Testleaf
Always Ahead

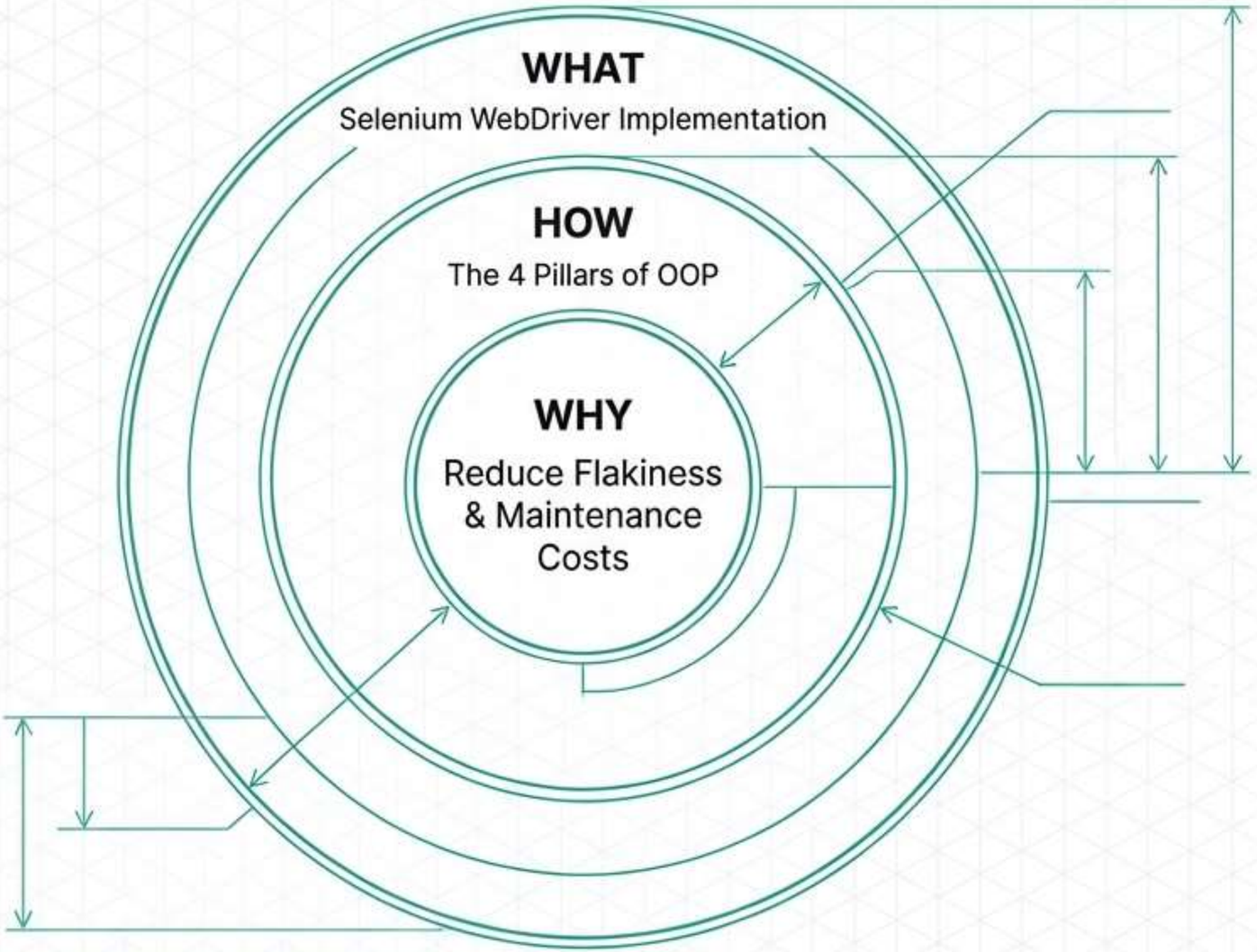
Architecting Robust Automation

OOP Principles in Selenium WebDriver

A Strategic Guide to Scalable Test Frameworks

Testleaf
Always Ahead

The Golden Circle of Automation Strategy



WHY:

To improve coding standards and performance. `JetBrains Mono`

HOW:

By implementing the 4 Pillars of Object Orientation. `JetBrains Mono`

WHAT:

Binding data and functionalities together using classes and objects.

Strategic Note:

In modern CI/CD pipelines, **maintainability is not a luxury; it is a requirement.**

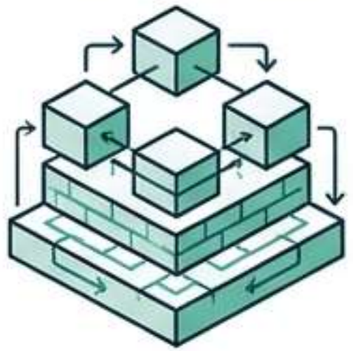
Testleaf
Always Ahead

The Four Pillars of Object Orientation

ENCAPSULATION

The Shield

Determining scope & protection.



INHERITANCE

The Foundation

Reusing code & attributes.

POLYMORPHISM

The Shapeshifter

One interface, many forms.

ABSTRACTION

The Interface

Hiding complexity.

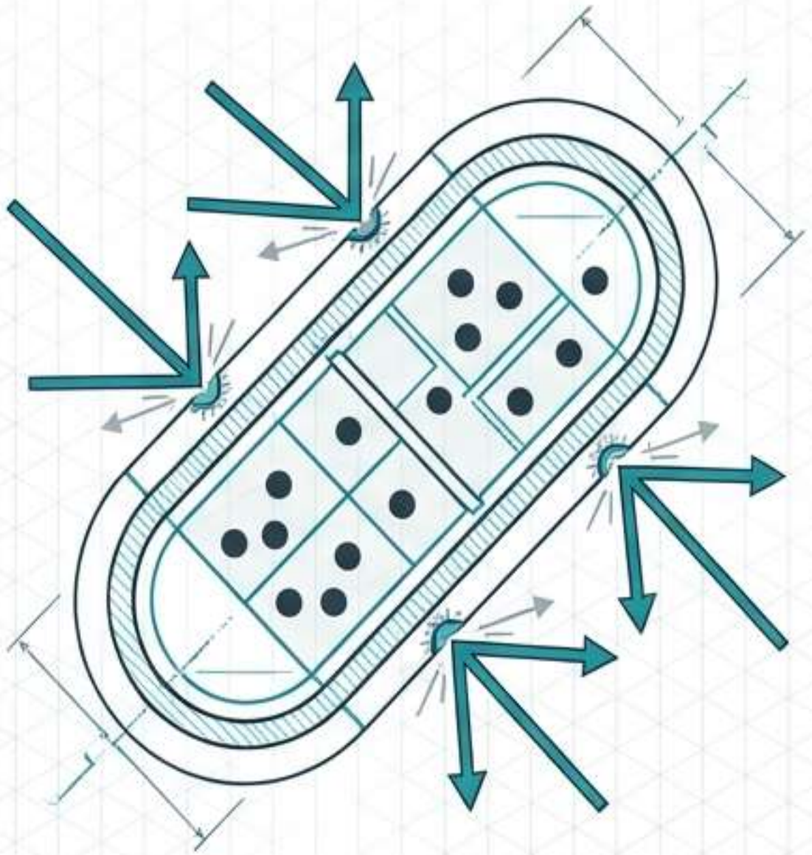
We will explore how each pillar transforms a fragile Selenium script into a robust framework architecture.

Testleaf
Always Ahead

Encapsulation: The Shield

Concept

Definition: Bundle code into a single unit where you can determine the scope of each piece of data.



Strategic Application: Page Object Model (POM)

Problem: Public WebElements allow tests to manipulate page state unpredictably.

Solution: Keep locators private. Expose only business actions.

```
private By username = By.id('user');
public void enterUsername(String u) {
   driver.findElement(username).sendKeys(u);
}
```

Modern Context

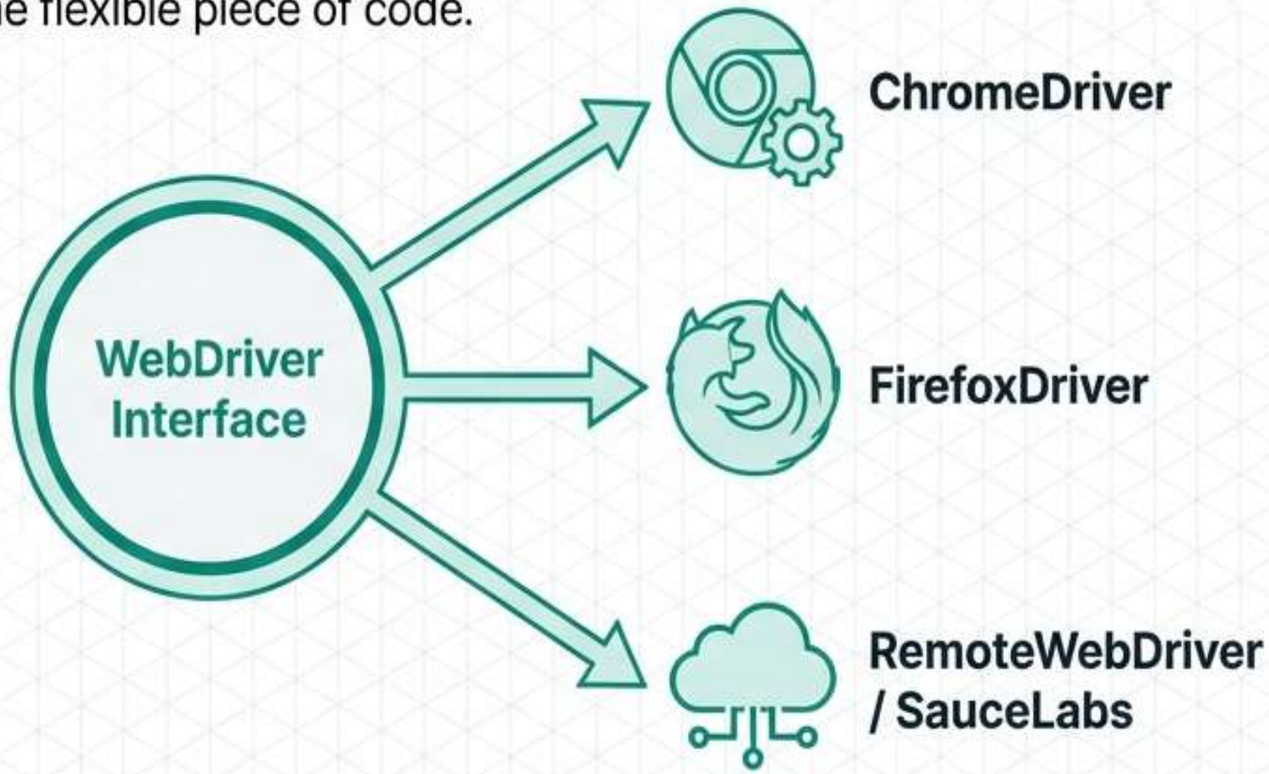
Encapsulation is essential for parallel execution to ensure thread safety.

Testleaf
Always Ahead

Polymorphism: The Shapeshifter

Concept

Definition: One class can be used to create many objects, all from the same flexible piece of code.



Cross-Browser Strategy

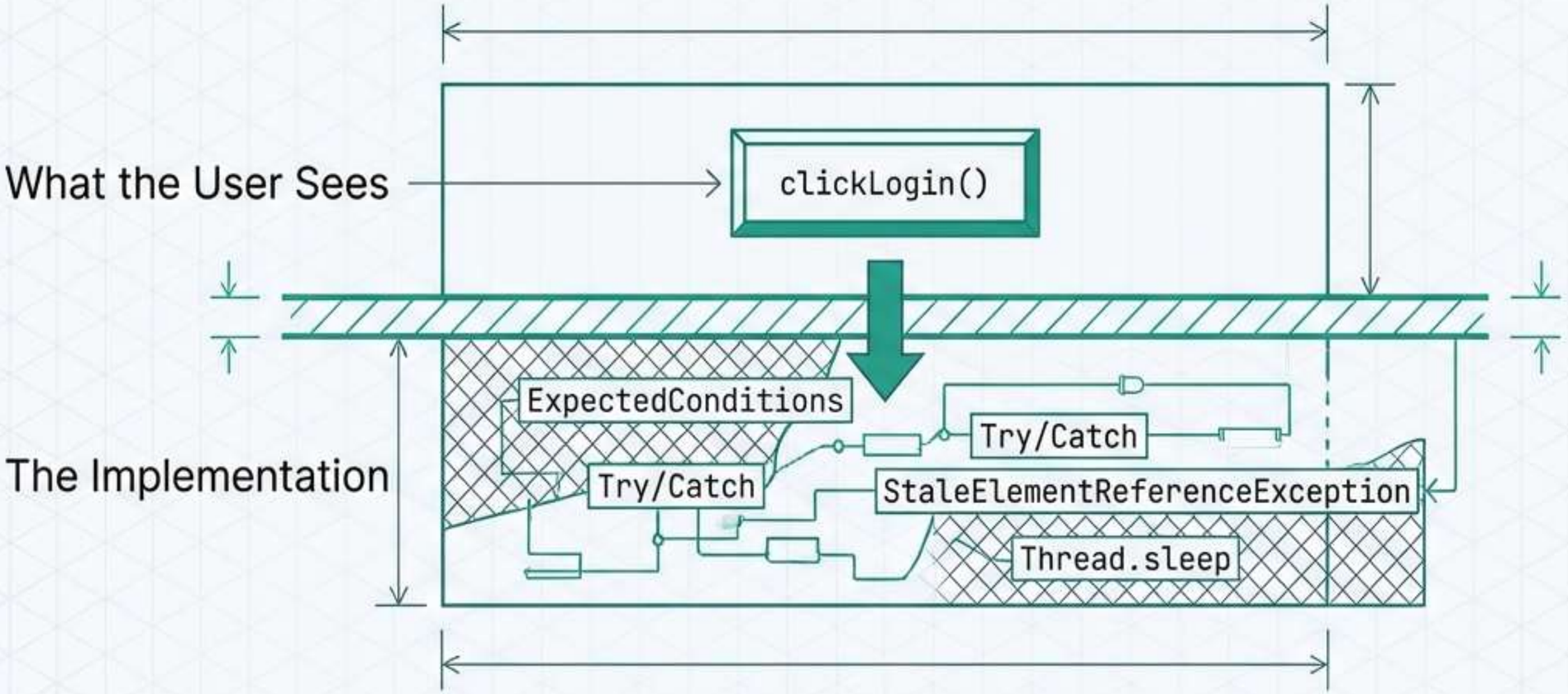
Switching between local execution and cloud grids (SauceLabs, BrowserStack) requires zero changes to the test logic when Polymorphism is used correctly.

```
WebDriver driver;

if(browser.equals('chrome')) {
   driver = new ChromeDriver();
}

if(browser.equals('firefox')) {
   driver = new FirefoxDriver();
}
```

Testleaf
Always Ahead

Abstraction: The Interface

Definition: Show the details that your user like to see and not the unimportant features.



Fluent Interfaces & Helper Classes

Abstraction empowers the test writer. They want to perform a business action, not debug a synchronization issue.