Shell View

File   Edit   View   Help

Help   About

Home

Counter

Customers

Products

Settings

Window Name

File   Edit   View   Help

☺

Home

Counter

Customers

Products

Settings

About

100 (+)

ICustomerService

Window Name

File   Edit   View   Help

| Home |
| Counter |
| Customers |
| Products |
| Settings |

About

Full Name	Credit	Nickname	Removed
John Smith	40	Peldi	⊟
Ann Spider	38		⊟
Mariah Fox	41	Patata	☑
Viktor Novorsky	100	Val	⊟

John Smith

First name

John

Last name

Smith

Customer Type

Happy

Save

XAML = XML + { }

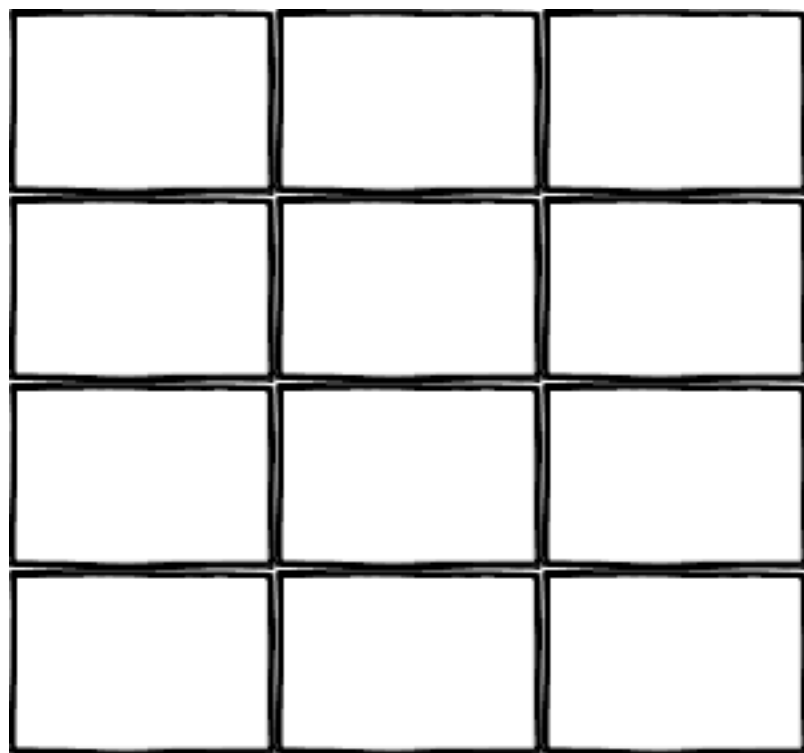$\underbrace{\text{XML}}_{\text{Markup}}$ $\underbrace{\{\ \}}_{\substack{\text{Markup} \\ \text{Extension}}}$

Window Name

File   Edit   View   Help

Button

Button
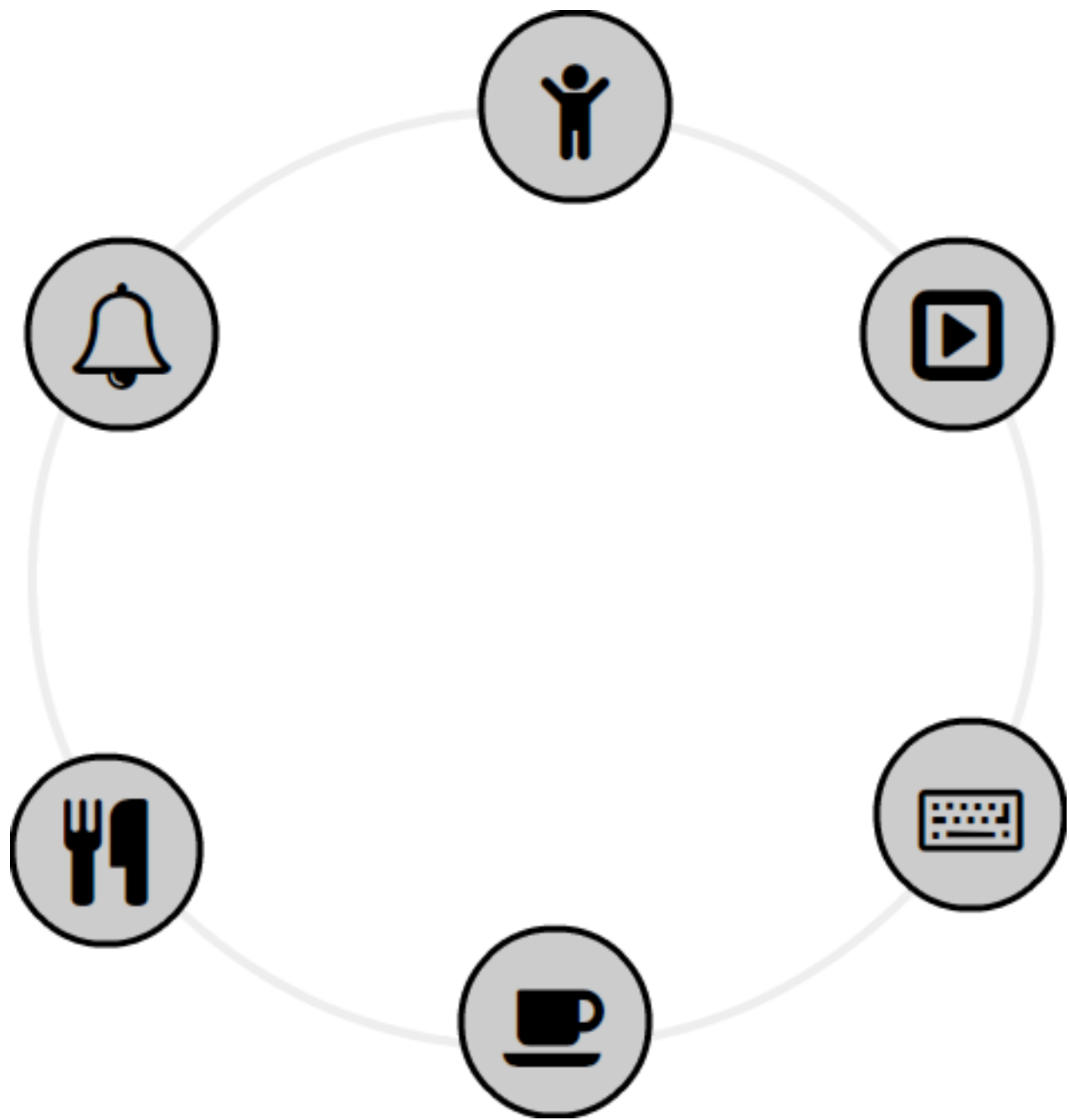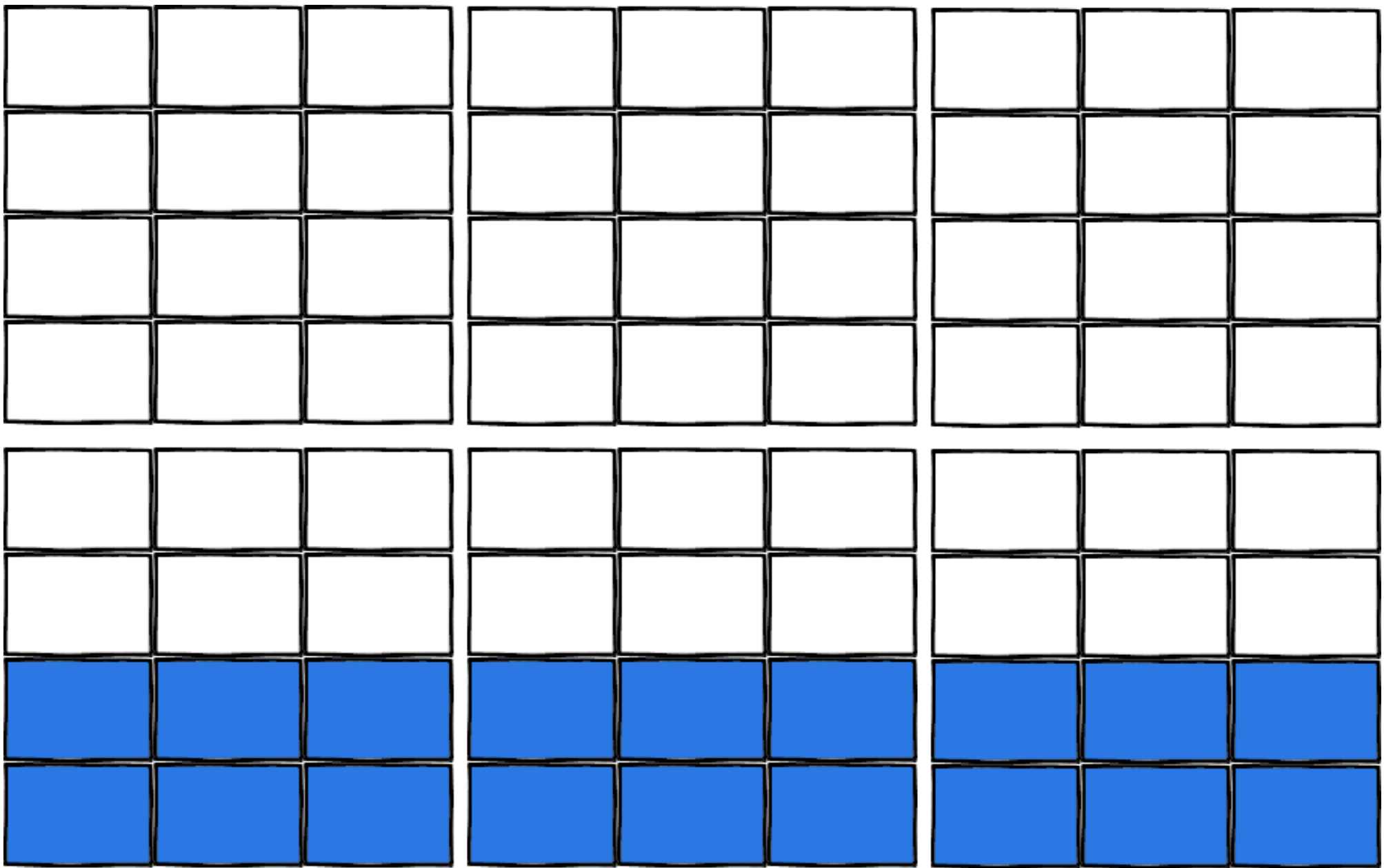
Double

MVVM (Model-ViewModel-View)

ViewModel-First

View-First

View

Send

INotifyPropertyChanged

ViewModel

Model

Service

View

| Window Name |
| File Edit View Help |

B I U S style ≡ ≡ ⟳ C ▥ ☺

Sent. Send

code-behind

messageService.Send()

textBlock1.Text = "Sent."

label.Content = "Sent."

button1.Enabled = false;

menuItem5.Enabled = false;

View   Controller   Model
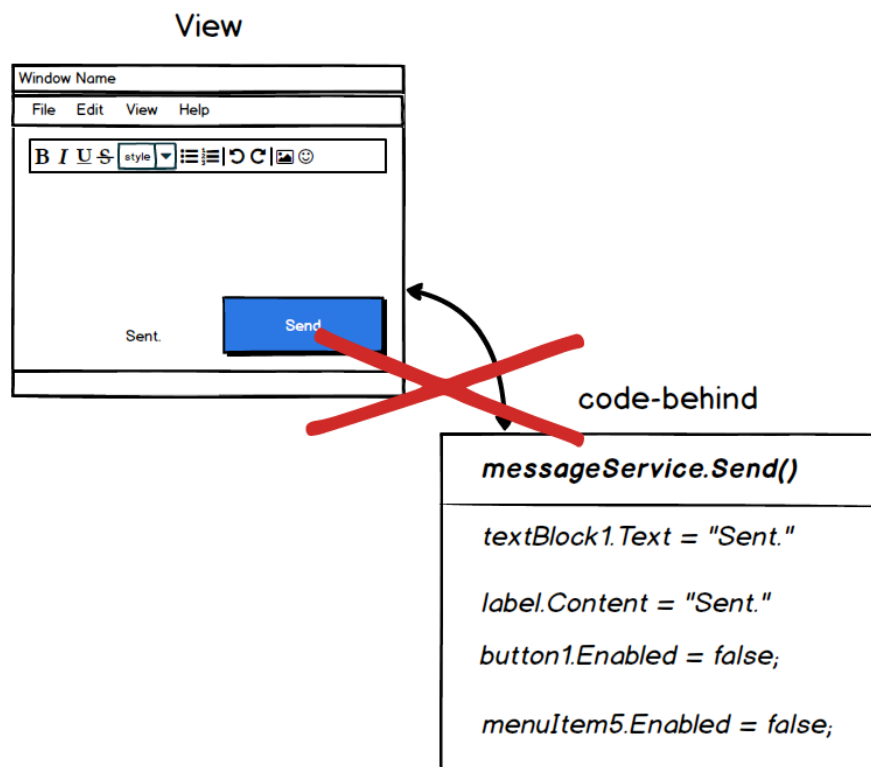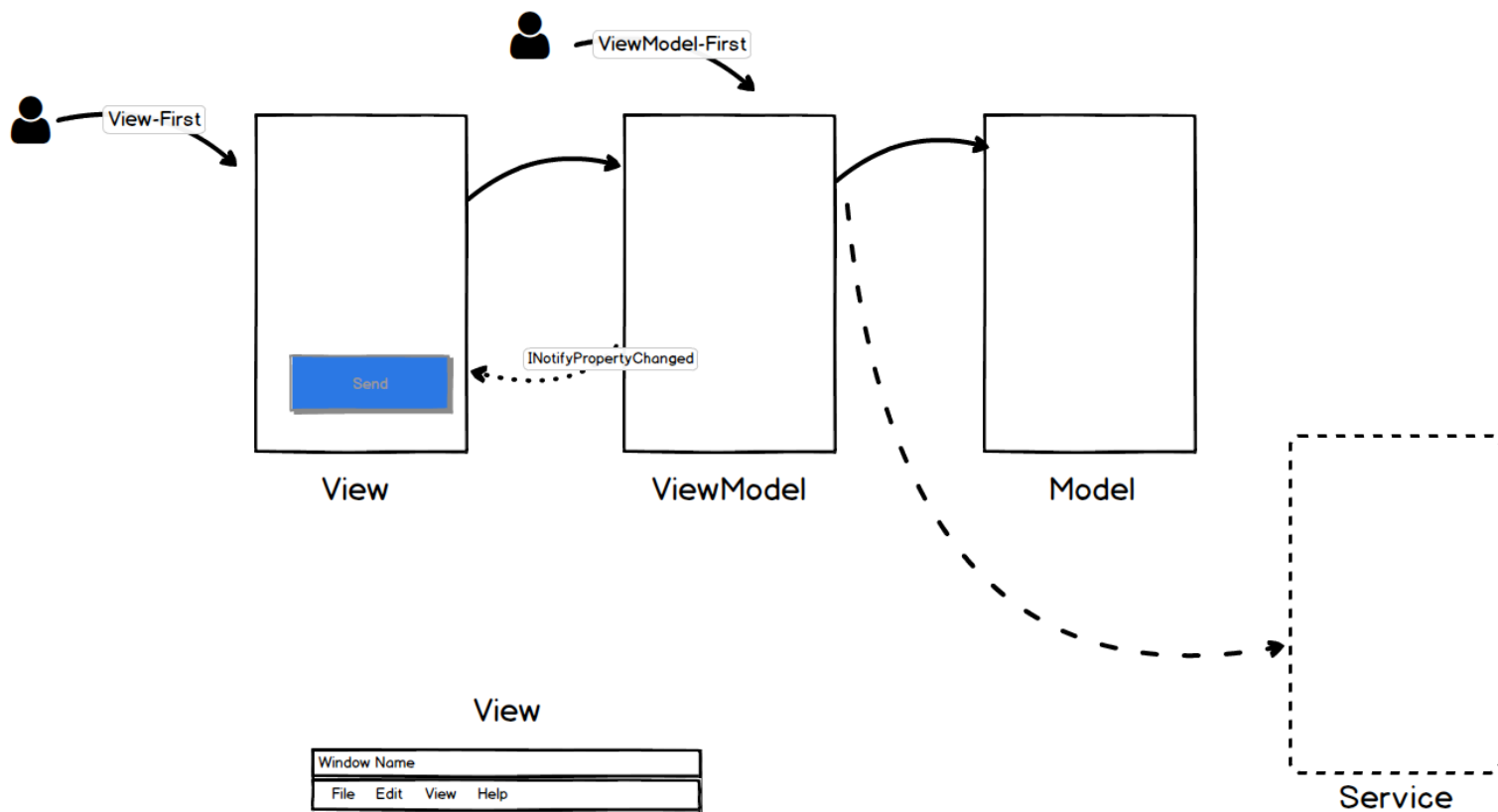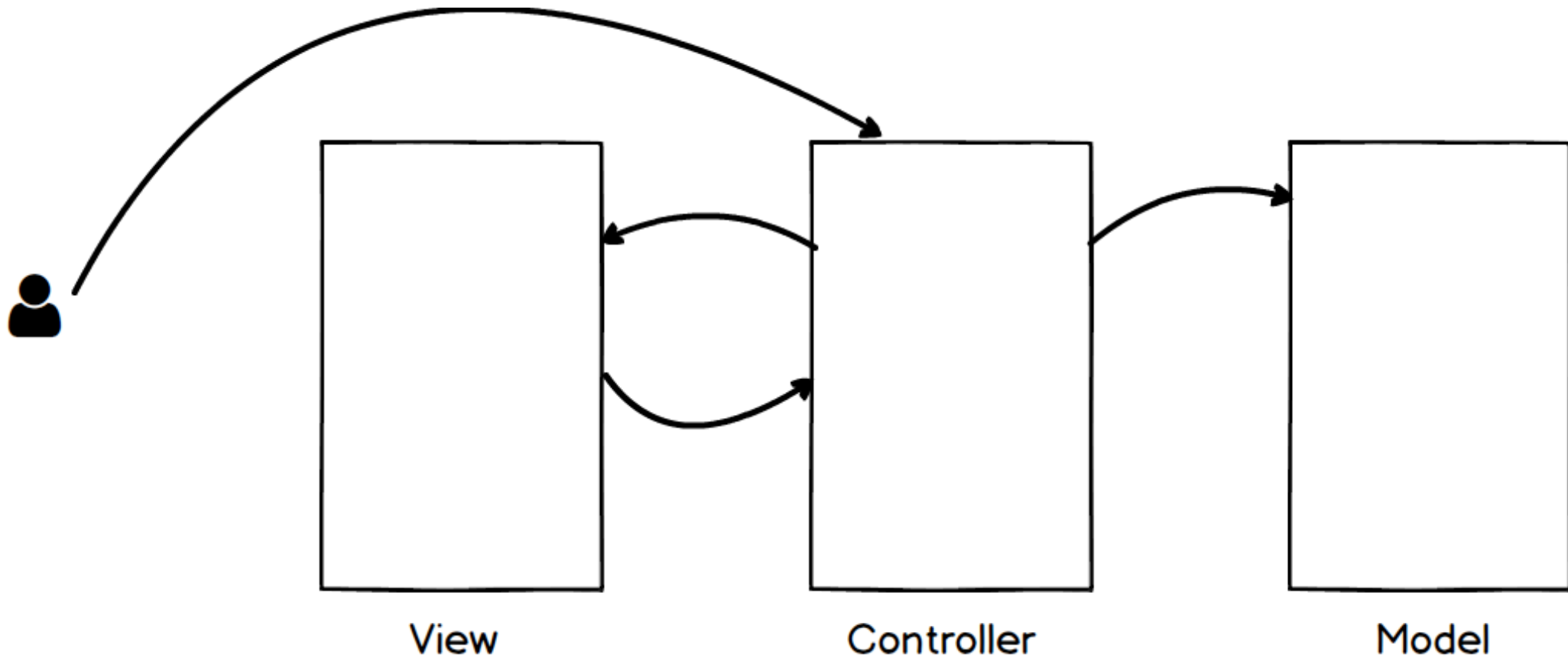
```
 public CustomersViewModel CustomersViewModel => new CustomersViewModel(new FakeCustomerService(new
CustomerFaker()));
```

class Factory

T Create(Type type);

Container

Key	Value	cykl życia
CustomersViewModel	CustomersViewModel	Instance
ICustomerService	FakeCustomerService	Instance
ICustomerService	DbCustomerService	Instance
Faker<Customer>	CustomerFaker	Singleton

T Resolve<T>

IEnumerable<T> ResolveAll<T>

Order
───────────────

Number

Print() { ... }

bool CanPrint() { .... }

Send() { .... }

bool CanSend() { .... }

interface ICommand

     void Execute() { ... }
     bool CanExecute() { ... }

class PrintCommand : ICommand

     Execute() { ... }

class SendCommand : ICommand

     Execute() { ... }

ICollection<ICommand> commands;   ICollection<ICommand> commands;

commands.Add(new PrintCommand(5));

commands.Add(new SendCommand("marcin.sulecki@sulmar.pl"));

foreach(ICommand command in commands)

     command.Execute();

CounterViewModel

Binding

Binding

public Counter { get; set; }

public Name { get; set; }

public Add() => Counter++;

Refresh()

INotifyPropertyChanged.PropertyChanged

PropertyChanged+= { ... }