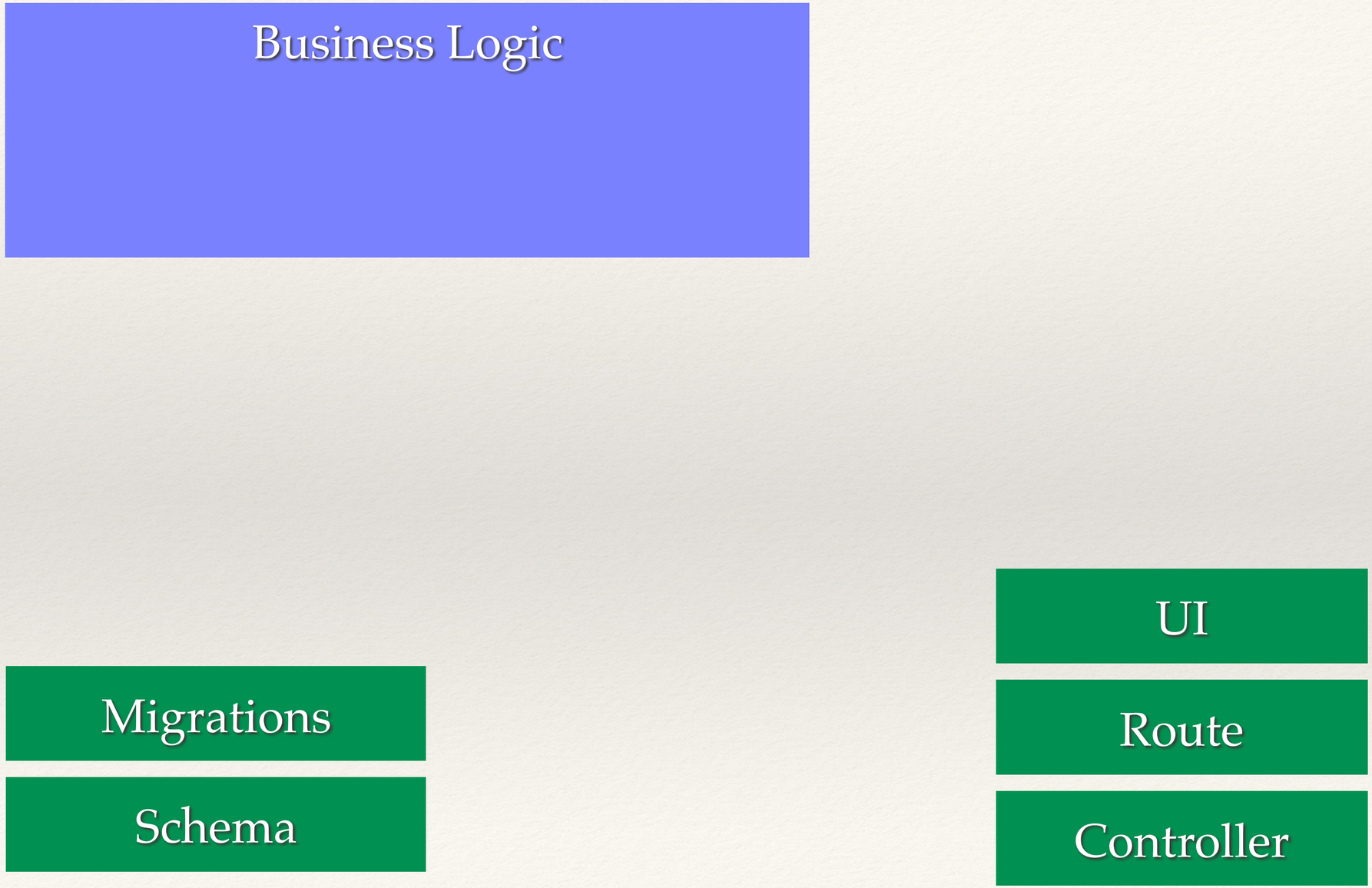

testCreateAndSaveUser
newUser()
getById()
testAssignRoleToUser
(Passenger, User)
assignRole
(Passenger, User)
testCreateLocation
newLocation()
testNewRide
(Passenger, Pickup)


testCreateAndSaveUser
newUser()
getById()
testAssignRoleToUser
(Passenger, User)
assignRole
(Passenger, User)
testCreateLocation
newLocation()
testNewRide
(Passenger, Pickup)


testCreateAndSaveUser
newUser()
getById()
testAssignRoleToUser
(Passenger, User)
assignRole
(Passenger, User)
testCreateLocation
newLocation()
testNewRide
(Passenger, Pickup)
newRide
(Passenger, Pickup)

Workflow

TDD Workflow



TDD Workflow

Migrations

Schema

UI

Route

Controller

TDD Workflow

Migrations

Schema

UI

Route

Controller

TDD Workflow

Migrations

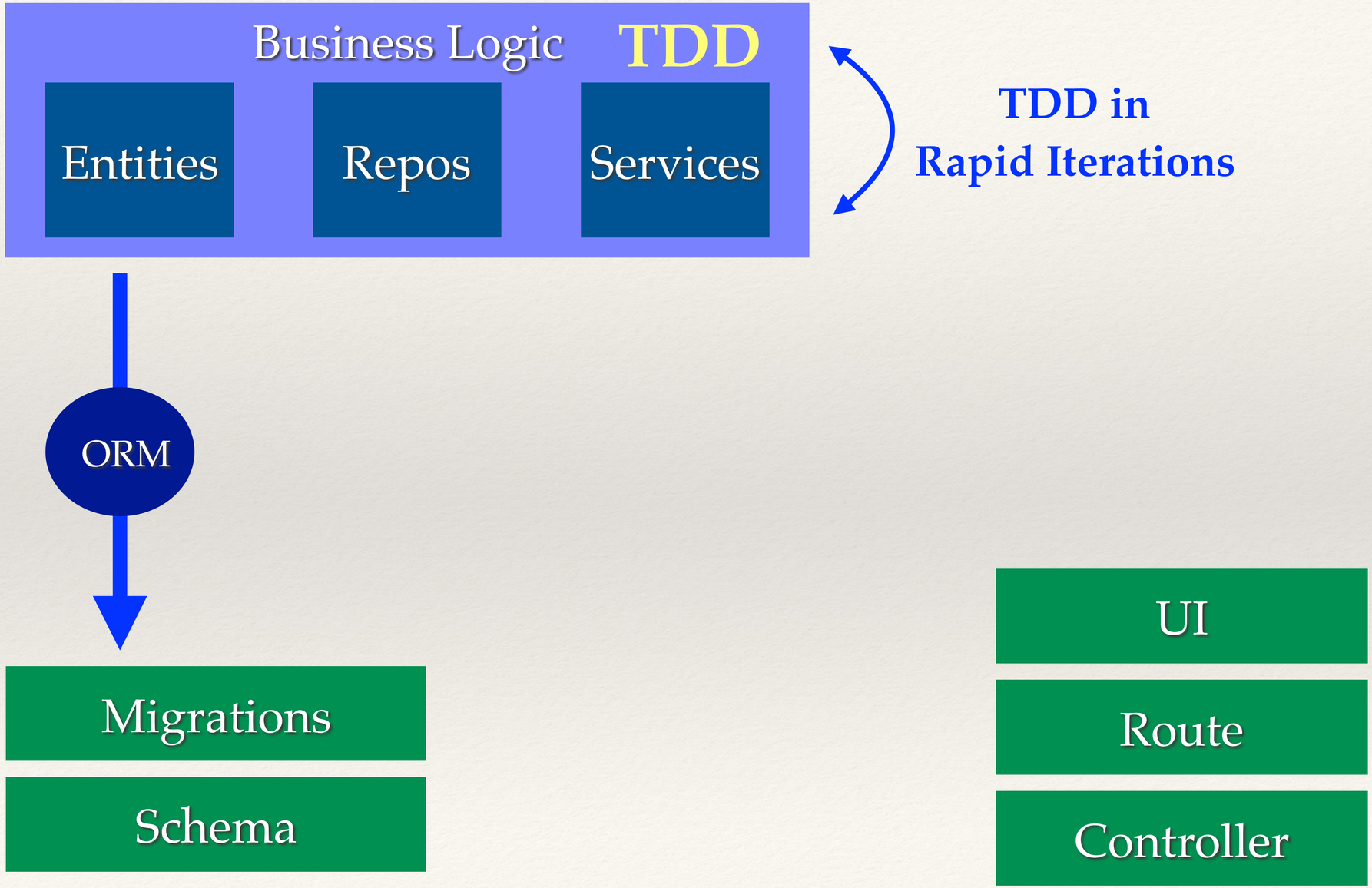
Schema

UI

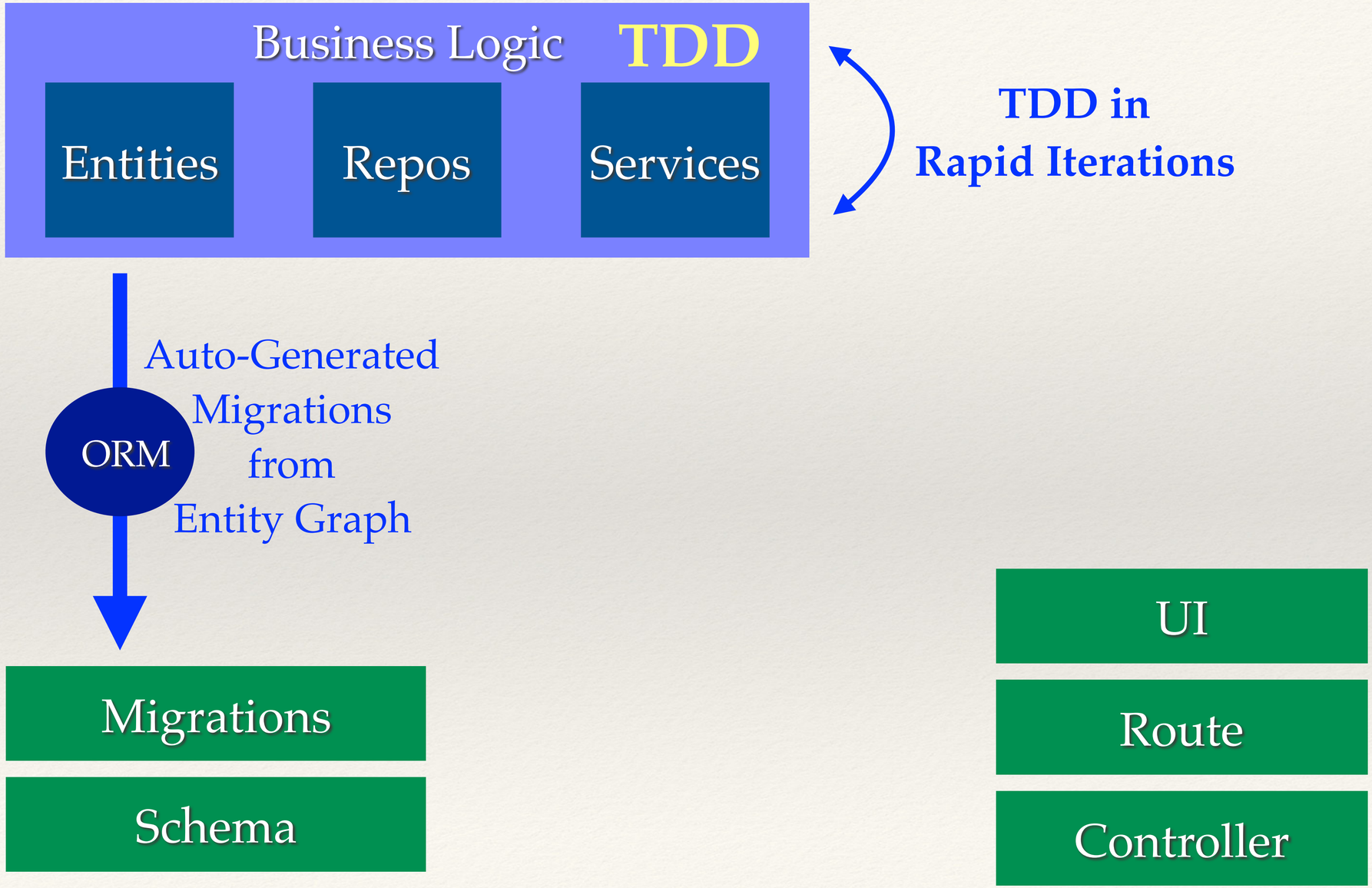
Route

Controller

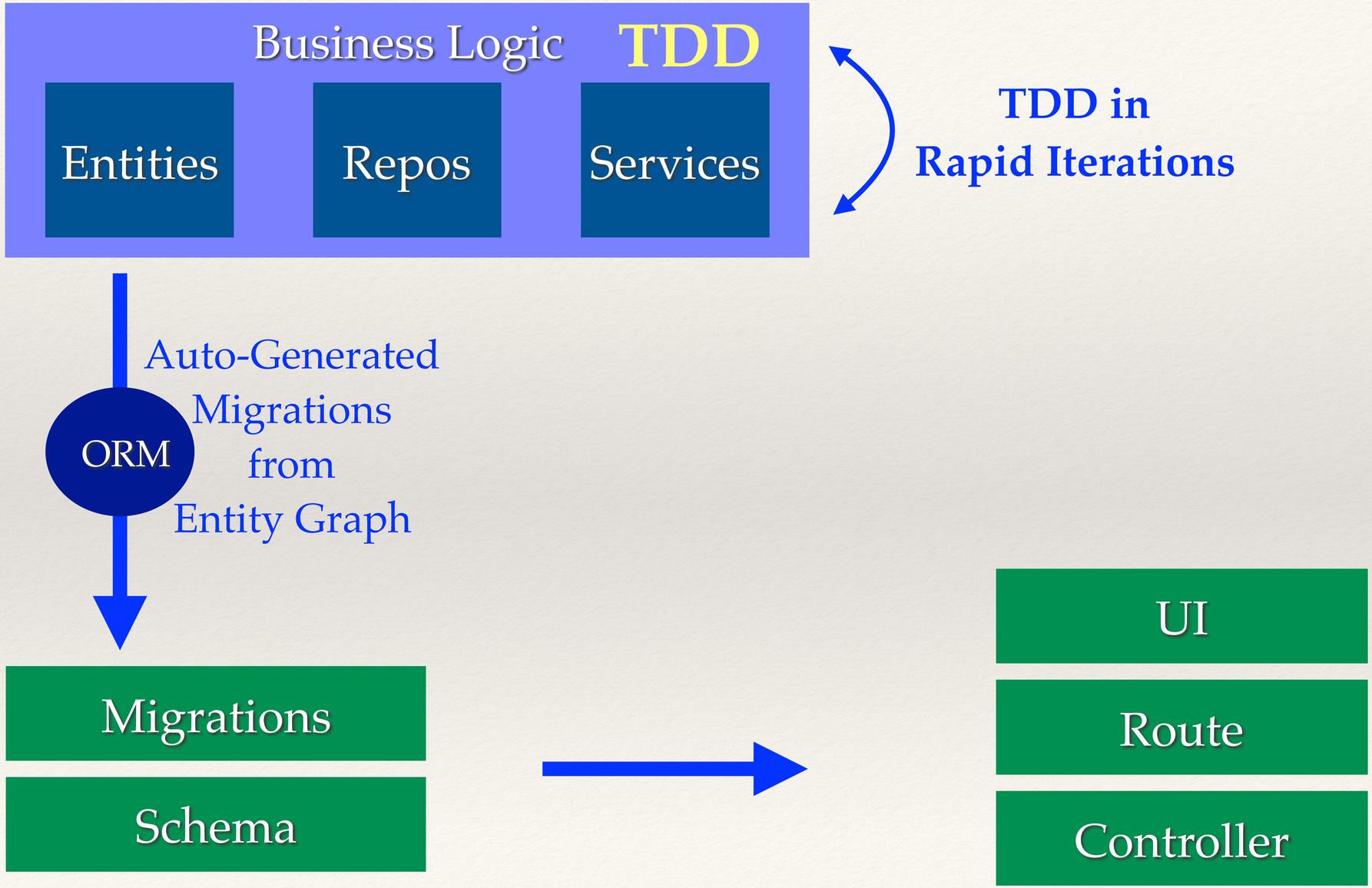
TDD Workflow



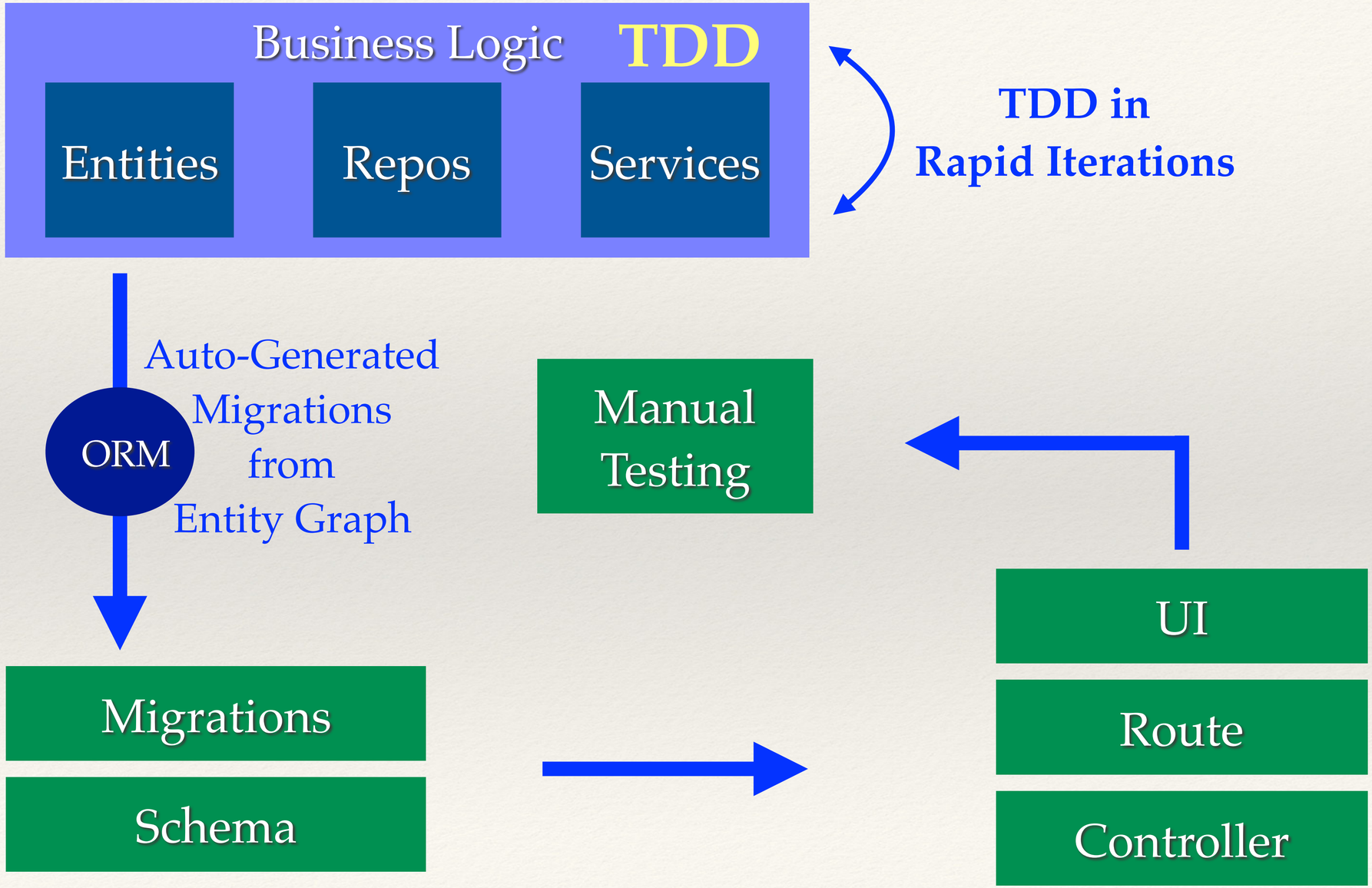
TDD Workflow



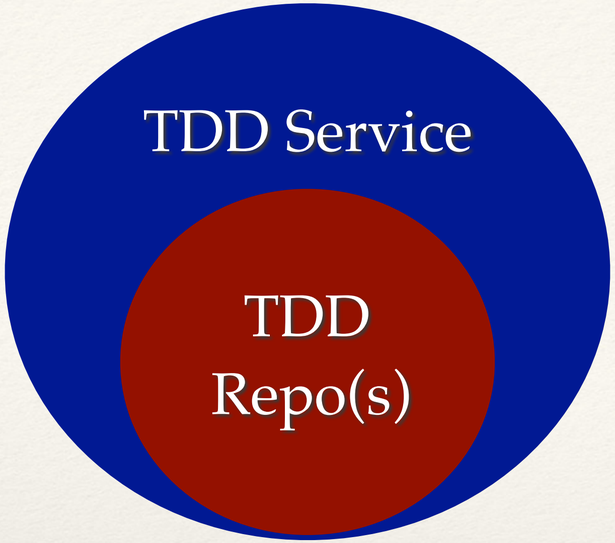
TDD Workflow

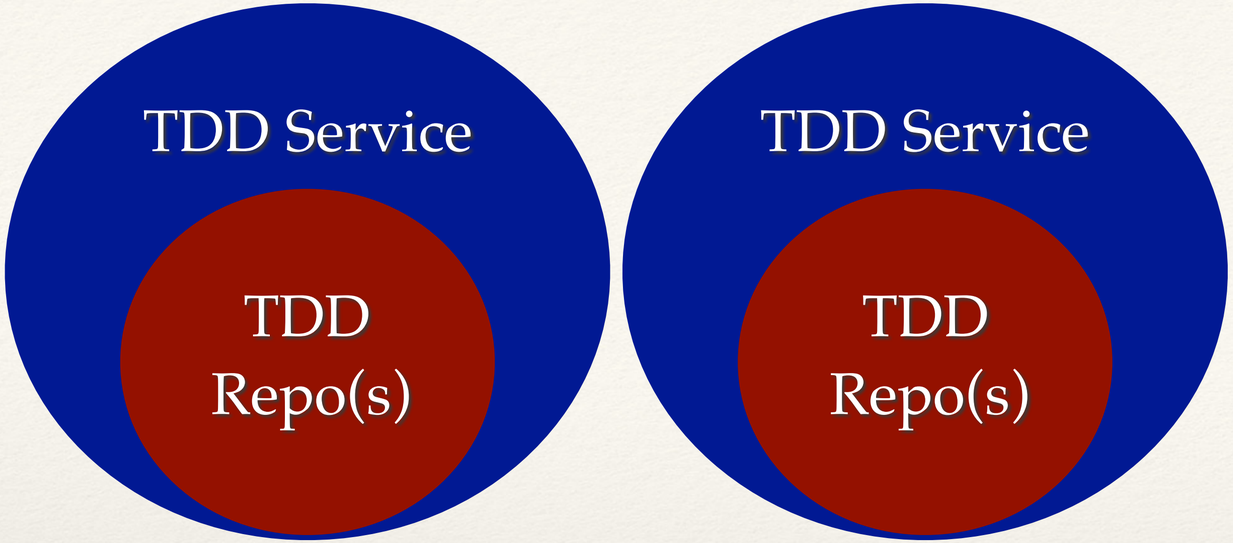


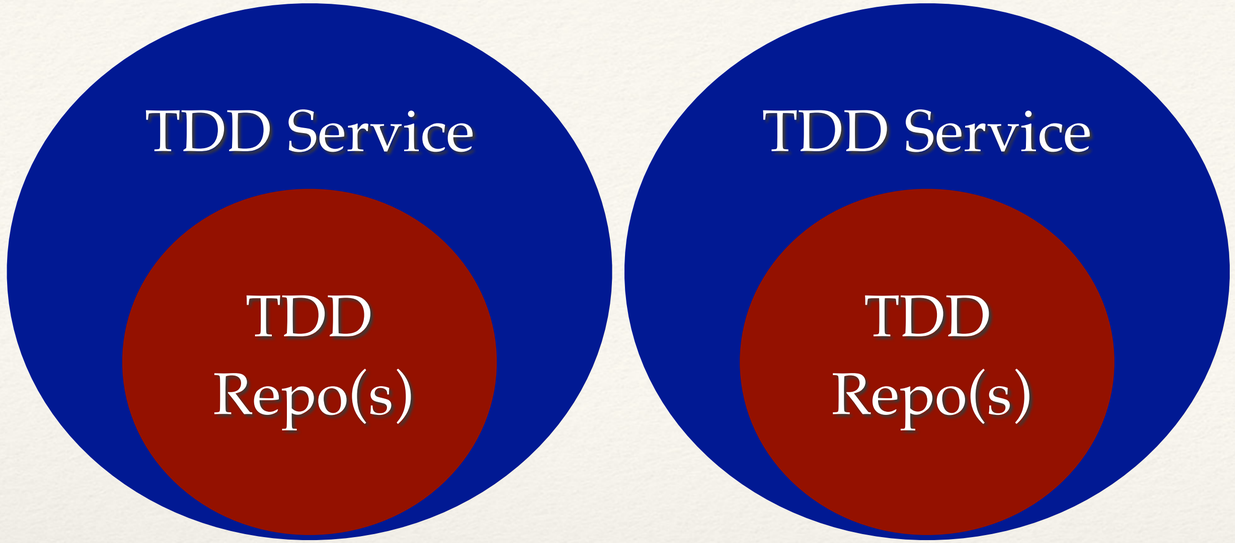
TDD Workflow



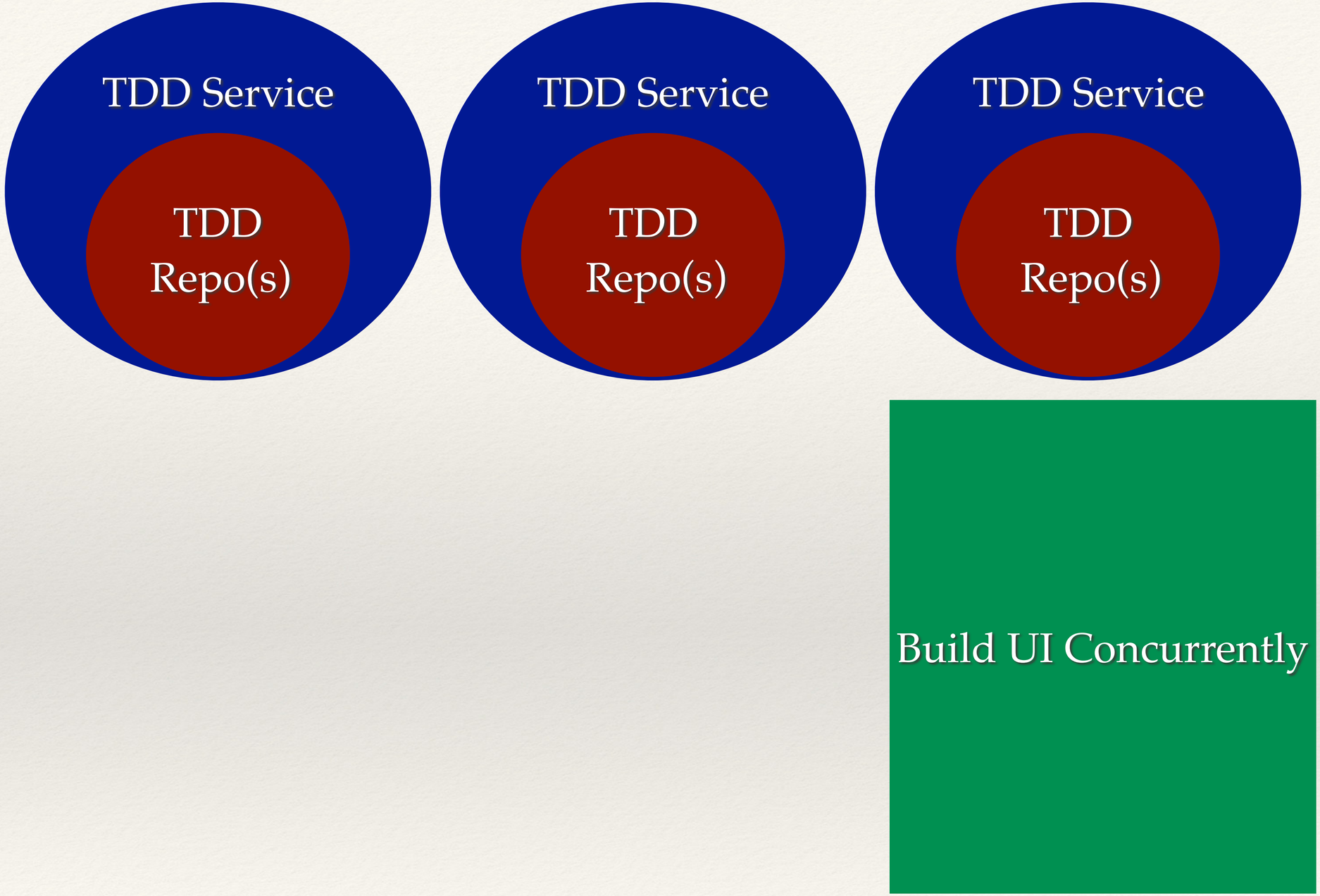
TDD
Repo(s)

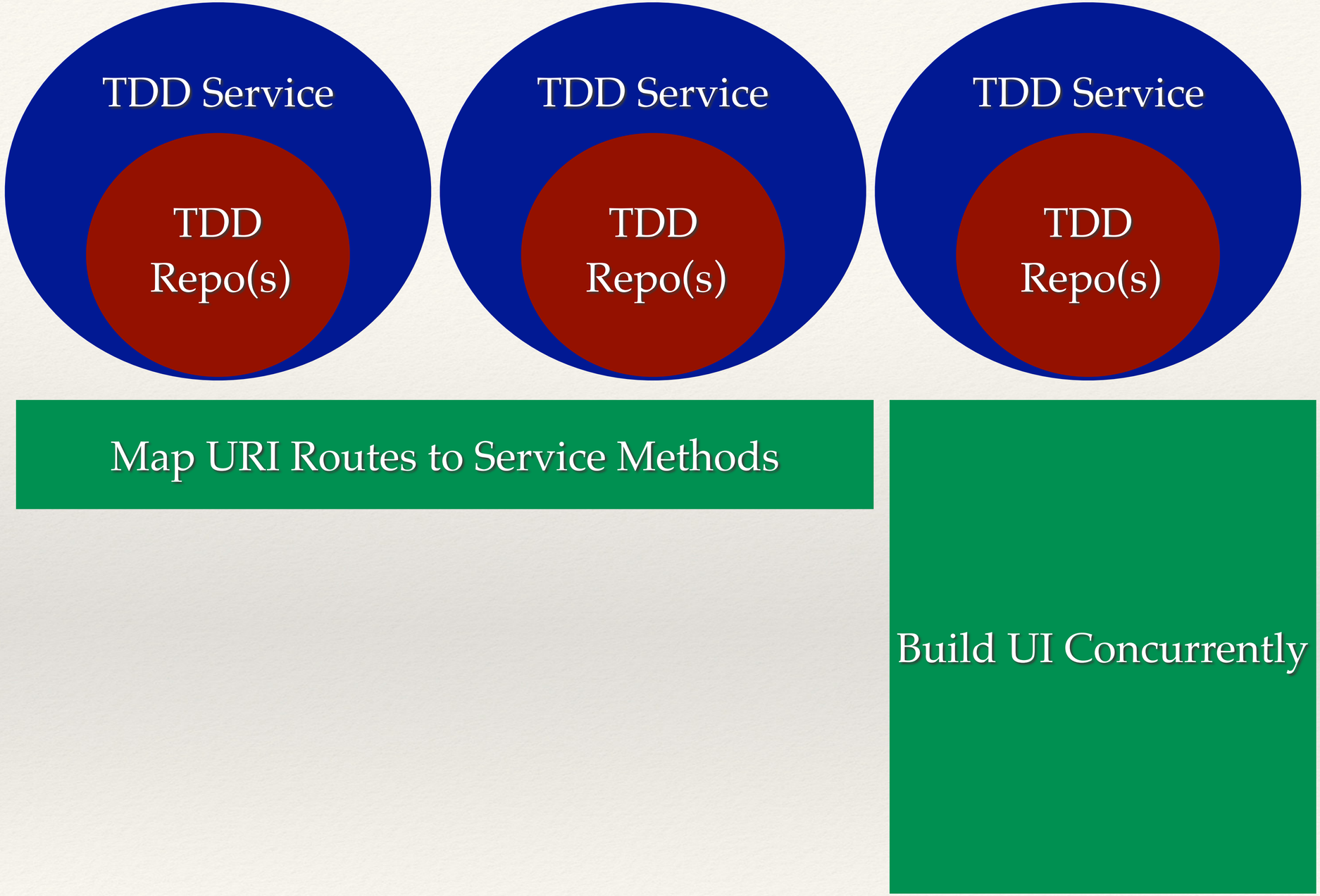

TDD Service
TDD
Repo(s)

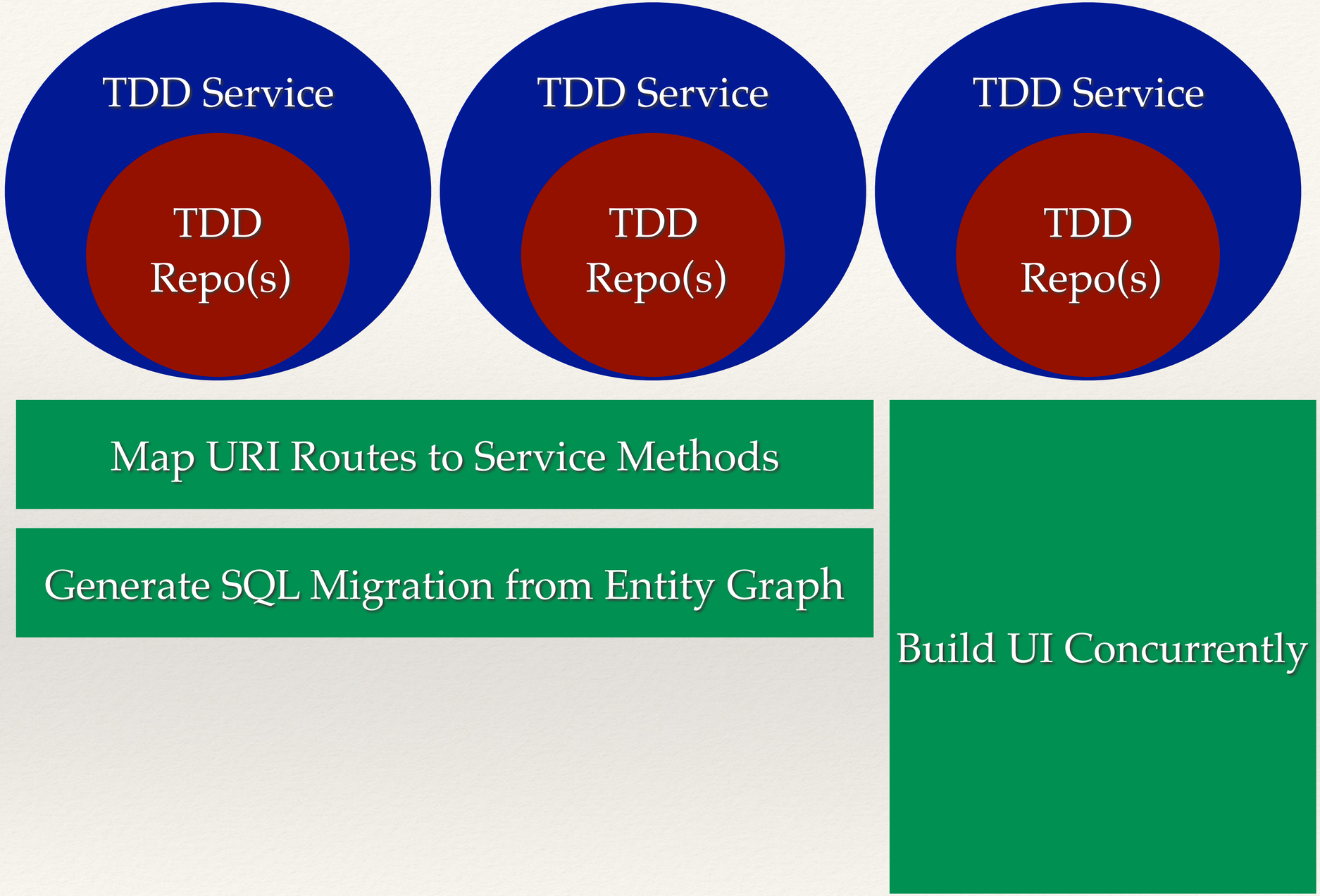

TDD Service
TDD
Repo(s)
TDD Service
TDD
Repo(s)

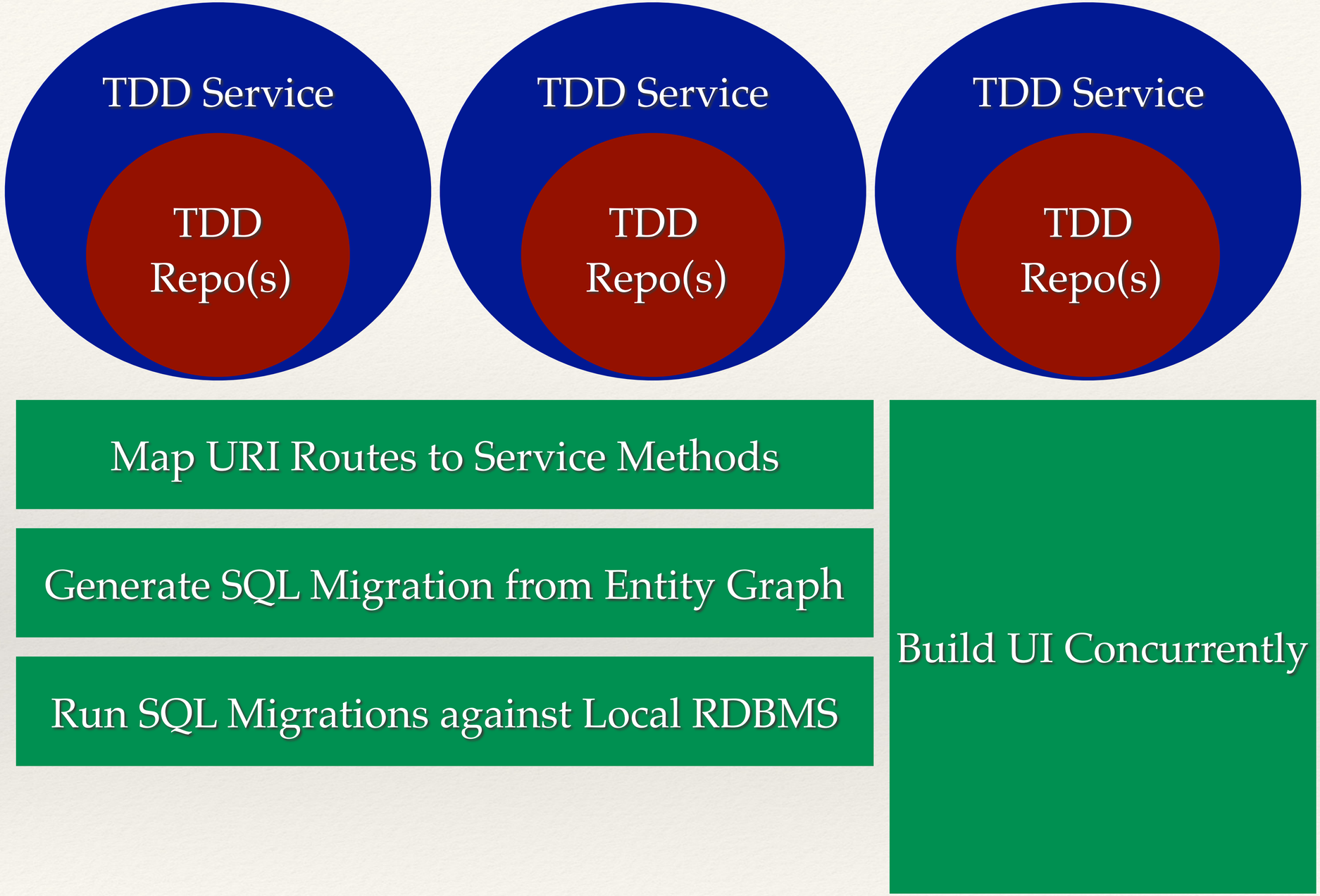


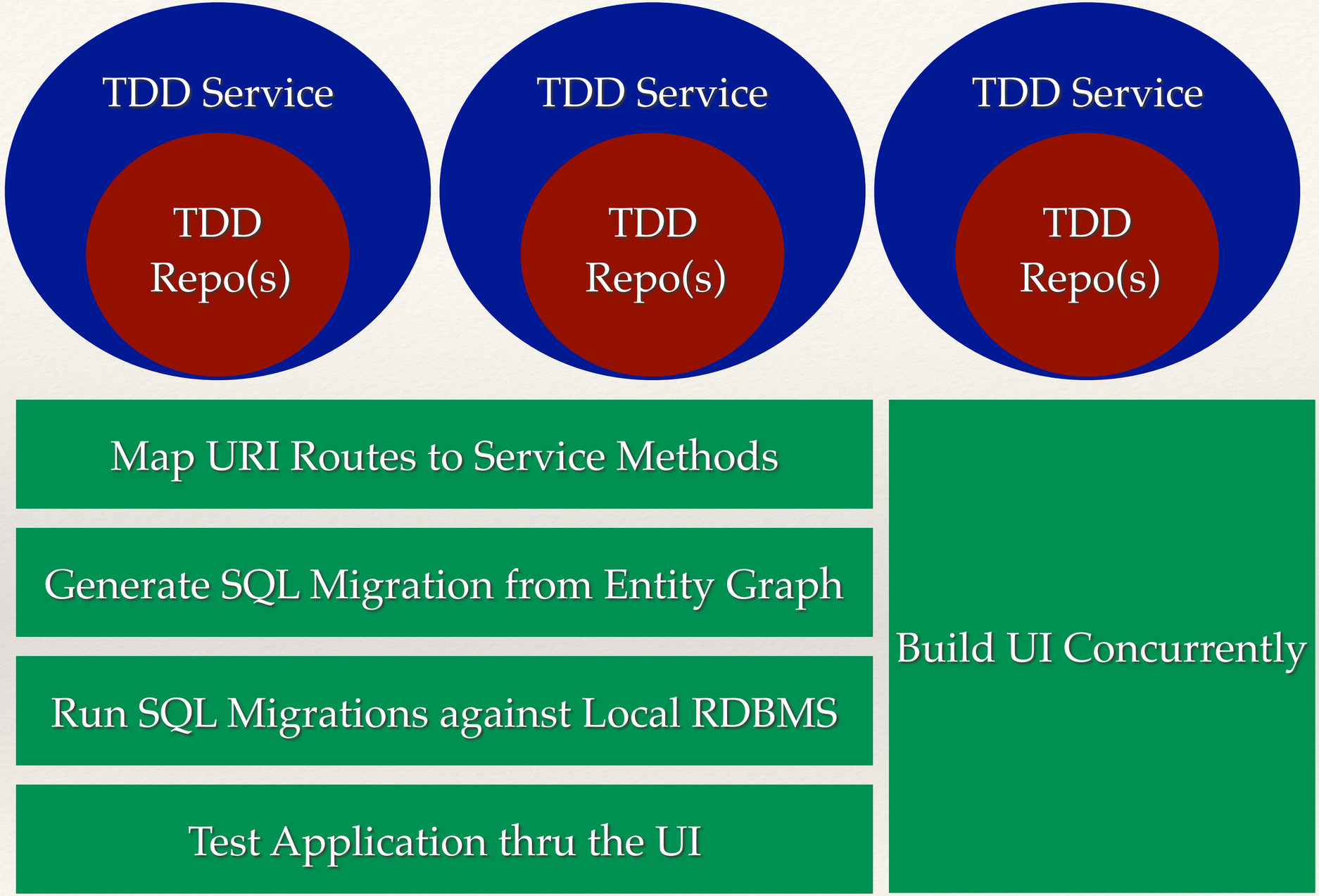
Build UI Concurrently

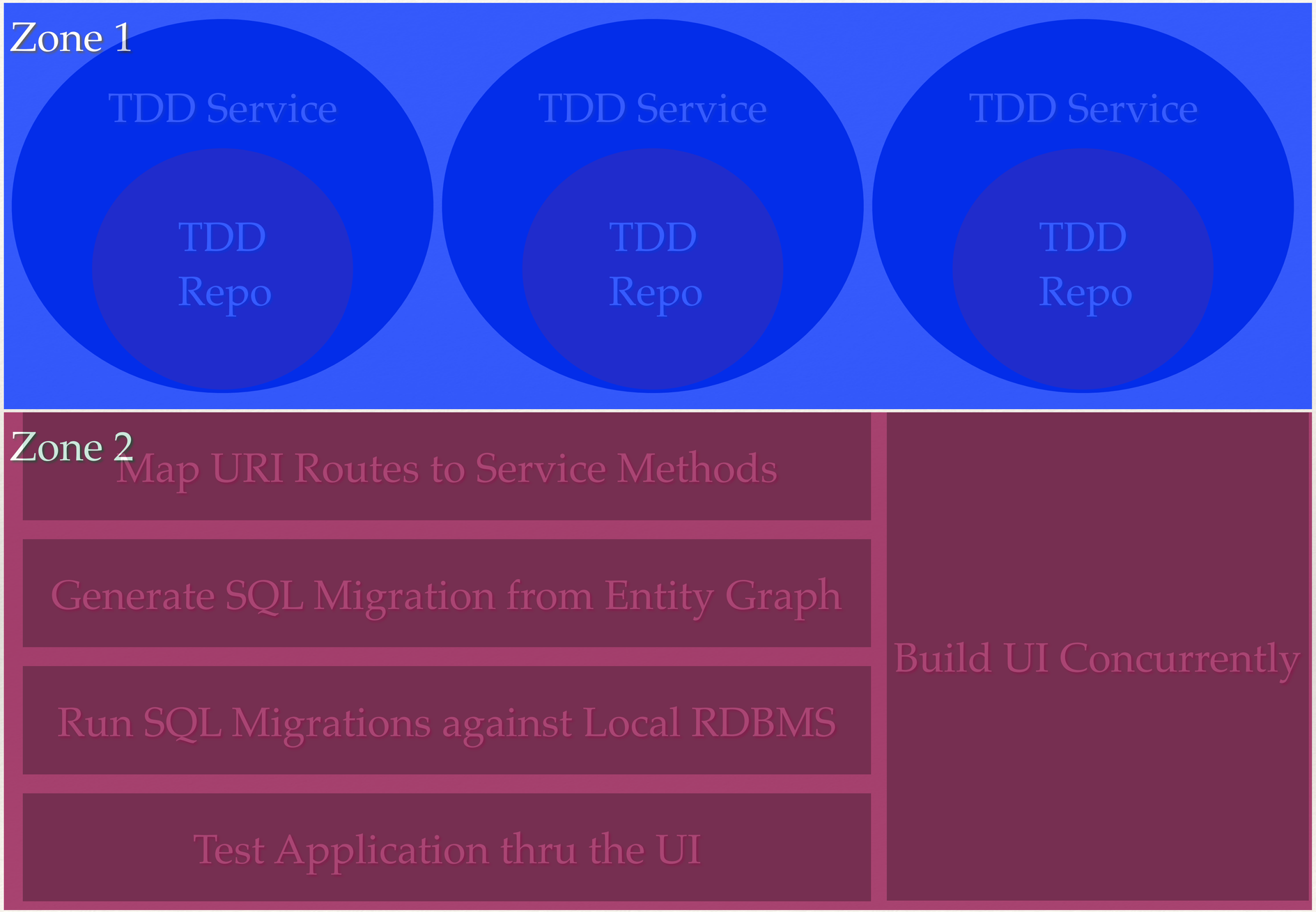

TDD Service
TDD
Repo(s)
TDD Service
TDD
Repo(s)
TDD Service
TDD
Repo(s)
Build UI Concurrently

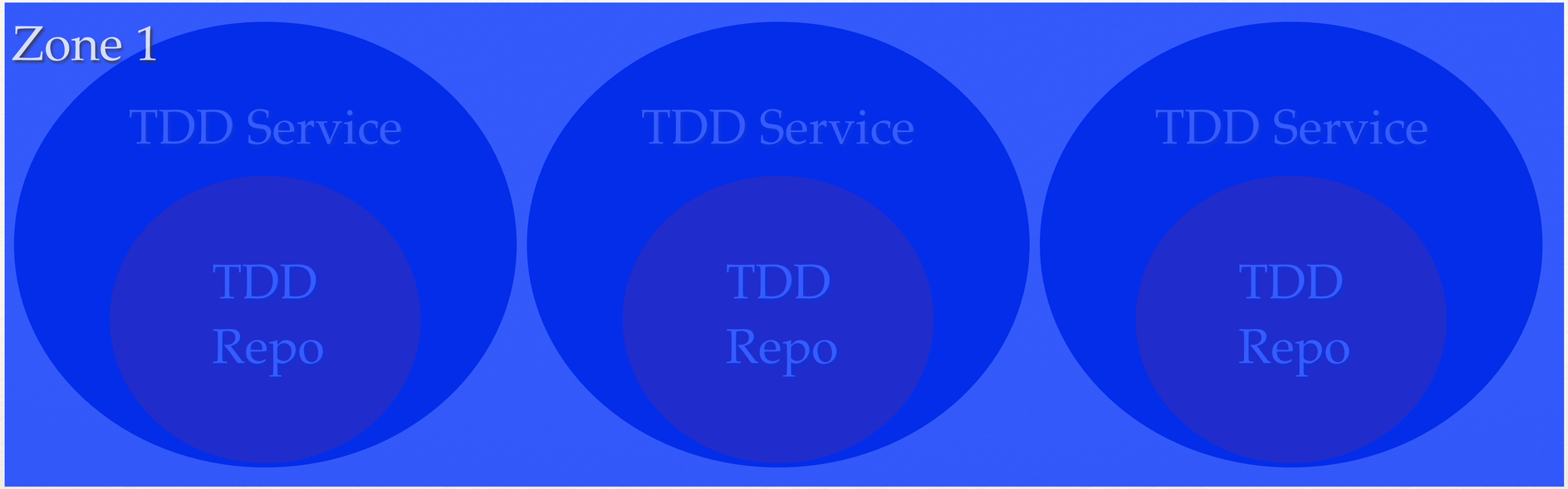

TDD Service
TDD Repo(s)
TDD Service
TDD Repo(s)
TDD Service
TDD Repo(s)
Map URI Routes to Service Methods
Build UI Concurrently

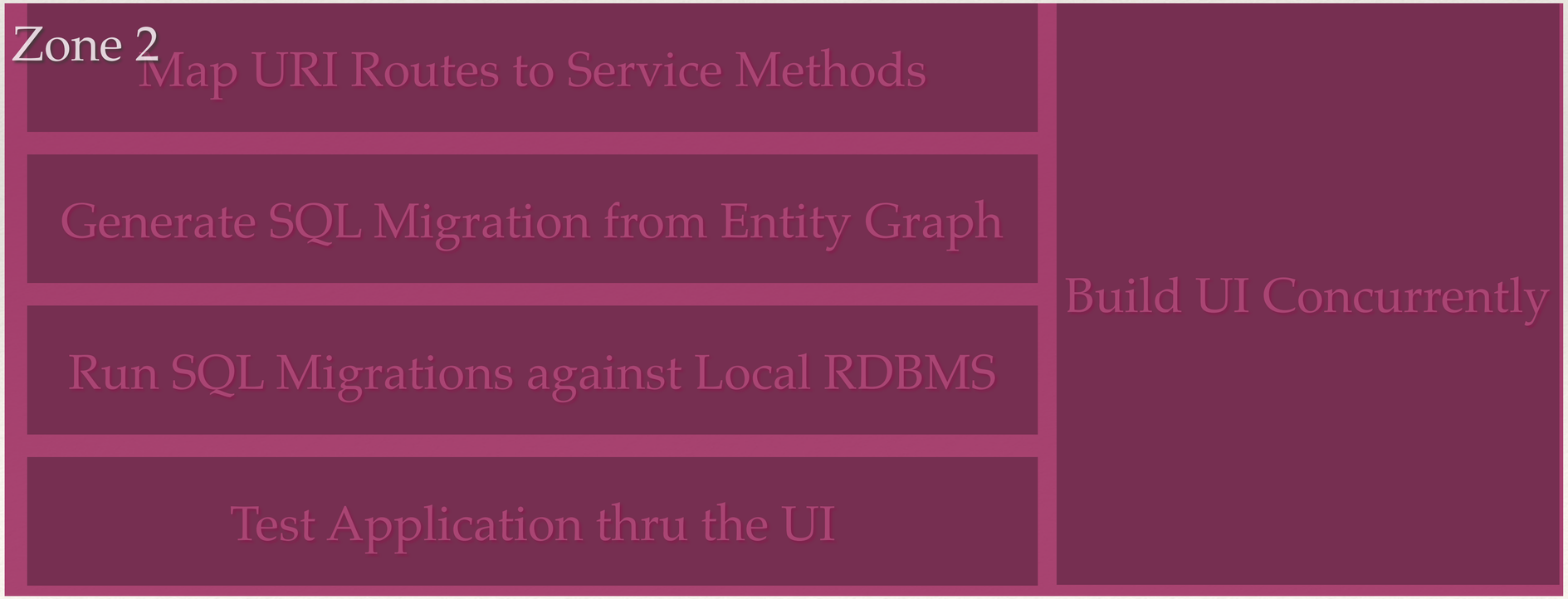

TDD Service
TDD Repo(s)
TDD Service
TDD Repo(s)
TDD Service
TDD Repo(s)
Map URI Routes to Service Methods
Generate SQL Migration from Entity Graph
Build UI Concurrently

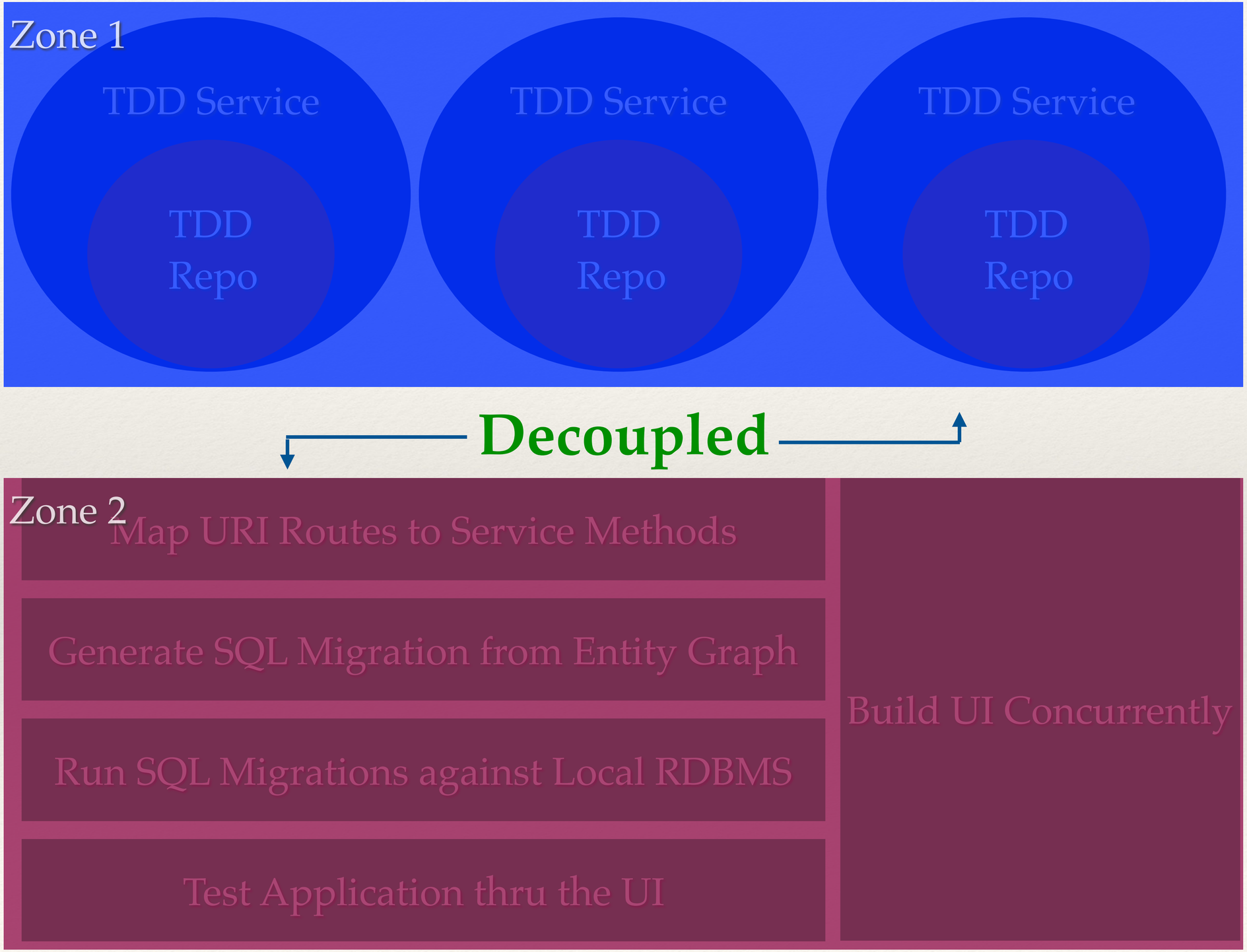

TDD Service
TDD Repo(s)
TDD Service
TDD Repo(s)
TDD Service
TDD Repo(s)
Map URI Routes to Service Methods
Generate SQL Migration from Entity Graph
Run SQL Migrations against Local RDBMS
Build UI Concurrently


TDD Service
TDD Repo(s)
TDD Service
TDD Repo(s)
TDD Service
TDD Repo(s)
Map URI Routes to Service Methods
Generate SQL Migration from Entity Graph
Run SQL Migrations against Local RDBMS
Test Application thru the UI
Build UI Concurrently


Zone 1
TDD Service
TDD Repo
TDD Service
TDD Repo
TDD Service
TDD Repo
Zone 2
Map URI Routes to Service Methods
Generate SQL Migration from Entity Graph
Run SQL Migrations against Local RDBMS
Test Application thru the UI
Build UI Concurrently


Zone 1
TDD Service
TDD Repo
TDD Service
TDD Repo
TDD Service
TDD Repo


Zone 2
Map URI Routes to Service Methods
Generate SQL Migration from Entity Graph
Run SQL Migrations against Local RDBMS
Test Application thru the UI
Build UI Concurrently


Zone 1
TDD Service
TDD Repo
TDD Service
TDD Repo
TDD Service
TDD Repo
Decoupled
Zone 2
Map URI Routes to Service Methods
Generate SQL Migration from Entity Graph
Run SQL Migrations against Local RDBMS
Test Application thru the UI
Build UI Concurrently

TDD Workflow in Action

http://bit.ly/tdd-vids

Putting Together:
PHPUnit, Doctrine, SQLite

Doctrine

Doctrine / Uuid

- composer require ramsey/uuid-doctrine

config.yml: Doctrine / Uuid

```yaml
# Doctrine Configuration
doctrine:
    dbal:
        driver:   pdo_mysql
        host:     "%database_host%"
        port:     "%database_port%"
        dbname:   "%database_name%"
        user:     "%database_user%"
        password: "%database_password%"
        charset:  UTF8
        # if using pdo_sqlite as your database driver:
        #   1. add the path in parameters.yml
        #     e.g. database_path: "%kernel.root_dir%/data/data.db3"
        #   2. Uncomment database_path in parameters.yml.dist
        #   3. Uncomment next line:
        #     path:     "%database_path%"
        types:
            uuid: Ramsey\Uuid\Doctrine\UuidType

    orm:
        auto_generate_proxy_classes: "%kernel.debug%"
        naming_strategy: doctrine.orm.naming_strategy.underscore
        auto_mapping: true

        # TODO: ADD
        mappings:
          AppBundle:
            mapping: true
            prefix: AppBundle\Entity
            alias: E
```

PHPUnit, Doctrine, SQLite

PHPUnit, Doctrine, SQLite

- config_test.yml <— used by **PHPUnit**
  - **doctrine**:
    - dbal:
      - driver: **pdo_sqlite**

AppTestCase
extends WebTestCase

```php
protected function setUp()
{
    parent::setUp();
    self::bootKernel();
    $this->em = static::$kernel->getContainer()->get( id: 'doctrine')->getManager();
    $this->setUpEntityManager();
}

protected function em()
{
    return $this->em;
}

private function setUpEntityManager()
{
    $classes = $this->em()->getMetadataFactory()->getAllMetadata();
    $tool = new SchemaTool($this->em);
    $tool->dropSchema($classes);
    try {
        $tool->createSchema($classes);
    } catch (ToolsException $e) {
    }
}
```

Forward-Engineering DB Schema

Forward-Engineering DB Schema

- composer require doctrine/doctrine-migrations-bundle

Forward-Engineering DB Schema

- ./bin/console doctrine:migrations:diff
- ./bin/console doctrine:migrations:migrate

Beyond Doctrine & Repos …

Frameworks …

… Functionality for Free !

Frameworks

- FOSRestBundle
  - composer require friendsofsymfony/rest-bundle
- FOSUserBundle
  - composer require friendsofsymfony/user-bundle
- FOSOAuthServerBundle
  - composer require friendsofsymfony/oauth-server-bundle

Frameworks vs Tests

Frameworks vs Tests

- Avoid loading frameworks not needed in PHPUnit
  - FOSRestBundle
  - FOSUserBundle
  - FOSOAuthBundle
- use config segregation
- use AppKernel segregation

Config Segregation

Symfony Config Files

- config_**test**.yml —> PHPUnit
- config_**dev**.yml —> localhost:8000
- config_**prod**.html —> yoursite.com

config_test.yml

config.yml

config_dev.yml

config.yml

frameworks.yml

config_prod.yml

config.yml

frameworks.yml

frameworks.yml

fos_rest

fos_user

fos_oauth_server

config.yml

doctrine

Frameworks vs Tests: AppKernel

```
if (! in_array($this->getEnvironment(), ['test'], true)) {
    $bundles[] = new FOS\RestBundle\FOSRestBundle();
    $bundles[] = new JMS\SerializerBundle\JMSSerializerBundle();
    $bundles[] = new Nelmio\CorsBundle\NelmioCorsBundle();
    $bundles[] = new Symfony\Bundle\WebServerBundle\WebServerBundle();
    $bundles[] = new FOS\UserBundle\FOSUserBundle();
    $bundles[] = new FOS\OAuthServerBundle\FOSOAuthServerBundle();
}
```

FOSRestBundle: frameworks.yml

FOSRestBundle: frameworks.yml

```yaml
# FOSRest Configuration
fos_rest:
    routing_loader:
        default_format: json
        include_format: false
    exception:
      enabled: true
    body_listener: true
    format_listener:
        rules:
            - { path: '^/api|oauth/token', priorities: ['json'], fallback_format: json, prefer_extension: false }
            - { path: '^/login', priorities: ['html'], fallback_format: html, prefer_extension: false }
            - { path: '^/oauth/v2/auth', priorities: ['html'], fallback_format: html, prefer_extension: false }
            - { path: '^/', priorities: ['html'], fallback_format: html, prefer_extension: false }
    param_fetcher_listener: true
    view:
        view_response_listener: 'force'
        formats:
          json: true
```

FOSRestBundle: frameworks.yml

```yaml
# FOSRest Configuration
fos_rest:
    routing_loader:
        default_format: json
        include_format: false
    exception:
      enabled: true
    body_listener: true
    format_listener:
        rules:
            - { path: '^/api|oauth/token', priorities: ['json'], fallback_format: json, prefer_extension: false }
            - { path: '^/login', priorities: ['html'], fallback_format: html, prefer_extension: false }
            - { path: '^/oauth/v2/auth', priorities: ['html'], fallback_format: html, prefer_extension: false }
            - { path: '^/', priorities: ['html'], fallback_format: html, prefer_extension: false }
    param_fetcher_listener: true
    view:
        view_response_listener: 'force'
        formats:
          json: true
```

<— possibly set to **FALSE**

FOSUserBundle: frameworks.yml

FOSUserBundle: frameworks.yml

```yaml
fos_user:
    db_driver: orm # other valid values are 'mongodb', 'couchdb' and 'propel'
    firewall_name: api
    user_class: AppBundle\Entity\AppUser
    from_email:
          address: "noreply@yourcompany.com"
          sender_name: "No Reply"
```

FOSUserBundle: services.yml

```yaml
services:
    fos_user.user_manager.public:
        alias: fos_user.user_manager
        public: true
```

FOSOauthServerBundle: frameworks.yml

OAuthServerBundle: frameworks.yml

```yaml
fos_oauth_server:
    db_driver:            orm
    client_class:         AppBundle\Entity\Client
    access_token_class:   AppBundle\Entity\AccessToken
    refresh_token_class:  AppBundle\Entity\RefreshToken
    auth_code_class:      AppBundle\Entity\AuthCode
    service:
        user_provider: fos_user.user_provider.username
```

OAuthServer Entities

OAuth: AccessToken

```php
namespace AppBundle\Entity;

use FOS\OAuthServerBundle\Entity\AccessToken as BaseAccessToken;
use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Table("oauth2_access_tokens")
 * @ORM\Entity
 */
class AccessToken extends BaseAccessToken
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    protected $id;

    /**
     * @ORM\ManyToOne(targetEntity="Client")
     * @ORM\JoinColumn(nullable=false)
     */
    protected $client;

    /**
     * @ORM\ManyToOne(targetEntity="AppUser")
     */
    protected $user;
}
```

OAuth: AuthCode

```php
namespace AppBundle\Entity;

use FOS\OAuthServerBundle\Entity\AuthCode as BaseAuthCode;
use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Table("oauth2_auth_codes")
 * @ORM\Entity
 */
class AuthCode extends BaseAuthCode
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    protected $id;

    /**
     * @ORM\ManyToOne(targetEntity="Client")
     * @ORM\JoinColumn(nullable=false)
     */
    protected $client;

    /**
     * @ORM\ManyToOne(targetEntity="AppUser")
     */
    protected $user;
}
```

OAuth: Client

```php
namespace AppBundle\Entity;

use FOS\OAuthServerBundle\Entity\Client as BaseClient;
use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Table("oauth2_clients")
 * @ORM\Entity
 */
class Client extends BaseClient
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    protected $id;
}
```

OAuth: RefreshToken

```php
namespace AppBundle\Entity;

use FOS\OAuthServerBundle\Entity\RefreshToken as BaseRefreshToken;
use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Table("oauth2_refresh_tokens")
 * @ORM\Entity
 */
class RefreshToken extends BaseRefreshToken
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    protected $id;

    /**
     * @ORM\ManyToOne(targetEntity="Client")
     * @ORM\JoinColumn(nullable=false)
     */
    protected $client;

    /**
     * @ORM\ManyToOne(targetEntity="AppUser")
     */
    protected $user;
}
```

security.yml

```yaml
# To get started with security, check out the documentation:
# http://symfony.com/doc/current/security.html
security:
    encoders:
        AppBundle\Entity\AppUser: bcrypt

    # http://symfony.com/doc/current/security.html#b-configuring-how-users-are-loaded
    providers:
        in_memory:
            memory: ~
        fos_userbundle:
                id: fos_user.user_provider.username
```

```yaml
firewalls:
    # disables authentication for assets and the profiler, adapt it according to your needs
    dev:
        pattern: ^/(_(profiler|wdt)|css|images|js)/
        security: false

    oauth_token:                                # Everyone can access the access token URL.
        pattern: ^/oauth/v2/token
        security: false

    api-public:
        pattern: ^/api/v1/register-user
        security: false

    api:
        pattern: ^/api/v1                       # All URLs are protected
        fos_oauth: true                         # OAuth2 protected resource
        stateless: true                         # Do no set session cookies
        anonymous: false
        provider: fos_userbundle
```

```
main:
    anonymous: ~
    # activate different ways to authenticate

    # http_basic: ~
    # http://symfony.com/doc/current/security.html#a-configuring-how-you
    # http://symfony.com/doc/current/cookbook/security/form_login_setup.

    pattern: ^/
    form_login:
        provider: fos_userbundle
        check_path: fos_user_security_check
        #csrf_provider: security.csrf.token_manager
    logout:       true
    anonymous:    true
    logout_on_user_change: true
```

FOSUser Entity

AppUser

```php
namespace AppBundle\Entity;

use AppBundle\DTO\UserDto;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Criteria;
use Doctrine\ORM\Mapping as ORM;
use Ramsey\Uuid\Uuid;
use FOS\UserBundle\Model\User as BaseUser;

/**
 * Class AppUser
 * @package AppBundle\Entity
 *
 * @ORM\Entity()
 * @ORM\Table(name="users")
 */
class AppUser extends BaseUser
{
```

```php
public function __construct(
    $firstName = null,
    $lastName = null,
    $email = null,
    $username = null,
    $password = null
) {
    parent::__construct();
    $this->id = Uuid::uuid4();
    $this->firstName = $firstName;
    $this->lastName = $lastName;
    $this->email = $email;
    $this->username = $username;
    $this->setPlainPassword($password);
    $this->appRoles = new ArrayCollection();
    $this->created = new \DateTime( time: null, new \DateTimeZone( timezone: 'UTC'));
}
```

UserRepository

UserRepository: Goals

- Leverages FOSUser's UserManager
- Same code in Test vs Production
- Inject a FakeUserManager in Test
- Inject FOS's Real UserManager in Production Code

UserRepository

```
/**
 * @var UserManagerInterface
 */
private $userManager;

public function __construct(
    EntityManagerInterface $entityManager,
    UserManagerInterface $userManager = null
) {
    parent::__construct($entityManager);
    $this->userManager = $userManager;
}
```

UserRepository

```
/**
 * @param AppUser $passedUser
 * @return AppUser
 */
public function saveNewUser(AppUser $passedUser): AppUser
{
    /** @var AppUser $user */
    $user = $this->userManager->createUser();
    $user->setFirstName($passedUser->getFirstName());
    $user->setLastName($passedUser->getLastName());
    $user->setUsername($passedUser->getUsername());
    $user->setEmail($passedUser->getEmail());
    $user->setPlainPassword($passedUser->getPlainPassword());
    $user->setEnabled( boolean: true);
    $this->userManager->updateUser($user);
    return $user;
}
```

UserRepository

UserManager

```
/**
 * @param AppUser $passedUser
 * @return AppUser
 */
public function saveNewUser(AppUser $passedUser): AppUser
{
    /** @var AppUser $user */
    $user = $this->userManager->createUser();
    $user->setFirstName($passedUser->getFirstName());
    $user->setLastName($passedUser->getLastName());
    $user->setUsername($passedUser->getUsername());
    $user->setEmail($passedUser->getEmail());
    $user->setPlainPassword($passedUser->getPlainPassword());
    $user->setEnabled( boolean: true);
    $this->userManager->updateUser($user);
    return $user;
}
```

UserRepository

- In Test Harness:
  - Inject:
    FakeUserManager implements UserManagerInterface
- In Production Code:
  - Inject:
  - $this->container->get('**fos_user.user_manager.public**');

FakeUserManager
implements UserManagerInterface

```php
class FakeUserManager implements UserManagerInterface
{
    /** @var UserInterface */
    private $user;

    /**
     * @var EntityManagerInterface
     */
    private $entityManager;

    public function __construct(EntityManagerInterface $entityManager)
    {

        $this->entityManager = $entityManager;
    }

    /**
     * Creates an empty user instance.
     *
     * @return UserInterface
     * @throws \Exception
     */
    public function createUser()
    {
        $this->user = new AppUser();
        return $this->user;
    }
```

```php
/**
 * Updates a user.
 *
 * @param UserInterface $user
 */
public function updateUser(UserInterface $user)
{
    $this->entityManager->persist($user);
    $this->entityManager->flush();
}
```

FakeUserManager

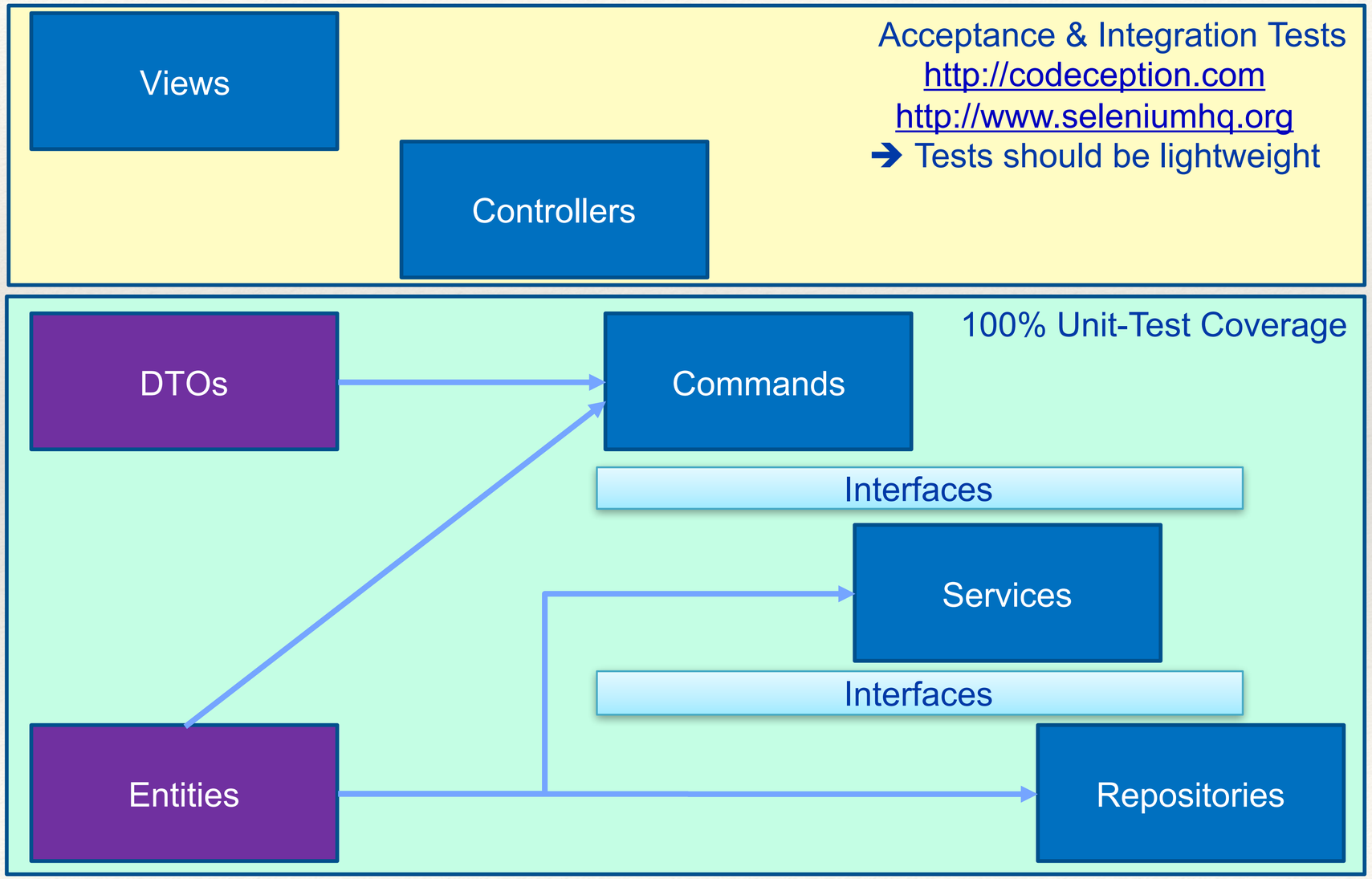
- override createUser & saveUser methods
  - gets rid of overhead incompatible with tests
  - gets rid of bcrypt password hashing
- Tests don’t care about other methods
  - as those pertain to use-cases …
    - … not in scope of our Domain

routing.yml

```yaml
app:
    resource: "@AppBundle/Controller/"
    type:     annotation

fos_oauth_server_token:
    resource: "@FOSOAuthServerBundle/Resources/config/routing/token.xml"

fos_oauth_server_authorize:
    resource: "@FOSOAuthServerBundle/Resources/config/routing/authorize.xml"

fos_user_security:
    resource: "@FOSUserBundle/Resources/config/routing/security.xml"

fos_user_register:
    resource: "@FOSUserBundle/Resources/config/routing/registration.xml"
    prefix: /register

fos_user_profile:
    resource: "@FOSUserBundle/Resources/config/routing/profile.xml"
    prefix: /profile

fos_user_resetting:
    resource: "@FOSUserBundle/Resources/config/routing/resetting.xml"
    prefix: /resetting

fos_user_change_password:
    resource: "@FOSUserBundle/Resources/config/routing/change_password.xml"
    prefix: /profile
```

Codeception

TDD-Friendly Architecture



CodeCeption

- composer require “codeception/codeception” —dev

CodeCeption: acceptance.suite.yml

```yaml
actor: AcceptanceTester
modules:
    enabled:
        - PhpBrowser:
            url: http://127.0.0.1:8000
        - \Helper\Acceptance
```

CodeCeption

- When testing OAuth “Auth Code Grant” Flow …
  - APITester must be able to:
    - do HTML for user “login” and “accept”
    - do JSON for API Requests

CodeCeption: api.suite.yml

```yaml
actor: ApiTester
modules:
    enabled:
        - \Helper\Api
        - PhpBrowser:
            url: http://127.0.0.1:8000/
        - REST:
            url: http://127.0.0.1:8000/api/v1
            depends: PhpBrowser
            part: Json
```

Testing OAuth Manually

OAuth: “Auth Code Grant” Flow

- http://localhost:8000/oauth/v2/auth?client_id=1_3bcbxd9e24g0gk4swg0kwgcwg4o8k8g4g888kwc44gcc0gwwk4&redirect_uri=http://localhost:8000/&response_type=token
  - redirects to /login if user not logged-in
  - token request: user Reject or Accept
    - redirects to / with “token” in the URL
  - grab “token” & use in subsequent API requests

http://localhost:8000/oauth/v2/auth?client_id=1_3bcbxd9e24g0gk4swg0kwgcwg4o8k8g4g888kwc44gcc0gwwk4&redirect_uri=http://localhost:8000/&response_type=token

layout.login

security.login.username chriswuzhere security.login.password

security.login.remember_me security.login.submit

authorize.accept

authorize.reject

localhost:8000/#state&access_token=N2ZiYTUzNTVhOWJjMWRjNzNlMzBiNGM

Logged in as	chriswuzhere
Authenticated	Yes
Token class	UsernamePasswordToken
Firewall name	main
Actions	Logout

chriswuzhere 1 ms 1 in

Testing OAuth w/ Codeception

Codeception
“Auth Code Grant”

```php
protected function loginWithUserName($username) : array
{

    $this->amOnPage('/logout');
    $authUrl =
        '/oauth/v2/auth?client_id='
        .UserApi::CLIENT_ID
        .'&redirect_uri=http://localhost:8000/&response_type=token';

    $this->amOnPage($authUrl);
    $this->canSeeInFormFields('form', [
        '_username' => '',
        '_password' => ''
    ]);
    $this->submitForm('form', [
        '_username' => $username,
        '_password' => 'password'
    ], '_submit');
    $this->seeResponseCodeIs(HttpCode::OK);
    $this->submitForm('form[name=fos_oauth_server_authorize_form]', [], 'accepted');
    $token = $this->grabFromCurrentUrl('~.*&access_token=([^&]+)~');
    $this->token = $token;
    return [
        'token_type' => 'bearer',
        'access_token' => $token
    ];
}
```

Codeception
“Auth Code Grant”

https://github.com/elchris/kata_tdd_php_symfony/blob/master/tests/_support/ApiTester.php#L335

It’s all in my GitHub

https://github.com/elchris/kata_tdd_php_symfony

https://github.com/elchris/kata_tdd_php_symfony

(master branch)

(Pull Requests are Welcome)

See it in action

- git clone https://github.com/elchris/kata_tdd_php_symfony.git
- start mysql && create database named “symfony”
- cd kata_tdd_php_symfony && composer install
- ./bin/console doctrine:migrations:migrate
- ./bin/console server:start (make sure it’s :8000)
- ./vendor/bin/phpunit
- ./vendor/bin/codecept run

See it in action

- http://localhost:8000/
- http://localhost:8000/register/
- http://localhost:8000/profile/
- http://localhost:8000/logout
- http://localhost:8000/login
- http://localhost:8000/oauth/v2/auth?client_id=1_3bcbxd9e24g0gk4swg0kwgcwg4o8k8g4g888kwc44gcc0gwwk4&redirect_uri=http://localhost:8000/&response_type=token

Welcome to
Symfony 3.4.17

Your application is now ready. You can start workin

`/Users/chris/workarea/chris/demo/kata`

What's next?

Read the documentation to learn

How to create your first page in Symfony

Logged in as	chriswuzhere
Authenticated	Yes
Token class	UsernamePasswordToken
Firewall name	main
Actions	Logout

MB 1 49 in 2.14 ms chriswuzhere 2 ms 1 in 0.50 ms

Live-Coding TDD Videos:
http://bit.ly/tdd-vids

Takeaways

Test-Drive Application ...

… Focus on Core Business Logic

… Test-Drive Data Operations (Repositories)

… Lean on Frameworks for …

RESTful End-Points

RESTful End-Points

User Management

RESTful End-Points

User Management

OAuth

... Without interfering
with Unit Tests!

Merci Bien!

Merci Bien!

Slides:

http://bit.ly/real-world-tdd

Suivez-moi sur Touitteure:
https://twitter.com/chrisholland

Merci Bien!

Slides:
http://bit.ly/real-world-tdd