



iORM2

Il punto della situazione



SPEAKER: MAURIZIO DEL MAGNO





MAURIZIO DEL MAGNO DEVELOPER



Lev@nte software



**i-ORM
DJSON**

github.com/mauriziodm/iORM

github.com/mauriziodm/DJSON



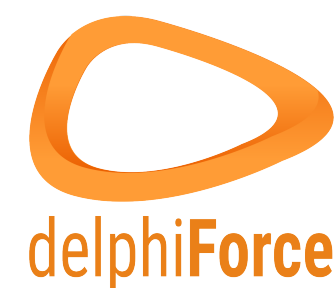
mauriziodm@levantesw.it

mauriziodelmagno@gmail.com



facebook.com/maurizio.delmagno

[iORM + DJSON \(group\)](#)



Membro fondatore

eInvoice4D

<https://github.com/delphiforce/eInvoice4D>



iORM2

Il punto della situazione



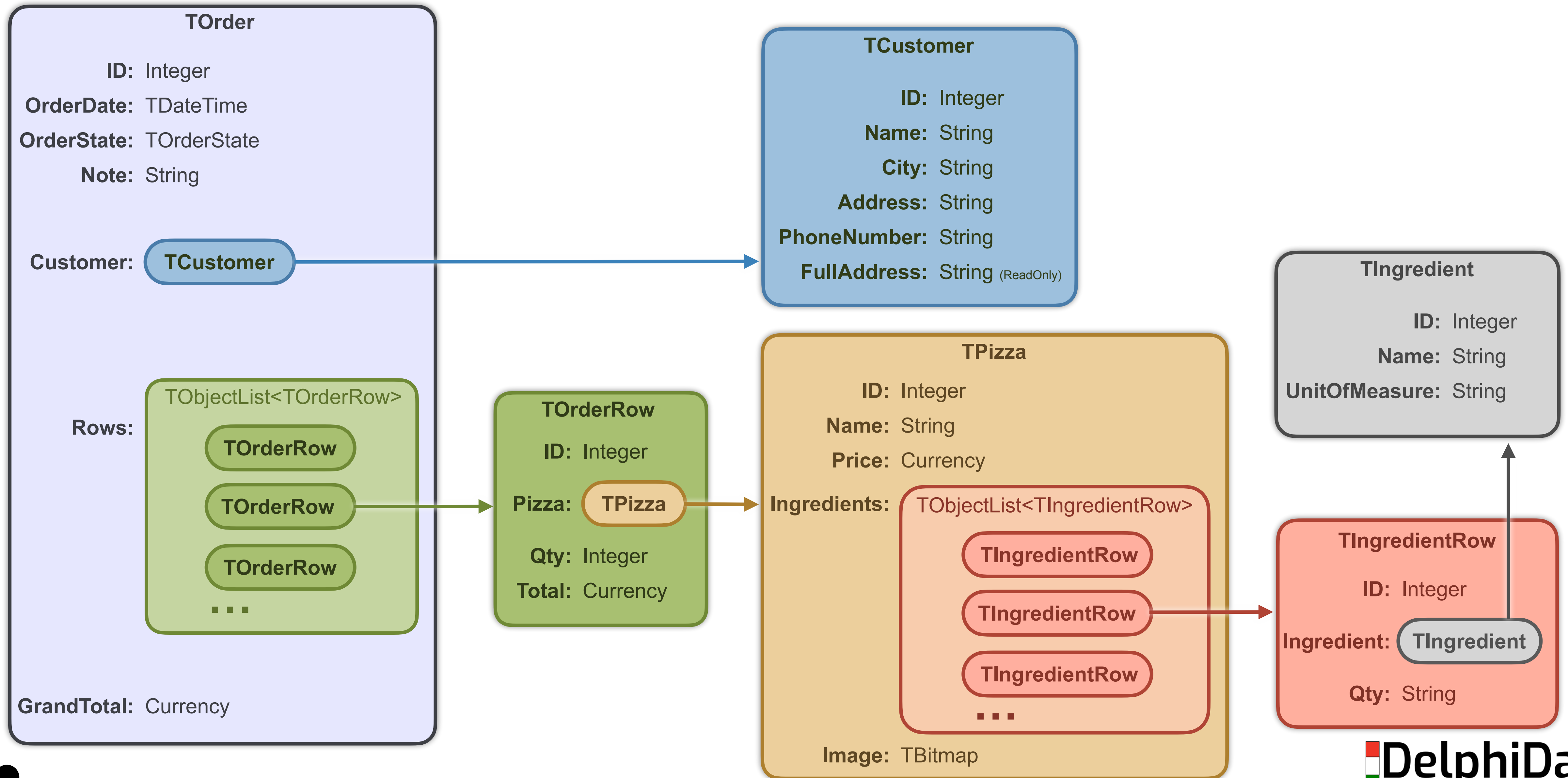
SPEAKER: MAURIZIO DEL MAGNO

Let'sCode



Model

Model





Data**binding** + Dependency**injection**

Databinding + Dependencyinjection



Databinding + Dependencyinjection Register

```
[diSimpleViewFor(TPizza)]  
TPizzaForm = class(TForm);
```

The screenshot shows a Delphi application window titled "Pizz'Amore". Inside the window is a form titled "Pizza". The form has a "Back" button in the top left corner. Below the title bar, there are four labeled input fields: "ID", "Name", "Price", and "Image". The "Image" field is a large rectangular area with a small green button containing three dots to its right. A red rectangular box with the text "BSPizza Master" is overlaid on the left side of the form. Below the form, there is a green rectangular button labeled "Actions". At the bottom of the window, there are two buttons: "Revert" on the left and "Persist" on the right.

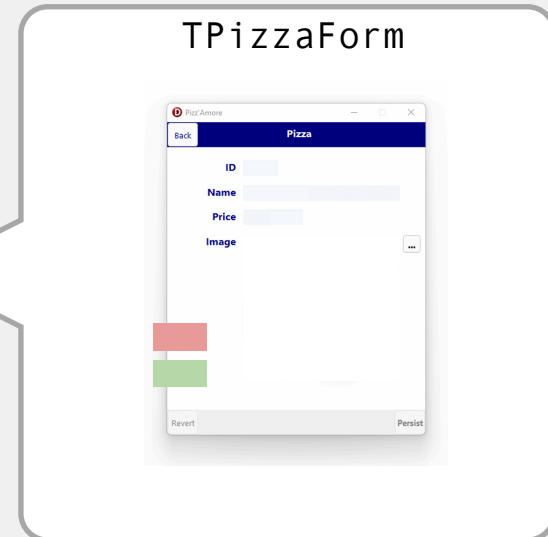


*SimpleView for
TPizza*

Databinding + Dependencyinjection Register

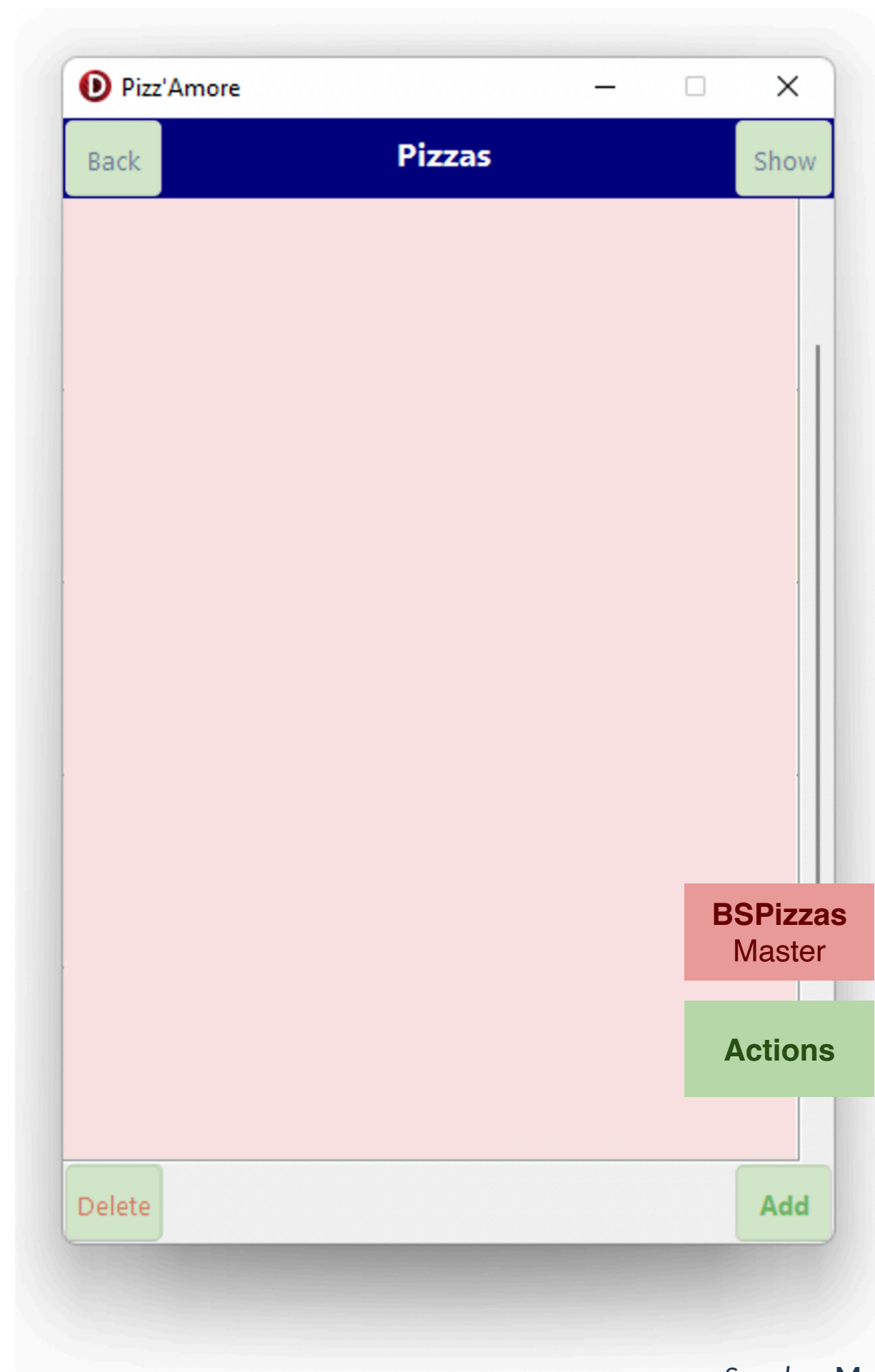


SimpleView for
TPizza



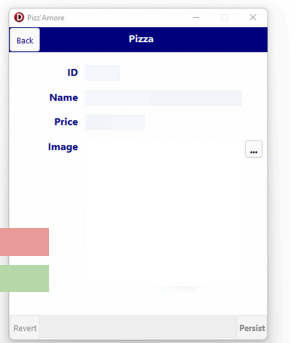
Databinding + Dependencyinjection Register

```
[diSimpleViewFor(TPizza, 'LIST')]  
TPizzaListForm = class(TForm);
```



*SimpleView for
TPizza*

TPizzaForm



*SimpleView for
TPizza
alias LIST*

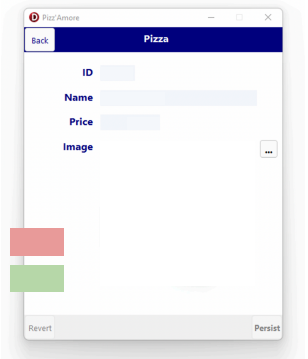


Databinding + Dependencyinjection



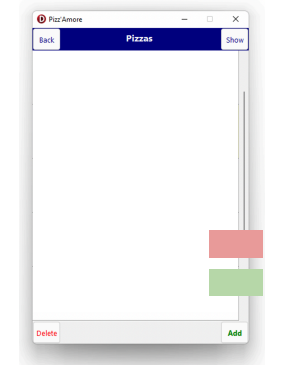
*SimpleView for
TPizza*

TPizzaForm

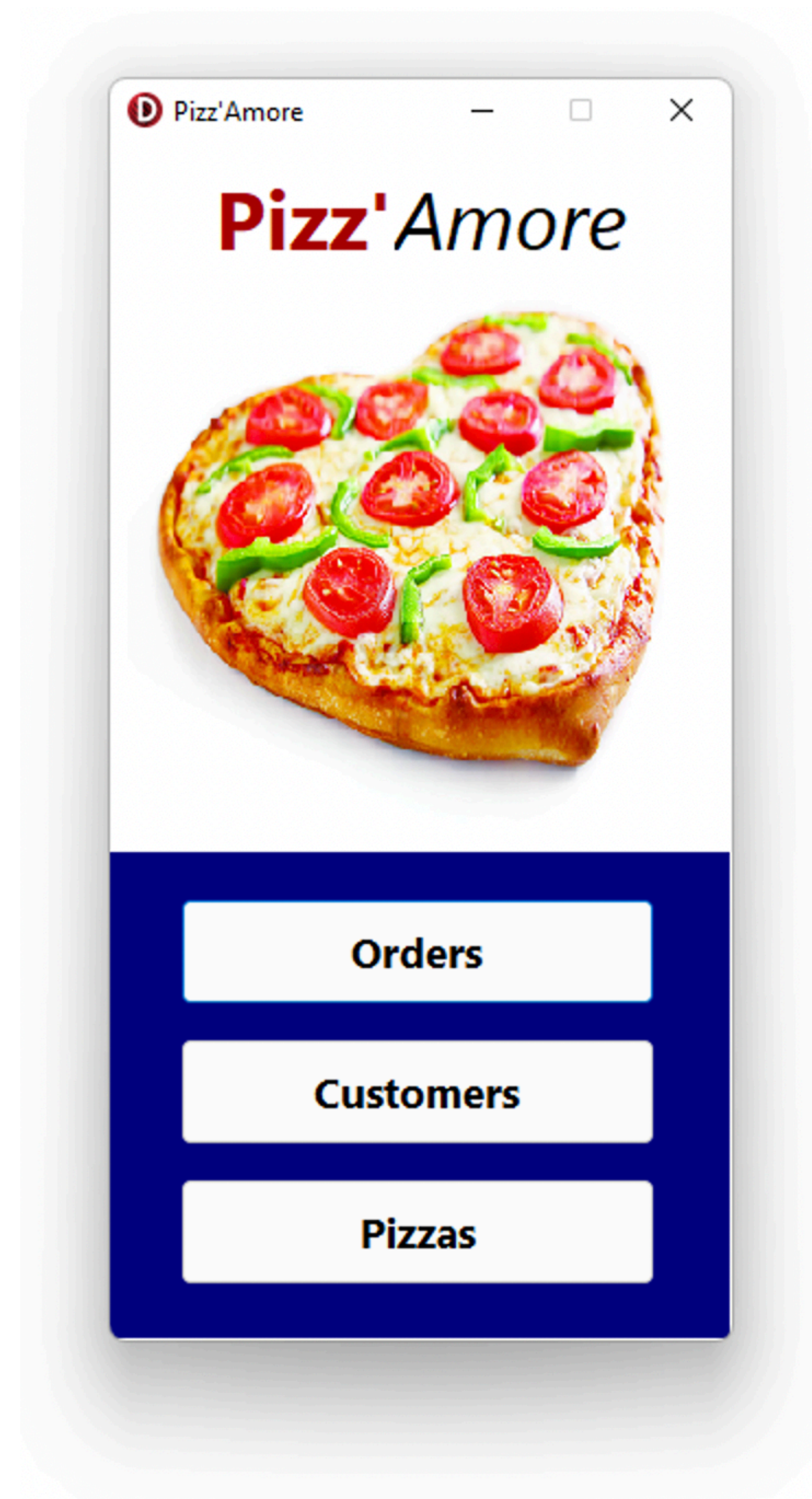
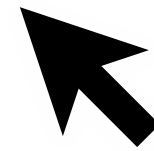


*SimpleView for
TPizza
alias LIST*

TPizzaListForm

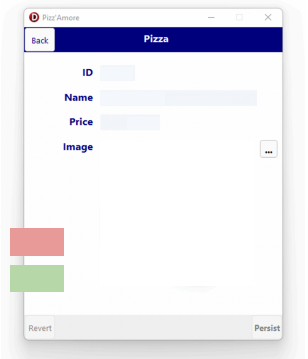


Databinding + Dependency Injection



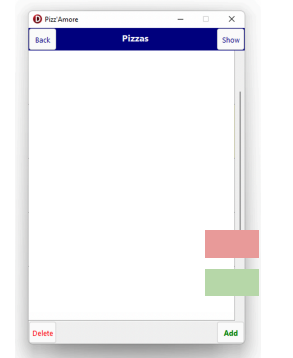
*SimpleView for
TPizza*

TPizzaForm

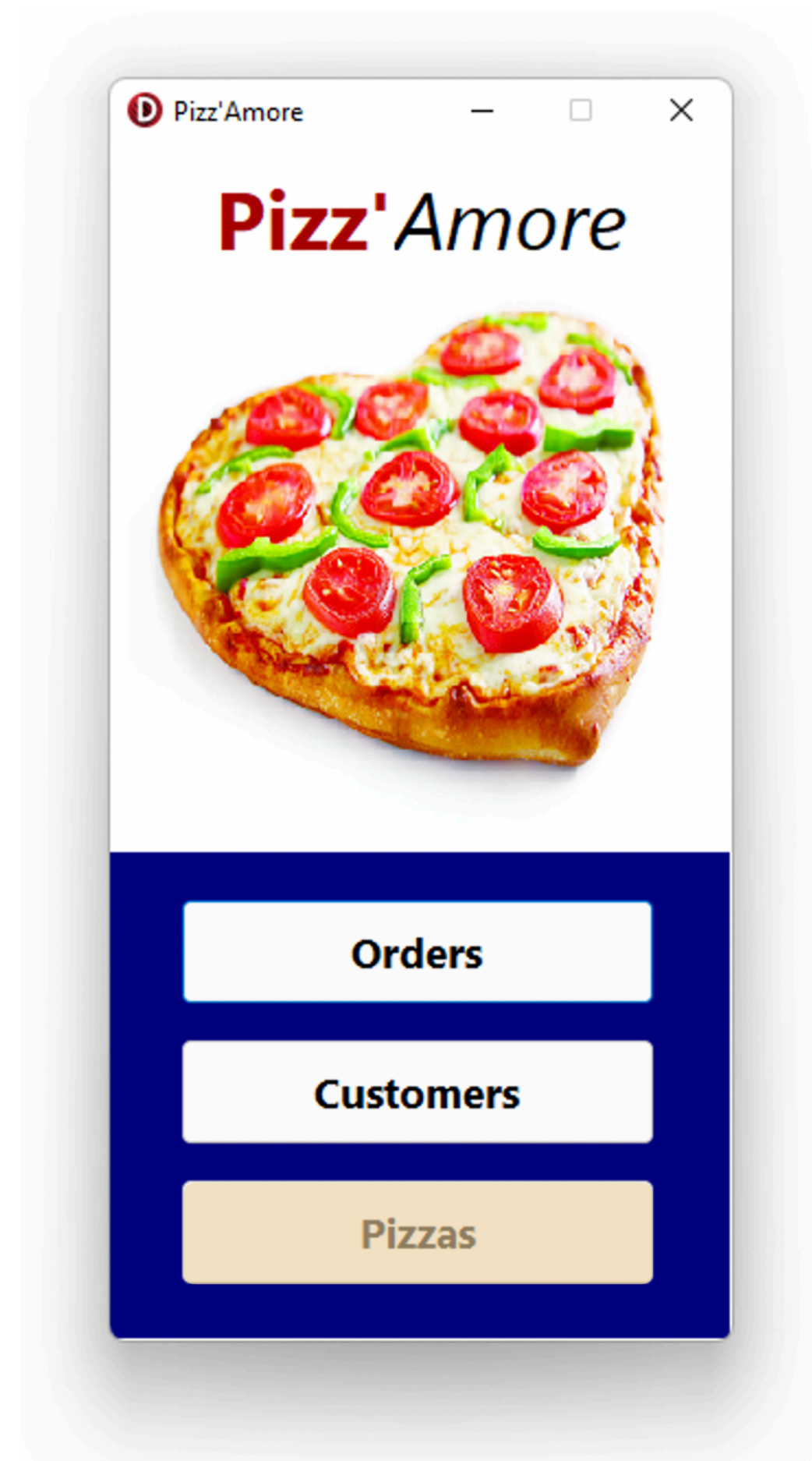


*SimpleView for
TPizza
alias LIST*

TPizzaListForm

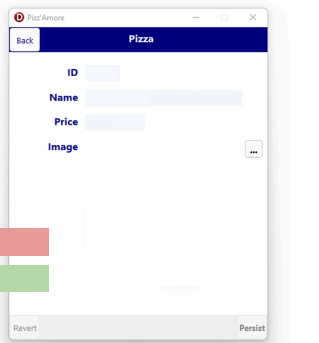


Databinding + Dependency Injection



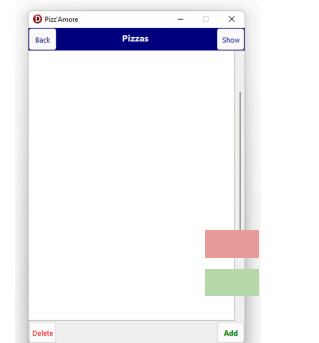
*SimpleView for
TPizza*

TPizzaForm



*SimpleView for
TPizza
alias LIST*

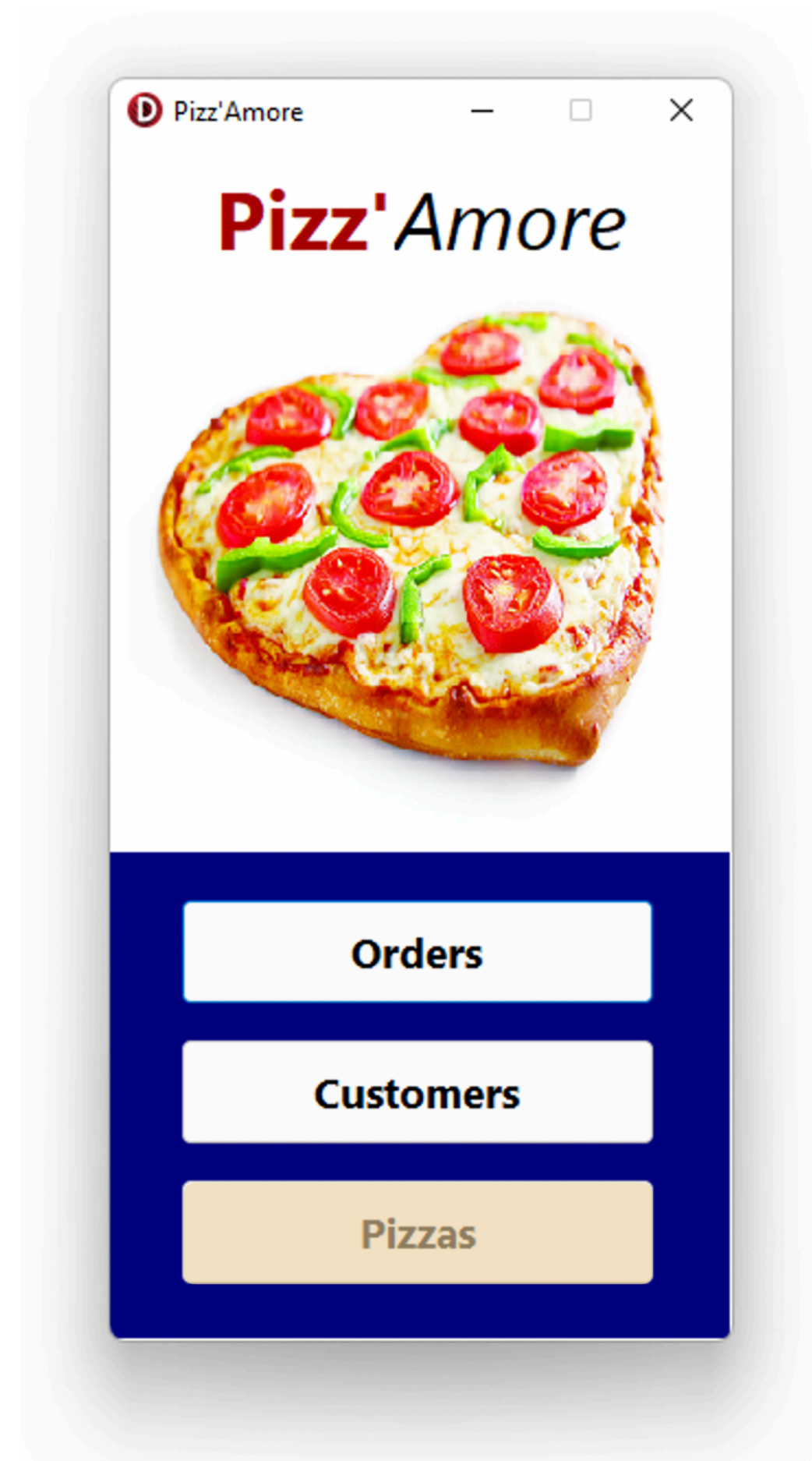
TPizzaListForm



```
io.Show(TPizza, 'LIST');
```


Databinding + Dependency Injection

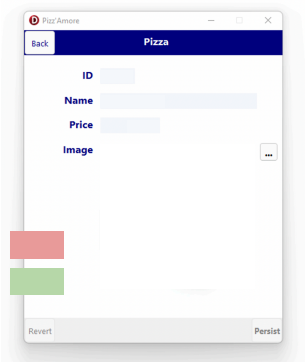
Injection



```
io.Show(TPizza, 'LIST');
```

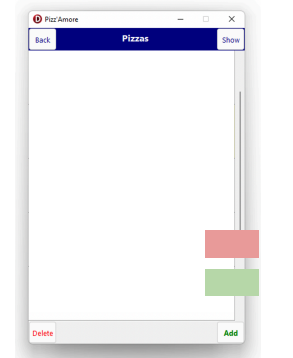
*SimpleView for
TPizza*

TPizzaForm

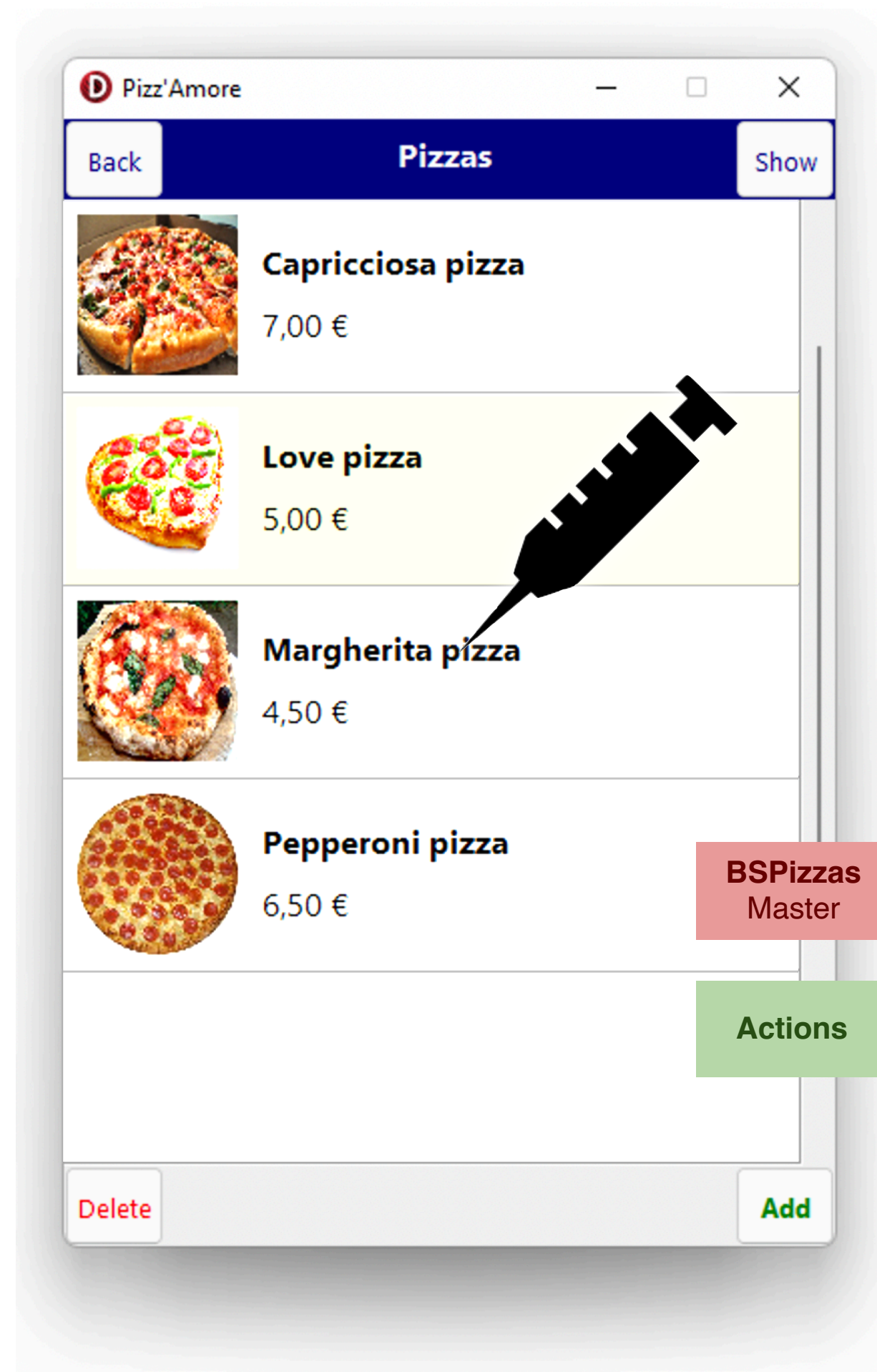


*SimpleView for
TPizza
alias LIST*

TPizzaListForm



Databinding + Dependency Injection



property TypeName := TPizza
property LoadType := ItAuto

ORM



oppure

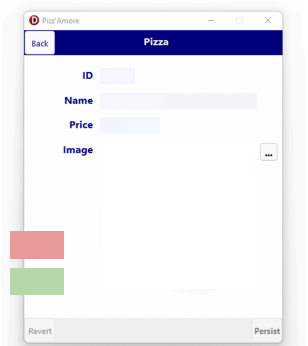
property TypeName := TPizza
property LoadType := ItManual

BSPizzas.SetDataObject(Obj);
Obj := BSPizzas.DataObject;

???

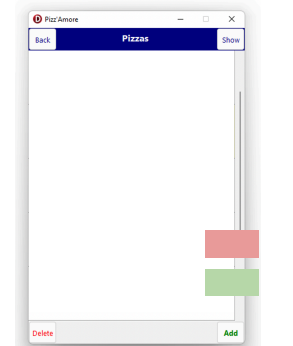
SimpleView for
TPizza

TPizzaForm

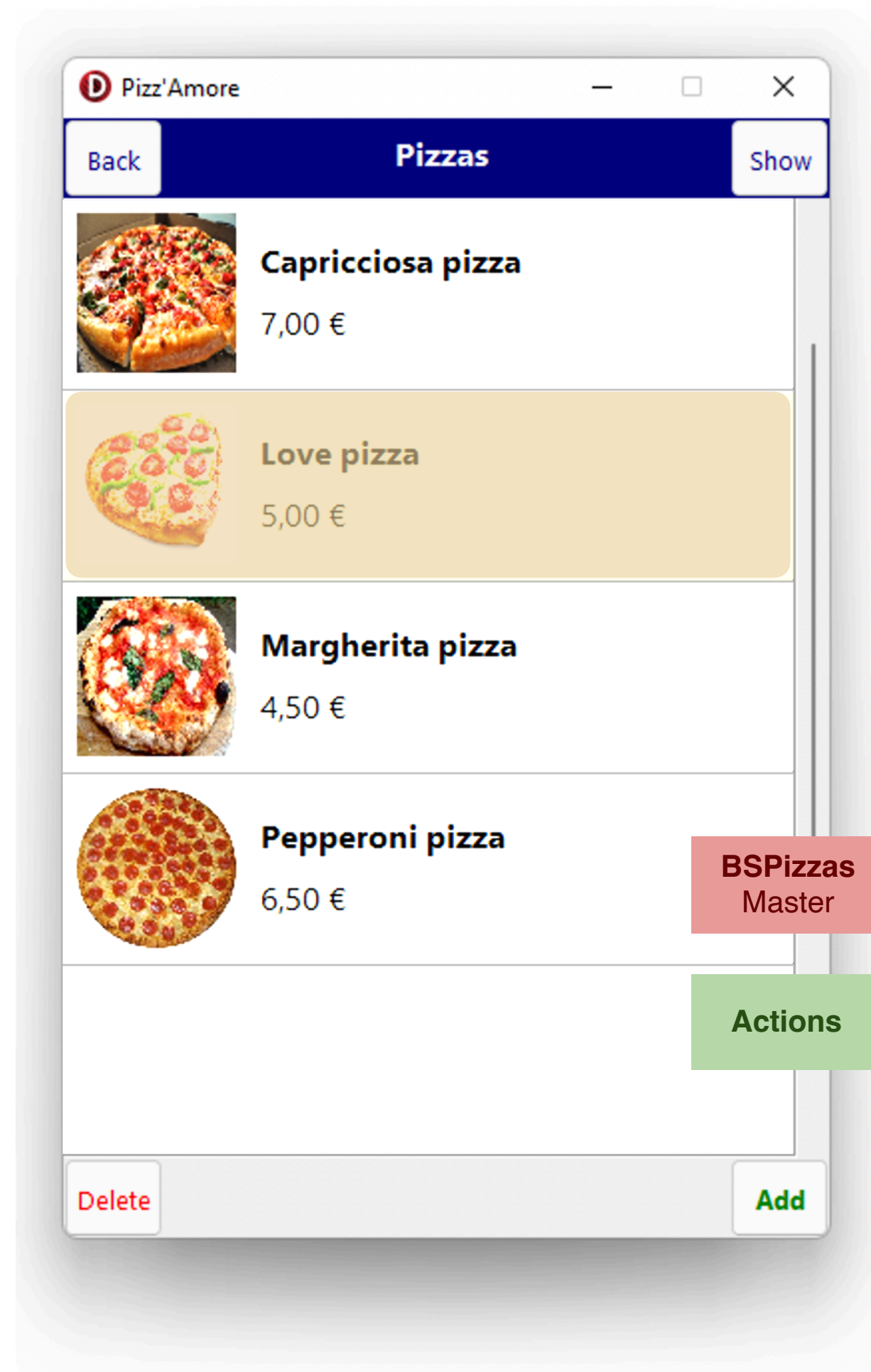
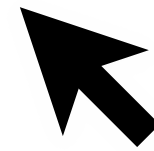


SimpleView for
TPizza
alias LIST

TPizzaListForm

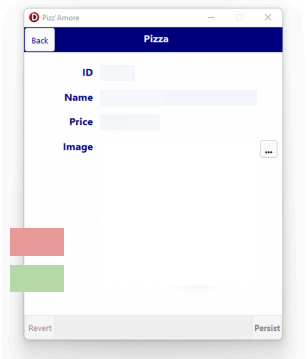


Databinding + Dependency Injection



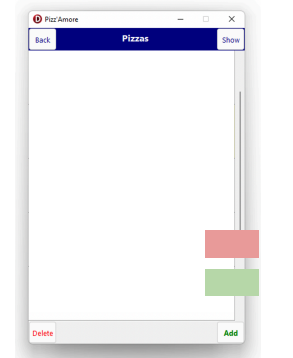
SimpleView for
TPizza

TPizzaForm



SimpleView for
TPizza
alias LIST

TPizzaListForm

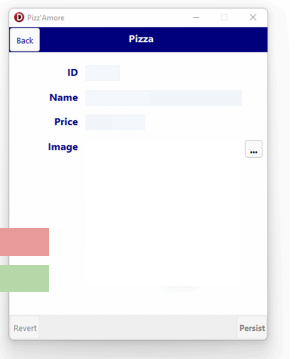


Databinding + Dependency Injection



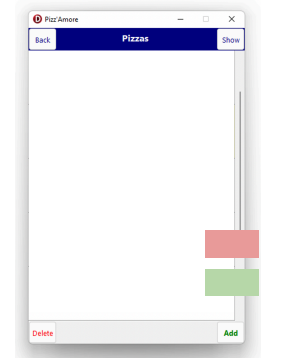
SimpleView for
TPizza

TPizzaForm



SimpleView for
TPizza
alias LIST

TPizzaListForm

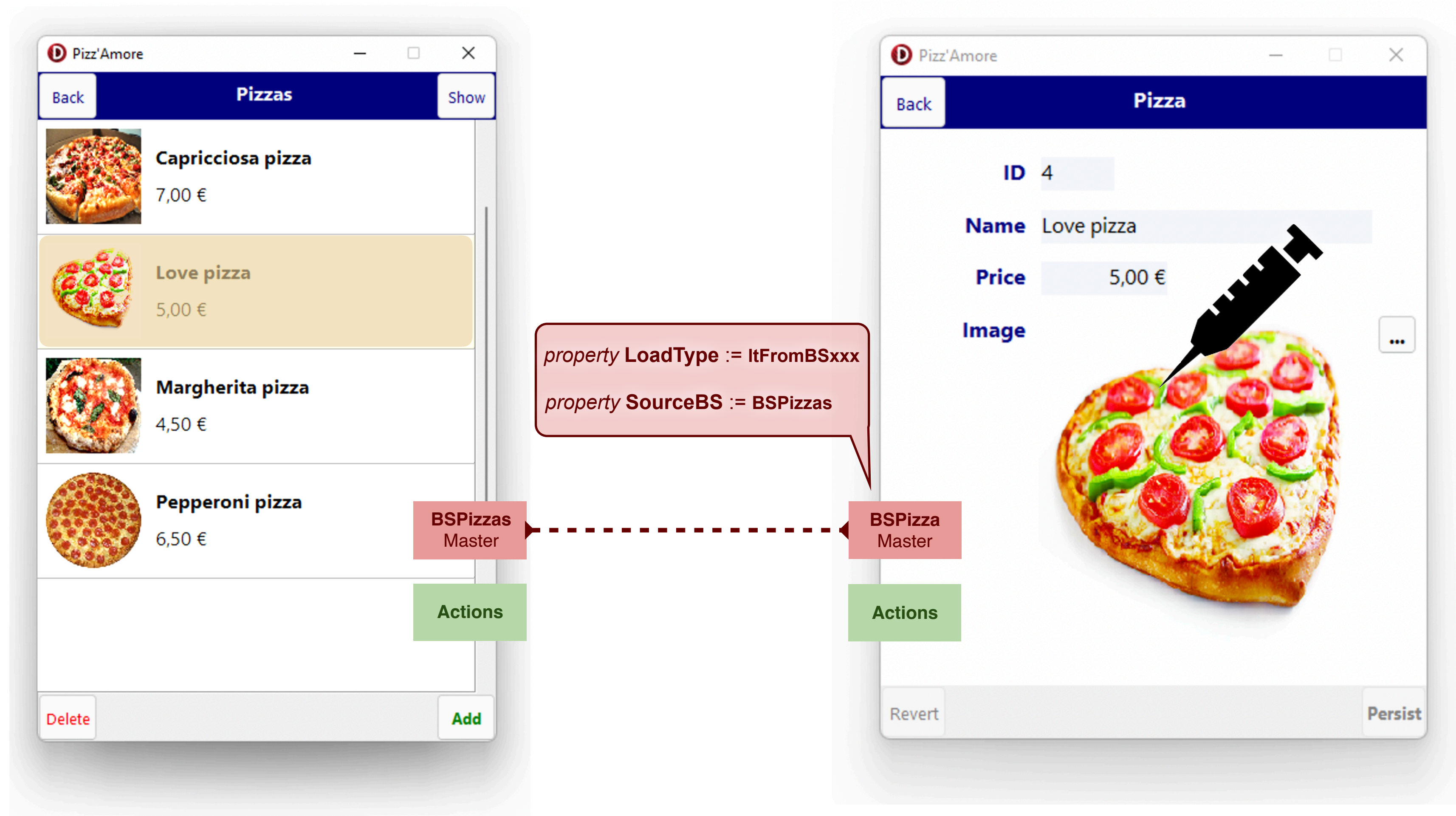


BSPizzas.ShowCurrent;

NB: BSPizzas.Current = **TPizza**

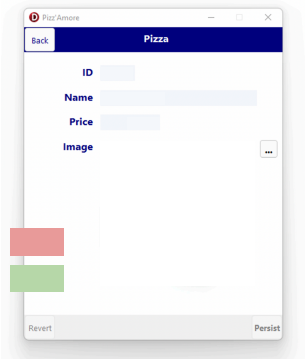
NB: Nessun **alias** specificato

Databinding + Dependency Injection



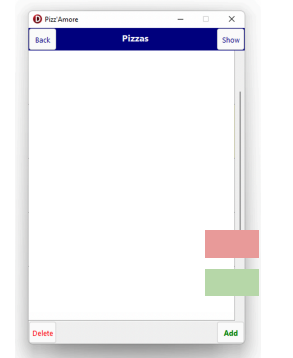
*SimpleView for
TPizza*

TPizzaForm



*SimpleView for
TPizza
alias LIST*

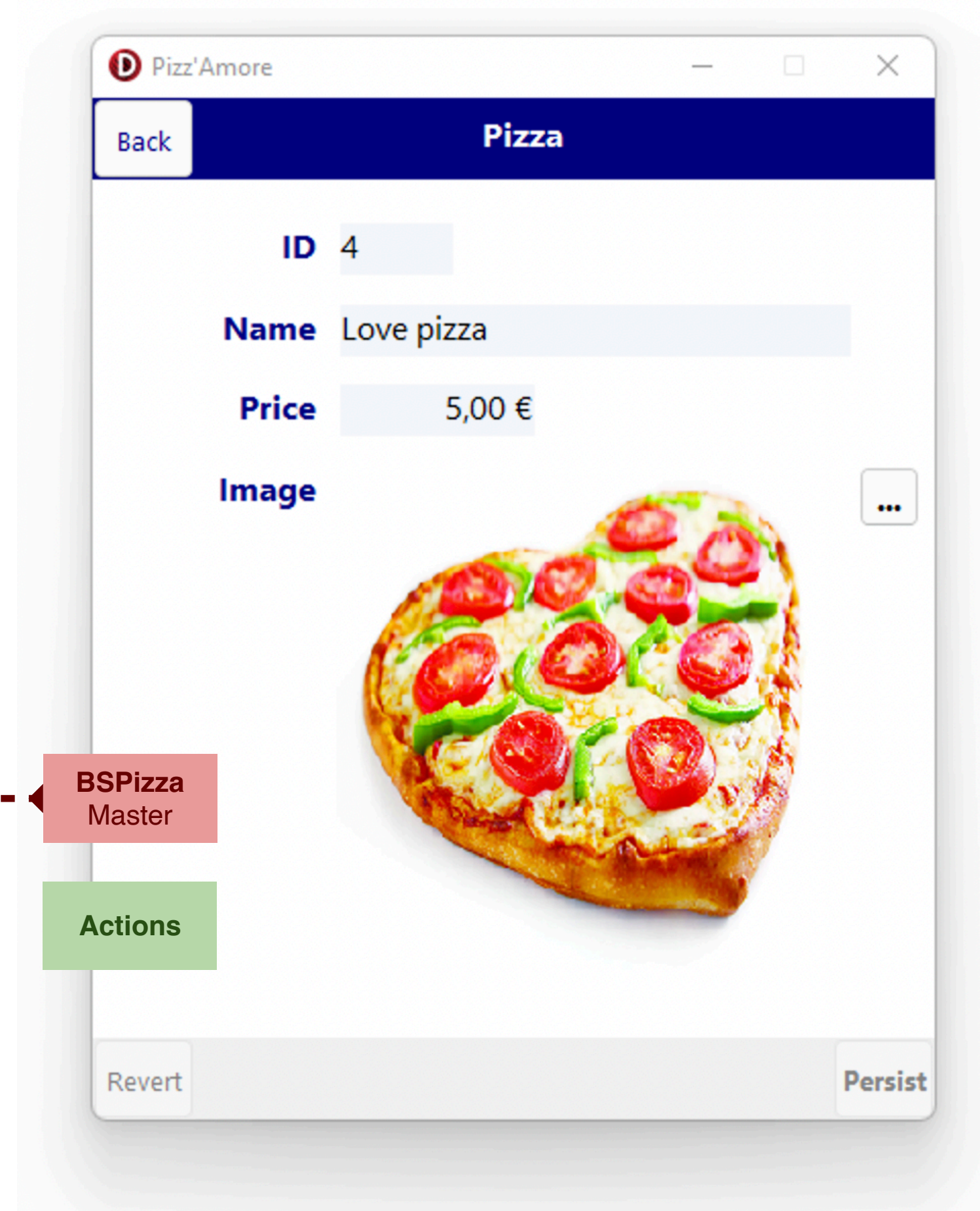
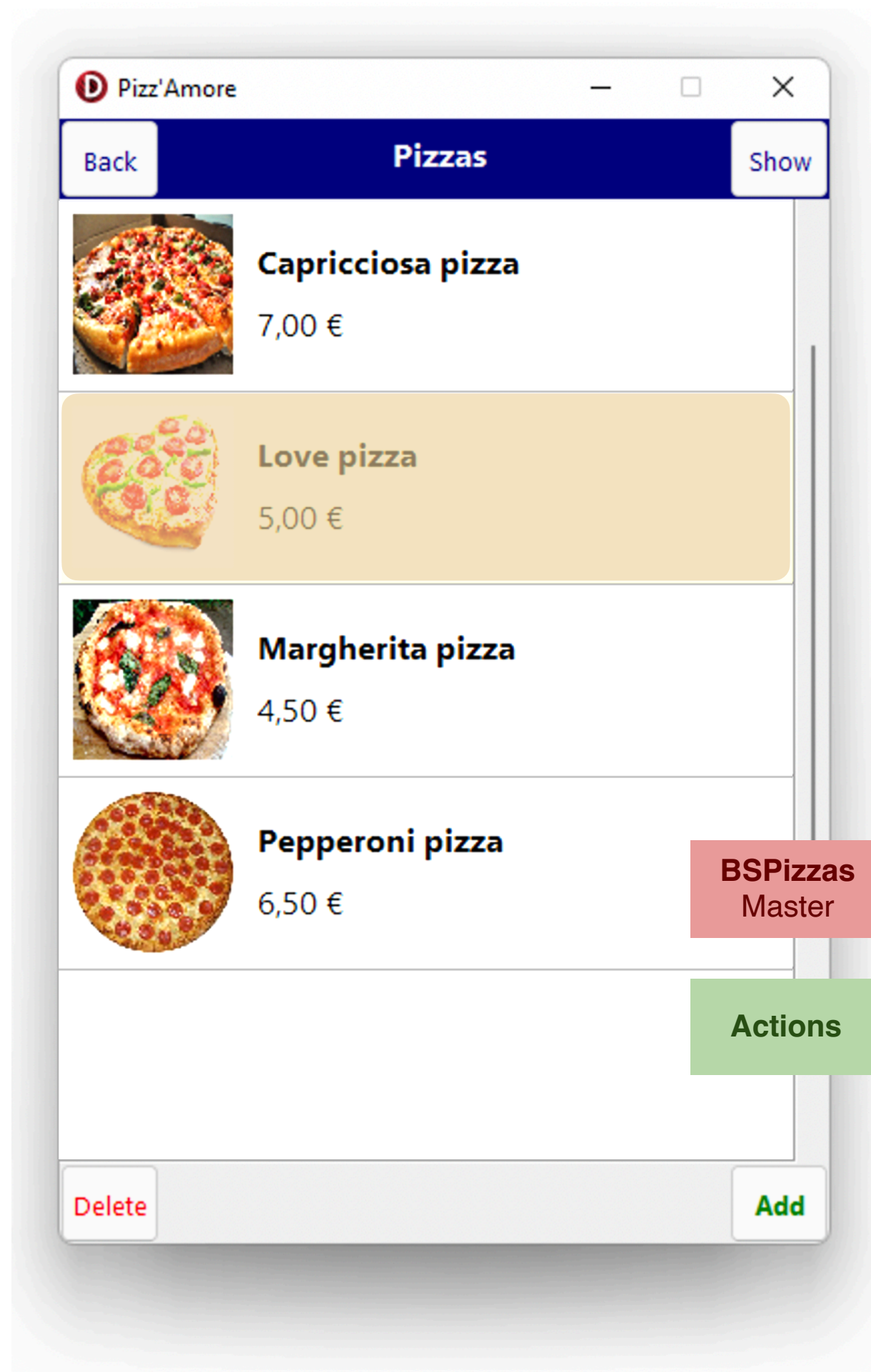
TPizzaListForm



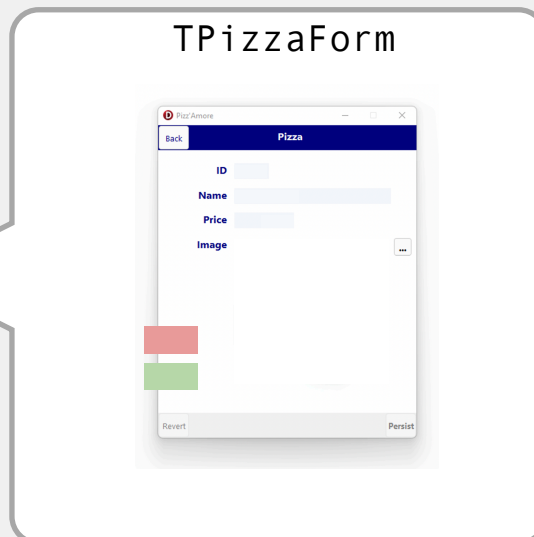
* TioLoadType = (ltManual, ltFromBSAsIs, ltFromBSReload, ltFromBSReloadNewInstance, ltAuto, ltCreate);

Databinding + Dependency Injection

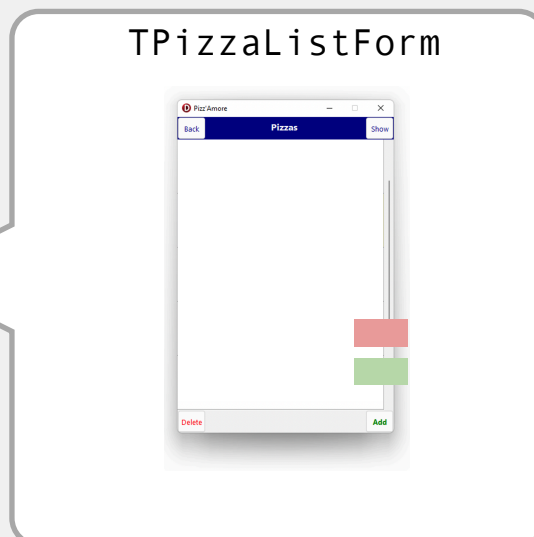
Injection



SimpleView for
TPizza



SimpleView for
TPizza
alias LIST



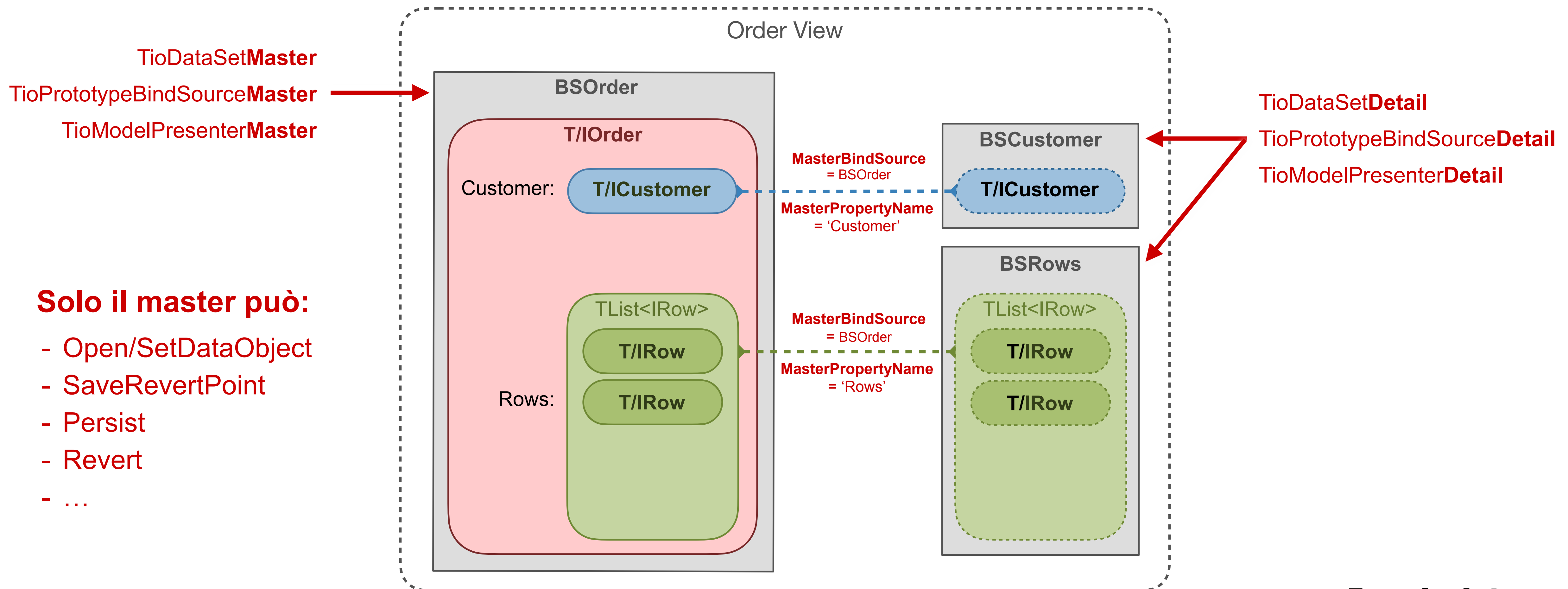
Bindsources

DataSet, PrototypeBindSource, ModelPresenter (Master-Detail)

Bindsources

Master - Detail

DataSet, PrototypeBindSource, ModelPresenter (Master-Detail)



Solo il master può:

- Open/SetDataObject
- SaveRevertPoint
- Persist
- Revert
- ...

Let'sCode



deepwherecriteria
keepsimple...again

deepwherecriteria

keepsimple...again

var

```
LCustomer: TCustomer;  
LOrders: TObjectList<TOrder>;  
LWhere: IioWhere;
```

begin

```
// Load customer with ID=1
```

```
LCustomer := io.LoadObject<TCustomer>(1);
```

```
// Compose where object
```

```
LWhere := io.Where('OrderState', coGreater, osPrepared) ← enumerated
```

```
  ._And('Customer', coEquals, LCustomer) ← instance
```

```
  ._And('Rows.Pizza.Ingredients.Ingredient.Name', coLike, 'Ham'); ← deep where
```

```
// Load orders
```

```
LOrders := io.LoadList<TObjectList<TOrder>>(LWhere);
```

end;

Let'sCode



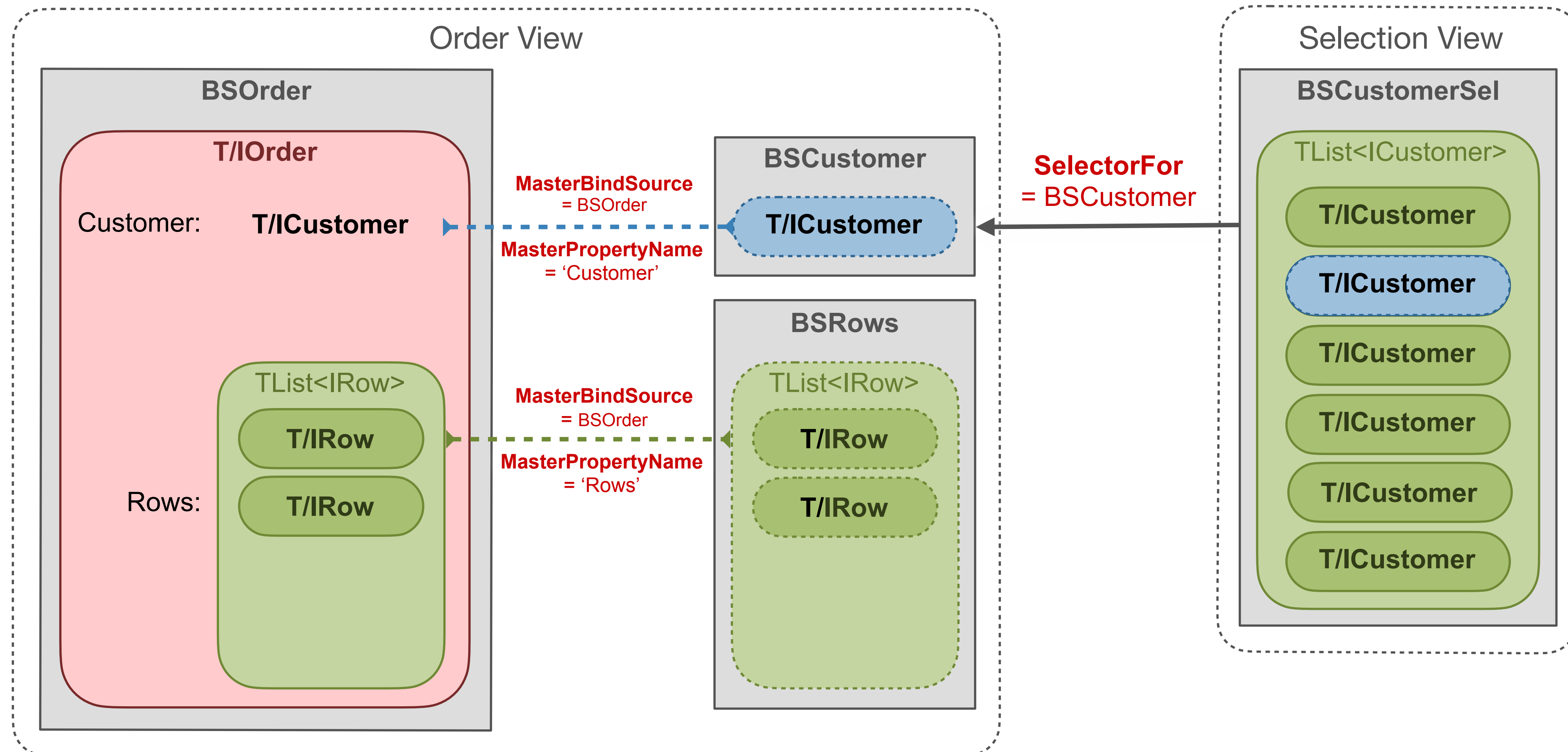
Selectors

DataSet, PrototypeBindSource, ModelPresenter

Selectors

DataSet, PrototypeBindSource, ModelPresenter

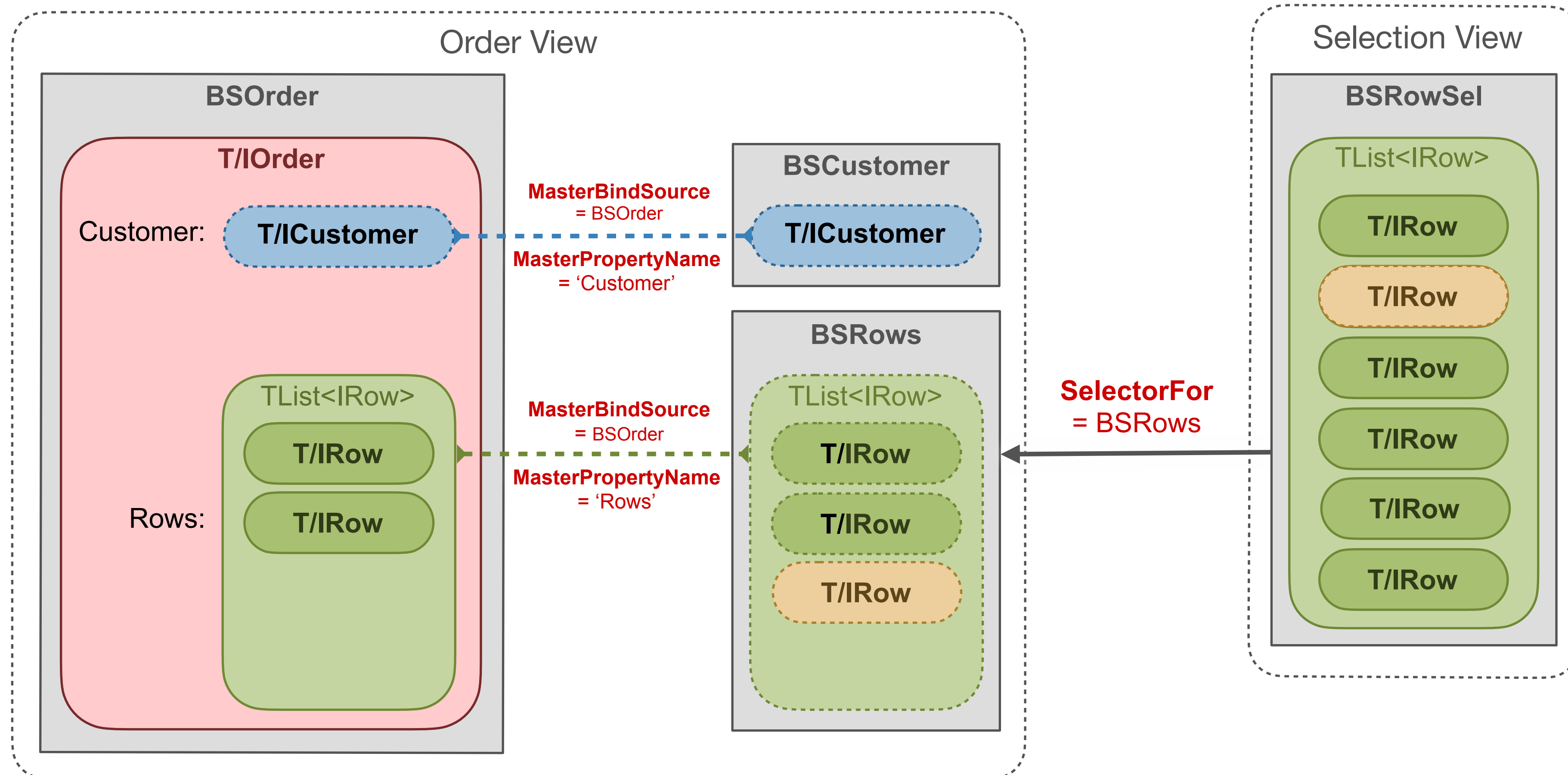
```
io.ShowAsSelector<ICustomer>(BSCustomer, nil);
```



Selectors

DataSet, PrototypeBindSource, ModelPresenter

```
io.ShowAsSelector<IRow>(BSRows, nil);
```



Let'sCode



Httpconnection

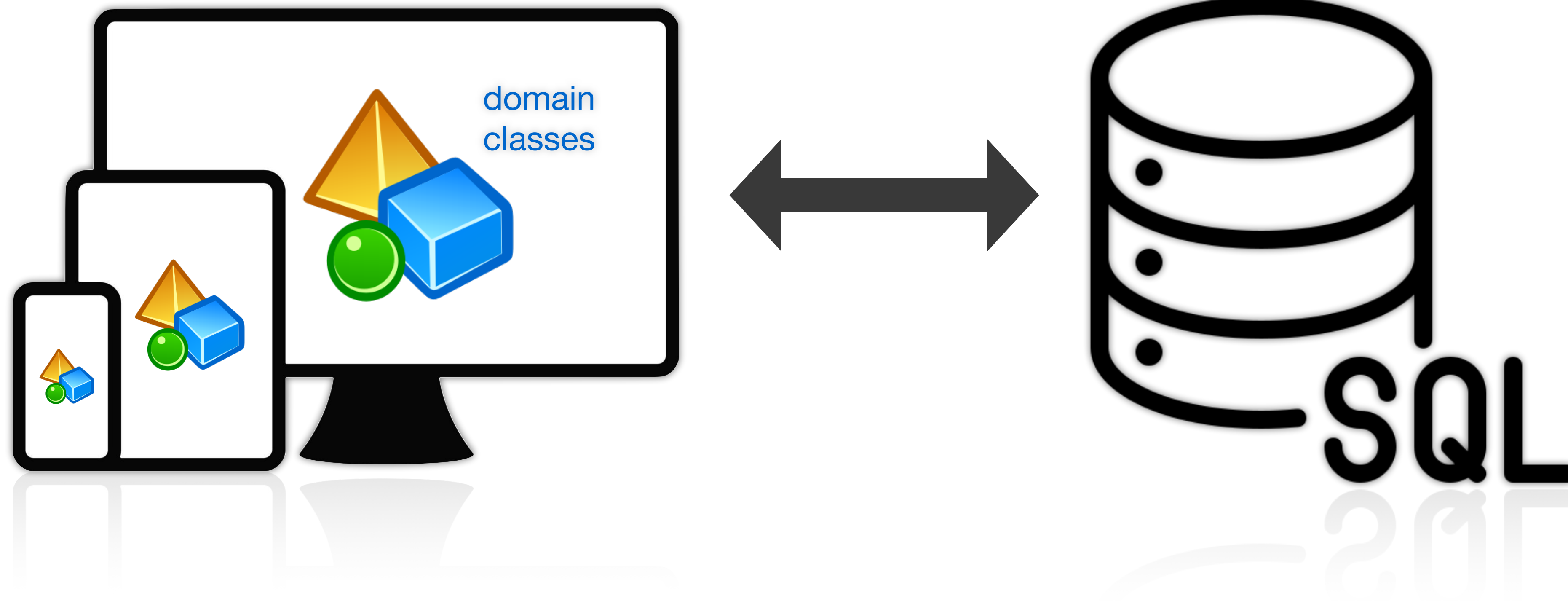
beconnected

Httpconnection

beconnected

TioFirebirdConnectionDef
TioSQLServerConnectionDef
TioSQLiteConnectionDef

Firebird
MSSQLServer
SQLite



Httpconnection



Httpconnection

beconnected

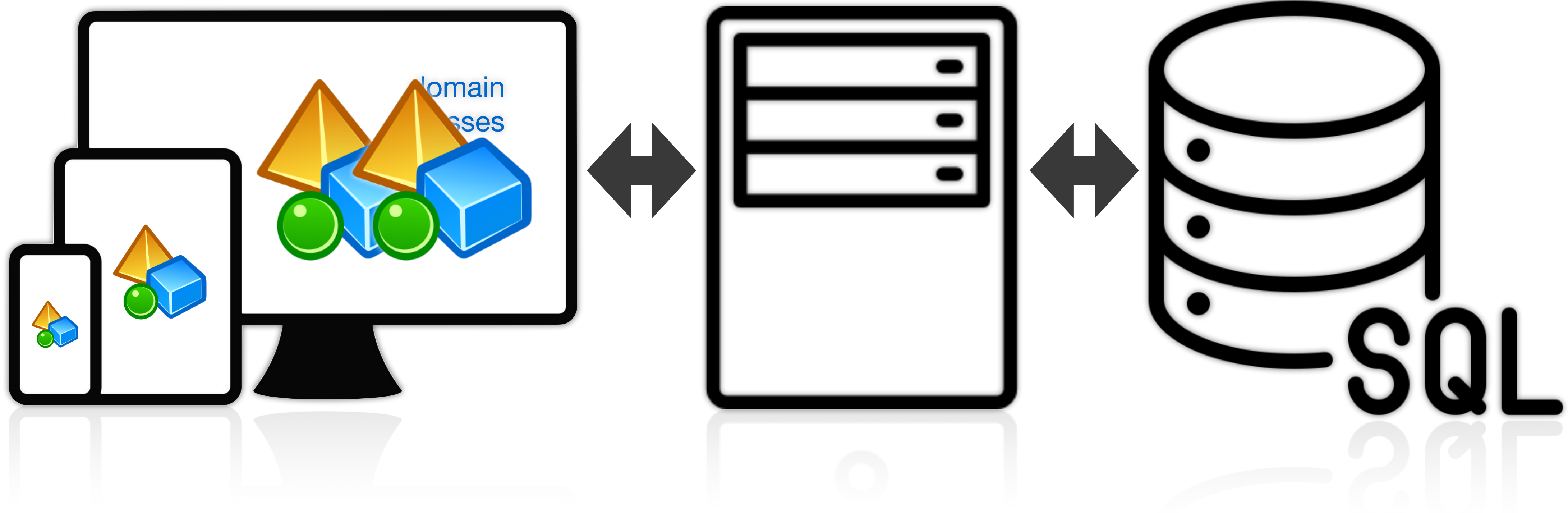
TioHttpConnectionDef

TioDMVCController

TioWirlApp

TioWebBrokerProducer

Firebird
MSSQLServer
SQLite



Let'sCode





MAURIZIO DEL MAGNO DEVELOPER



Lev@nte software



**i-ORM
DJSON**

github.com/mauriziodm/iORM

github.com/mauriziodm/DJSON



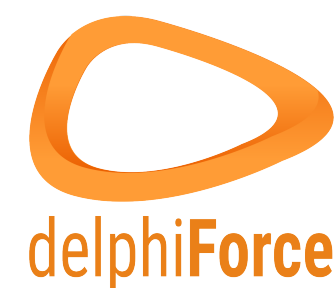
mauriziodm@levantesw.it

mauriziodelmagno@gmail.com



facebook.com/maurizio.delmagno

[iORM + DJSON \(group\)](#)



Membro fondatore

eInvoice4D

<https://github.com/delphiforce/eInvoice4D>



iORM2

Il punto della situazione

