



From Chaos to Order

Using the Repository Pattern the Right Way



From Chaos to Order

Using the Repository Pattern the Right Way

(... According to Me)

What are Repositories?

Been around for a bit

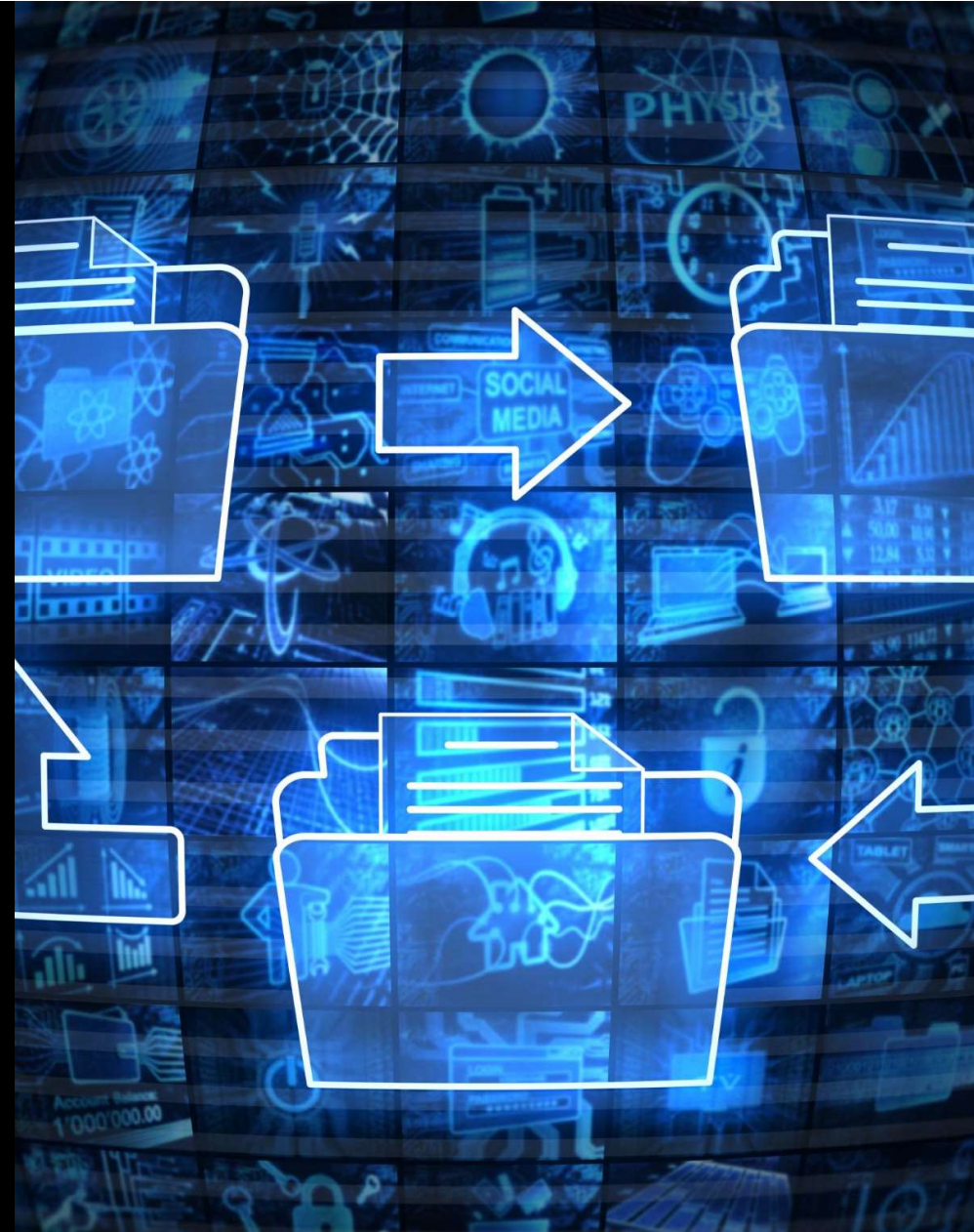
Three Layer Architecture

Repositories – DAL – Data Access Layer

Façades – BLL – Business Logic Layer

Views & ViewModels - UI Layer

Used for exactly that, accessing data



Hi, I'm Eric
This is my boss Zelda

Author of NoDoxx and dotnet Bacon

Live coder on Twitch

Progress Telerik Champion





Why NOT to Use Repositories

Yet another abstraction layer

Denormalized data stores

Performance concerns

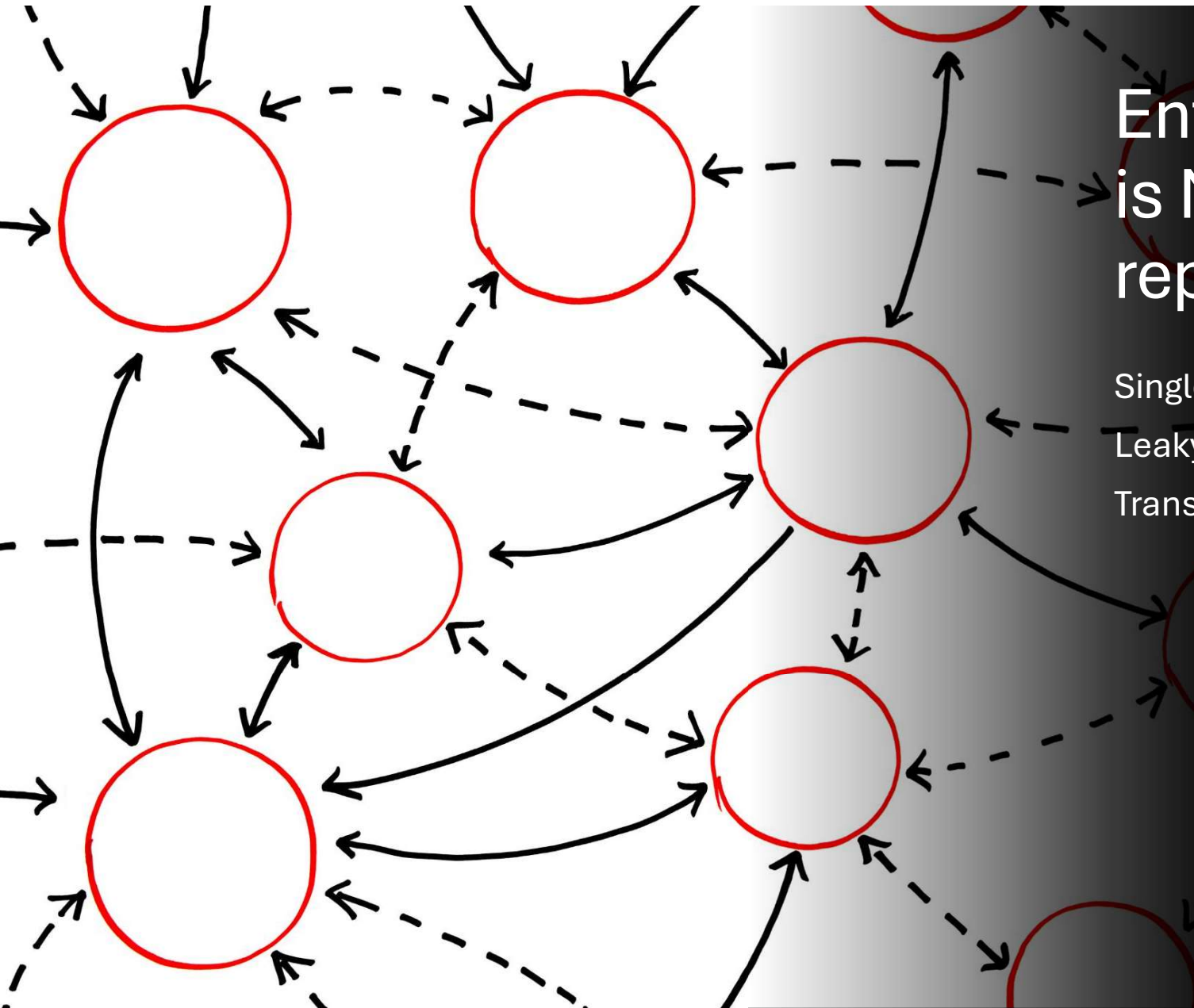
Complexity



Why Use Repositories

Entity Framework is NOT a repository

Single data source
Leaky inserts/Updates
Transactions and partial saves





Interface-itis

Basic CRUD

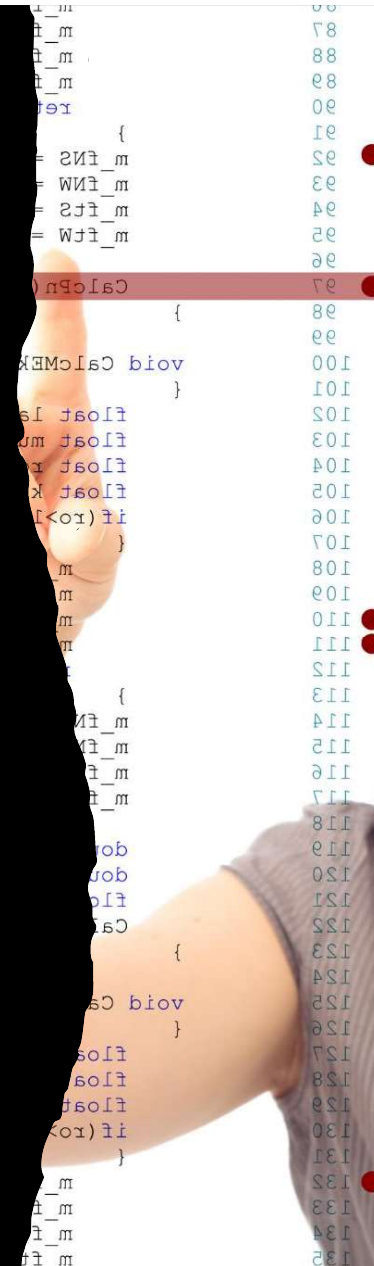
Generics

Abstract classes

Shared code for each datasource

Specific interfaces

Access data in specific ways





TIME FOR SOME
CODE

Unit of Work

Transactions

Data reference integrity

Data consistency

Retry policy

Order by risk





Different Unit of Work Patterns

.RepositoryTypeProperty

Get a new repository from the Unit of Work class by accessing a property

.GetRepository<T>()

Get a new repository by calling a function to activate or create one

.Attach(IRepositoryType instance)

Track an existing repository for handling transactions

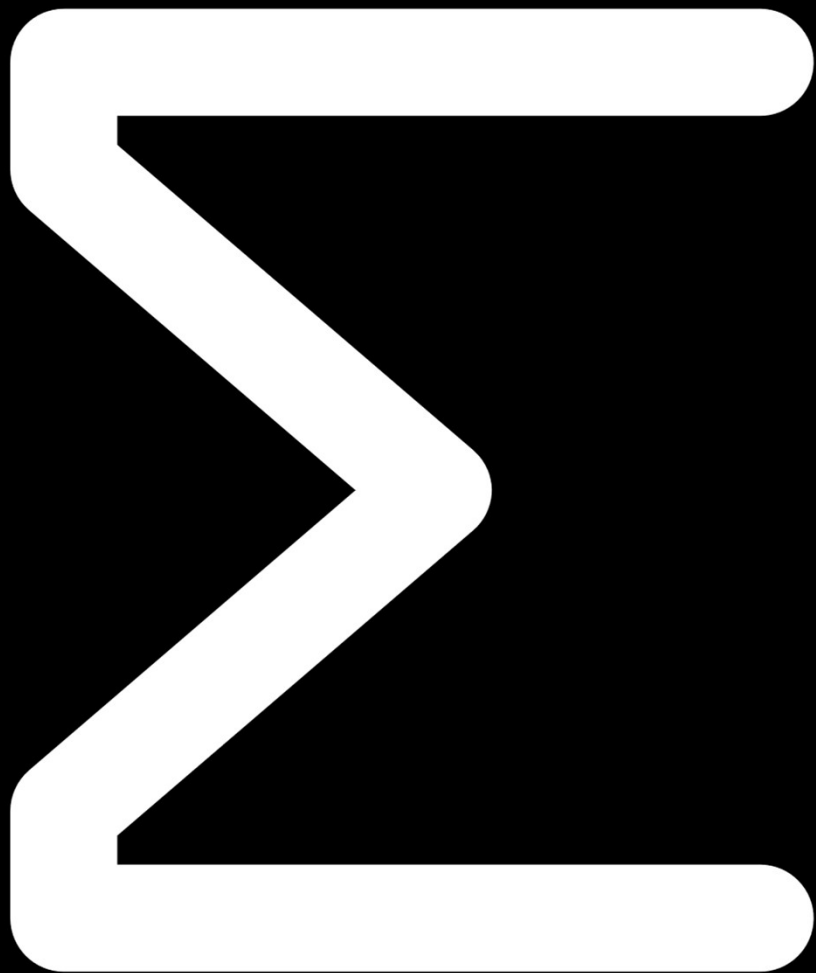
```
object to mirror  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select= 1  
mirror_ob.select= 1  
context.scene.objects[one.name].select  
("Selected" + str(notifier))  
mirror_ob.select = 0  
= bpy.context.selected_objects  
data.objects[one.name].select  
print("please select exactly one")
```

```
-- OPERATOR CLASSES --
```

```
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"
```

SOME MORE CODE



Summing it up

Denormalize your data in ways that make sense

Let your repositories grow with specific access and search functions

Interface-itis isn't always a bad thing

Base classes will save you time



Where to find me

Twitch every week at <https://twitch.tv/thindal>

Send me an email – eric@ceriq.se

LinkedIn, [eric-r-johansson](#)

Twitter aka X, [@ericjohansson](#)

Slides and more on GitHub, <https://github.com/thindaltv>

THANK YOU!

Please leave feedback and help me do it better next time!