



Open Mic Night 2024

11 dicembre

Impact Hub Via Panciatichi, 16, Fi



Clean Architecture: dal caos all'ordine

Giuseppe De Luca





GIUSEPPE DE LUCA



delucagiuseppe@outlook.it

<https://www.linkedin.com/in/giuseppe-dl/>

AVETE MAI LAVORATO IN UN PROGETTO IN CUI...



LEGGENDO IL CODICE SI FATICA A CAPIRE LA
LOGICA DI BUSINESS CHE E' NASCOSTA DA
DETTAGLI TECNICI?

AVETE MAI LAVORATO IN UN PROGETTO IN CUI...

NON CI SONO LAYER CHE SUDDIVIDONO
L'APPLICAZIONE?



AVETE MAI LAVORATO IN UN PROGETTO IN CUI...



OPPURE LAYER CI SONO MA
SONO CONFUSI,
COMPLICATI, RIGIDAMENTE
INTERCONNESSI?

AVETE MAI LAVORATO IN UN PROGETTO IN CUI...

TEST DIFFICILI O FRAGILI?



AVETE MAI LAVORATO IN UN PROGETTO IN CUI...



CAMBIARE E' UN INCUBO E
CAPIRE E' UN SOGNO??

AVETE MAI LAVORATO IN UN PROGETTO IN CUI...

LOGICA DI BUSINESS NASCOSTA DA DETTAGLI TECNICI?

NON CI SONO LAYER CHE SUDDIVIDONO L'APPLICAZIONE?

LAYER CONFUSI, COMPLICATI, RIGIDAMENTE INTERCONNESSI?

TEST DIFFICILI O FRAGILI?

CAMBIARE E' UN INCUBO E CAPIRE E' UN SOGNO??



VEDIAMO ALCUNI ESEMPI



SOLUZIONE?



ESEMPIO 1: REQUISITI

Implementare la funzionalità di inserimento di un nuovo utente sulla nostra applicazione:

- Accessibile tramite Web API Rest
- Gli attributi da gestire sono nome, cognome, email (obbligatori)
- Deve salvare i dati nel database
- Deve inviare un messaggio di benvenuto
- Se richiesto, deve iscrivere l'utente alla newsletter
- Infine, deve configurare la libreria di documenti dell'utente

ESEMPIO 1: INSERIMENTO DI UN UTENTE

HttpRequest validation



```
public class User{  
  
    public HttpResponseMessage InsertUser(HttpRequest httpRequest){  
  
        if(httpRequest.Body == null) ReturnError("Empty body");  
        if(IsValidJson(httpRequest.Body) == false) ReturnError("Body is not a valid json");  
        // other request validation  
    }  
}
```

ESEMPIO 1: INSERIMENTO DI UN UTENTE

Data binding



```
string name = jsonRequest.GetFieldValue("name");  
string surname = jsonRequest.GetFieldValue("surname");  
string email = jsonRequest.GetFieldValue("email");  
bool subscribeNewsletter = jsonRequest.GetFieldValue("subscribeNewsletter");  
//...other fields...
```

ESEMPIO 1: INSERIMENTO DI UN UTENTE

Data validation



```
if (string.IsNullOrEmpty(name)) ReturnError("Name is required");  
if (string.IsNullOrEmpty(surname)) ReturnError("Surname is required");  
// ...other fields validation...
```

ESEMPIO 1: INSERIMENTO DI UN UTENTE

Data persistence



```
int id = database.Insert("INSERT INTO Users (Name, Surname, Email) VALUES @p1, @p2, @p3", name, surname, email);
```


ESEMPIO 1: INSERIMENTO DI UN UTENTE

Other functions



```
sendWelcomeMessage(email);
```

```
if (subscribeNewsletter){  
    var newsletterManager = new NewsletterManager();  
    var newsletterSubscription = new NewsletterSubscription(type := "default", userId := id);  
    newsletterManager.AddSubscription(newsletterSubscription);  
}
```

```
// user document library configuration  
var documentLibraryClient = new DocumentLibraryClient();  
documentLibraryClient.Configure();  
//...ecc
```

ESEMPIO 1: INSERIMENTO DI UN UTENTE

HttpResponse



```
var response = new HttpResponseMessage(HttpStatusCode.Ok, new { UserId = userId});  
return response;
```

ESEMPIO 1: INSERIMENTO DI UN UTENTE

```
1 public class User{
2
3     public HttpResponseMessage InsertUser(HttpRequest httpRequest){
4
5         if(httpRequest.IsJson()){
6             if(IsValidJson(httpRequest.Content.ReadAsStringAsync().Result, "json")){
7                 // other
8
9                 string name = httpRequest.GetHeader("name");
10                string surname = httpRequest.GetHeader("surname");
11                string email = httpRequest.GetHeader("email");
12                bool subscribed = httpRequest.GetHeader("subscribed") == "true";
13                // ...other
14
15                if (string.IsNullOrEmpty(name) || string.IsNullOrEmpty(surname) || string.IsNullOrEmpty(email))
16                {
17                    // ...other
18
19                    int id = 1;
20                    sendWelcomeEmail(id, @p2, @p3", name, surname, email);
21
22                }
23                if (subscribed)
24                {
25                    var user = new User { Name = name, Surname = surname, Email = email, Subscribed = subscribed };
26                    news = new News { Title = "New user", UserId = id };
27                }
28
29                // user
30                var document = new Document { Name = name, Surname = surname, Email = email, Subscribed = subscribed };
31                // ...other
32
33                var response = new HttpResponseMessage(HttpStatusCode.Ok, new { UserId = id });
34                return response;
35            }
36        }
37    }
38 }
```



Quante responsabilità ha questa funzione?



Dobbiamo sviluppare una nuova funzionalità che permetta l'importazione di utenti da file. Possiamo riutilizzare lo stesso codice?



Posso supportare tipologie di database diverse (Mysql, Postgress)?



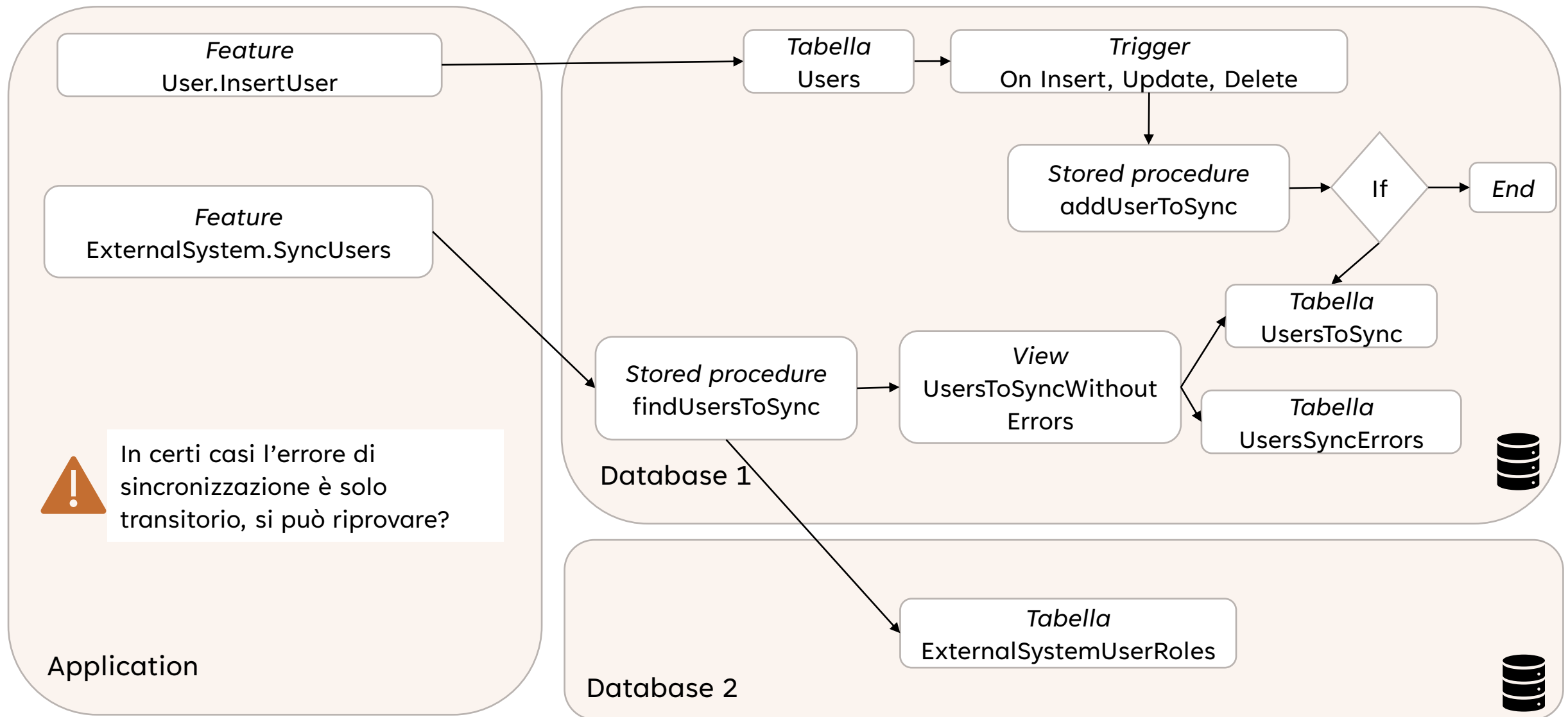
Posso scrivere test automatici???

ESEMPIO 2: REQUISITI

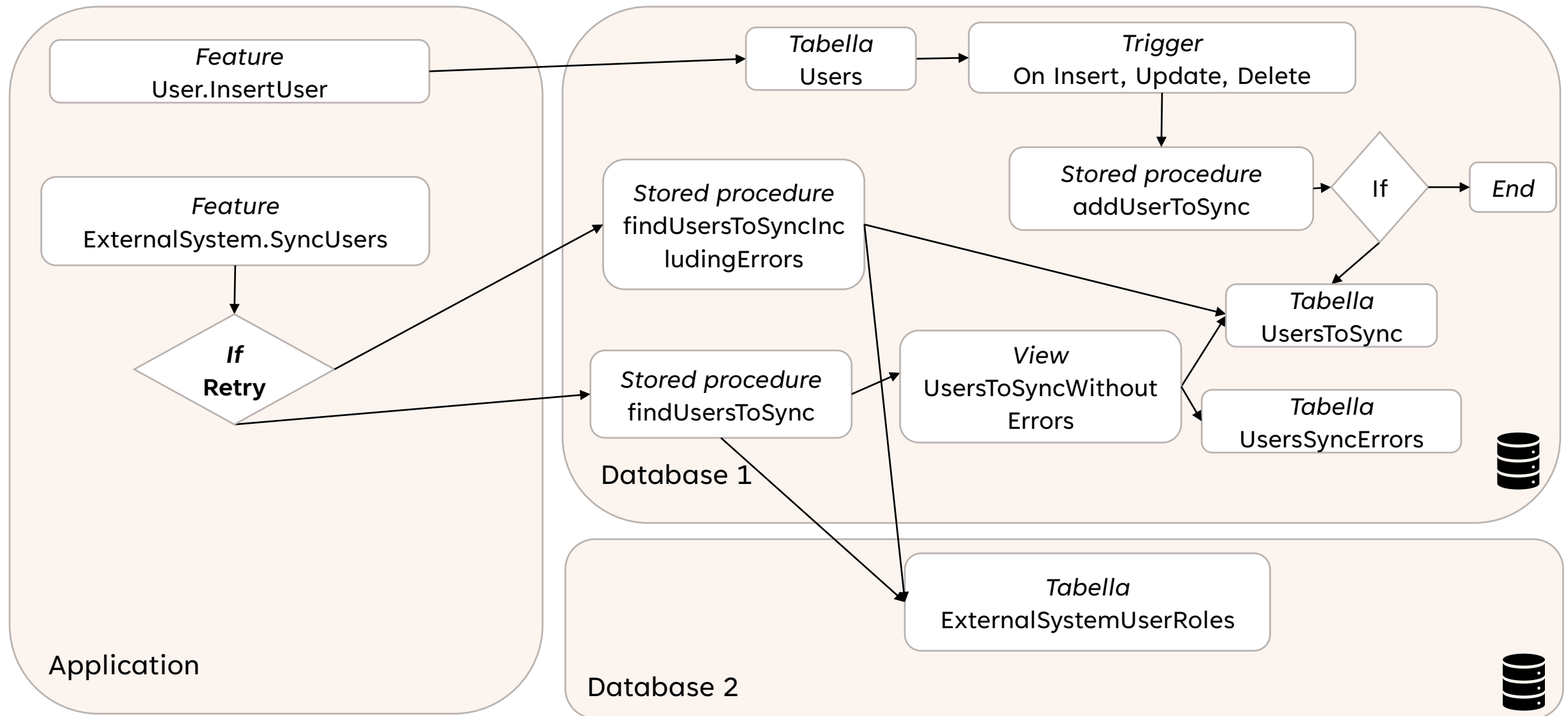
Implementare funzionalità di sincronizzazione degli utenti della nostra applicazione con una applicazione esterna:

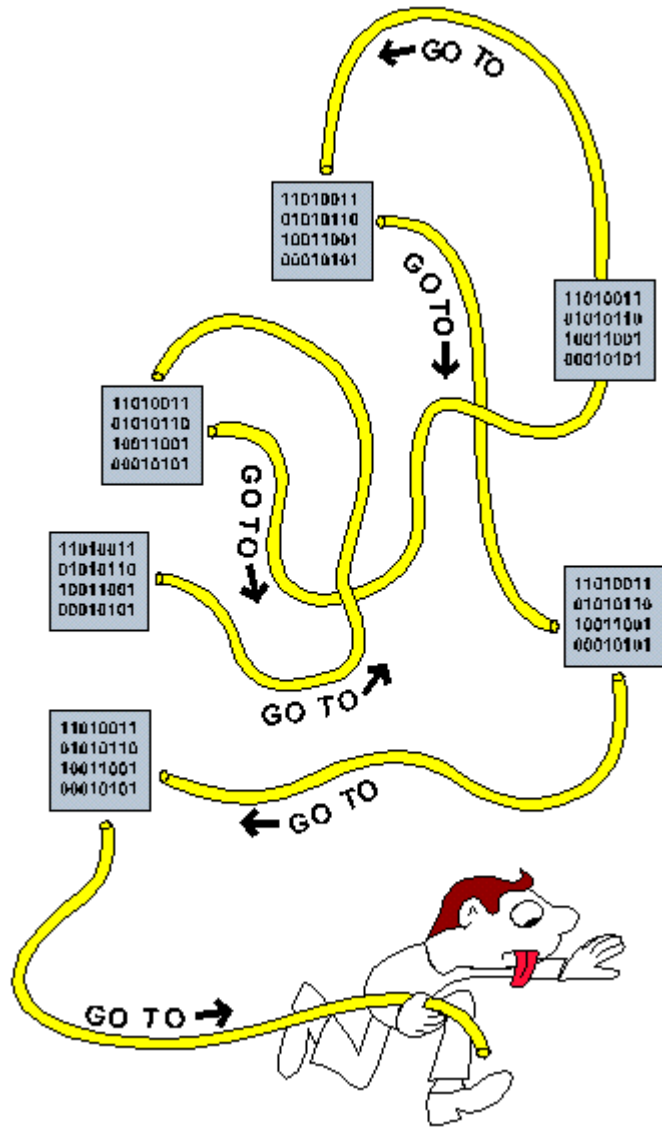
- Ogni operazione su un utente deve essere storicizzata (inserimento, modifica e cancellazione)
- La procedura di sincronizzazione deve raccogliere tutte le modifiche avvenute dopo l'ultima esecuzione e notificarle all'applicazione esterne
- La pianificazione della sincronizzazione deve essere configurabile
- La sincronizzazione riguarda solo gli utenti che soddisfano una certa condizione
- Infine, eventuali errori di sincronizzazione devono essere storicizzati e il relativo utente deve essere escluso dalle sincronizzazioni successive

ESEMPIO 2: SINCRONIZZAZIONE UTENTI SU APPLICAZIONE ESTERNA



ESEMPIO 2: SINCRONIZZAZIONE UTENTI SU APPLICAZIONE ESTERNA

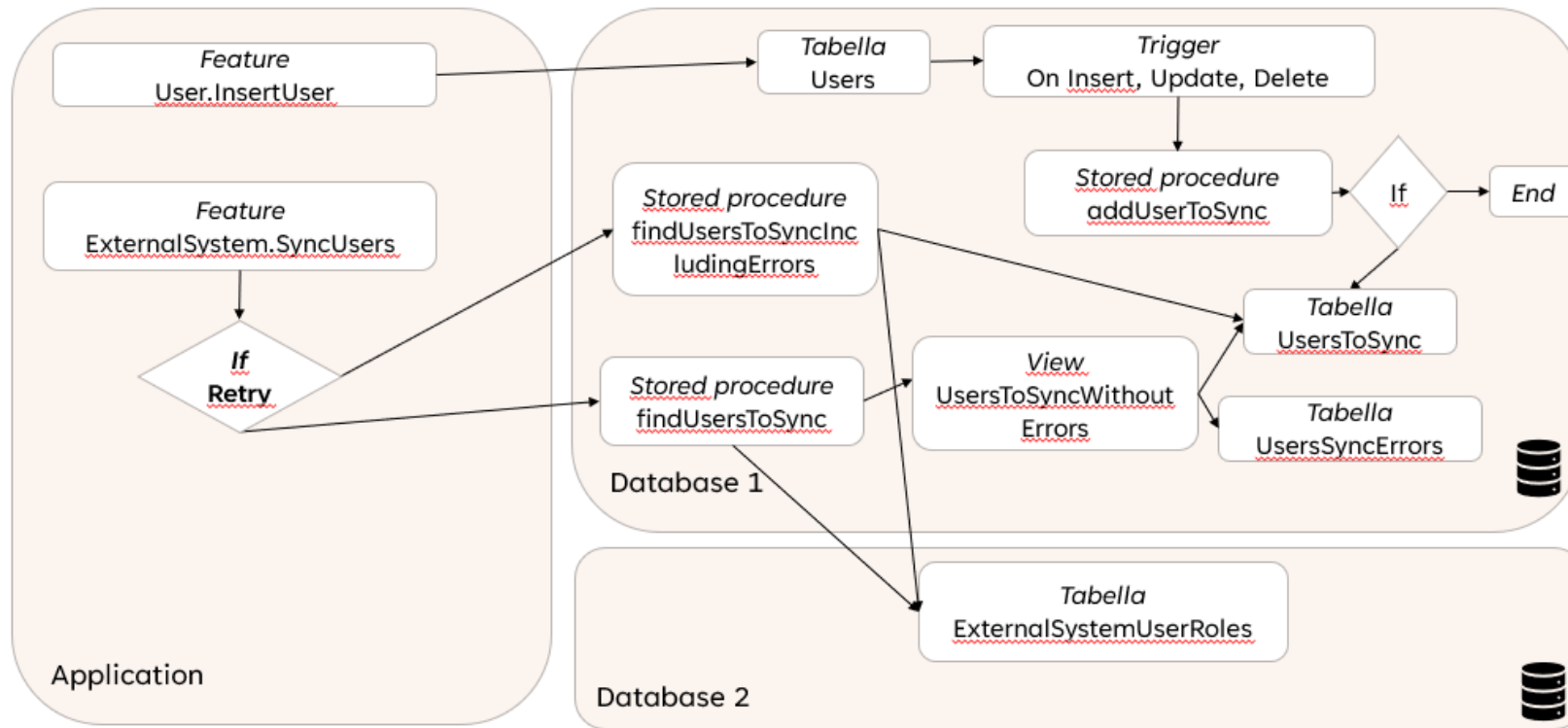




~~SPAGHETTI CODE~~

SPAGHETTI ARCHITECTURE

ESEMPIO 2: SINCRONIZZAZIONE UTENTI SU APPLICAZIONE ESTERNA



I layer ci sono però...

- ! Logiche di business sparse tra applicazione e database
- ! Manutenibilità: migliore a livello applicativo
- ! Scalabilità: più semplice a livello applicativo
- ! Testabilità: più semplice a livello applicativo



Robert Martin (Uncle Bob)

Robert C. Martin Series



Clean Architecture

A Craftsman's Guide to
Software Structure and Design

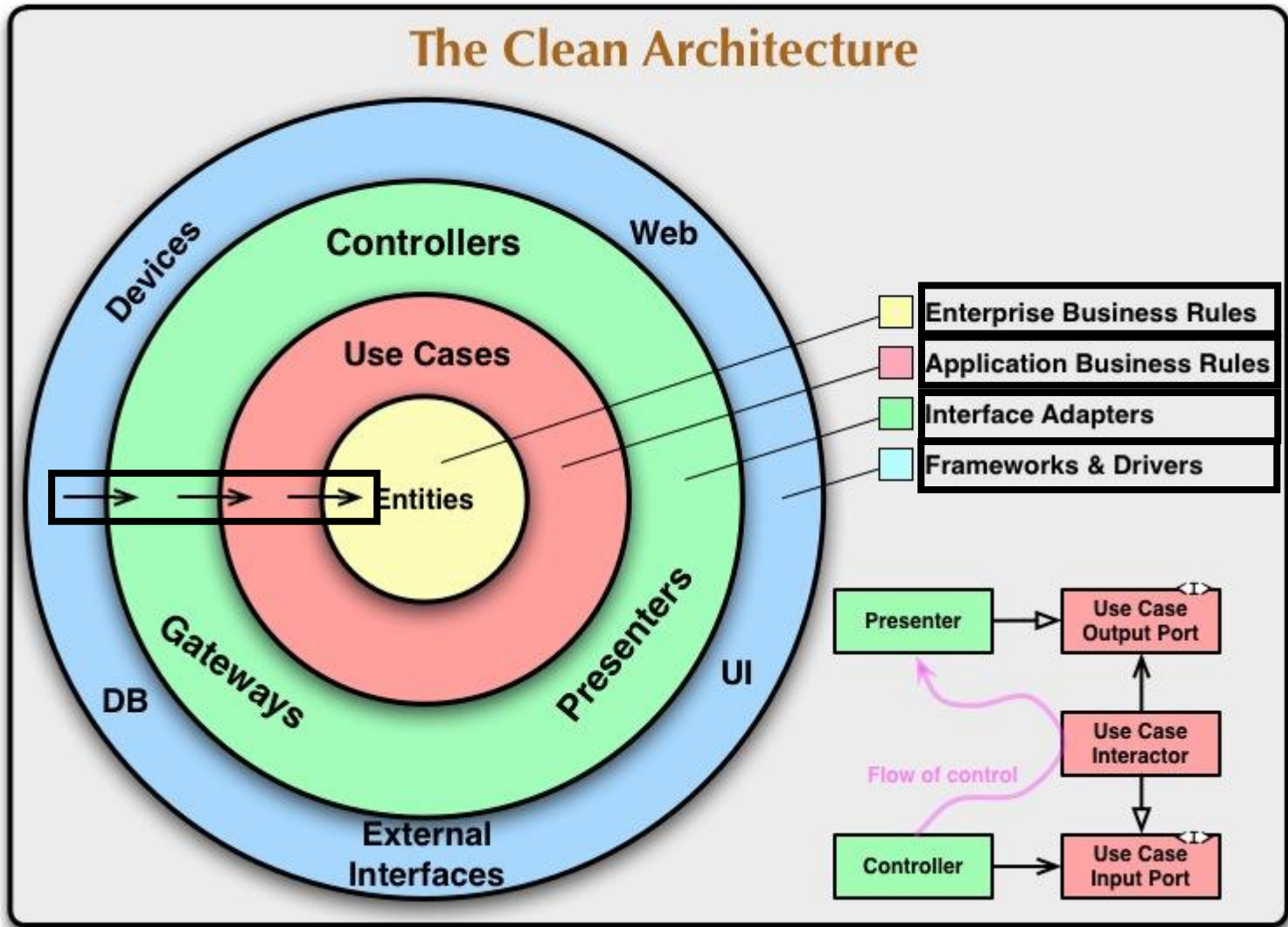
Robert C. Martin

With contributions by James Grenning and Simon Brown

Foreword by Kevlin Henney

Afterword by Jason Gorman





Suddivisione in layer

- Entities: logica di business del dominio

- Use Cases: orchestrazione tra dominio e componenti esterne

- Interface adapters: connettono use cases a input/output (API, database)

- Framework & driver: database, UI, ecc

Dipendenze verso l'interno

La applichiamo all'esempio 1!

ESEMPIO 1: INSERIMENTO DI UN UTENTE

```
1 public class User{
2
3     public HttpResponseMessage InsertUser(HttpRequest httpRequest){
4
5         if(httpRequest.Body == null) ReturnError("Empty body");
6         if(!IsValidJson(httpRequest.Body)) ReturnError("Body is not a valid json");
7         // other request validation
8
9         string name = jsonRequest.GetFieldValue("name");
10        string surname = jsonRequest.GetFieldValue("surname");
11        string email = jsonRequest.GetFieldValue("email");
12        bool subscribeNewsletter = jsonRequest.GetFieldValue("subscribeNewsletter");
13        //...other fields...
14
15        if (string.IsNullOrEmpty(name)) ReturnError("Name is required");
16        if (string.IsNullOrEmpty(surname)) ReturnError("Surname is required");
17        // ...other fields validation...
18
19        int id = database.Insert("INSERT INTO Users (Name, Surname, Email) VALUES @p1, @p2, @p3", name, surname, email);
20
21        sendWelcomeMessage(email);
22
23        if (subscribeNewsletter){
24            var newsletterManager = new NewsletterManager();
25            var newsletterSubscription = new NewsletterSubscription(type := "default", userId := id);
26            newsletterManager.AddSubscription(newsletterSubscription);
27        }
28
29        // user document library configuration
30        var documentLibraryClient = new DocumentLibraryClient();
31        documentLibraryClient.Configure();
32        //...ecc
33
34        var response = new HttpResponseMessage(StatusCode.Ok, new { UserId = userId});
35        return response;
36    }
37
38 }
```



Quante responsabilità ha questa funzione?



Dobbiamo sviluppare una nuova funzionalità che permetta l'importazione di utenti da file. Possiamo riutilizzare lo stesso codice?



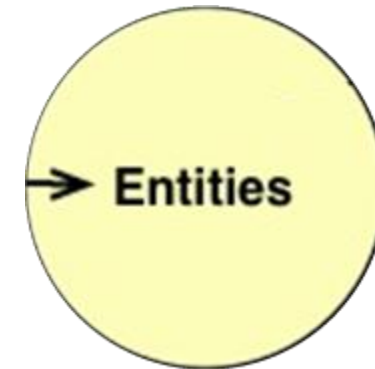
Posso supportare tipologie di database diverse (Mysql, Postgress)?



Posso scrivere test automatici???

ESEMPIO 1: INSERIMENTO DI UN UTENTE

```
//Entity
public class User
{
    public int Id { get; private set; }
    public string Name { get; private set; }
    public string Surname { get; private set; }
    public string Email { get; private set; }
}
```



ESEMPIO 1: INSERIMENTO DI UN UTENTE

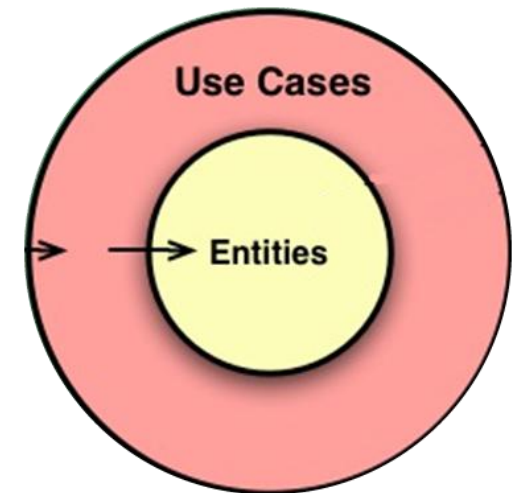
```
// Use case
public class UserService
{
    public UserService(IUserRepository userRepository, IEmailService emailService,
        INewsletterService newsletterService, IDocumentLibraryService documentLibraryService)
    {
        // ...
    }

    public int InsertUser(User user, bool subscribeNewsletter)
    {
        if (string.IsNullOrEmpty(user.Name)) throw new ArgumentException("Name is
required");
        if (string.IsNullOrEmpty(user.Surname)) throw new
ArgumentException("Surname is required");

        int id = _userRepository.Insert(user);

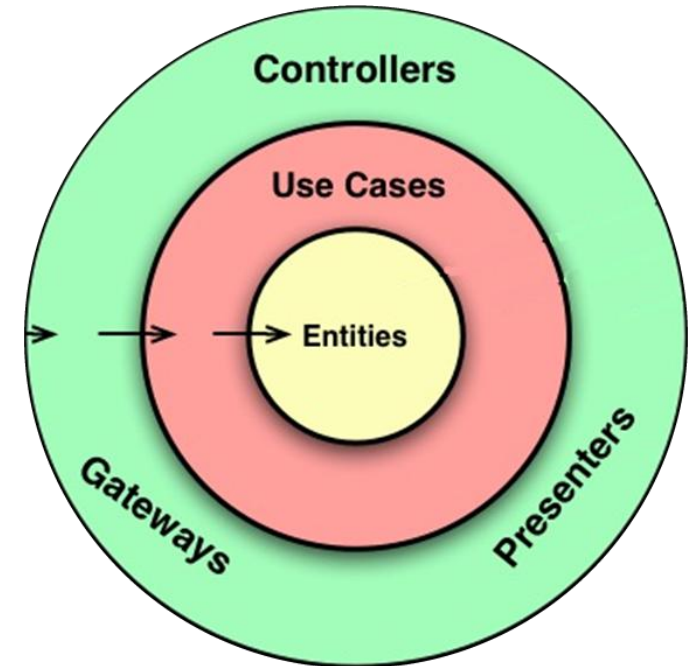
        _emailService.SendWelcomeMessage(user.Email);
        if (subscribeNewsletter)
            _newsletterService.AddUserToNewsletter(id);
        _documentLibraryService.Configure();

        return id;
    }
}
```



ESEMPIO 1: INSERIMENTO DI UN UTENTE

```
// Repository
public class UserRepository : IUserRepository
{
    public int Insert(User user)
    {
        // Implement the database insert logic
        return _database.Insert("INSERT INTO Users (Name, Surname, Email)
VALUES (@p1, @p2, @p3)", user.Name, user.Surname, user.Email);
    }
}
```



ESEMPIO 1: INSERIMENTO DI UN UTENTE

```
// Controller
public class UserController{

    public void InsertUser(HttpRequest httpRequest){

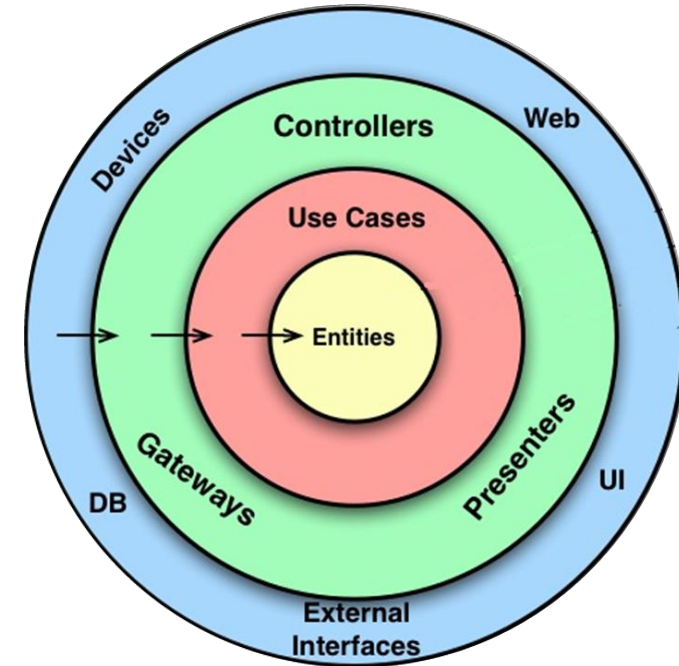
        if (httpRequest.Body == null) ReturnError("Empty body");
        if (IsValidJson(httpRequest.Body) == false) ReturnError("Body is not a valid
json");
        // other request validation

        string name = jsonRequest.GetFieldValue("name");
        string surname = jsonRequest.GetFieldValue("surname");
        string email = jsonRequest.GetFieldValue("email");
        bool subscribeNewsletter = jsonRequest.GetFieldValue("subscribeNewsletter");
        //...other fields...

        User user = new User(name, surname, email);

        var userId = _userService.InsertUser(user, subscribeNewsletter);

        var response = new HttpResponseMessage(StatusCode.Ok, new { UserId = userId});
        return response;
    }
}
```



ESEMPIO 1: INSERIMENTO DI UN UTENTE

Quante responsabilità ha questa funzione?



*Dobbiamo sviluppare una nuova funzionalità che permetta l'importazione di utenti da file.
Possiamo riutilizzare lo stesso codice?*



Posso supportare tipologie di database diverse (Mysql, Postgress)?



Posso scrivere test automatici???

UNIT TEST

```
[Fact]
public void InsertUser_MissingName_ThrowsArgumentException()
{
    // Arrange
    var user = new User("", "Doe", "john.doe@example.com");

    // Act & Assert
    var exception = Assert.Throws<ArgumentException>(() => _userService.InsertUser(user));
    Assert.Equal("Name is required", exception.Message);
}

[Fact]
public void InsertUser_MissingSurname_ThrowsArgumentException()
{
    // Arrange
    var user = new User("John", "", "john.doe@example.com");

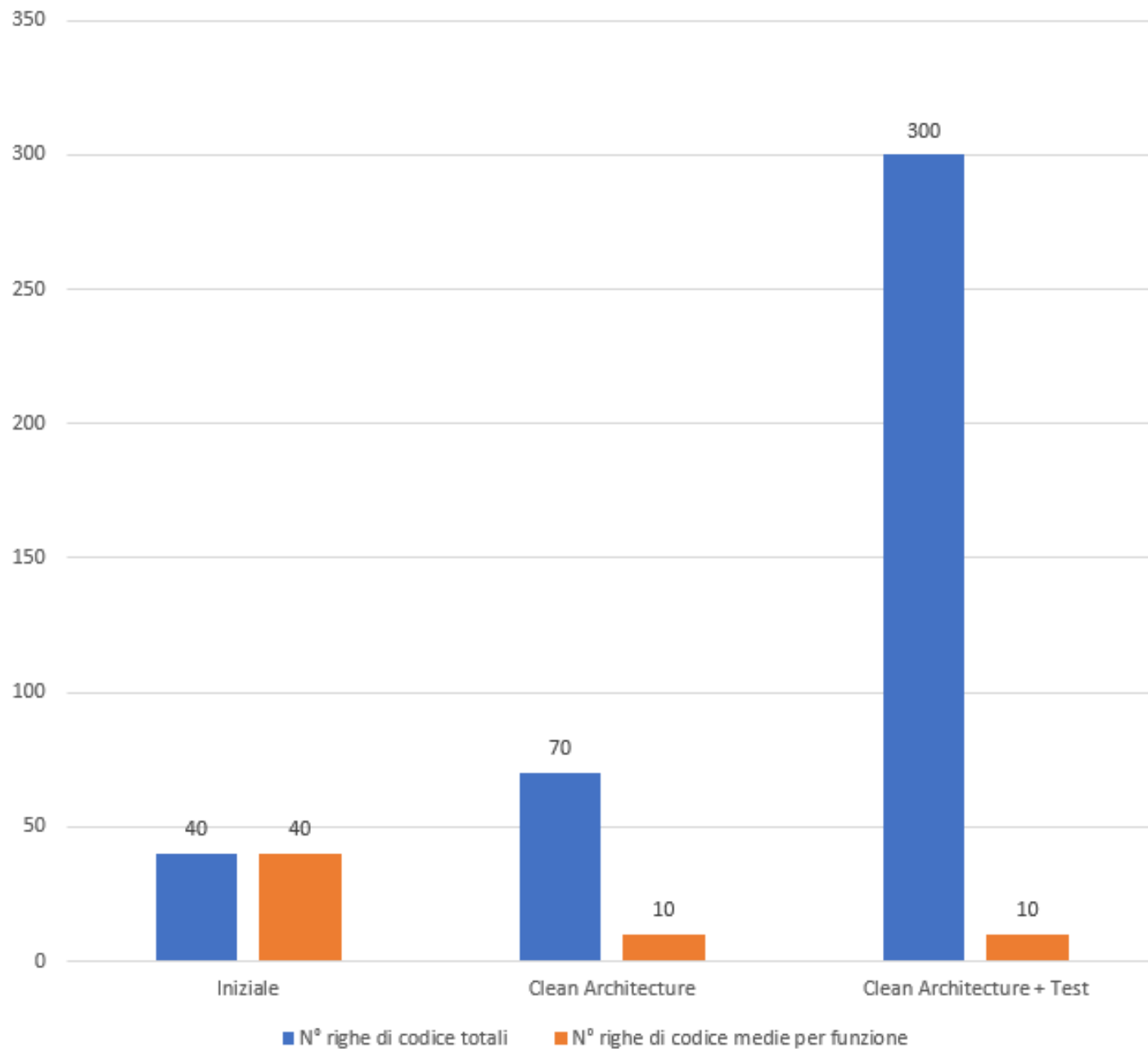
    // Act & Assert
    var exception = Assert.Throws<ArgumentException>(() => _userService.InsertUser(user));
    Assert.Equal("Surname is required", exception.Message);
}
```

INTEGRATION TEST

```
[Fact]
public void InsertUser_ValidUser_InsertsUserIntoDatabase()
{
    // Arrange
    var user = new User("John", "Doe", "john.doe@example.com");

    // Act
    var userId = _userService.InsertUser(user, false);

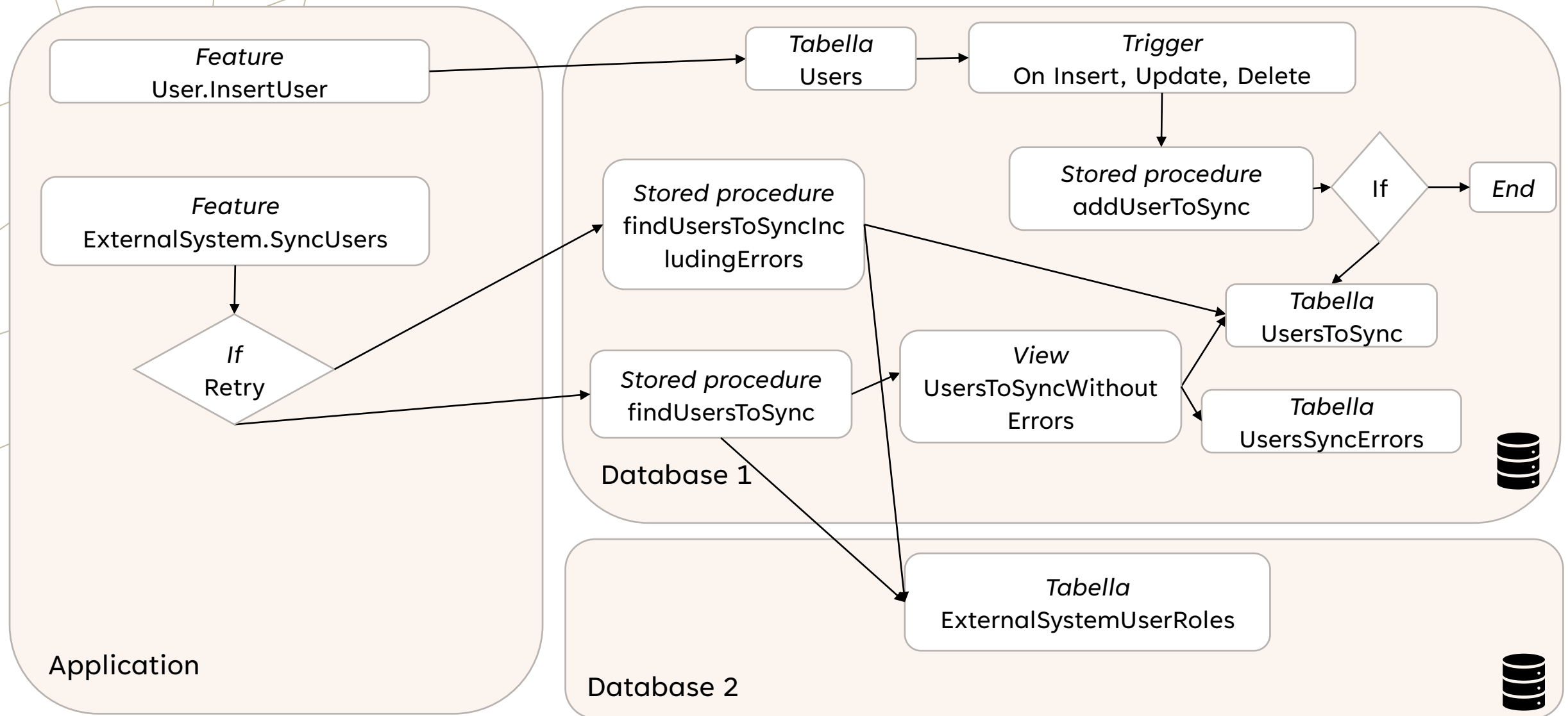
    // Assert
    var savedUser = _context.Users.Find(userId);
    Assert.NotNull(savedUser);
    Assert.Equal("John", savedUser.Name);
    Assert.Equal("Doe", savedUser.Surname);
    Assert.Equal("john.doe@example.com", savedUser.Email);
}
```



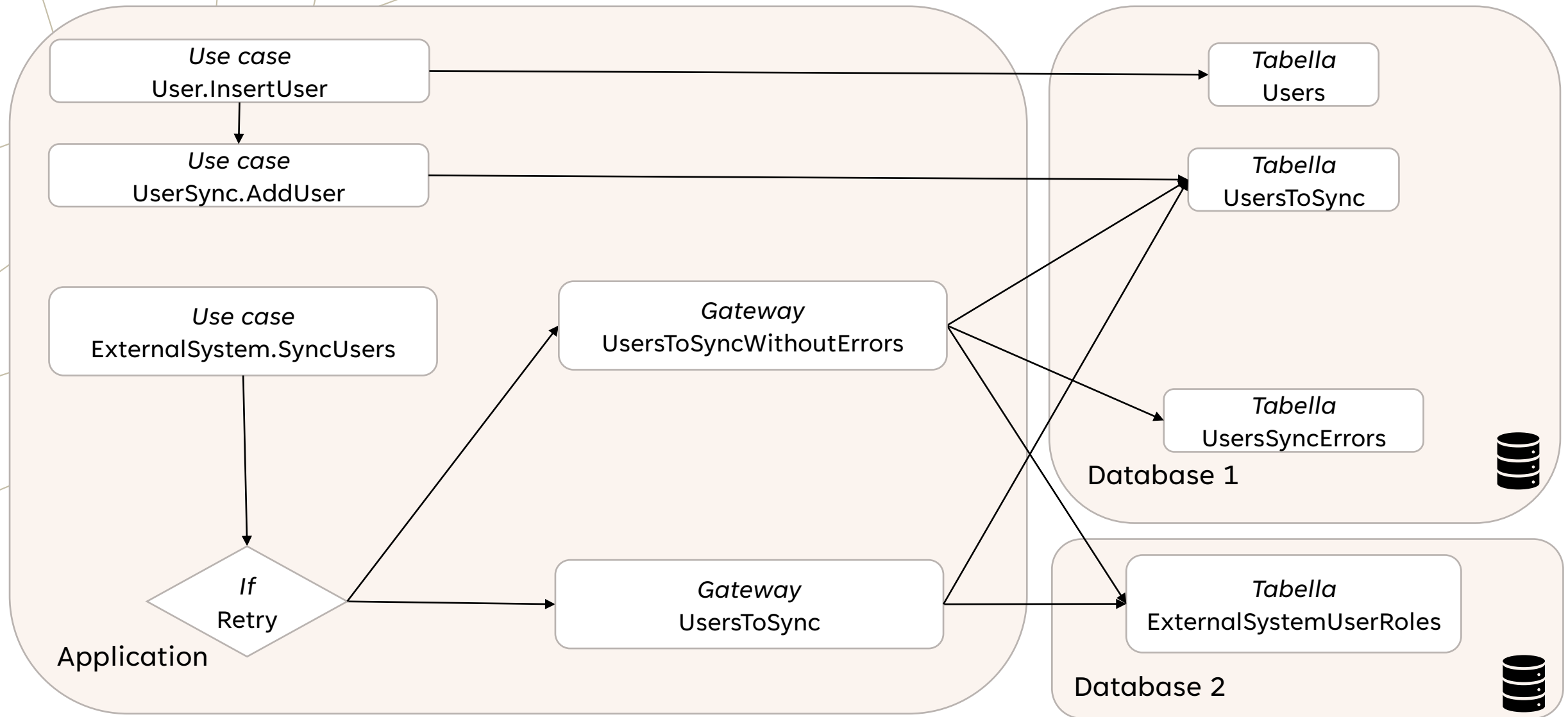
Più codice ma più
modulare, chiaro,
manutenibile e
riutilizzabile



ESEMPIO 2: SINCRONIZZAZIONE UTENTI SU APPLICAZIONE ESTERNA



ESEMPIO 2: SINCRONIZZAZIONE UTENTI SU APPLICAZIONE ESTERNA



Logiche di business nell'applicazione!



TAKEAWAY

SUDDIVIDIAMO LA
COMPLESSITA' IN
RESPONSABILITA'

IL DATABASE E' UN
DETTAGLIO

AFFIDIAMOCI A TEST
AUTOMATIZZATI





GRAZIE

Giuseppe De Luca

delucagiuseppe@outlook.it

<https://www.linkedin.com/in/giuseppe-dl/>