Introduction to Test Driven Development

Mohammad Azam
@azamsharp

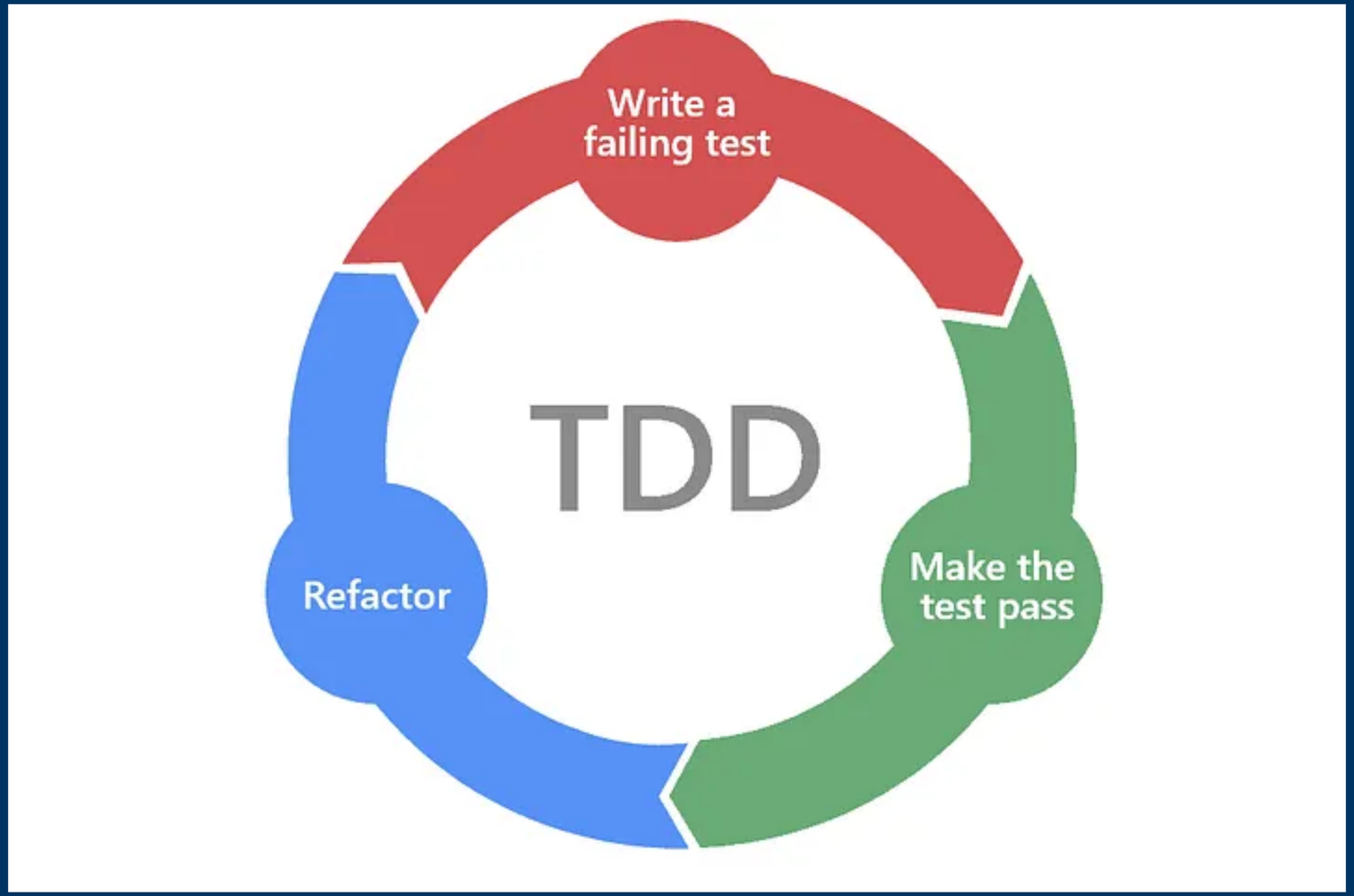
About Me

- Mohammad **Azam**
- Coding Bootcamp Instructor
- AzamSharp School (https://azamsharp.school)
- Author and speaker

Test Driven Development

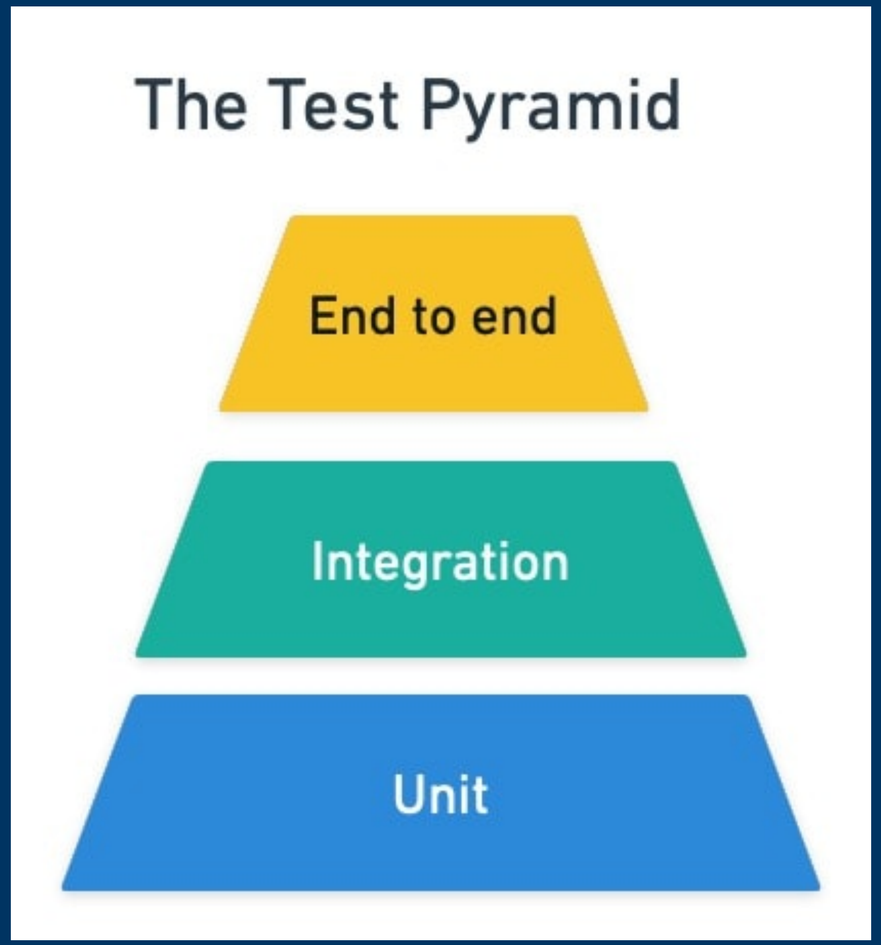
Write test before any line of production code.

Red, Green and Refactor



Source: https://marsner.com/blog/why-test-driven-development-tdd/

Test Pyramid



Source: https://semaphoreci.com/blog/testing-pyramid

Benefits of TDD

- It helps to understand a complicated domain.
- It gives confidence for future changes.
- It helps to solve algorithmic problems more efficiently.

Notes about Testing

- Make sure to test the domain. Domain consists of the business rules.
- Unit tests are allowed to access databases and filesystems.
- Features responsible for making $$ for the company needs to be throughly tested.
- Test the behavior not implementation details.
- During refactoring step, don’t write any new tests.
- End-to-End testing is best against regression.

Demo

Mocks and Stubs

Stubs

- A stub is an object that provides predetermined answers to method calls made during the test.
- It is used to simulate the behavior of a real dependency and provide specific responses to method calls.
- Stubs do not record the calls made to them or verify interactions; they simply return pre-defined responses.
- Stubs are useful when you want to isolate the code under test from the behavior of its dependencies and focus solely on the behavior being tested.

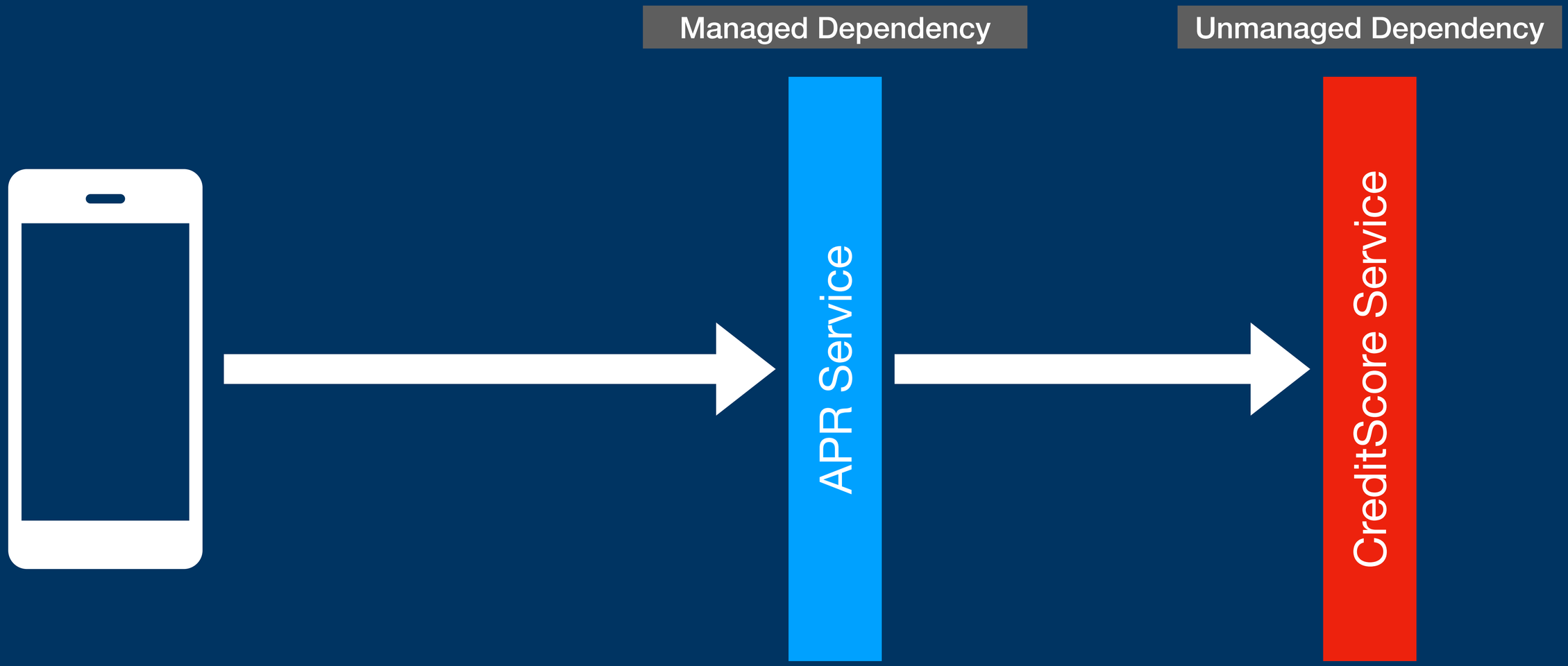
Mocks

- A mock is an object that mimics the behavior of a real object in a controlled way.
- It is used to verify interactions between the code under test and its dependencies.
- Mocks are typically used when you want to verify that certain methods are called with specific arguments or that they are called a certain number of times.

Loan App

Loan App

Feature: A user should be able to calculate APR based on their SSN


Managed Dependency
Unmanaged Dependency
APR Service
CreditScore Service

Managed Dependency

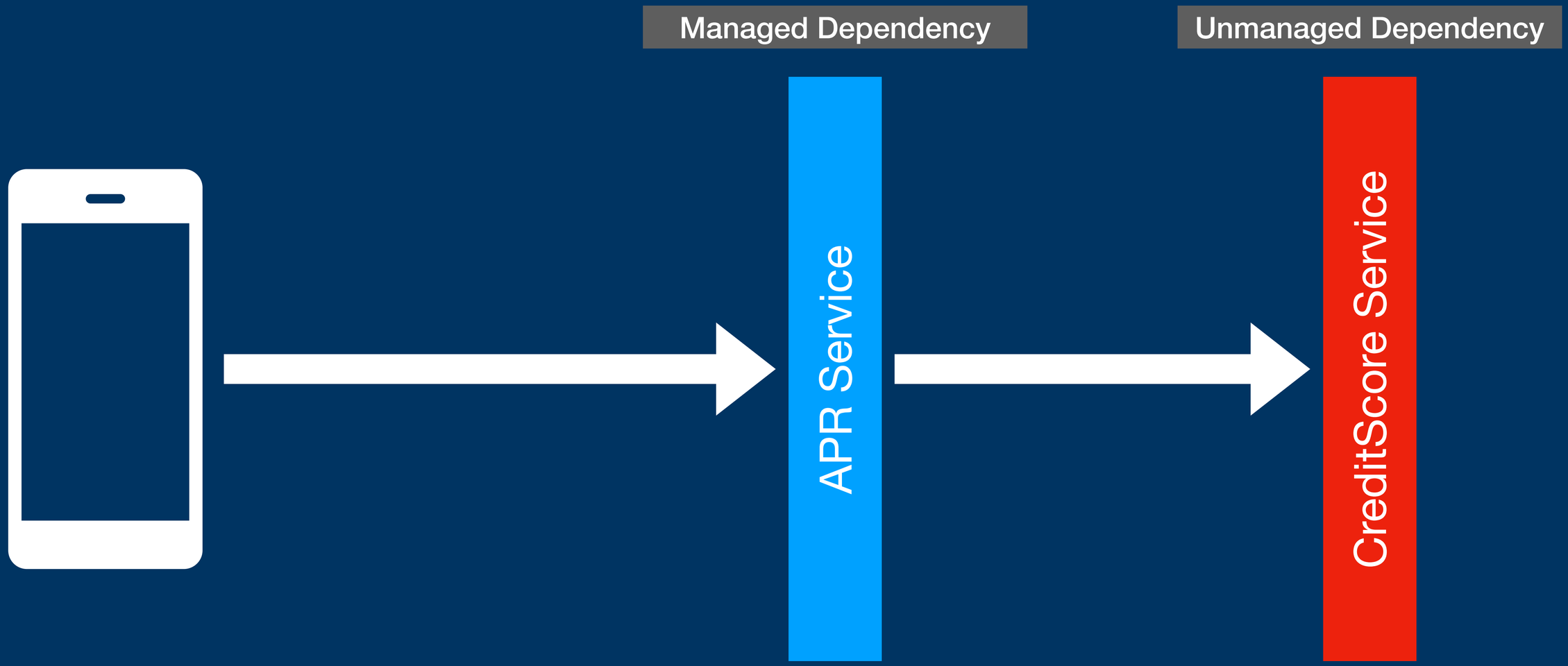
- You have full control over it
- Outside world cannot access it
- External systems can access it through the use of your API
- Examples include your application database

https://khalilstemmler.com/wiki/managed-vs-unmanaged-dependencies/

Unmanaged Dependency

- You don’t have full control over it
- It is also used by other applications
- Examples include Payment gateway like Stripe or PayPal
- Any other external API not owned by you or your team

https://khalilstemmler.com/wiki/managed-vs-unmanaged-dependencies/


Managed Dependency
Unmanaged Dependency
APR Service
CreditScore Service

Demo

Contact

- @azamsharp
- https://azamsharp.school
- azamsharp@gmail.com