Haute École Léonard de Vinci
SCIENCES ET TECHNIQUES

BINV314A
.NET Outils et Concepts d'Application d'Entreprise

Semaine 9

WPF - MVVM - LINQ

Sommaire : WPF - MVVM - LINQ

- Considérations sur le design
- Considérations sur la séparation en couches du MVVM
- Considérations sur le modèle
- Datacontext hérité

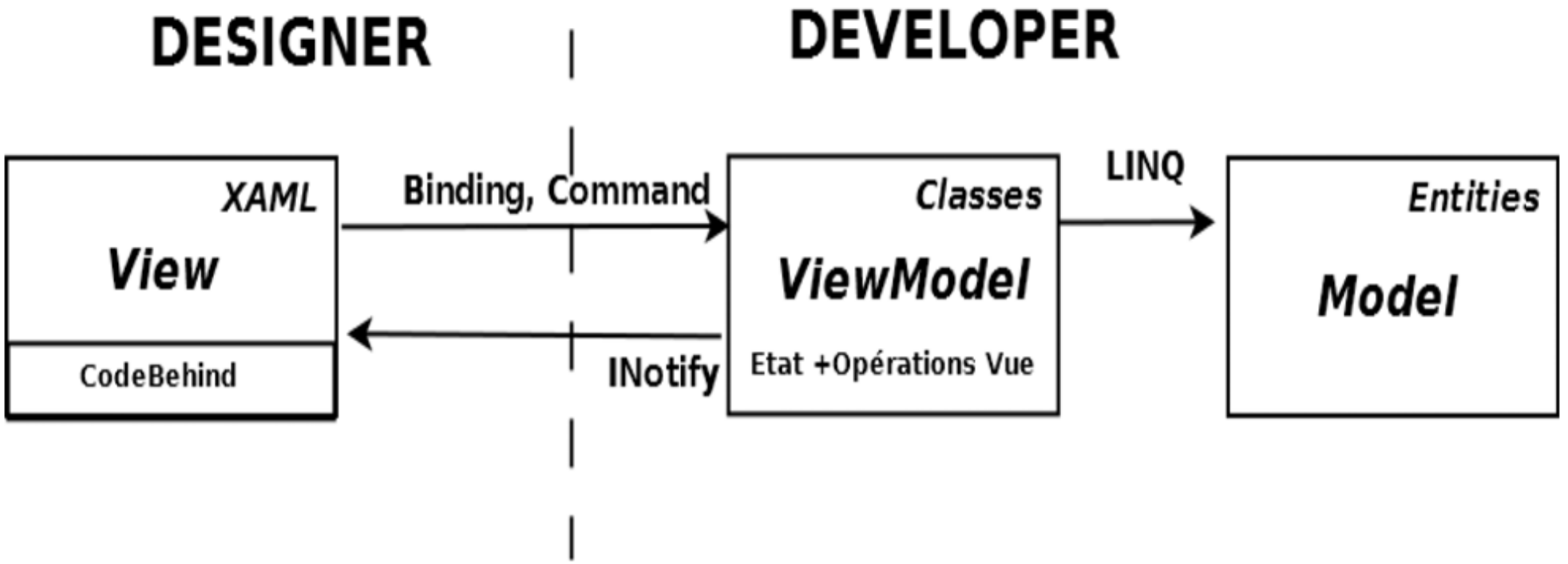
Design

- Grid
  - Elément le plus employé
  - Les composants peuvent s’adapter (*)
  - ShowGridLines = true

Design Exemple


```
<Grid>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="50"/>
        <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="50" />
        <RowDefinition Height="*" />
    </Grid.RowDefinitions>
    <Canvas Background="Red" />
    <Canvas Background="Blue" Grid.Column="1" />
    <Canvas Background="Yellow" Grid.Row="1" />
    <Canvas Background="Green" Grid.Column="1" Grid.Row="1" />
</Grid>
```

MVVM : Séparation en couches



Considérations sur le(s) modèle(s)

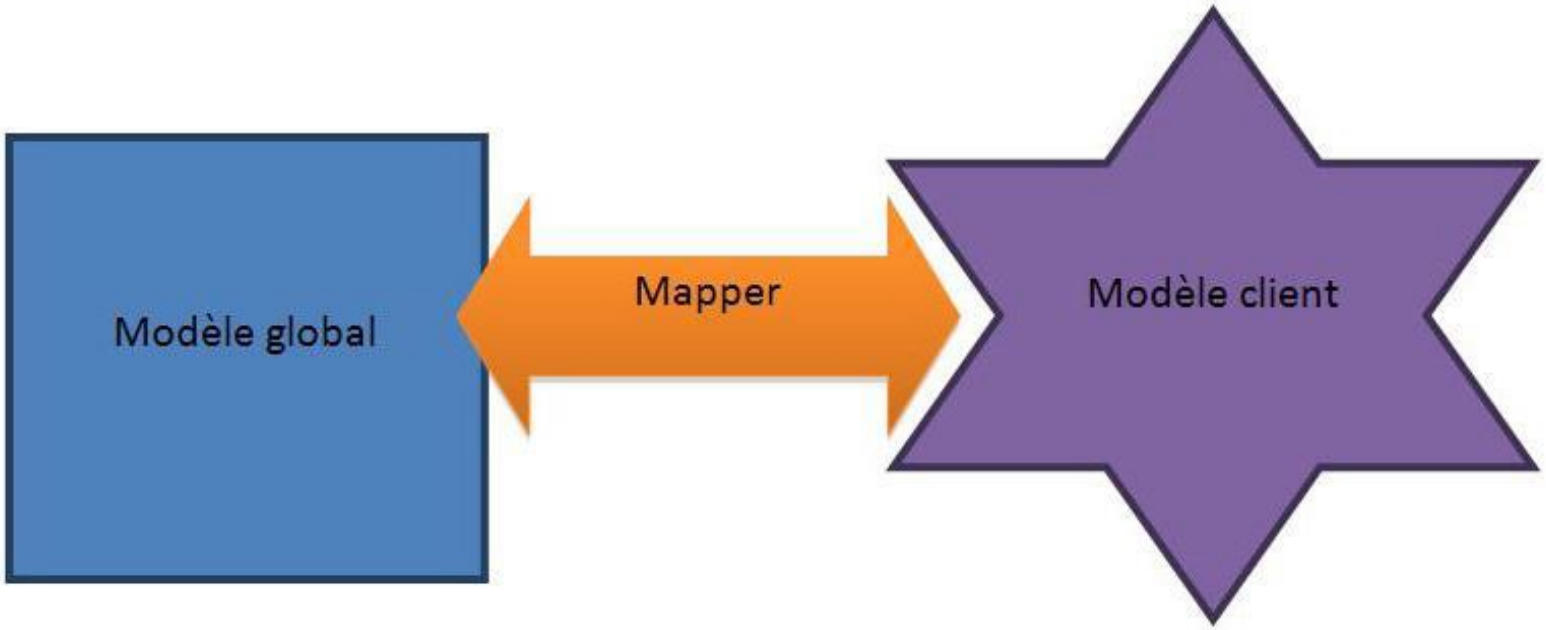
- Modèle global == modèle métier (svt Entities)
- Modèle client == UI modèle
- Modèle client aussi appelée Modèle de présentation
- Ne pas confondre Modèle global et Modèle client
- Modèle global indépendant du modèle client
- Rôle du modèle client : présenter les données, notifier la vue, état de la vue, gérer les actions, … .

Considérations sur le(s) modèle(s)

- Quid si le modèle global change ?
  - Quels impacts sur les Viewmodels ?
    - Dépend de la technique utilisée
      - Mapping
      - Wrapping

Considérations sur le(s) modèle(s)

- Mapping

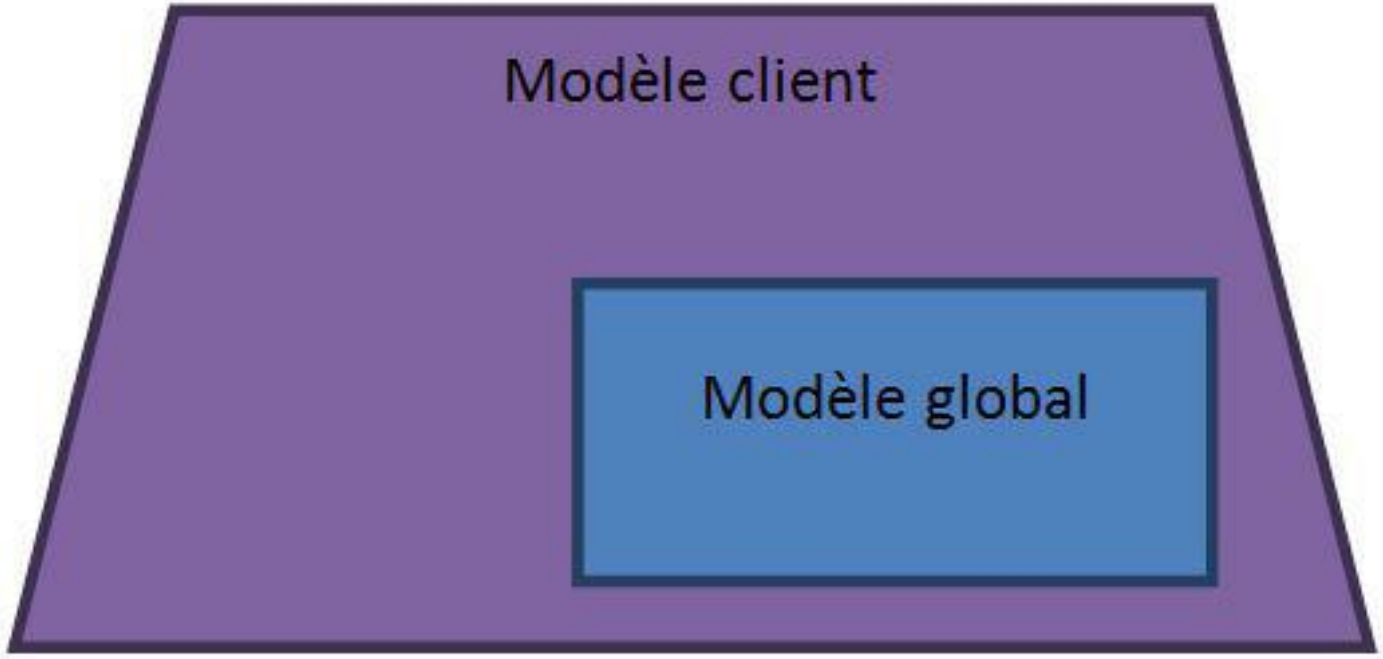


Considérations sur le(s) modèle(s)

- Mapping Avantages
  - Indépendance forte entre Modèle Global et Modèle Client
  - Maintenance facilitée
    - Réécriture centrée dans le mapper en cas de changement
- Mapping Inconvénients
  - Coût d'écriture initial
  - Légère puissance de calcul supplémentaire

Considérations sur le(s) modèle(s)

- Wrapping



Considérations sur le(s) modèle(s)

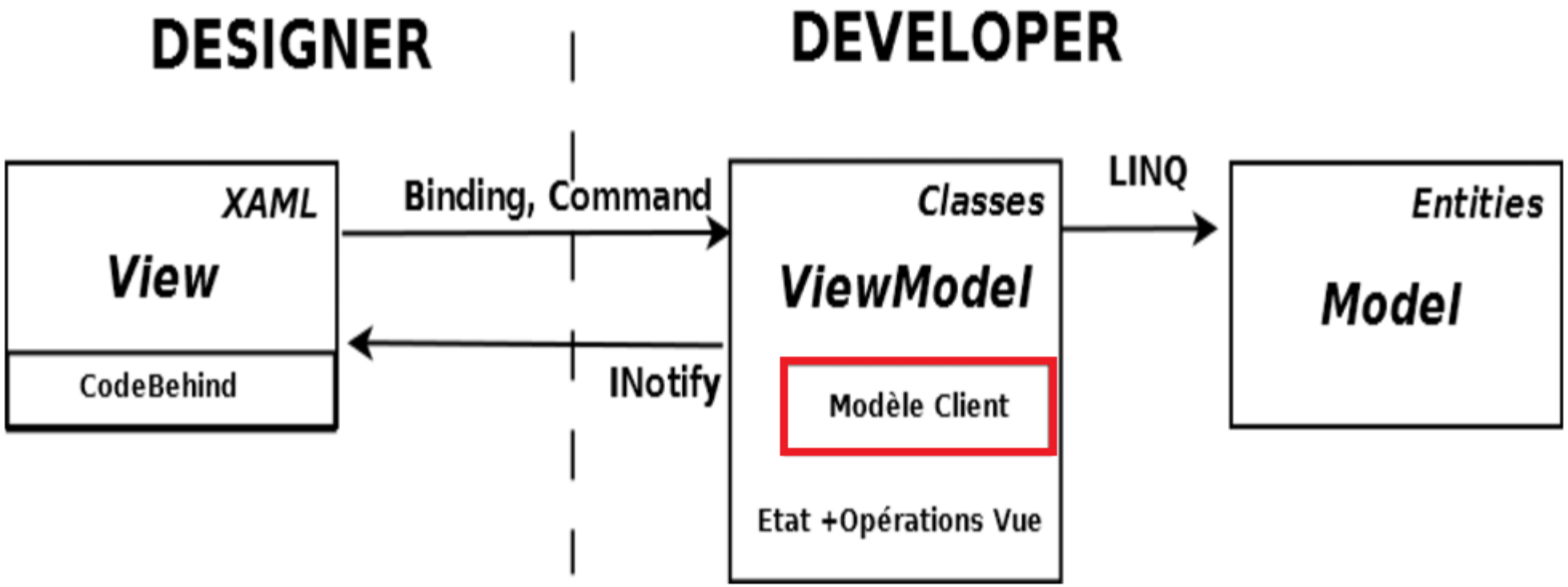
- Wrapping Avantages
  - Peu de puissance de calcul nécessaire
  - Ecriture facilitée
- Wrapping inconvénients
  - Forte Dépendance au modèle global
  - Nécessite connaissances et un contrôle du code du modèle global

Considérations sur le(s) modèle(s)

- Wrapping exemple
  - Création d'un ViewModel ….

```
public class ContactModel : ModelBase
{
    private readonly Contact _Contact;
```

Considérations sur le(s) modèle(s)



Datacontext hérité

Le datacontext est hérité. Un composant enfant dans le XAML héritera du datacontext de son parent.

Exemple :

1. Window →liste d'employés comme datacontext
2. Grid positionne son datacontext sur le SelectedItem →un employé
3. Les champs à l'intérieur du grid auront comme Datacontext par défaut l' employé sélectionné
4. Et si je veux que le champs aient le Datacontext de la fenêtre ?

```
"{Binding DataContext.ListEmployee,
        RelativeSource={RelativeSource FindAncestor,
        AncestorType={x:Type Window}}}"
```