

Testing with Dependencies



Richard Warburton

JAVA CHAMPION, AUTHOR AND PROGRAMMER

@richardwarburto www.monotonic.co.uk



What Are We Covering?

Dependencies

Dependency injection

Stubs

Mocks



Dependencies and Motivation

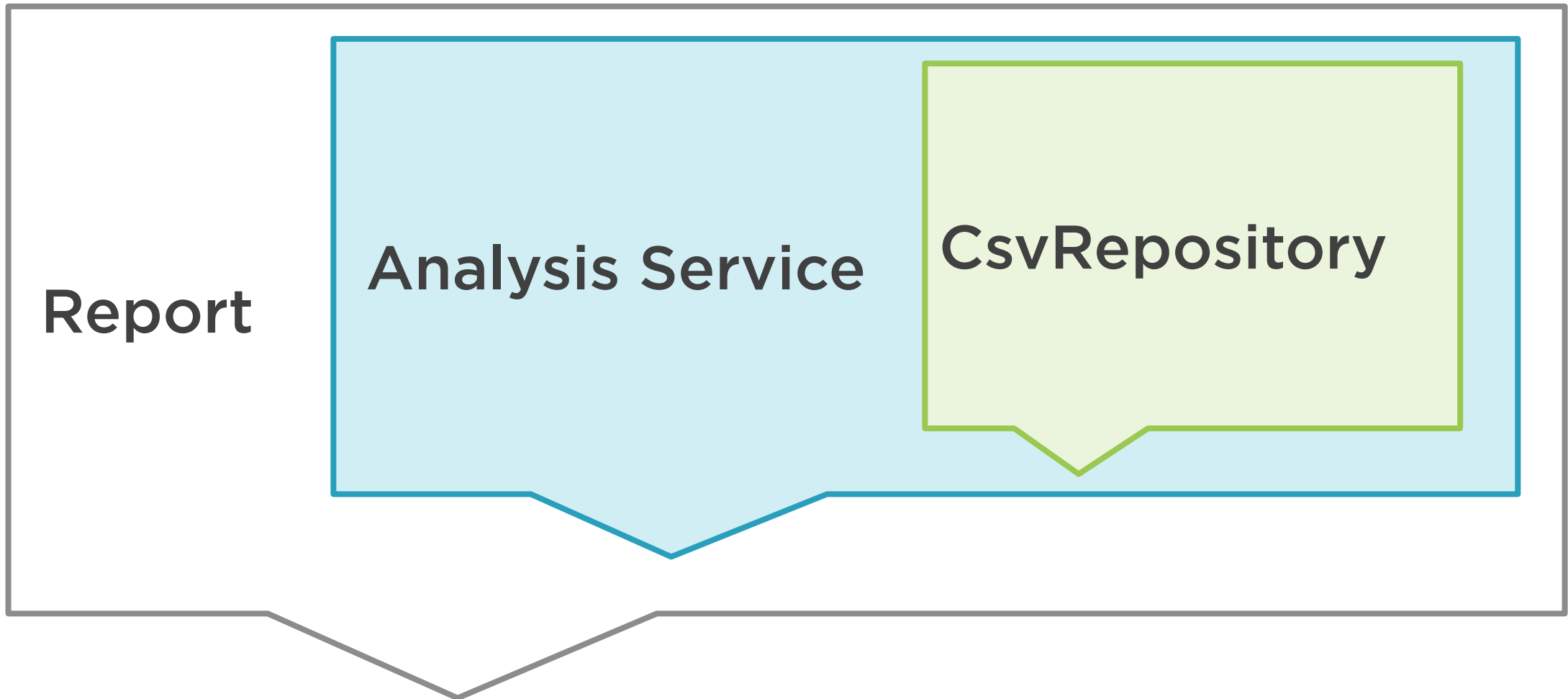


Dependency

A relationship between two components where the functionality of one component relies on another component



Sales Report Example



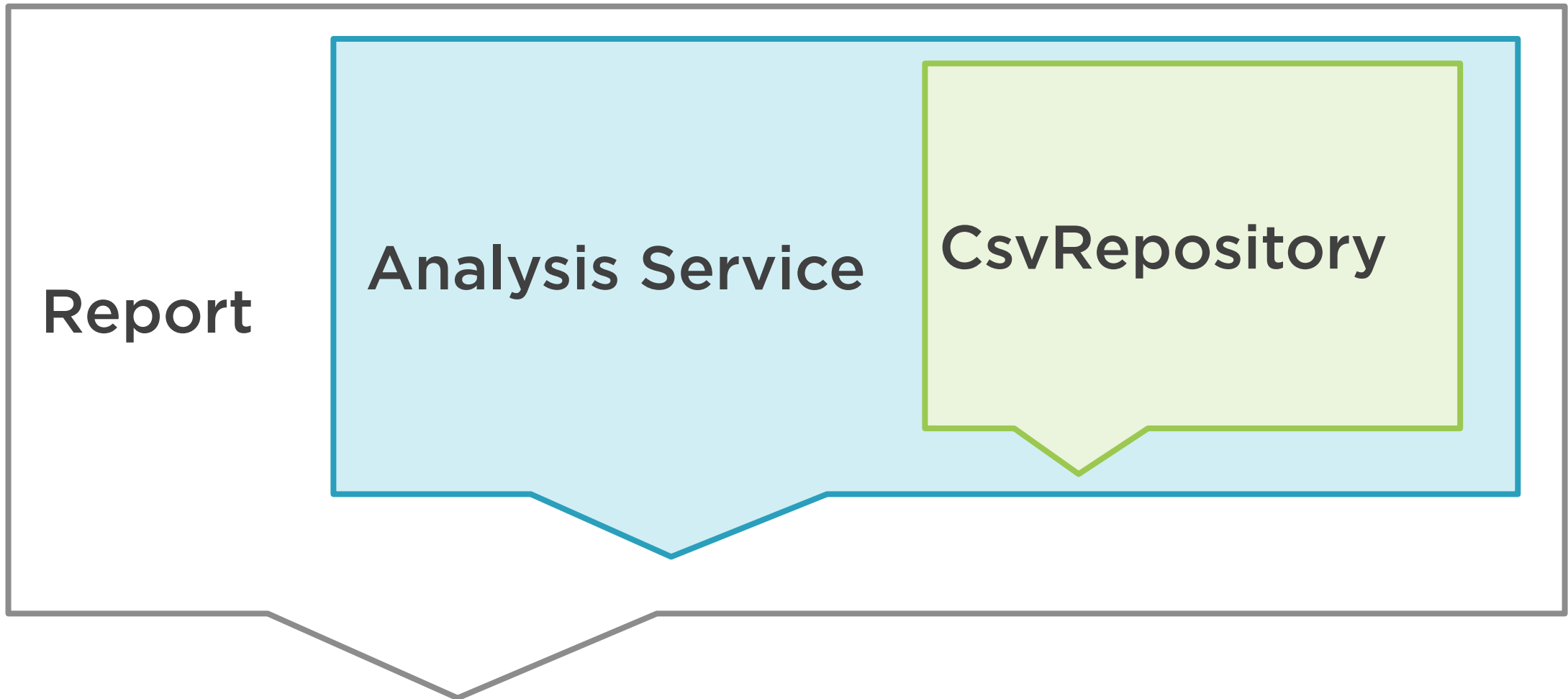




Dependency Injection



Sales Report Example



Goal: Decoupling



The diagram illustrates the goal of decoupling in software design. It features three distinct components arranged horizontally. Each component is represented by a speech bubble shape with a colored border and a matching fill. The first component, 'Report', is a white bubble with a gray border. The second, 'Analysis Service', is a light blue bubble with a blue border. The third, 'CsvRepository', is a light green bubble with a green border. All three bubbles have a small triangular tail pointing downwards at the bottom center. The text inside each bubble is bold and black.

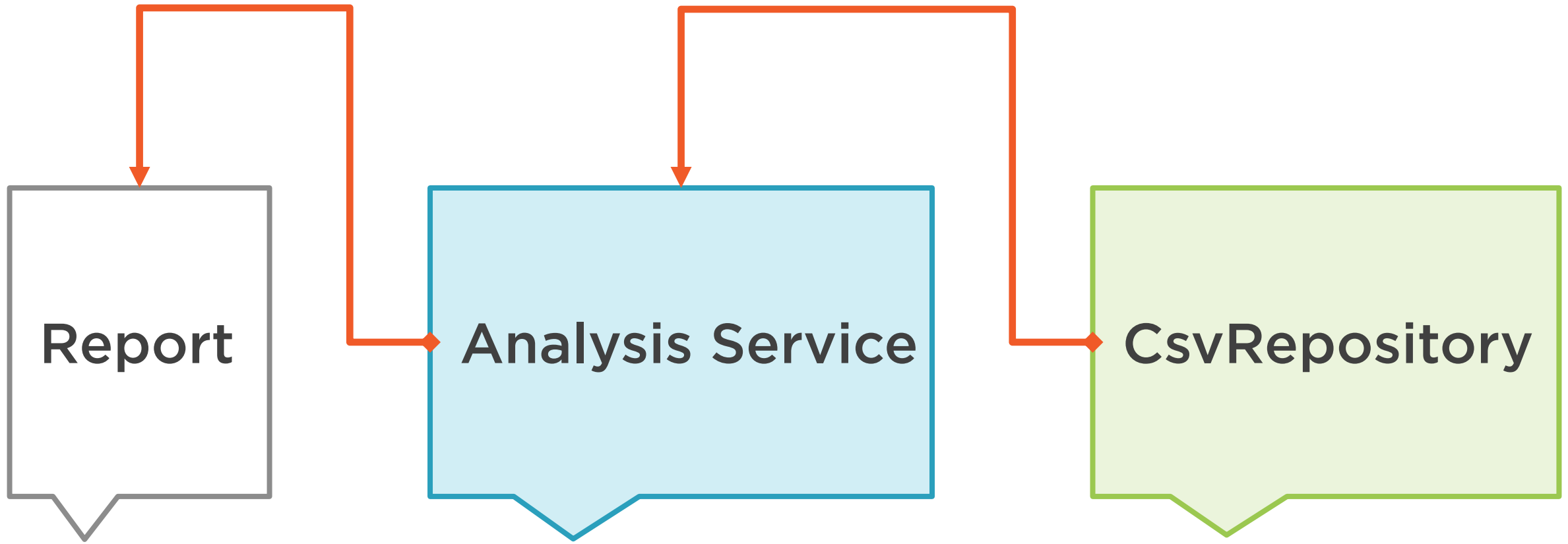
Report

Analysis Service

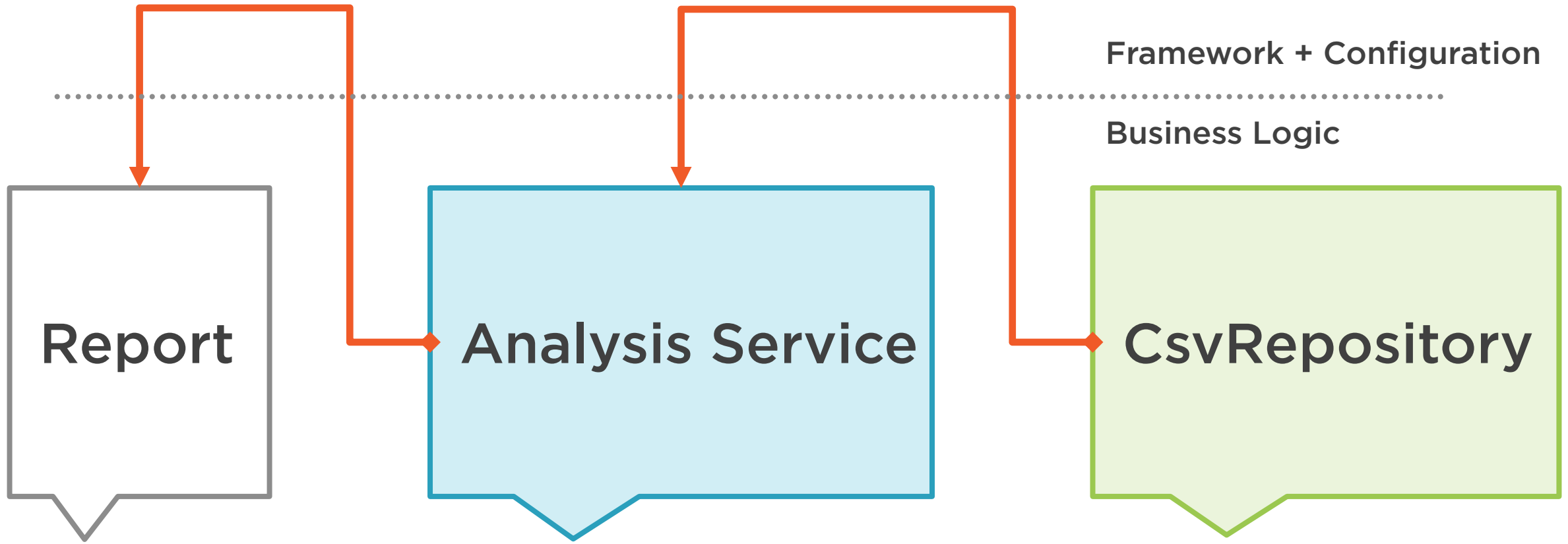
CsvRepository



Approach: Dependency Injection



Separation of Concerns

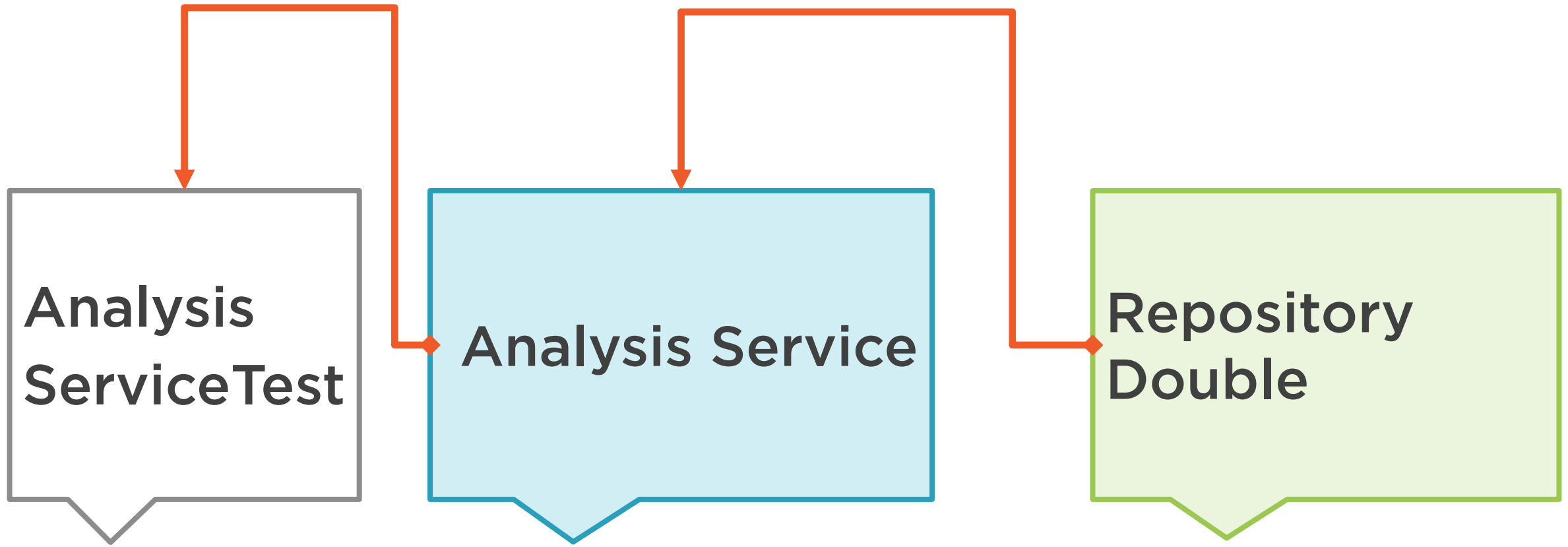


Test Doubles





Demo: Dependency Injection



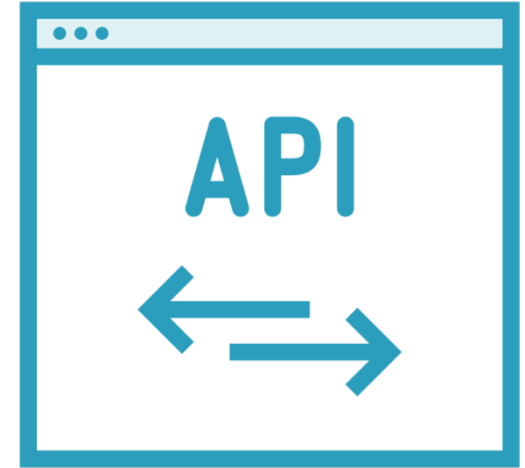
Types of Test Double



Fake



Stub



Mock

Fake

Working Implementation

Simpler than the real one

Example

In Memory Database



Stub

**Return values for
method calls**

Only what is required

Example

`size()` always returns 1



Mock

**Test interactions with
object**

Can also stub values

Example

Must invoke `add(1)`



Conclusions



Conclusions



Dependency injection pattern for decoupling

Mocks & stubs enable test isolation

It's really simple!

