Advanced C++ Mocking Using Google Mock

INTRODUCING GOOGLE MOCK

Dror Helper

@dhelper www.blog.drorhelper.com

Who is this course for?

C++ Developers

- Learn more about unit testing in the real world
- Use fake objects (mocks) in your tests
- Gain control over existing legacy code
- Beginner level knowledge of C++

Not an introduction to unit testing

- Check out: *C++ Unit Testing Fundamentals Using Catch*
  - Getting started with unit testing
  - Unit testing best practices

Course Overview

Introducing Google Mock

- Using fake objects/mocks
- Setting up GTest and GMock

Unit testing using Google Test

- Unit testing recap

Creating fake objects

- Why use mocks/fakes
- Creating your first fake

Setting behaviors and expectations

- Default behavior vs. test behavior
- How to avoid overusing fake behavior

Verifying methods were called

- State based testing vs. interaction testing
- Verifying Do and Don'ts

Using arguments and Matchers

- Beyond simple behaviors
- Improving GTest assertions

Getting your legacy code under control

a “Unit Test” is:

A method (Code)

Tests specific functionality

Clear pass/fail criteria

Runs in Isolation

Unit Test Example

```cpp
#include "gtest/gtest.h"


TEST(ThisIsATest)
{
    int result = 2 + 2;


    ASSERT_EQ(4, result);
}
```

Why Write Automated Tests?

Quick Feedback

Avoid Stupid Bugs

Immune to Regression

Change Your Code Without Fear

In Code Documentation

You're already testing your code!

Google Test

xUnit test framework

Test discovery

Test Runner

Assertions

Parameterized tests

Report generation

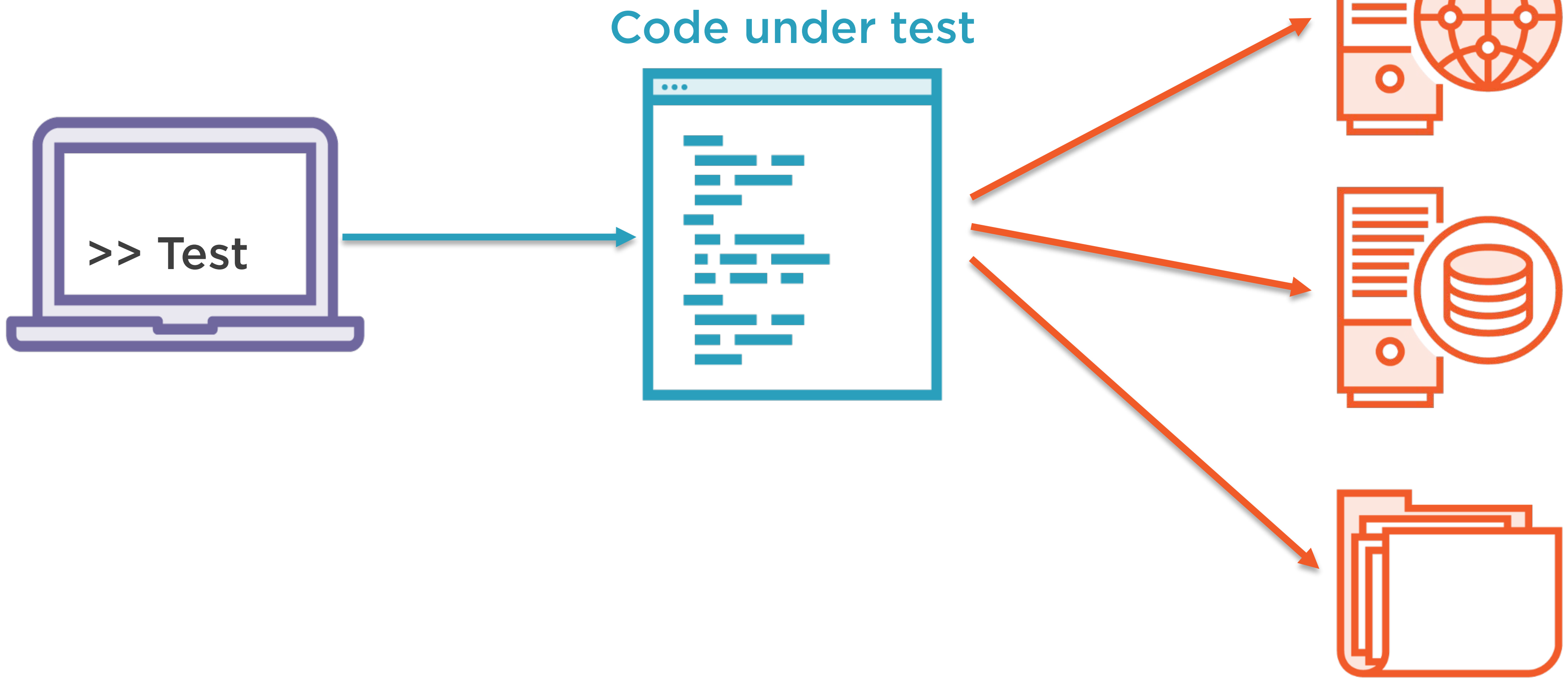
Real World Unit Testing Problems

Tests break due to external factors

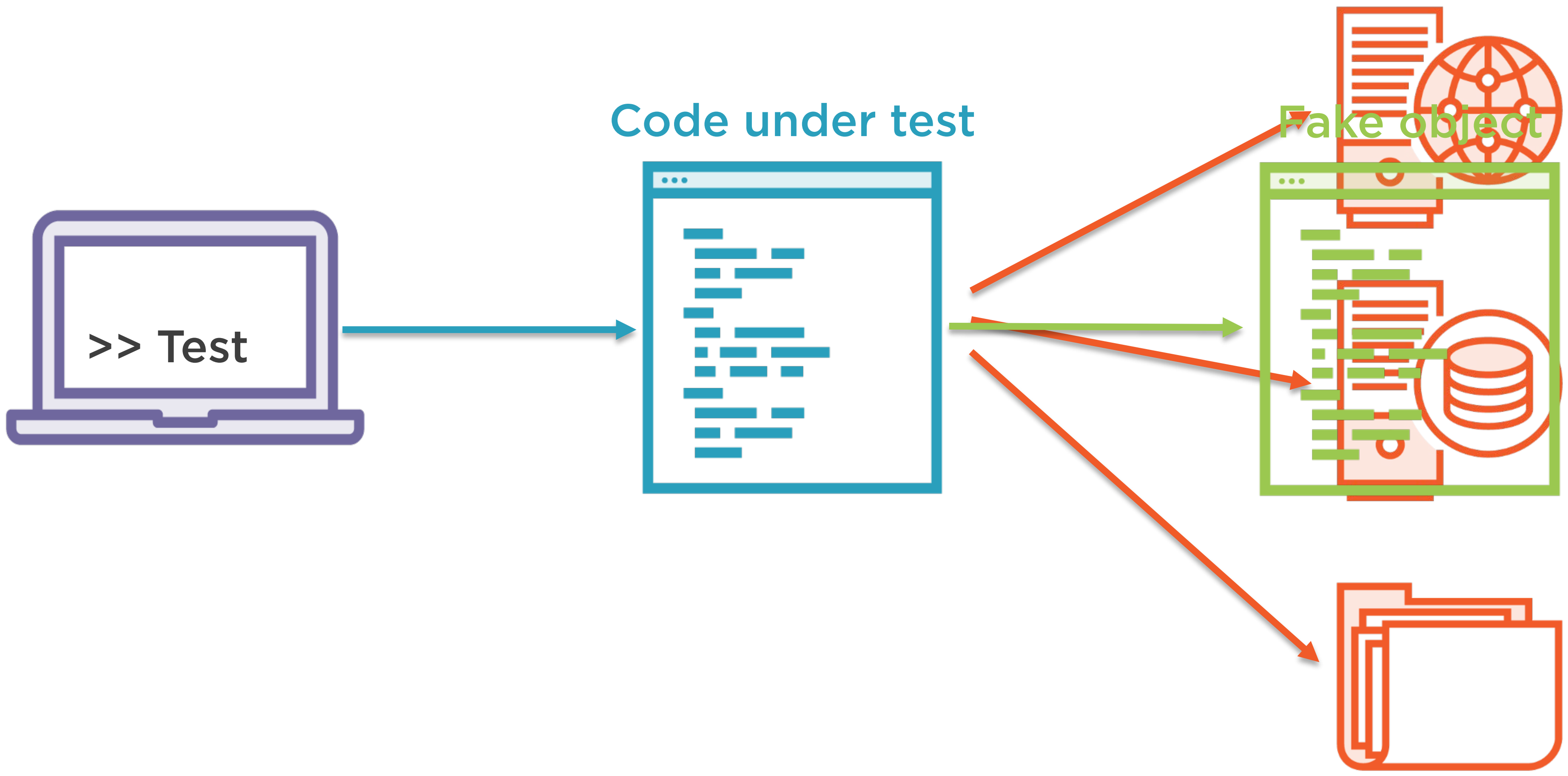
Test run for long time

Inconsistent results

The Problem == Dependencies



The solution → Fake Objects



Google Mock

Google C++ mocking framework (GMock)

- C++ Library
- Writing & using mock classes

Open source

- https://github.com/google/googletest
- With GTest inside

Can be used with any C++ unit testing framework

Benefits of Using Mocks

Remove dependencies

Reduce run time

Test hard to set scenarios

Test in isolation

Test system under development

Test failures

Getting Stated with GTest and GMock

Get sources from GitHub

Build

- Using Visual Studio
- Using Make

Create new console application project

Include headers in test project

- gtest/gtest.h
- gmock/gmock.h

Add GMock as project dependency

Add the *init* method call to main

- ::testing::InitGoogleTest
- ::testing::InitGoogleMock

Getting started with Google Mock

```
#include "gtest/gtest.h"
#include "gmock/gmock.h"


int main(int argc, char** argv)
{
    ::testing::InitGoogleMock(&argc, argv);
    return RUN_ALL_TESTS();
}
```

Demo

Getting started with Gtest/Gmock

- Where to get Gtest/Gmock
- Building required libraries
- Setting your dev environment

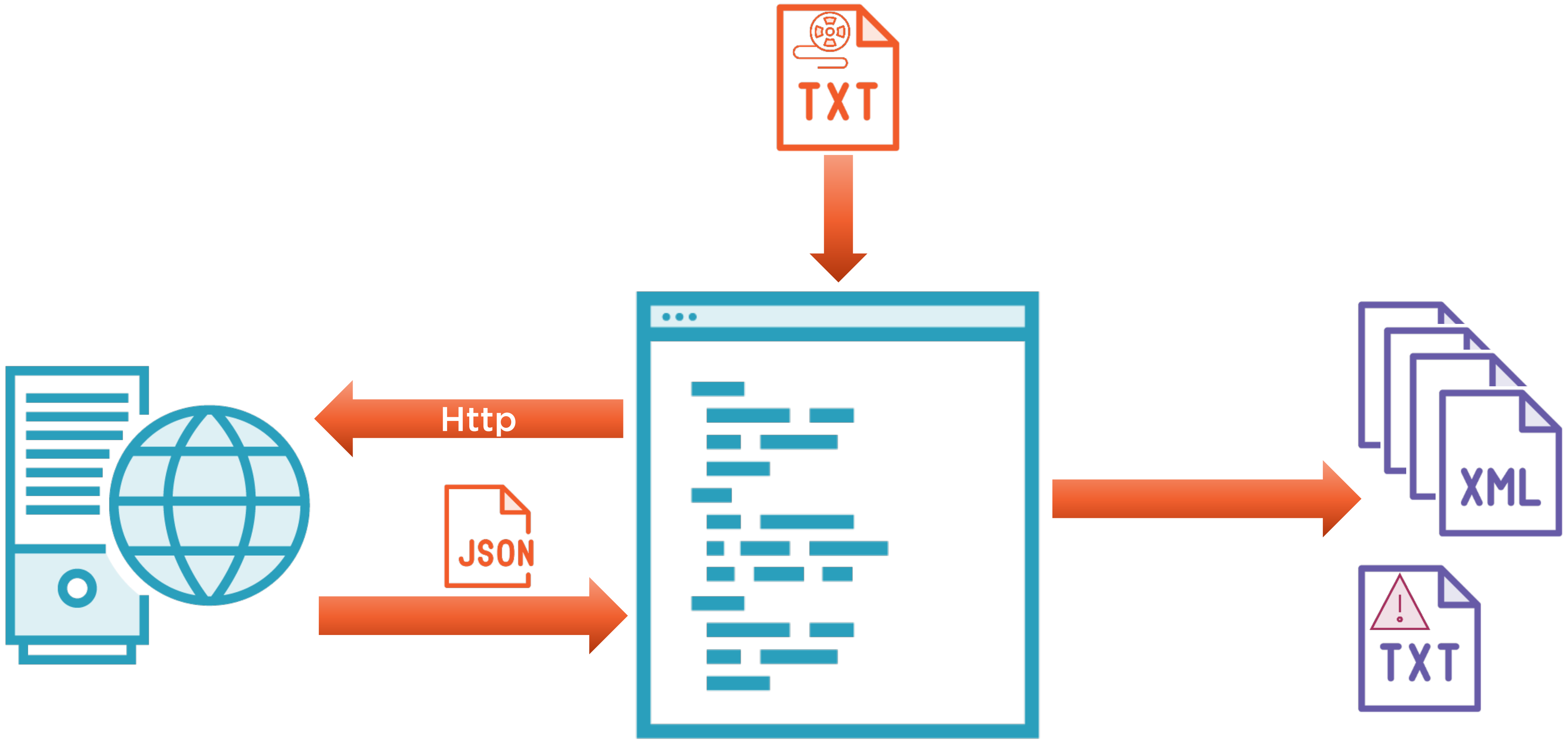
Writing Tests Using GTest

```
#include "gtest/gtest.h"

TEST(ThisIsATest)
{
    int result = 2 + 2;

    ASSERT_EQ(4, result);
}
```

Introducing the Sample Project



Demo

Writing your first test

- Defining a new test
- Running GTest
- Test failure

Using GMock

Summary

Why write unit tests

Why we need a mocking frameworks

How to set up GTest and GMock

Writing your first unit test