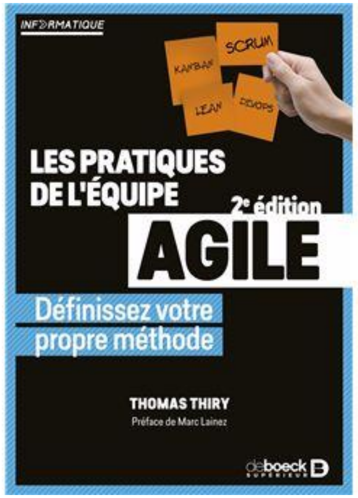
Enfin des tests que j’aime !

2024

Thomas THIRY

KEYTRADE BANK

Thomas THIRY



INFORMATIQUE

LES PRATIQUES DE L'ÉQUIPE AGILE

2e édition

Définissez votre propre méthode

THOMAS THIRY

Préface de Marc Lainez

deboeck supérieur

Coach agile technique

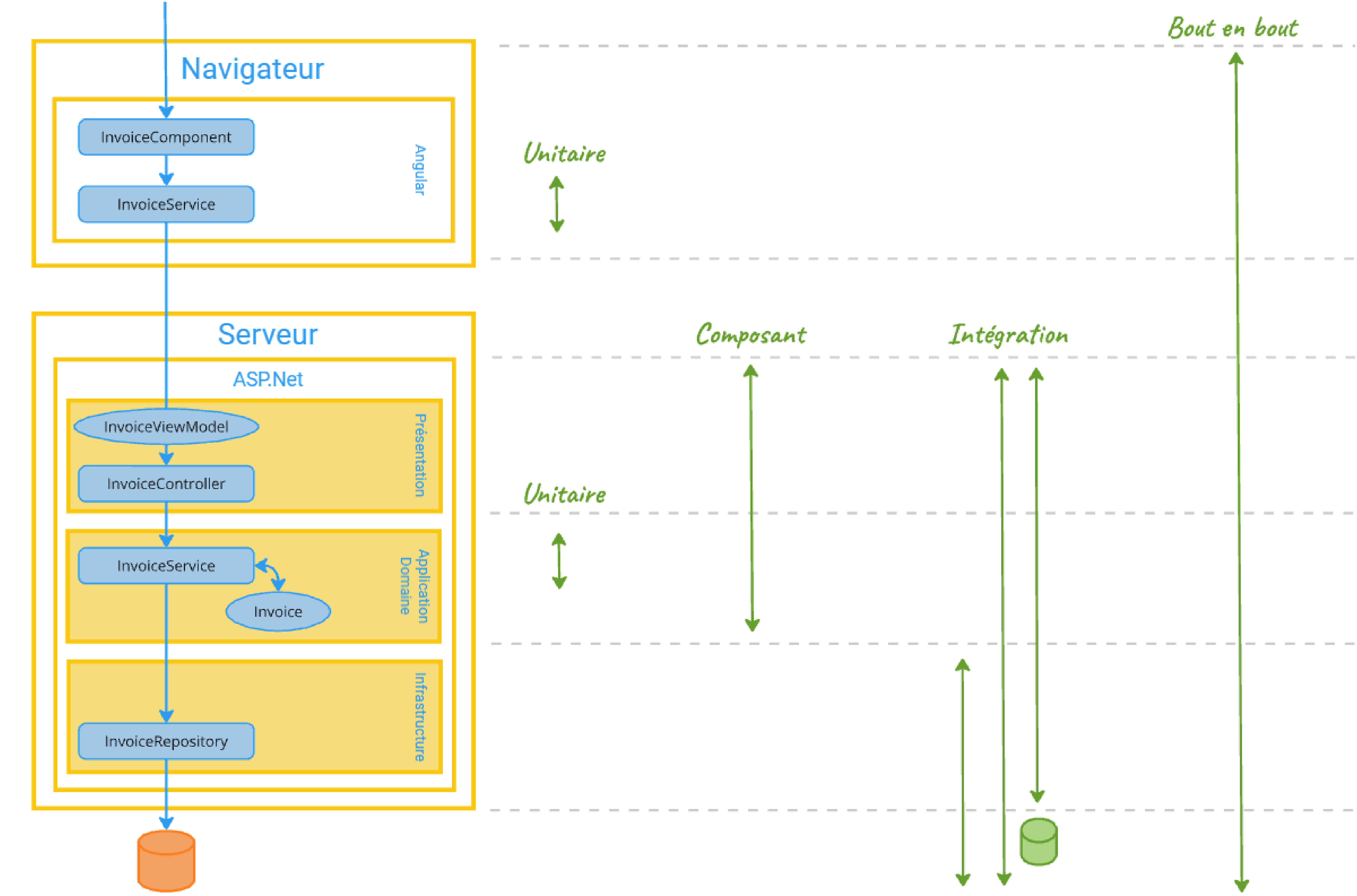
(extreme programming)

Tech lead

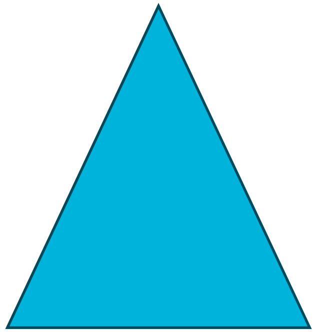
KEYTRADE BANK

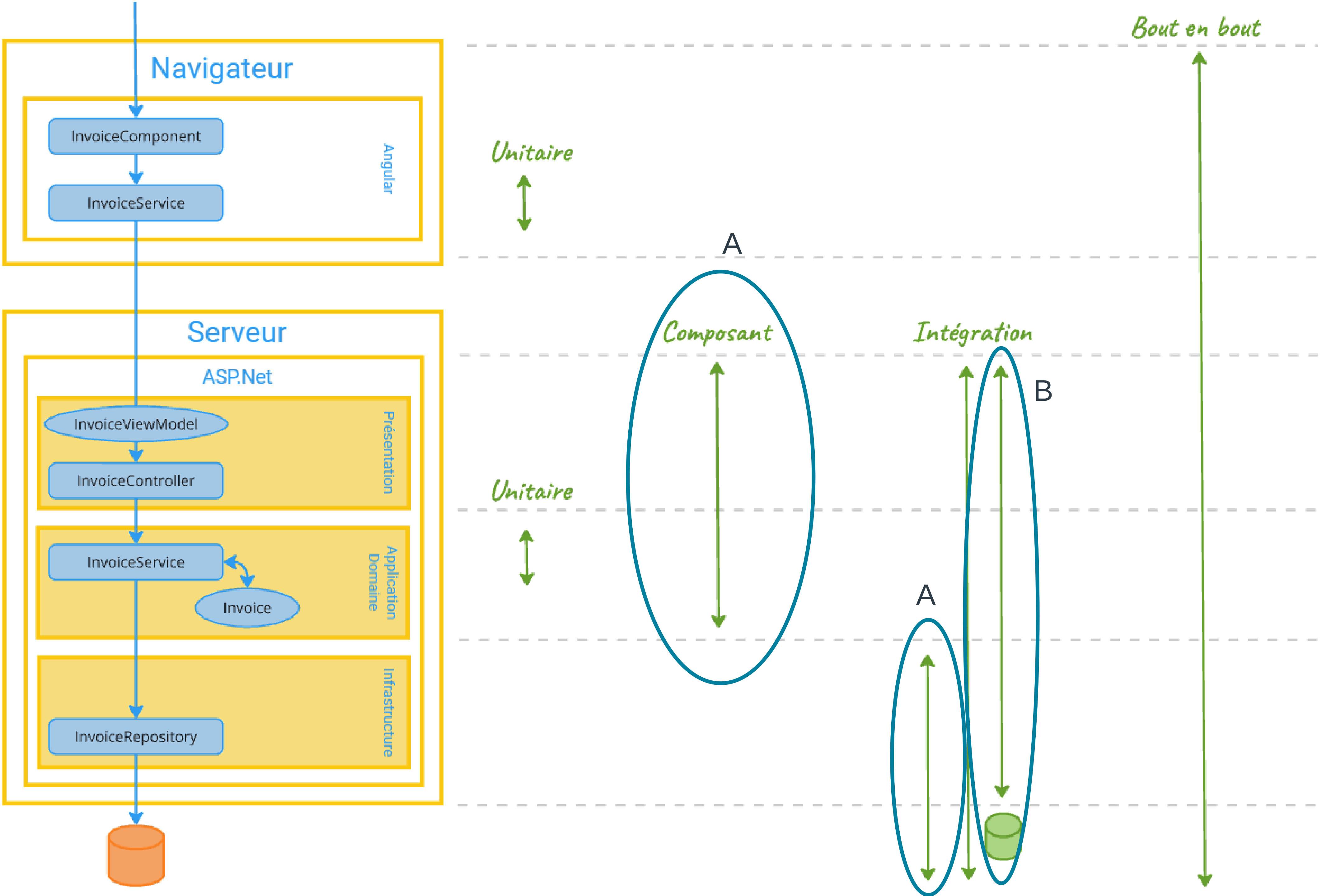
Propriétés désirables des tests (Kent Beck)

- Isolé
- Composable
- Rapide
- Inspirant
- Facile à écrire
- Lisible
- Comportemental
- Insensible à la structure
- Automatisé
- Spécifique
- Déterministe
- Prédictif



Isolé
Composable
Rapide
Inspirant
Facile à écrire
Lisible
Comportemental
Insensible à la structure
Automatisé
Spécifique
Déterministe
Prédictif





Navigateur

InvoiceComponent

InvoiceService

Angular

Serveur

ASP.Net

InvoiceViewModel

InvoiceController

Présentation

InvoiceService

Invoice

Application
Domaine

InvoiceRepository

Infrastructure

Bout en bout

Unitaire

A

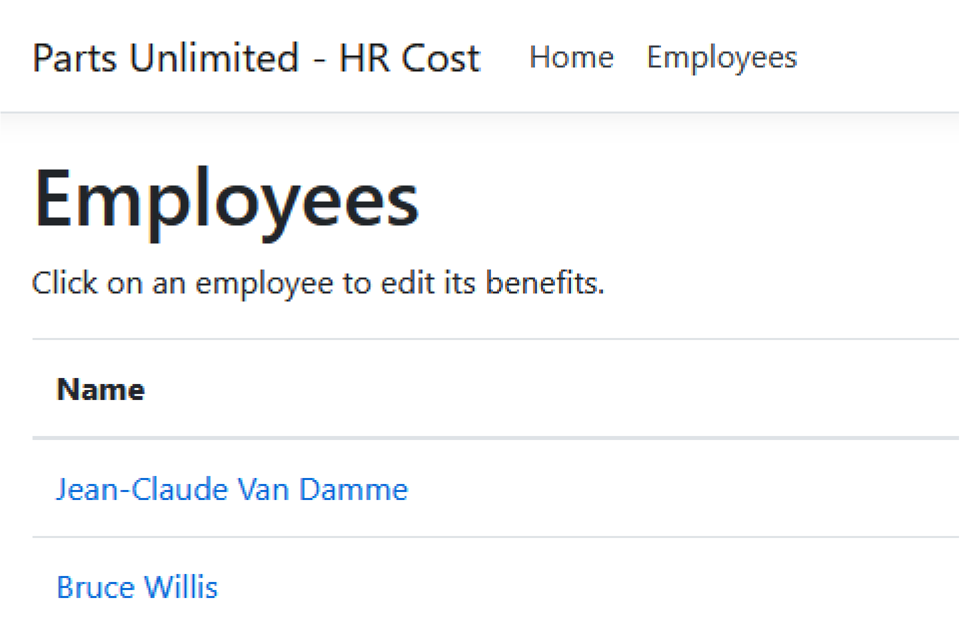
Composant

Intégration

B

Unitaire

A

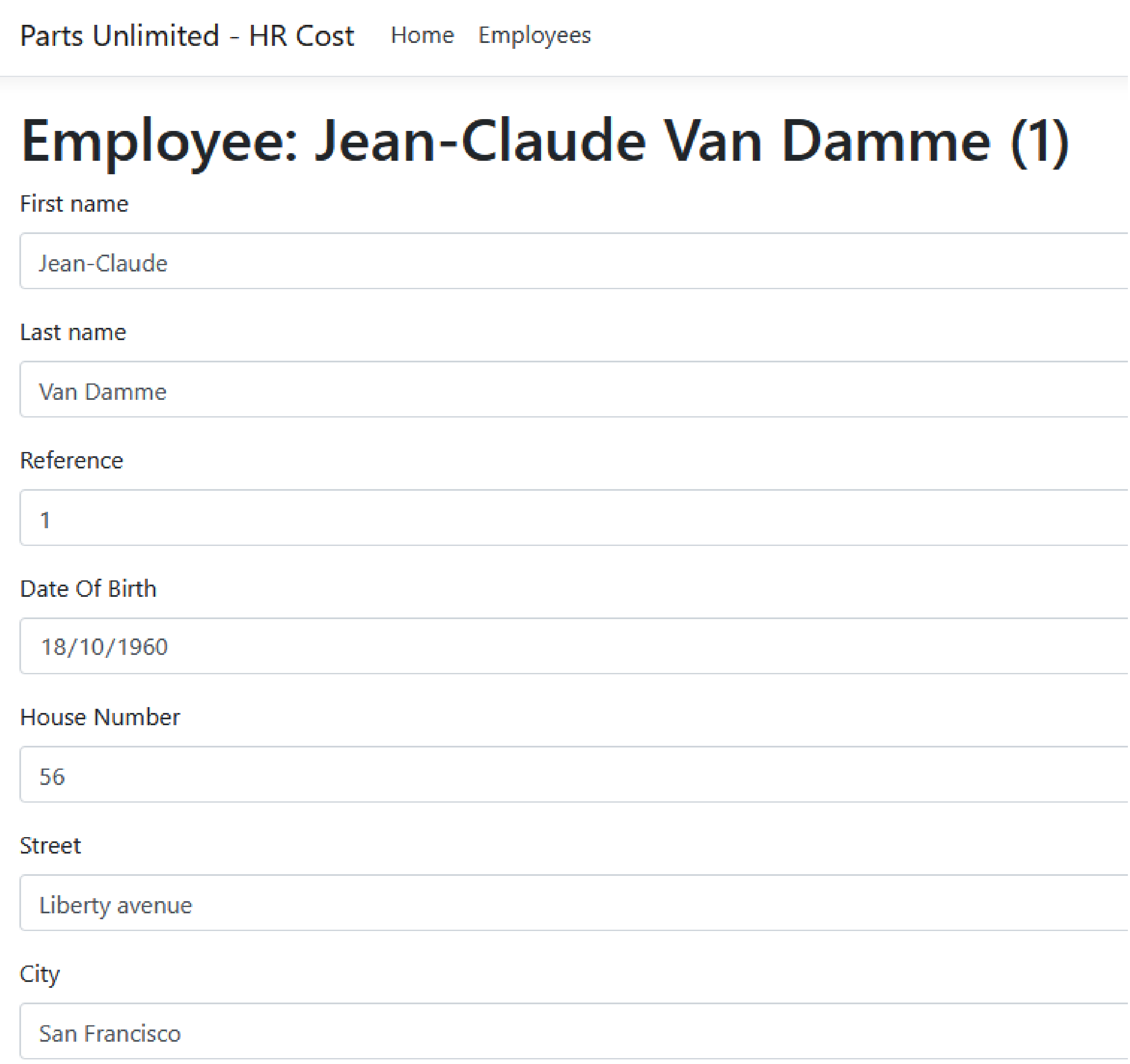


Parts Unlimited - HR Cost Home Employees

Employees

Click on an employee to edit its benefits.

Name
Jean-Claude Van Damme
Bruce Willis



Parts Unlimited - HR Cost Home Employees

Employee: Jean-Claude Van Damme (1)

First name

Jean-Claude

Last name

Van Damme

Reference

1

Date Of Birth

18/10/1960

House Number

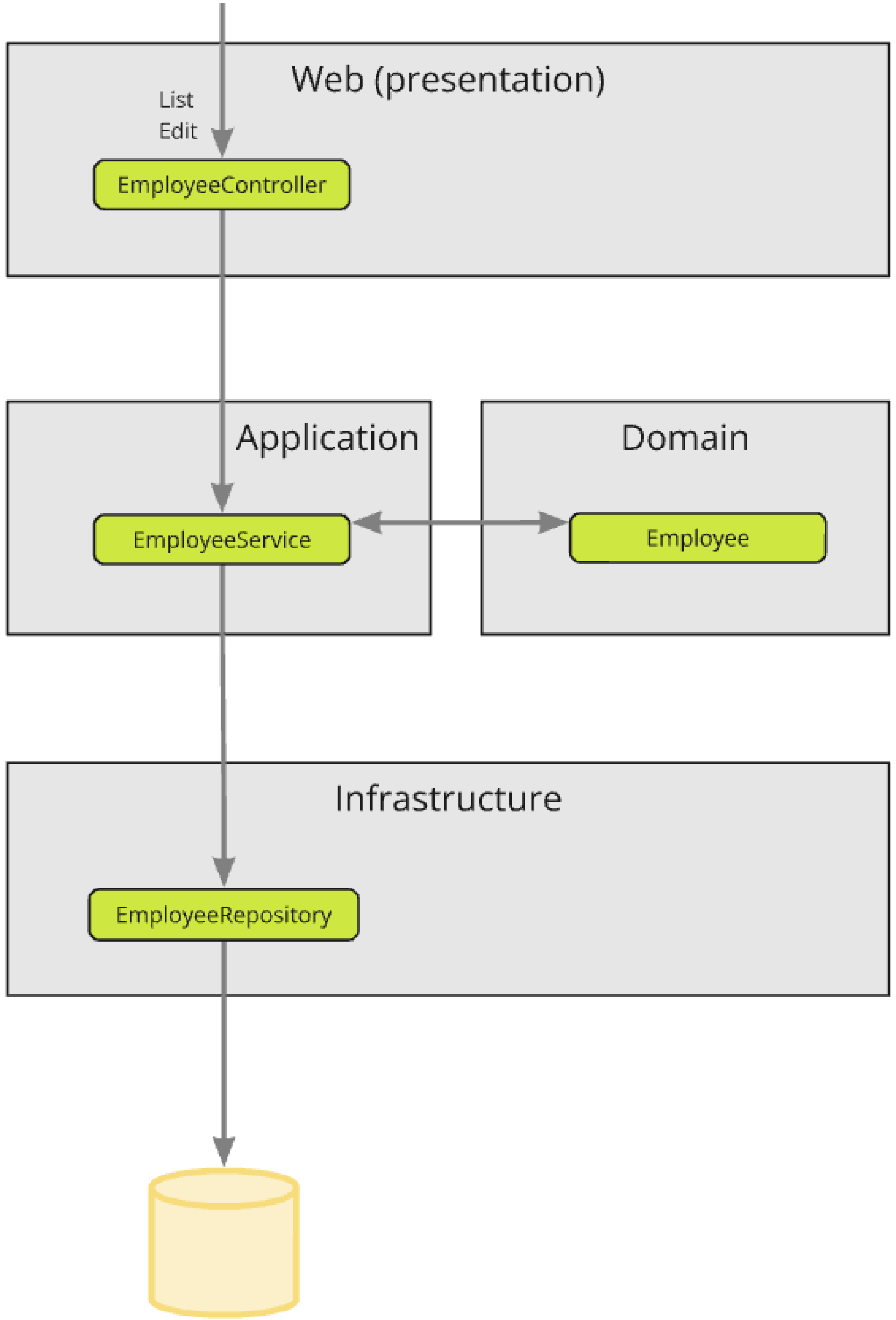
56

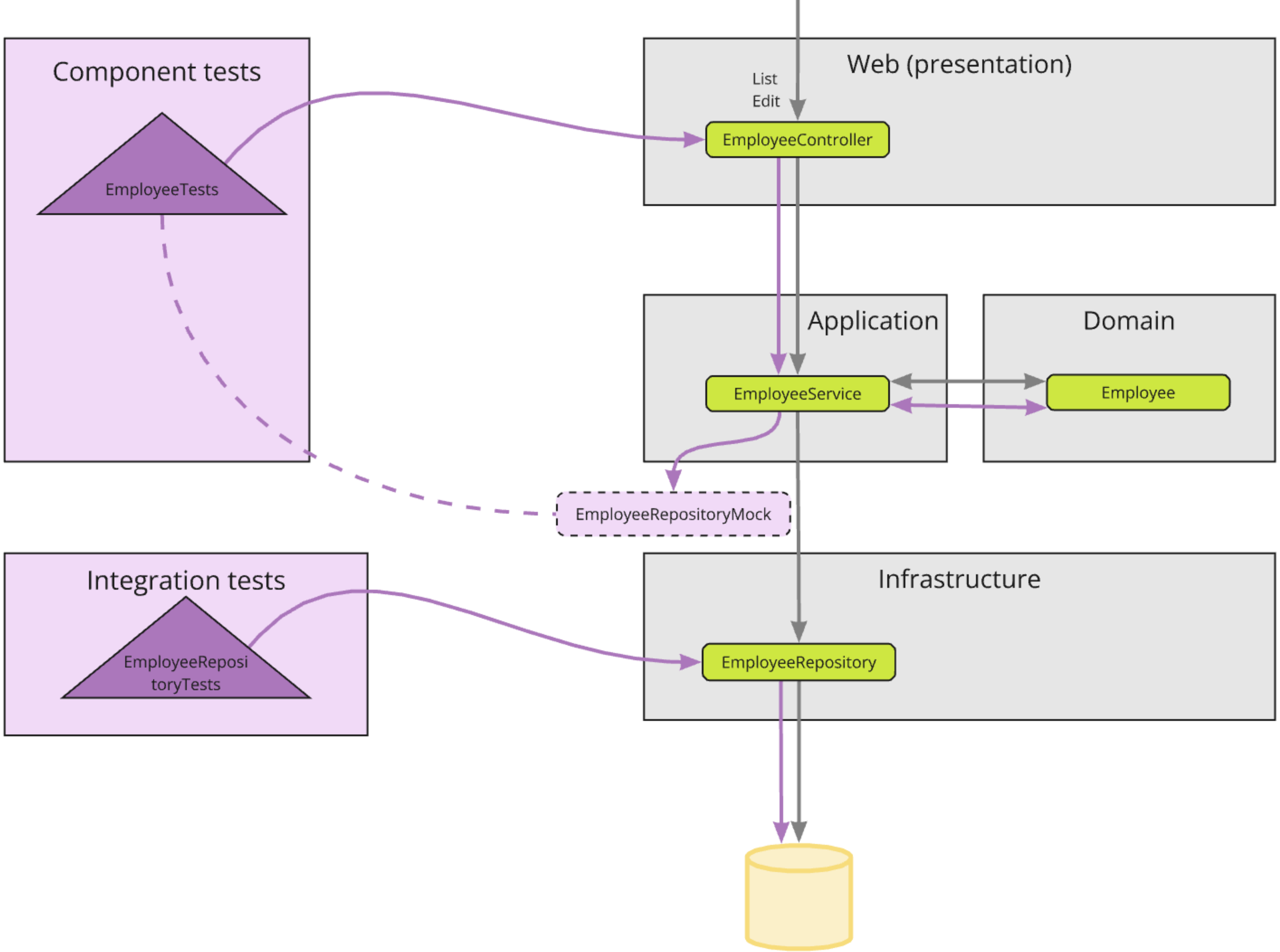
Street

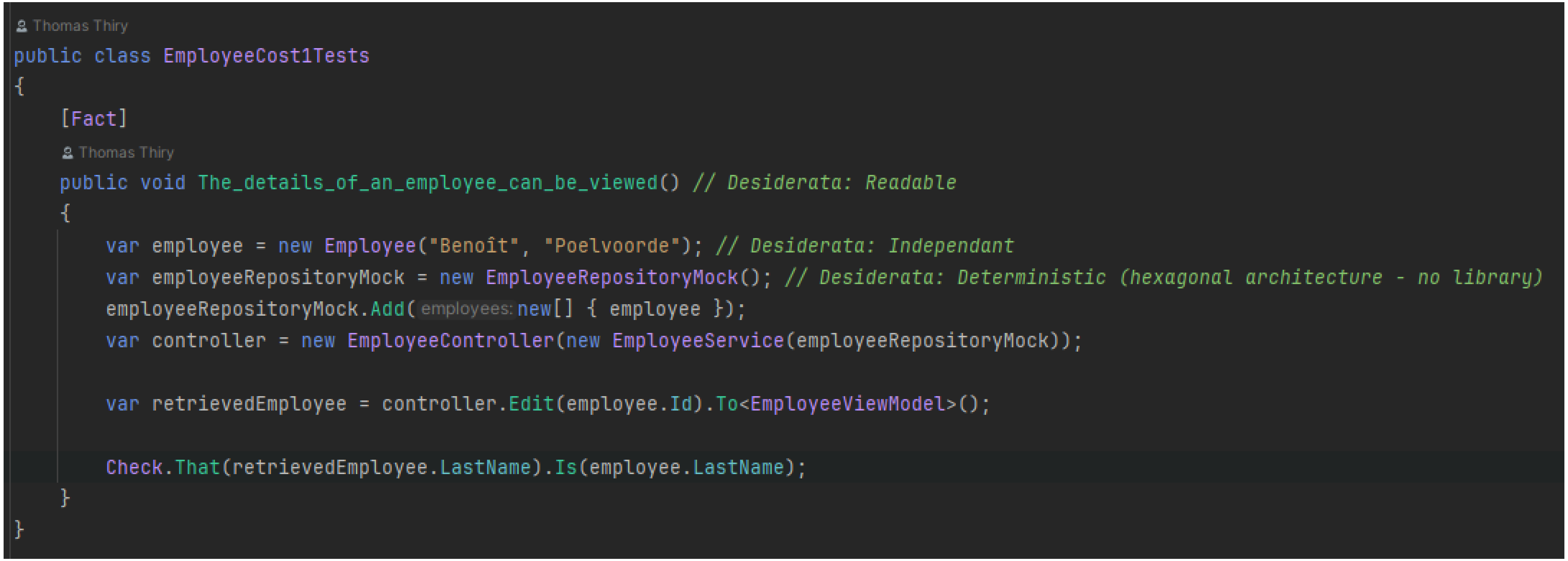
Liberty avenue

City

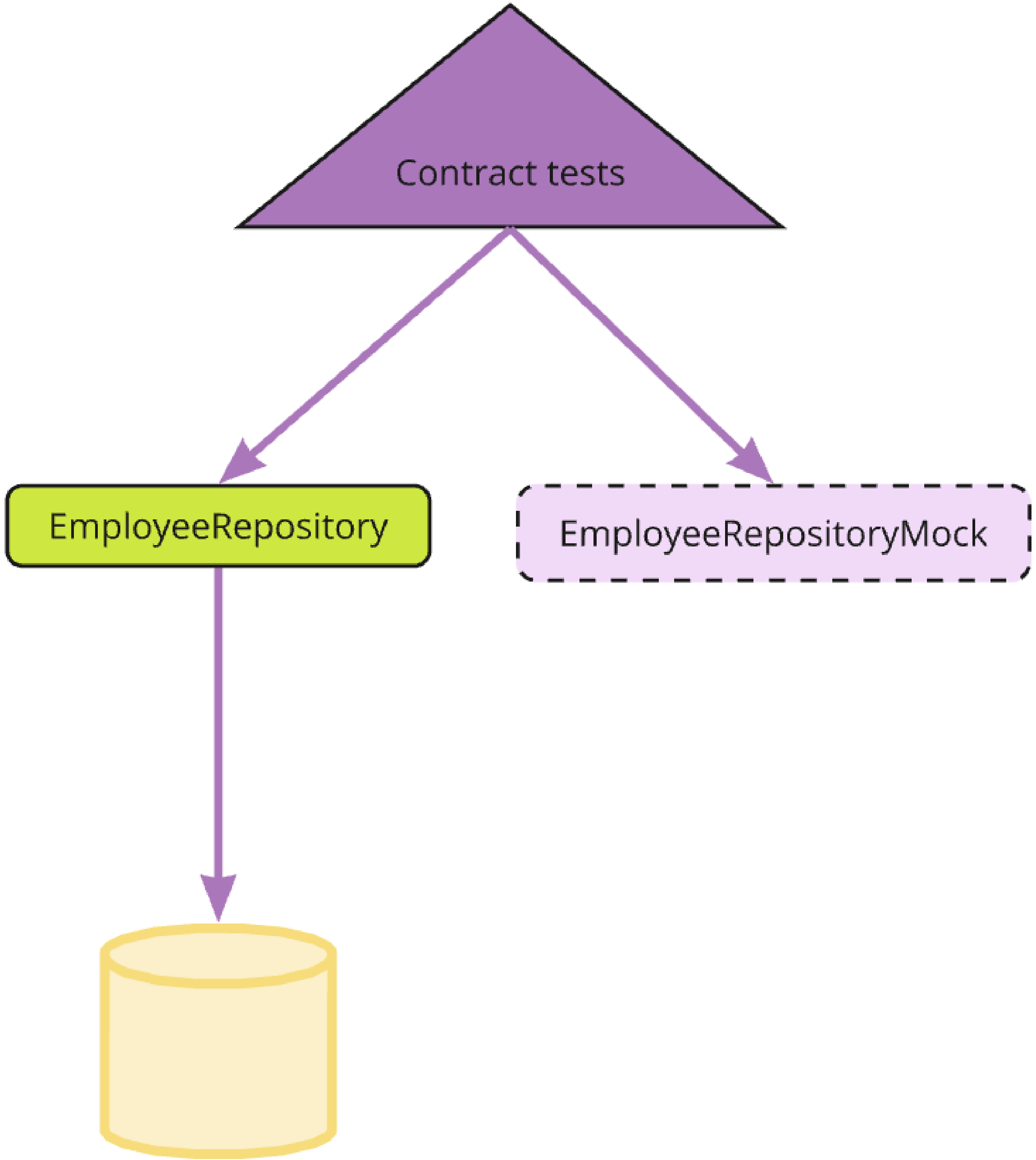
San Francisco


Web (presentation)
List
Edit
EmployeeController
Application
EmployeeService
Domain
Employee
Infrastructure
EmployeeRepository

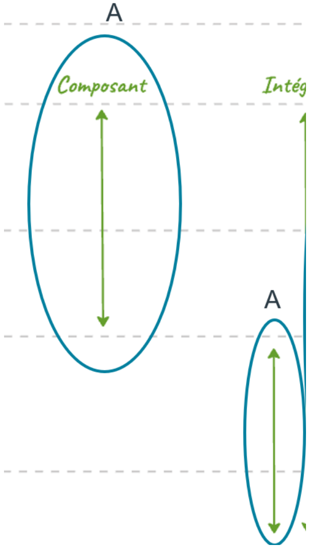

Component tests
EmployeeTests
Web (presentation)
List
Edit
EmployeeController
Application
EmployeeService
Domain
Employee
EmployeeRepositoryMock
Integration tests
EmployeeReposi
toryTests
Infrastructure
EmployeeRepository



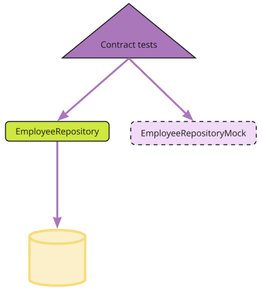
```
Thomas Thiry
public class EmployeeCost1Tests
{
    [Fact]
    Thomas Thiry
    public void The_details_of_an_employee_can_be_viewed() // Desiderata: Readable
    {
        var employee = new Employee("Benoît", "Poelvoorde"); // Desiderata: Independant
        var employeeRepositoryMock = new EmployeeRepositoryMock(); // Desiderata: Deterministic (hexagonal architecture - no library)
        employeeRepositoryMock.Add( employees: new[] { employee });
        var controller = new EmployeeController(new EmployeeService(employeeRepositoryMock));

        var retrievedEmployee = controller.Edit(employee.Id).To<EmployeeViewModel>();

        Check.That(retrievedEmployee.LastName).Is(employee.LastName);
    }
}
```


Contract tests
EmployeeRepository
EmployeeRepositoryMock

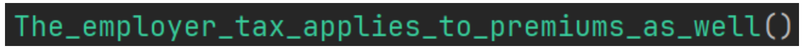
Conclusion



Tests de composant
+ intégration

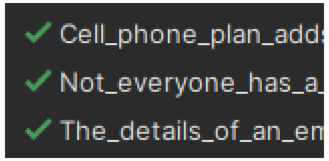


Tests de contrat



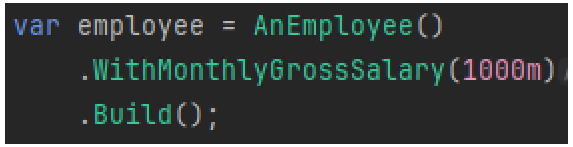
```
The_employer_tax_applies_to_premiums_as_well()
```

Noms



✓ Cell_phone_plan_add
✓ Not_everyone_has_a_
✓ The_details_of_an_em

Déterministe



```
var employee = AnEmployee()
    .WithMonthlyGrossSalary(1000m)
    .Build();
```

Test data builders

Pour aller plus loin…

FEEDBACK

GitHub

https://github.com/thomasthiry/PartsUnlimited.HRCost

https://kentbeck.github.io/TestDesiderata/

https://principal-it.eu/2023/02/contract-tests/

https://martinfowler.com/articles/microservice-testing/

Test data builders : http://natpryce.com/articles/000714.html