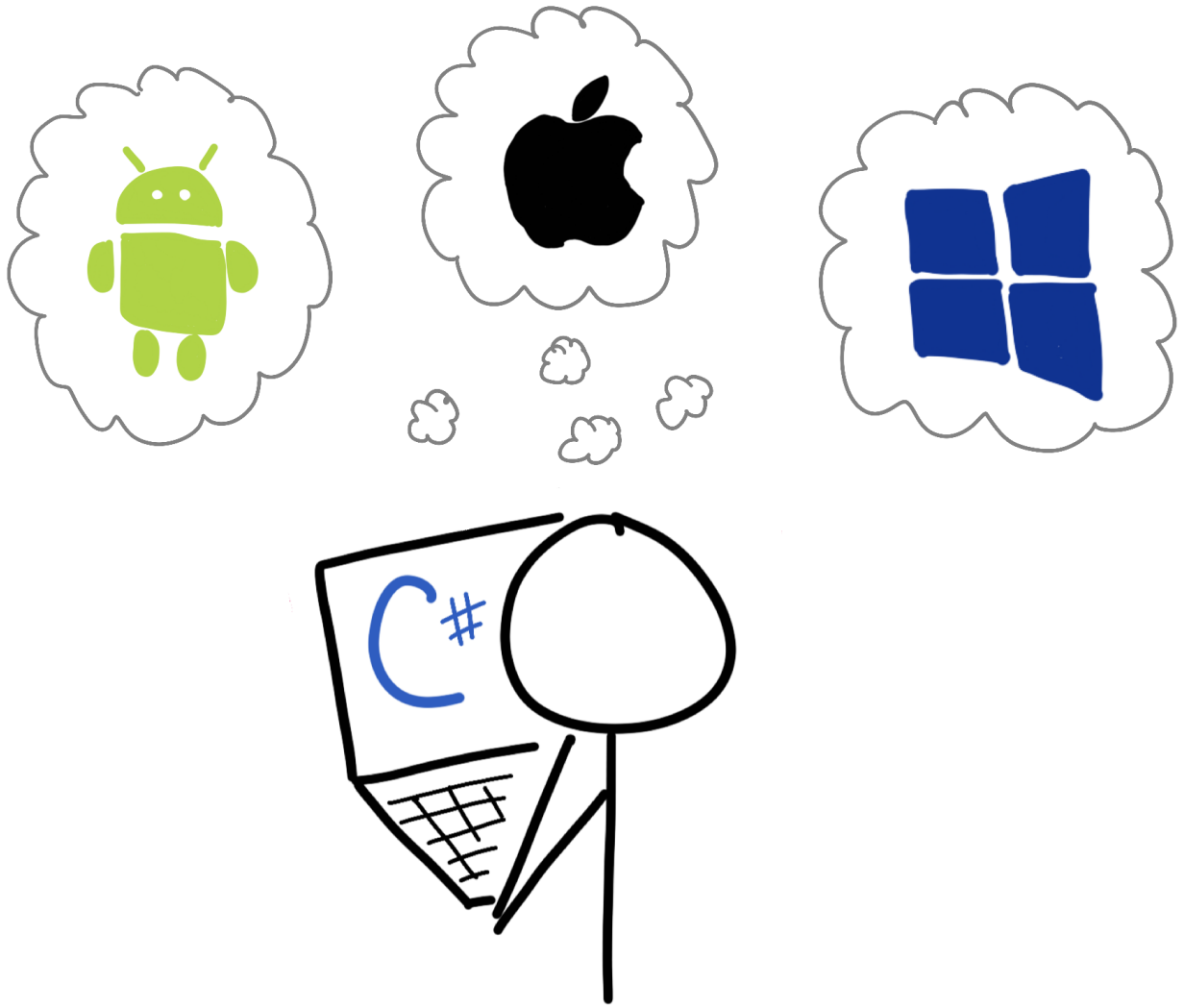
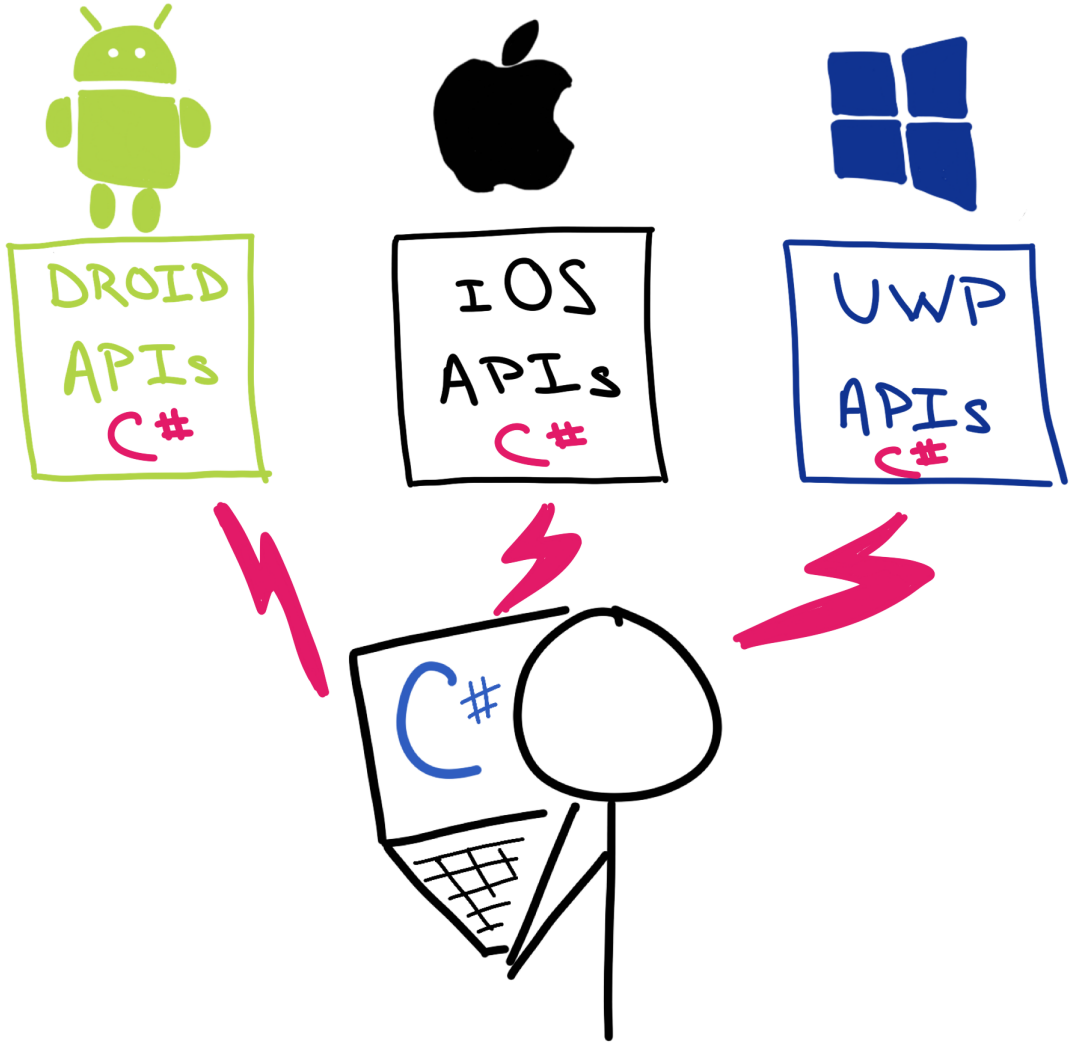
Creating Three Beautiful Apps at Once

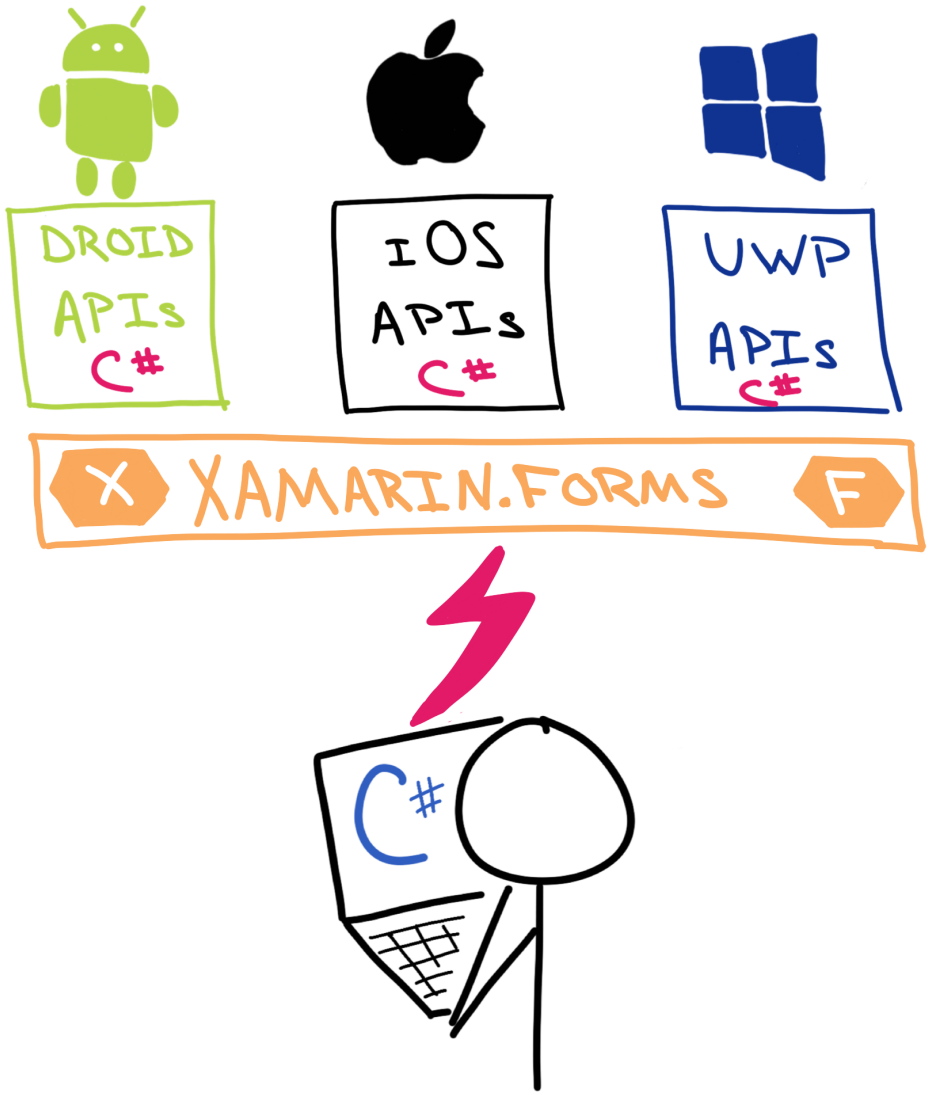
INTRODUCTION TO XAMARIN.FORMS

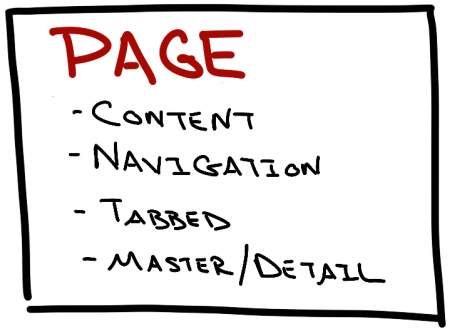
Matthew Soucoup
PRINCIPAL

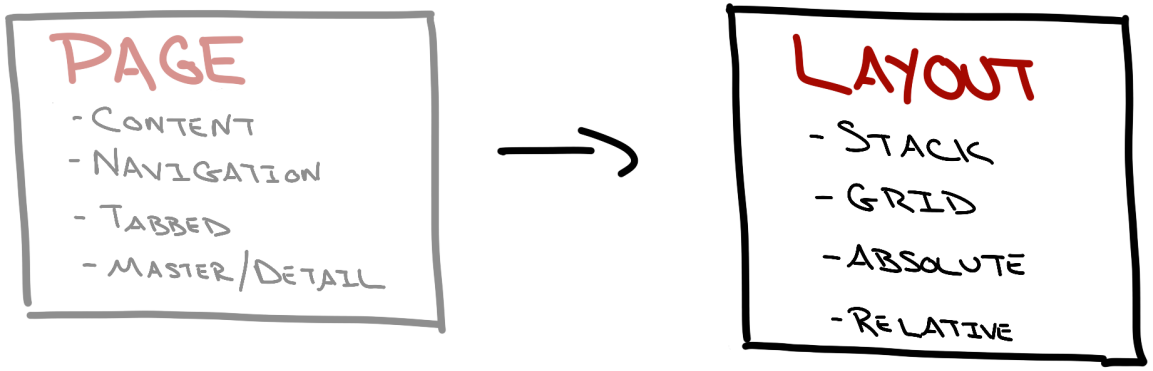
@codemillmatt codemilltech.com

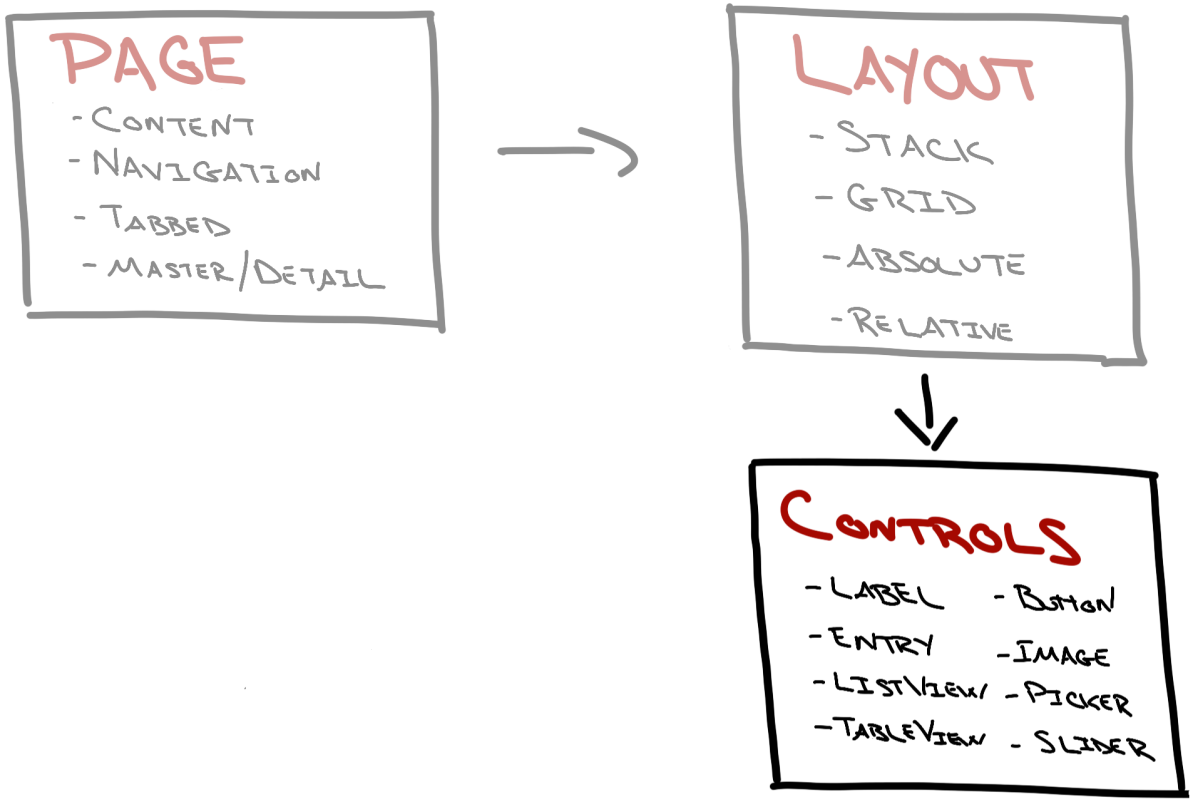

C#

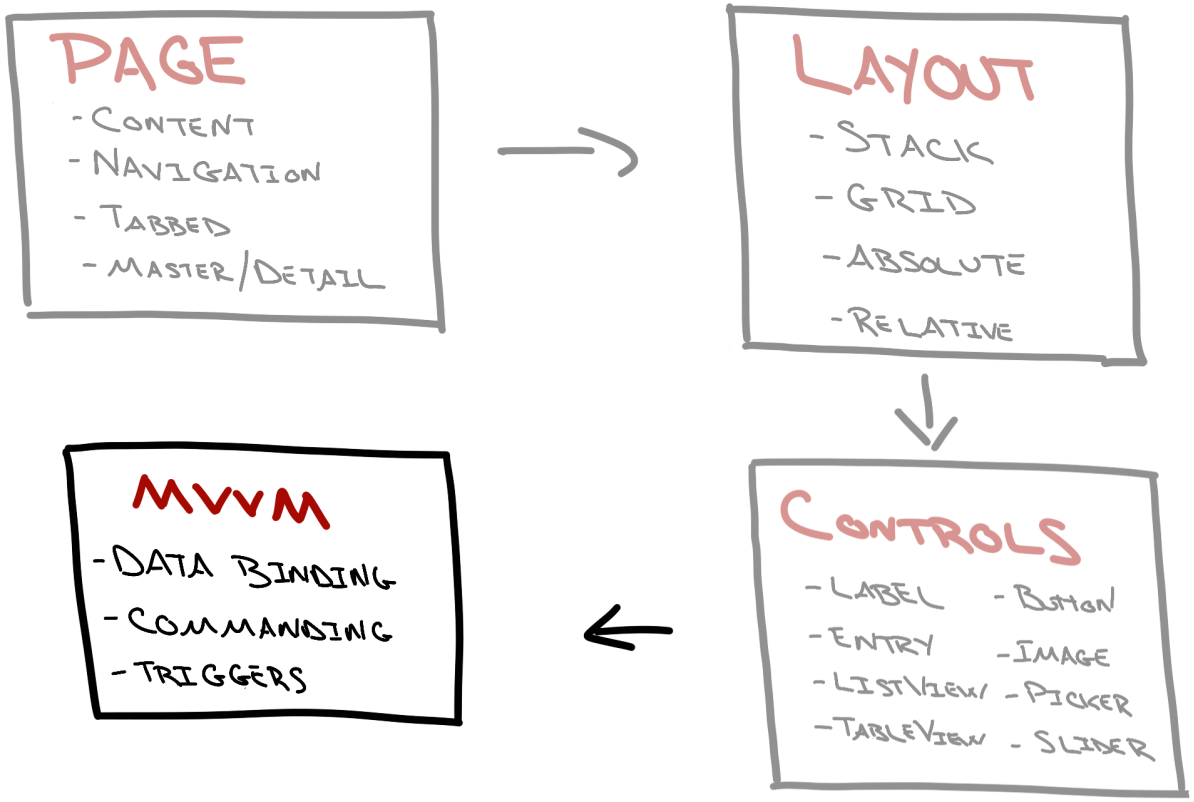

DROID
APIs
C#
iOS
APIs
C#
UWP
APIs
C#
C#

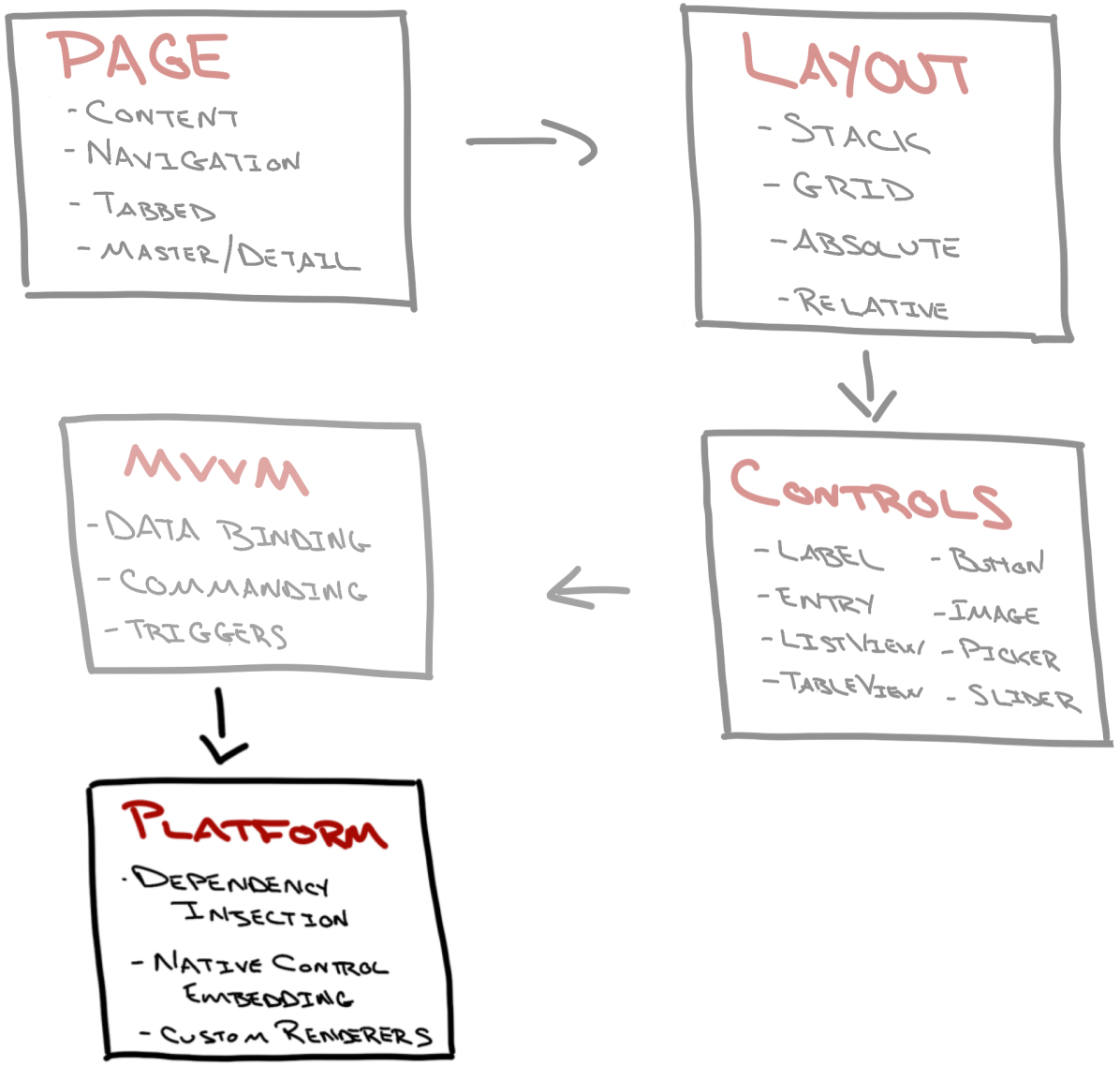

DROID
APIs
C#
iOS
APIs
C#
UWP
APIs
C#
XAMARIN.FORMS
C#

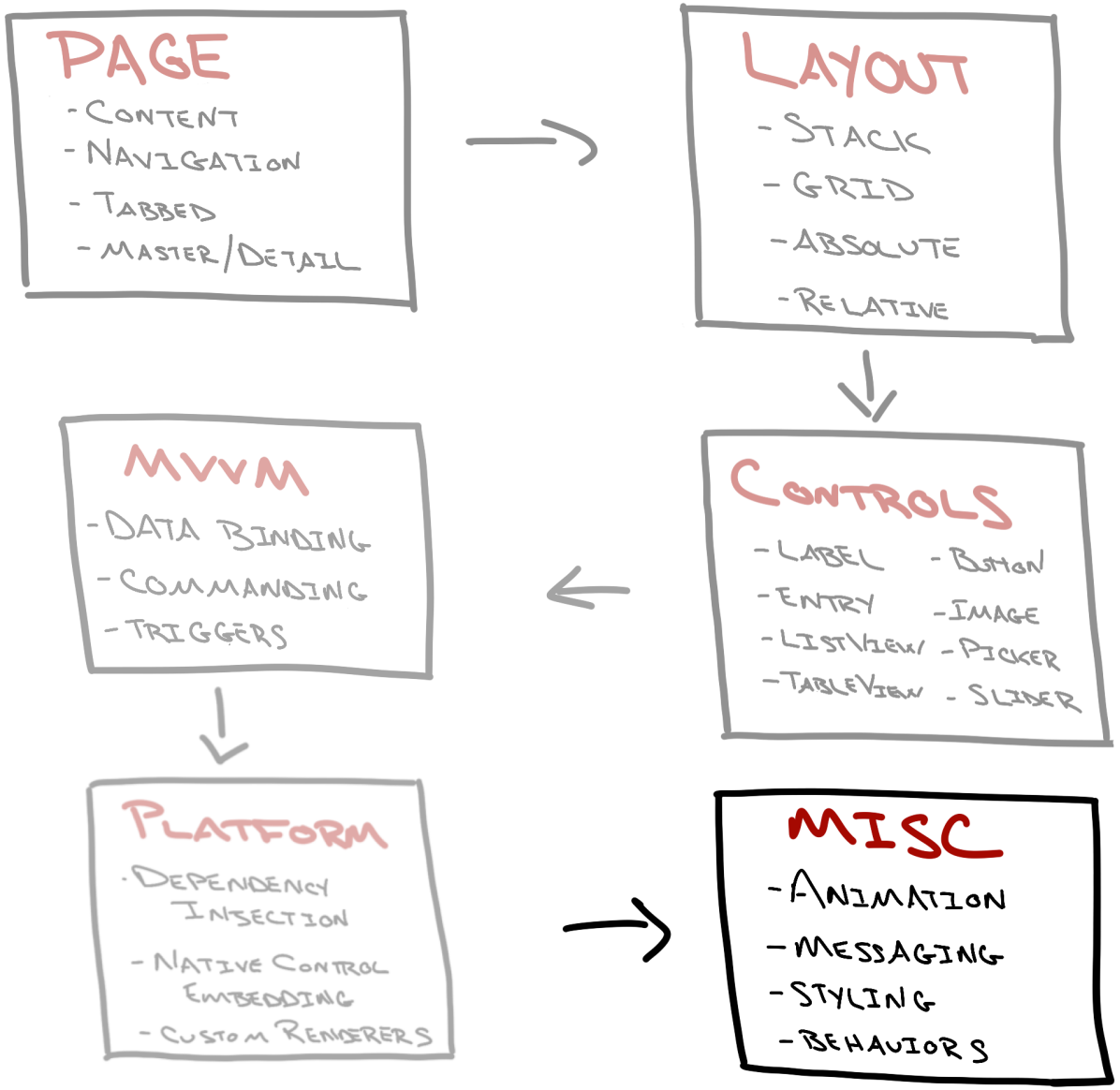

PAGE
- CONTENT
- NAVIGATION
- TABBED
- MASTER/DETAIL

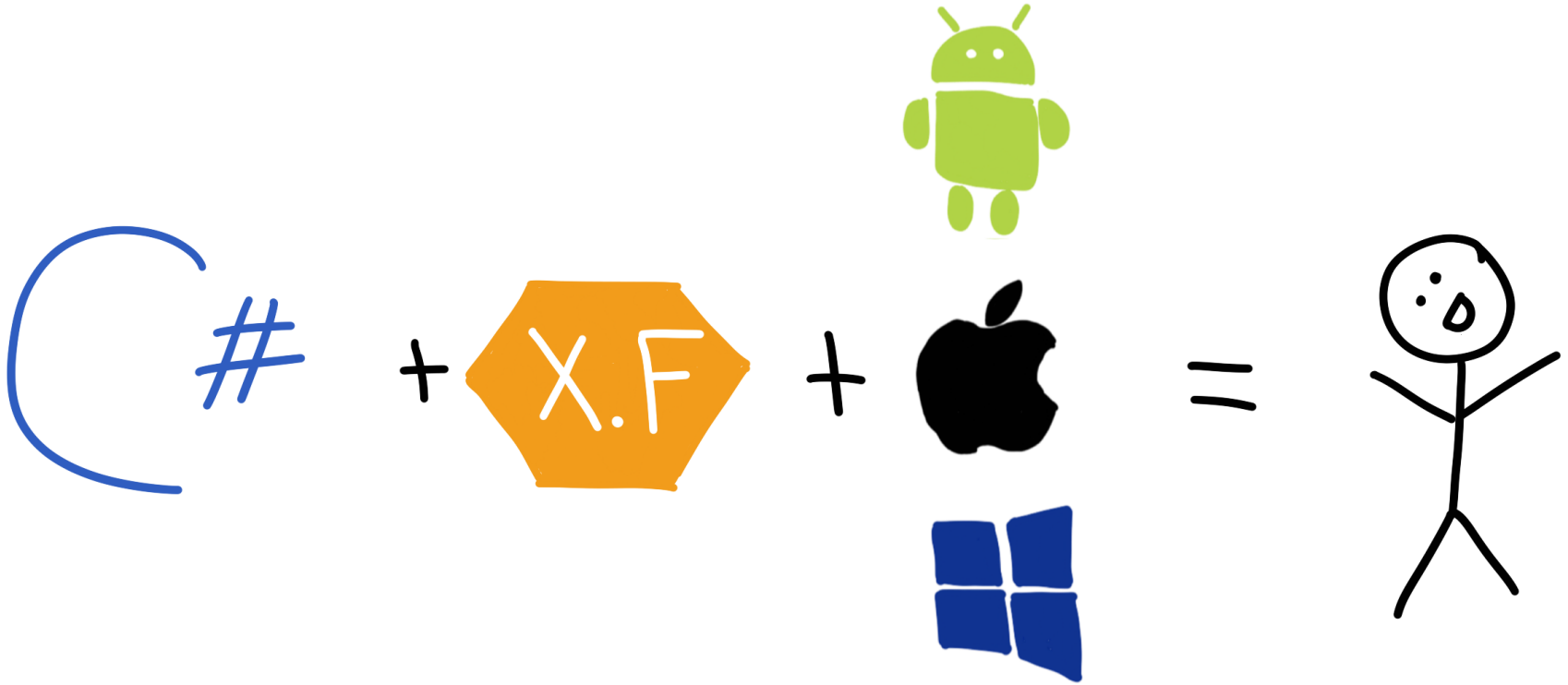

PAGE
- CONTENT
- NAVIGATION
- TABBED
- MASTER/DETAIL
LAYOUT
- STACK
- GRID
- ABSOLUTE
- RELATIVE


PAGE
- CONTENT
- NAVIGATION
- TABBED
- MASTER/DETAIL
LAYOUT
- STACK
- GRID
- ABSOLUTE
- RELATIVE
CONTROLS
- LABEL
- BUTTON
- ENTRY
- IMAGE
- LISTVIEW
- PICKER
- TABLEVIEW
- SLIDER


PAGE
- CONTENT
- NAVIGATION
- TABBED
- MASTER/DETAIL
LAYOUT
- STACK
- GRID
- ABSOLUTE
- RELATIVE
CONTROLS
- LABEL
- BUTTON
- ENTRY
- IMAGE
- LISTVIEW
- PICKER
- TABLEVIEW
- SLIDER
MVVM
- DATA BINDING
- COMMANDING
- TRIGGERS


PAGE
- Content
- Navigation
- Tabbed
- Master/Detail
LAYOUT
- Stack
- Grid
- Absolute
- Relative
CONTROLS
- Label
- Button
- Entry
- Image
- ListView
- Picker
- TableView
- Slider
MVVM
- Data Binding
- Commanding
- Triggers
PLATFORM
- Dependency Injection
- Native Control Embedding
- Custom Renderers


PAGE
- Content
- Navigation
- Tabbed
- Master/Detail
LAYOUT
- Stack
- Grid
- Absolute
- Relative
CONTROLS
- Label
- Button
- Entry
- Image
- ListView
- Picker
- TableView
- Slider
MVVM
- Data Binding
- Commanding
- Triggers
PLATFORM
- Dependency Insection
- Native Control Embedding
- Custom Renderers
MISC
- Animation
- Messaging
- Styling
- Behaviors


C#
X.F

Intro to Xamarin.Forms Agenda

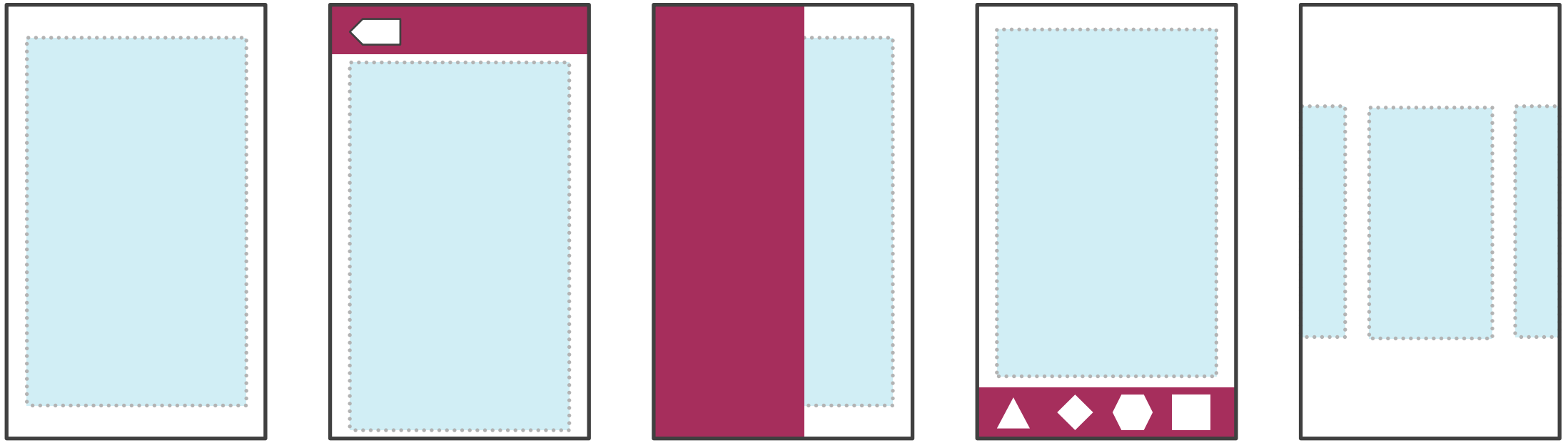
User Interface

MVVM

Additional Features and Platform Access

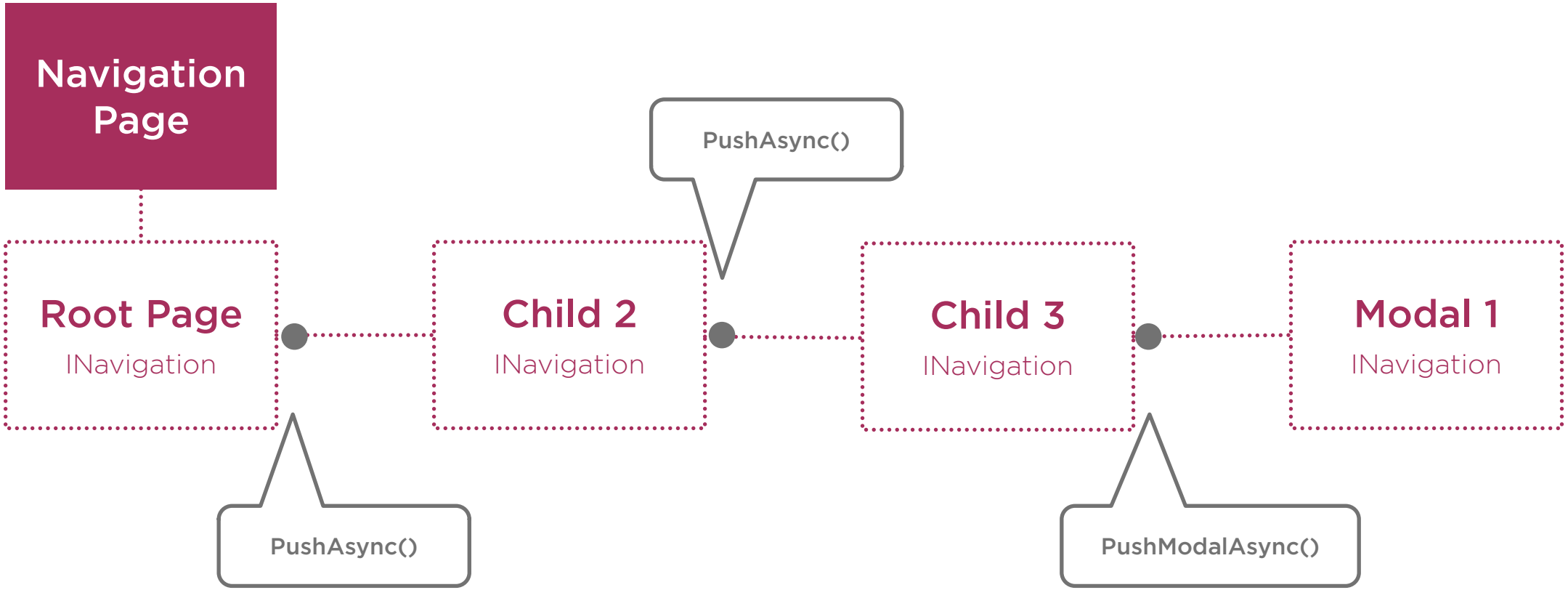
User Interface

Pages

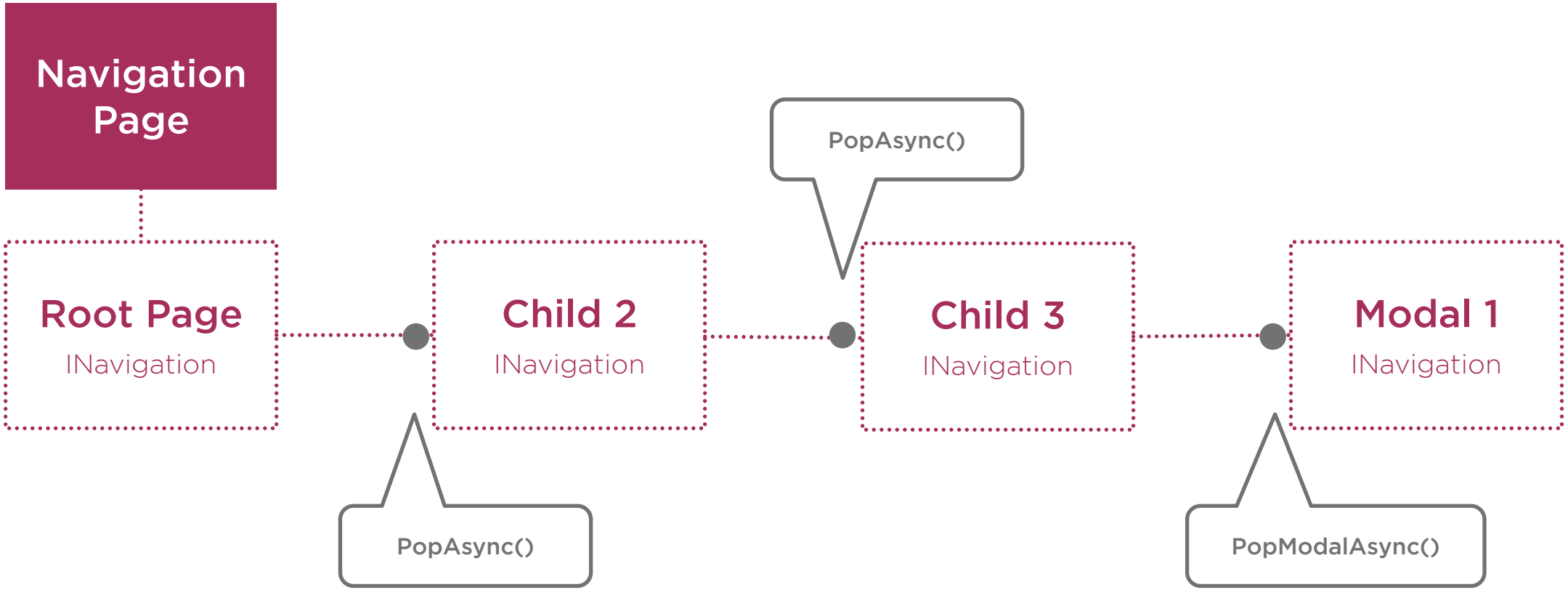


Content Navigation Master/Detail Tabbed Carousel

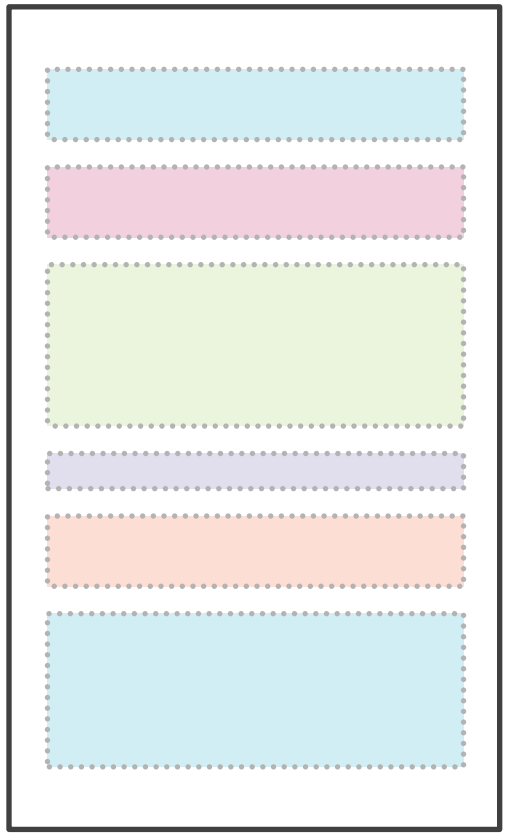
Navigation


Navigation Page
Root Page
INavigation
Child 2
INavigation
Child 3
INavigation
Modal 1
INavigation
PushAsync()
PushAsync()
PushModalAsync()

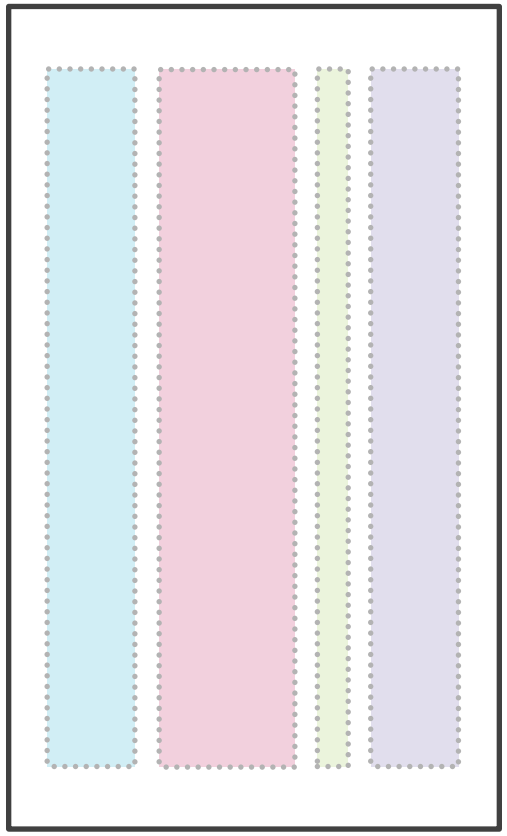
Navigation


Navigation Page
Root Page
INavigation
Child 2
INavigation
PopAsync()
Child 3
INavigation
PopAsync()
Modal 1
INavigation
PopModalAsync()

Layouts - Stack

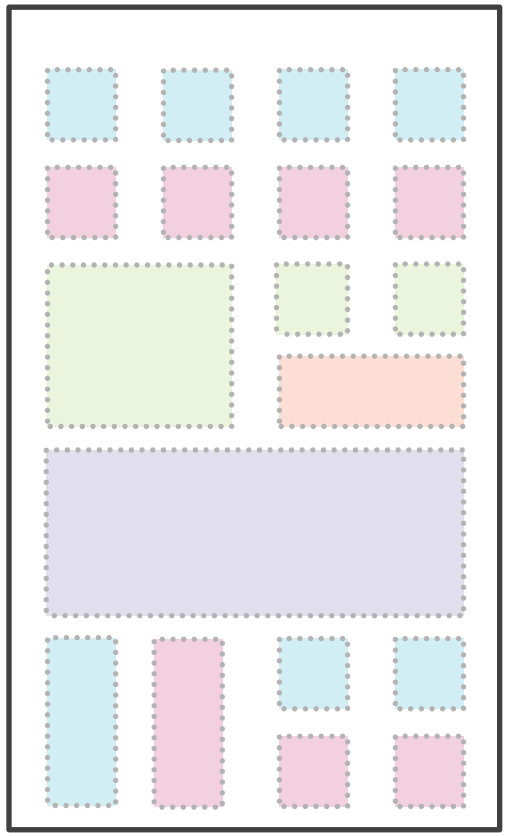


Vertical

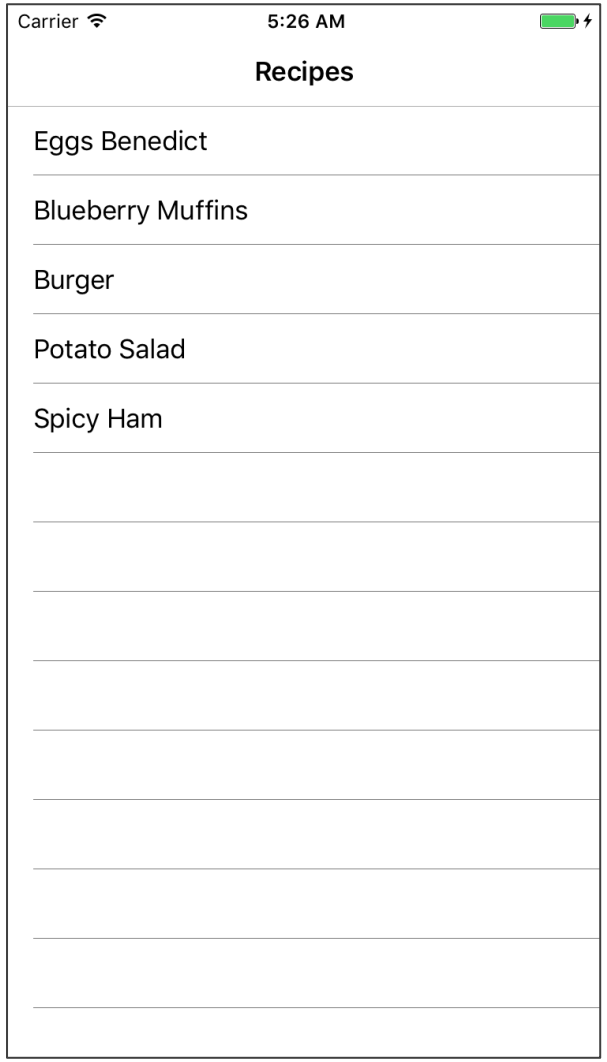


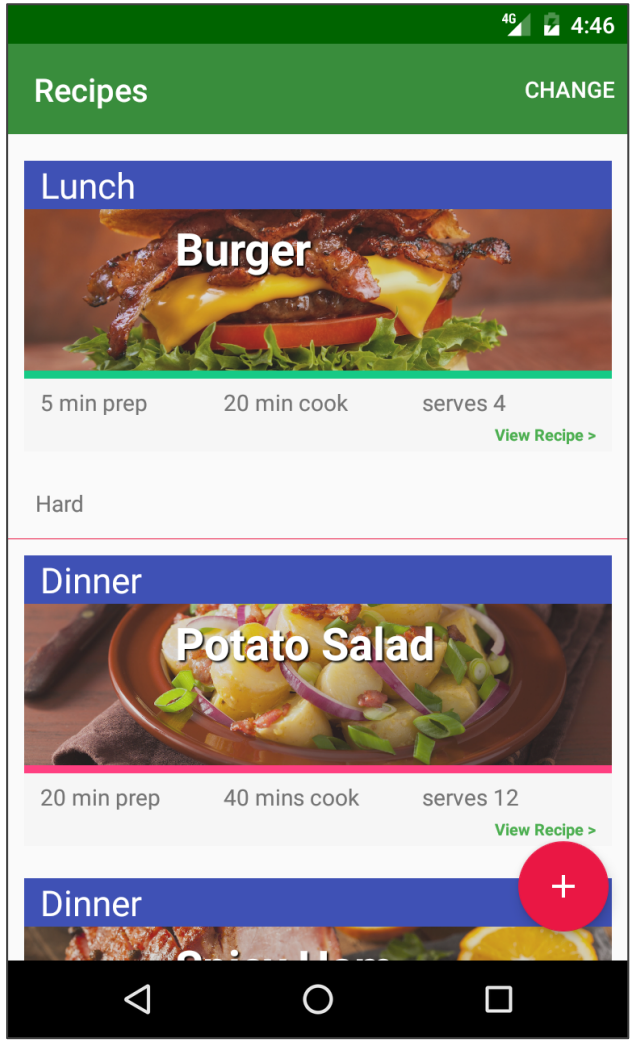
Horizontal

Layouts - Grid

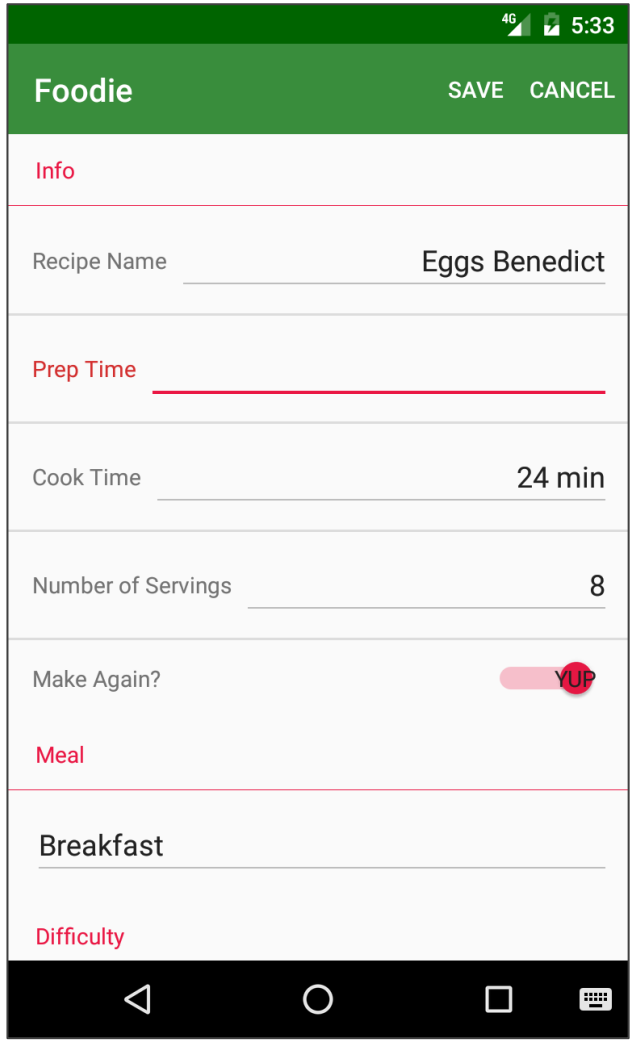


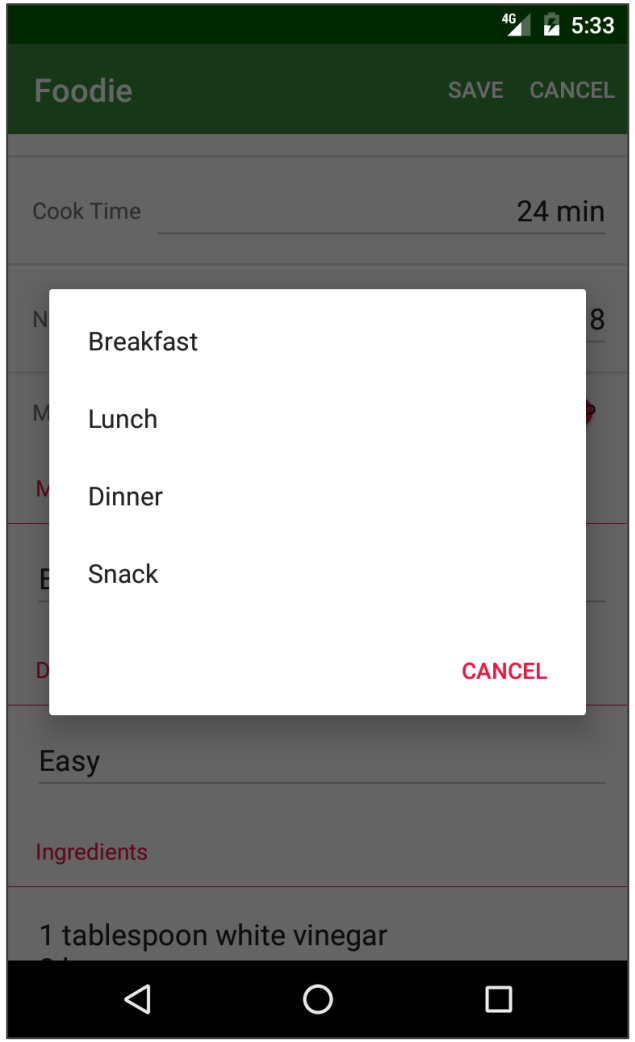
ListView


Carrier
5:26 AM
Recipes
Eggs Benedict
Blueberry Muffins
Burger
Potato Salad
Spicy Ham

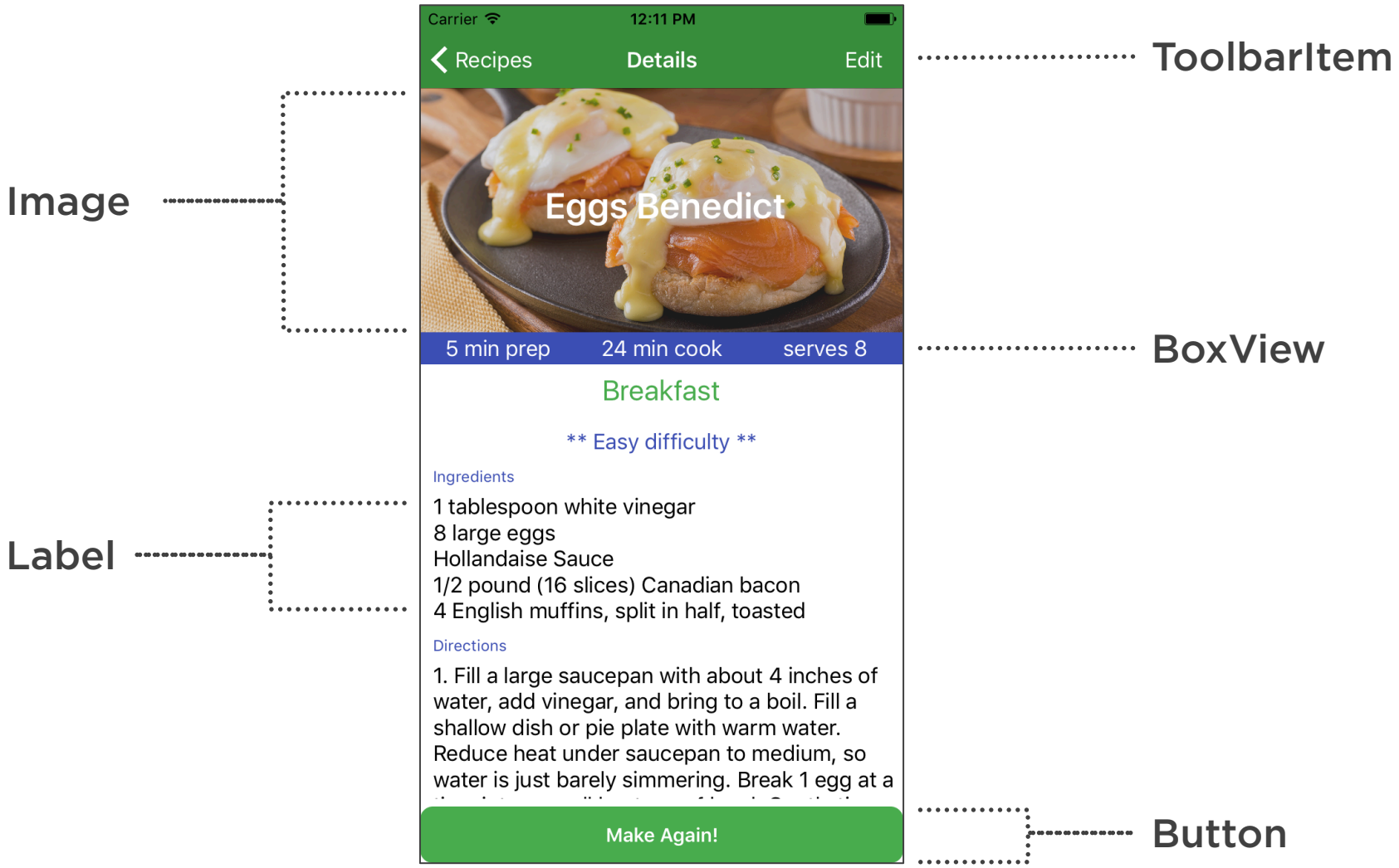

4:46
Recipes
CHANGE
Lunch
Burger
5 min prep
20 min cook
serves 4
View Recipe >
Hard
Dinner
Potato Salad
20 min prep
40 mins cook
serves 12
View Recipe >
Dinner

TableView

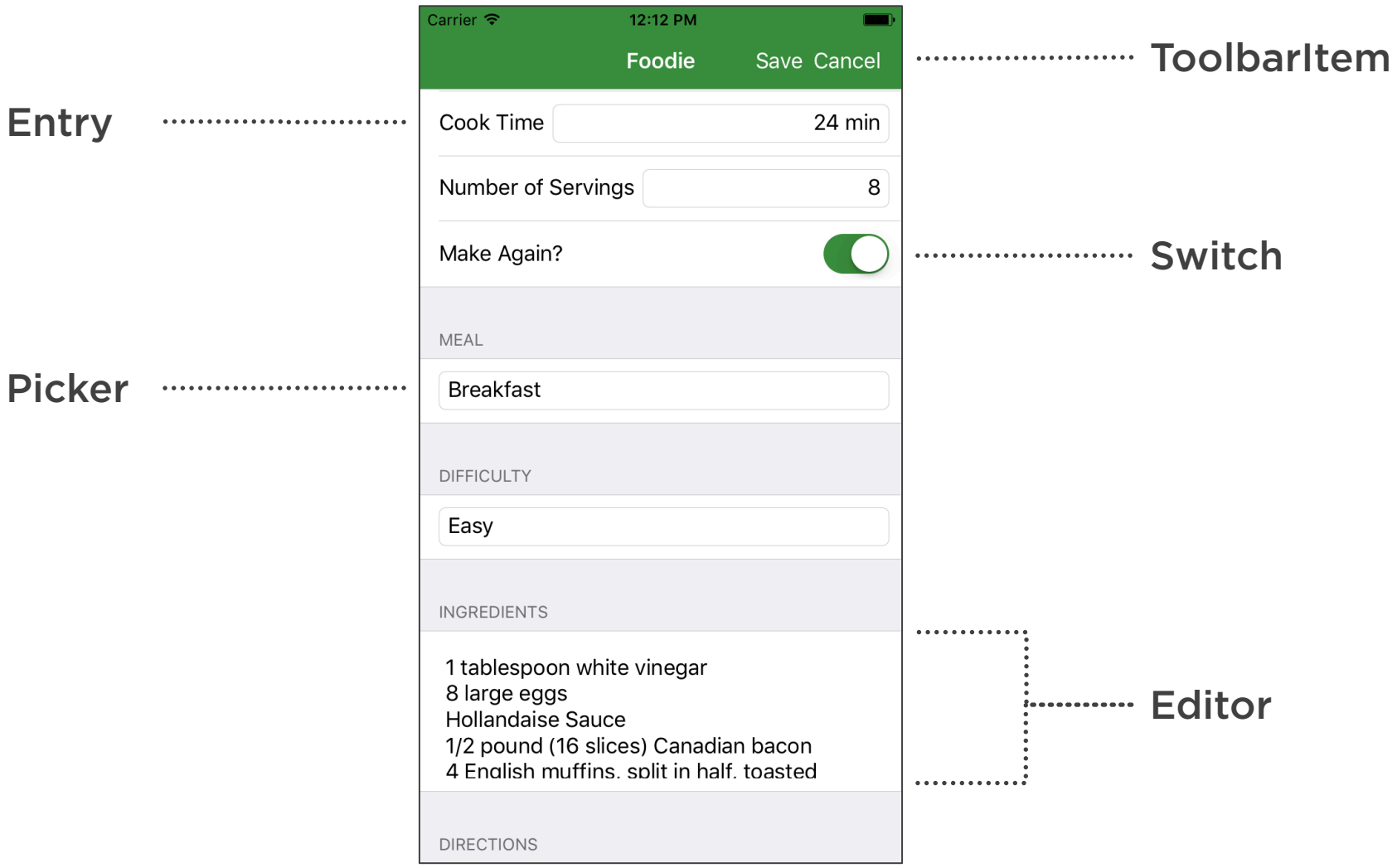

5:33
Foodie
SAVE
CANCEL
Info
Recipe Name
Eggs Benedict
Prep Time
Cook Time
24 min
Number of Servings
8
Make Again?
YUP
Meal
Breakfast
Difficulty


5:33
Foodie
SAVE
CANCEL
Cook Time
24 min
Breakfast
Lunch
Dinner
Snack
CANCEL
Easy
Ingredients
1 tablespoon white vinegar

Controls


Carrier
12:11 PM
Recipes
Details
Edit
Eggs Benedict
5 min prep
24 min cook
serves 8
Breakfast
** Easy difficulty **
Ingredients
1 tablespoon white vinegar
8 large eggs
Hollandaise Sauce
1/2 pound (16 slices) Canadian bacon
4 English muffins, split in half, toasted
Directions
1. Fill a large saucepan with about 4 inches of water, add vinegar, and bring to a boil. Fill a shallow dish or pie plate with warm water. Reduce heat under saucepan to medium, so water is just barely simmering. Break 1 egg at a
Make Again!
Image
Label
ToolbarItem
BoxView
Button

Controls

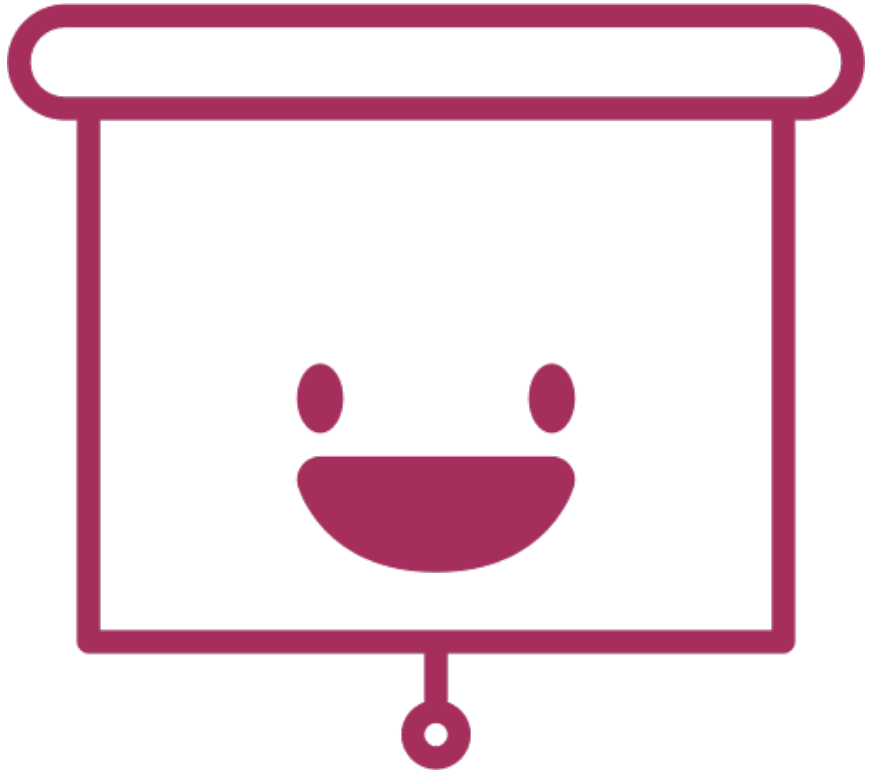

Carrier
12:12 PM
Foodie
Save
Cancel
Cook Time
24 min
Number of Servings
8
Make Again?
MEAL
Breakfast
DIFFICULTY
Easy
INGREDIENTS
1 tablespoon white vinegar
8 large eggs
Hollandaise Sauce
1/2 pound (16 slices) Canadian bacon
4 English muffins, split in half, toasted
DIRECTIONS
Entry
Picker
ToolbarItem
Switch
Editor

UI Demo

Controls

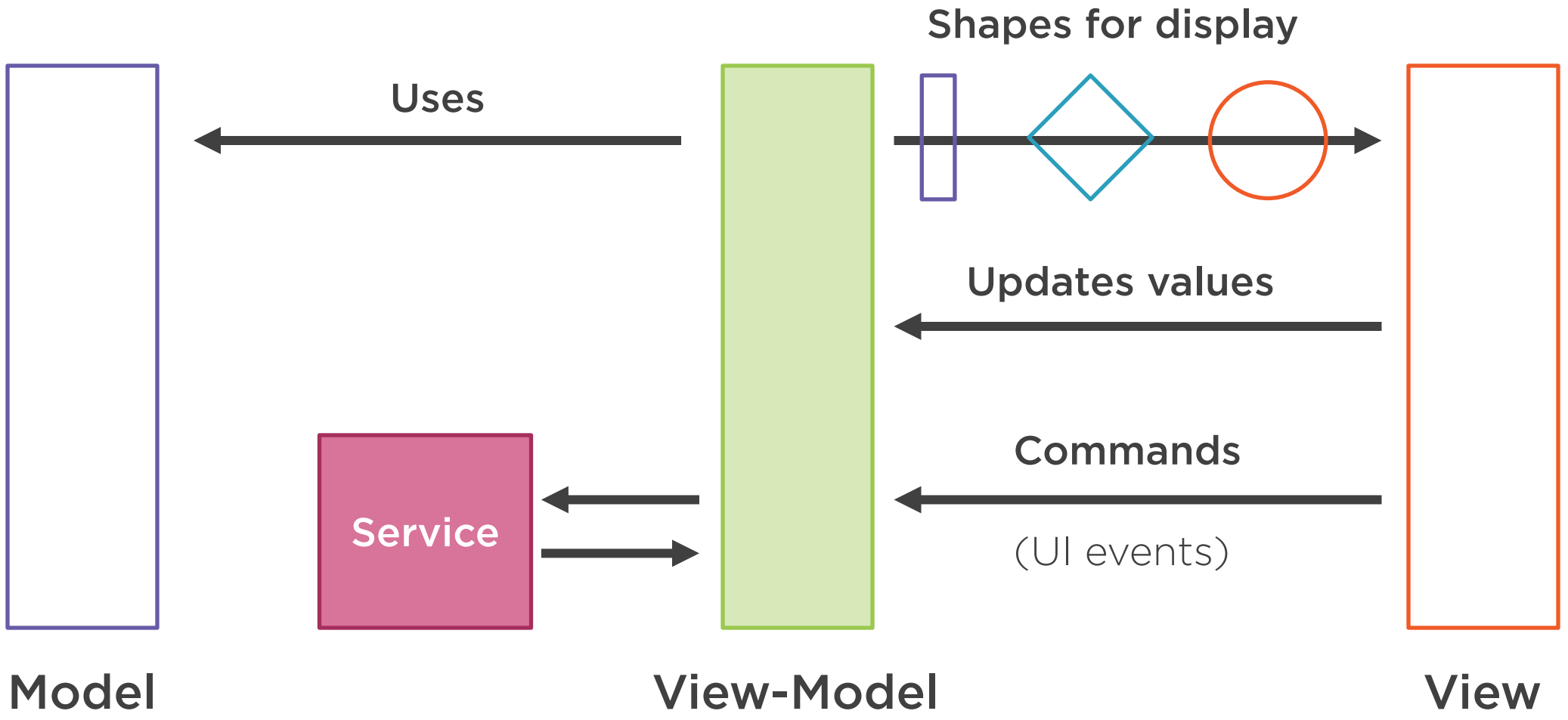
TableView

ListView

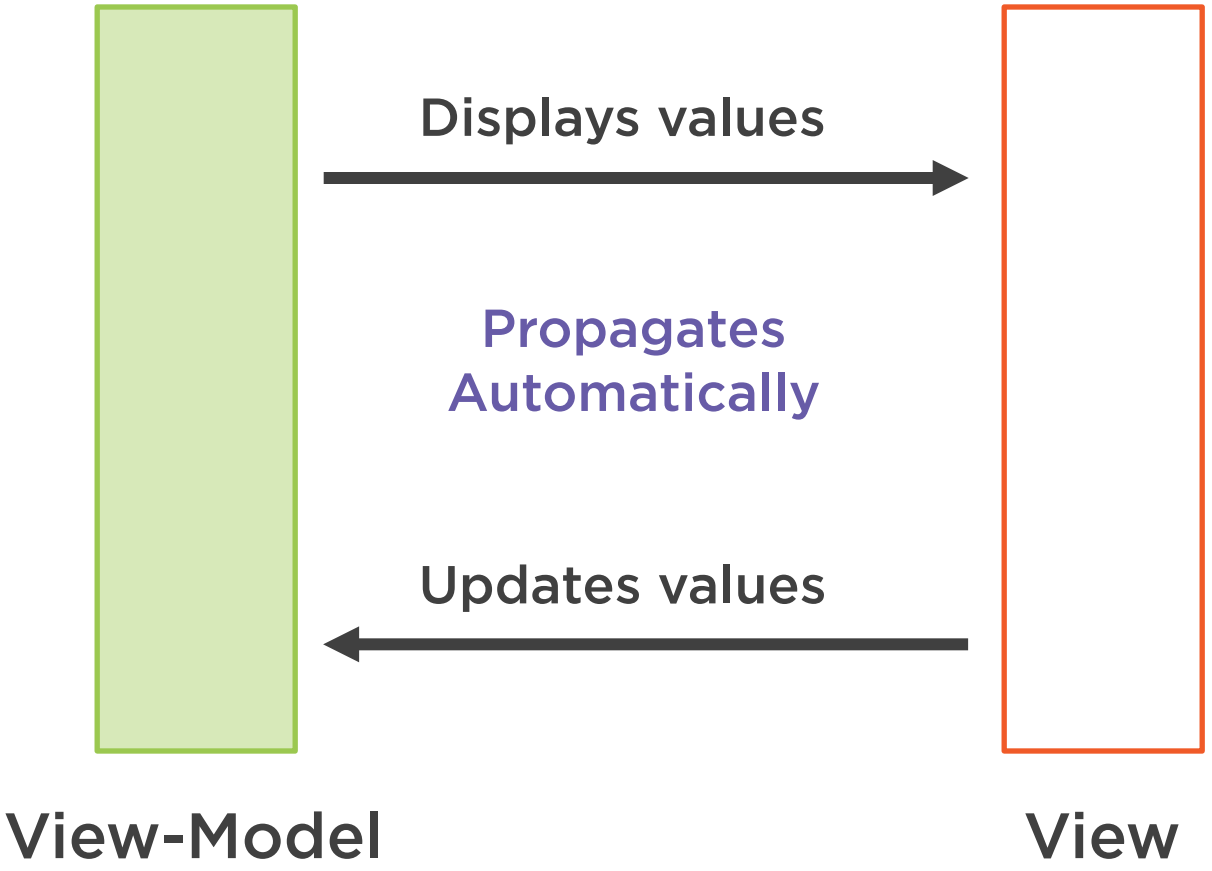


MVVM

MVVM


Shapes for display
Uses
Updates values
Commands
(UI events)
Service
Model
View-Model
View

Databinding



INotifyPropertyChanged

```
public class RecipeViewModel : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;

    string _recipeName;
    public string RecipeName
    {
        get { return _recipeName; }
        set
        {
            if (_recipeName == value)
                return;

            _recipeName = value;
            OnPropertyChanged();
        }
    }

    public void OnPropertyChanged([CallerMemberName]string memberName = "") =>
        PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(memberName));
}
```

```
<Label Text="{Binding RecipeName}" />

...

<Label Text="{Binding Time, StringFormat='{0} prep'}" />
```

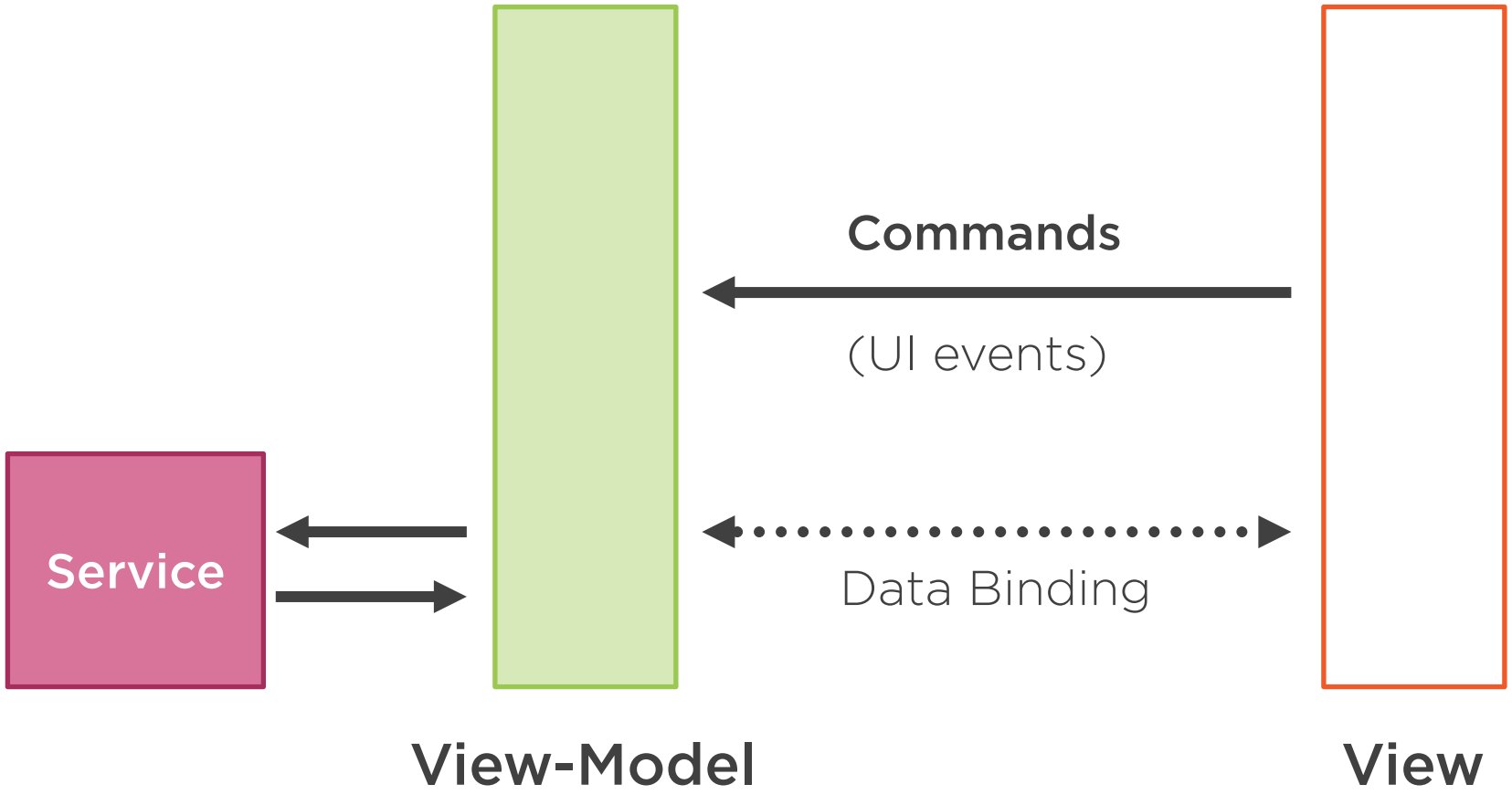
Data Binding - XAML

BindingContext set on parent or control

Binding keyword

Reference view model property

Commanding



```
public ICommand AddRecipeCommand { get; } =
    new Command(async (obj) => await Navigation.PushModalAsync(
        new NavigationPage(new EditRecipePage())))
    );
```

Command – View Model

ICommand interface

Action when invoked

Optional bool function for enabling

```
<Button Text="Add" Command="{Binding AddRecipeCommand}" />
```

Command - XAML

Command property

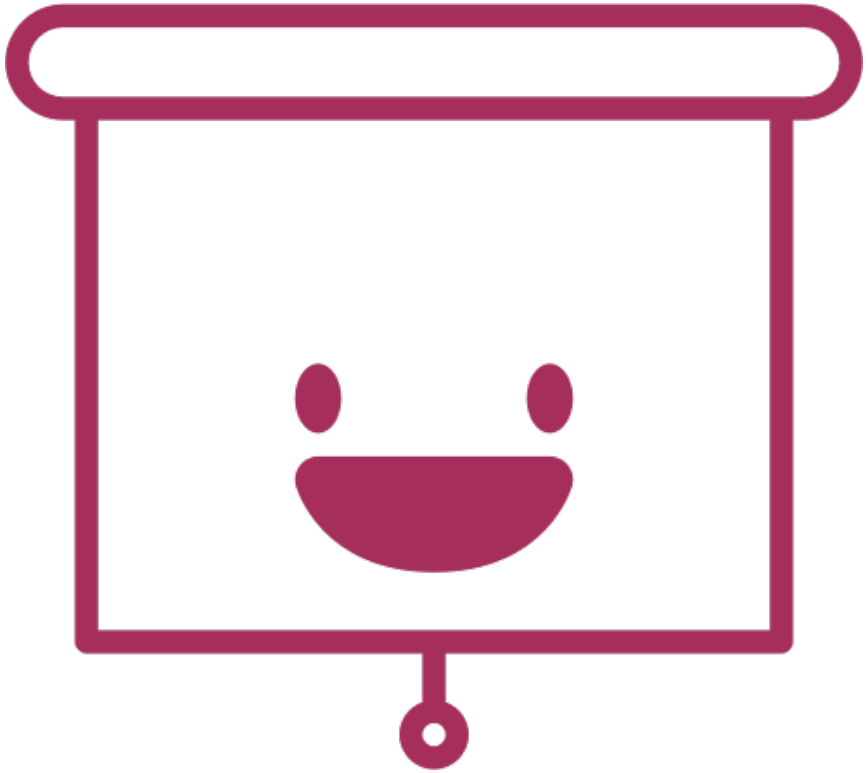
Binding keyword

One command per control

MVVM Demo

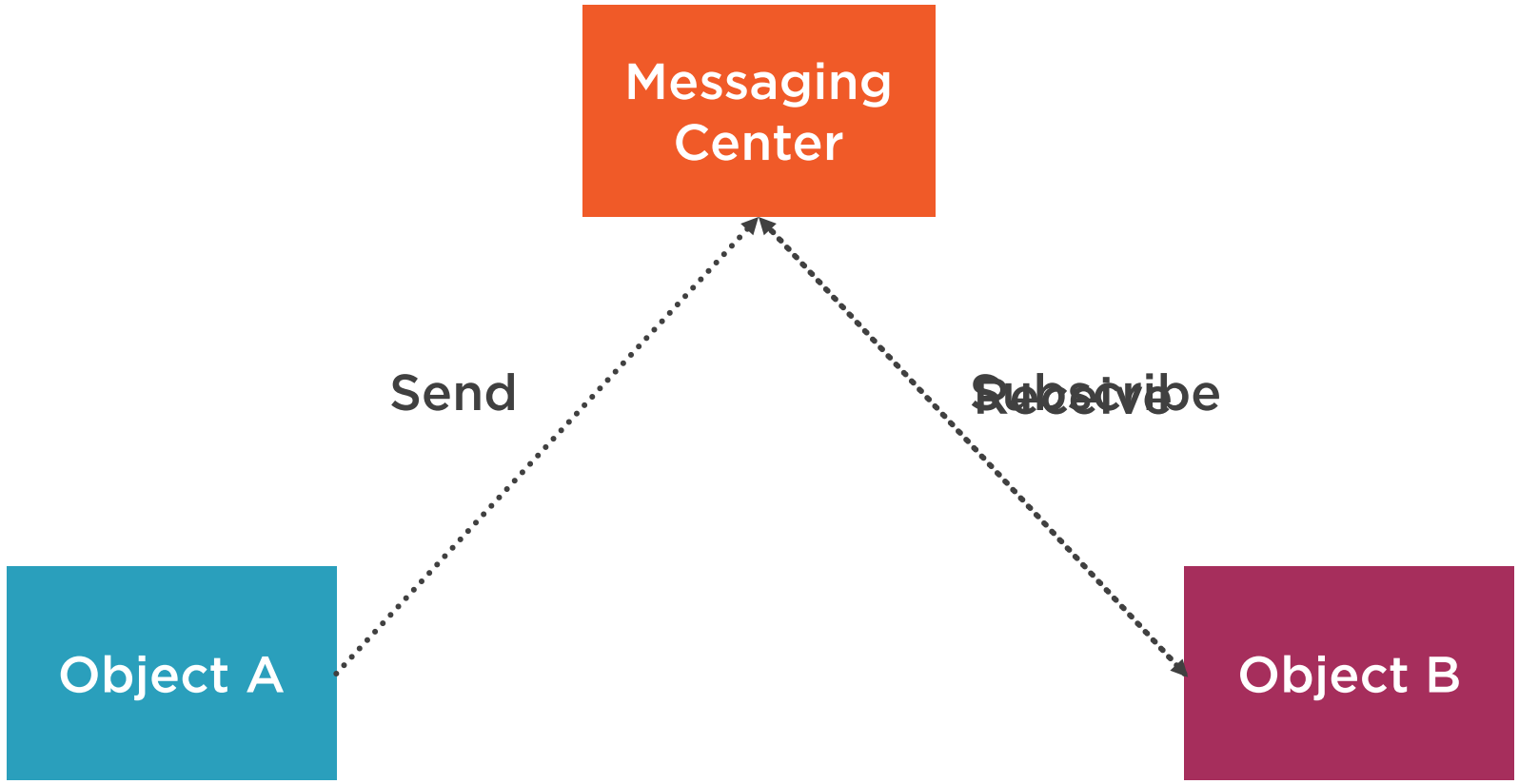
Data binding

Commanding



Additional Features & Platform Access

Messaging Center



Dependency Service

Platform Specific Functionality in Shared Code

Define interface

Implement on each platform

Register with DependencyService

Shared code calls DependencyService

```
public interface IFormsCamera
{
    void LaunchCamera();
}
```

Dependency Service

Define the interface

```
[assembly: Dependency(typeof(AppleCamera))]
namespace Foodie.iOS
{
    public class AppleCamera : IFormsCamera
    {
        public void LaunchCamera()
        {
            // Some camera stuff
        }
    }
}
```

Dependency Service

Platform implementation

```
public void TakePhoto()
{
    var camera = DependencyService.Get<IFormsCamera>();
    camera.LaunchCamera();
}
```

Dependency Service - Consume

DependencyService static object

Bindable Native Views

Native defined control

Referenced in XAML

Bindable properties

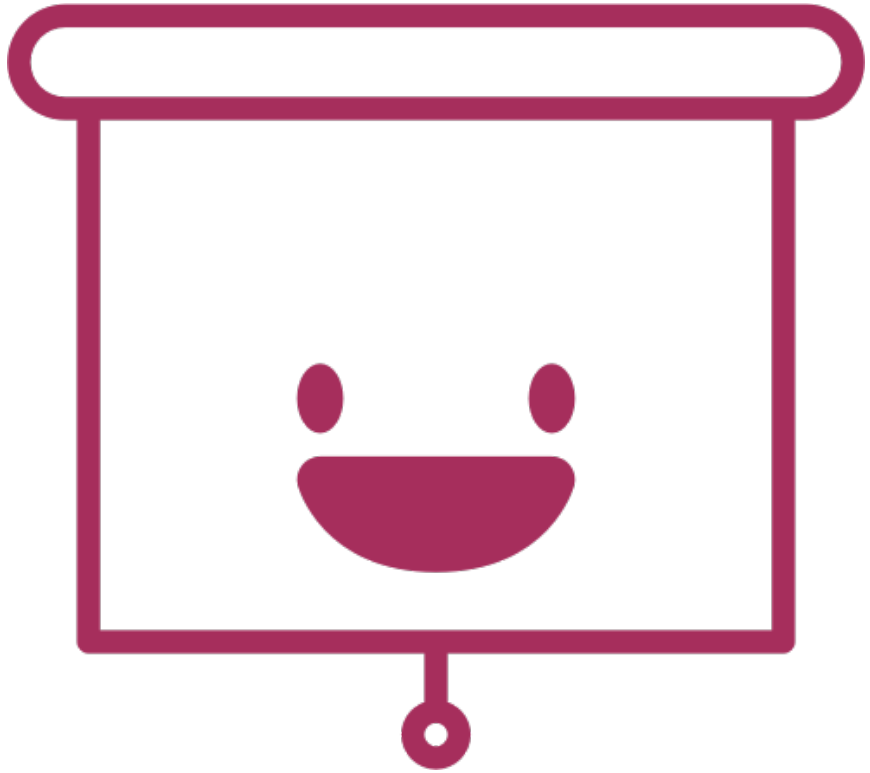
Can handle native events

Native View Demo

Referencing

Binding

MessagingCenter



Xamarin.Forms

User Interface

- Pages
- Navigation
- Layouts
- Controls

MVVM

- Data binding
- Commanding

Platform Access

- Messaging Center
- Dependency Service
- Bindable Native Views

Learn More!

Pluralsight: Moving Beyond the Basics with Xamarin.Forms

https://www.pluralsight.com/courses/xamarin-forms-moving-beyond-basics

Creating Three Beautiful Apps at Once

INTRODUCTION TO XAMARIN.FORMS

Matthew Soucoup
PRINCIPAL

@codemillmatt codemilltech.com