

# How to make Smoothie !



We want...

- × Find the perfect smoothie
- × Be rich
- × Not the hard way the smart way

Something cool !



!

# The smoothIA

Tell the perfect smoothie for YOU





**Here is Melissa**  
She love smoothie  
with **raspberry**  
but she is **allergic** to **lemon**

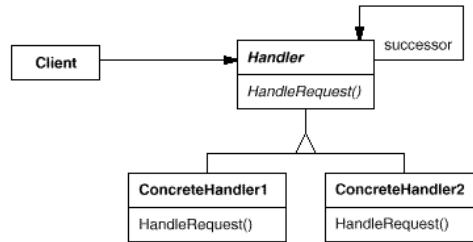
# Ready ?

# Chain of responsability

“

Chain :

Avoid coupling the sender of  
a... blablabla ...object handles it.



**BORING**





Big title really fun so you are focus

**Input aka  
Sender**

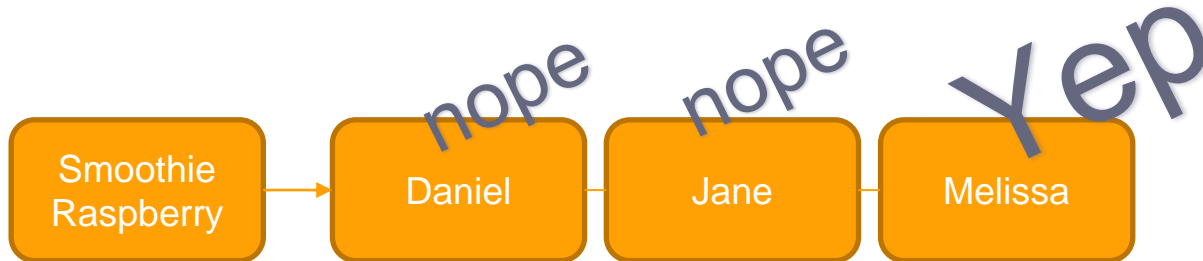
A smoothie

**People aka  
recievers**

Daniel,  
Jane,  
Melissa

**Output**

Does the  
smoothie has  
been choosen ?



Another title really catchy, go !

**Input aka  
Sender**

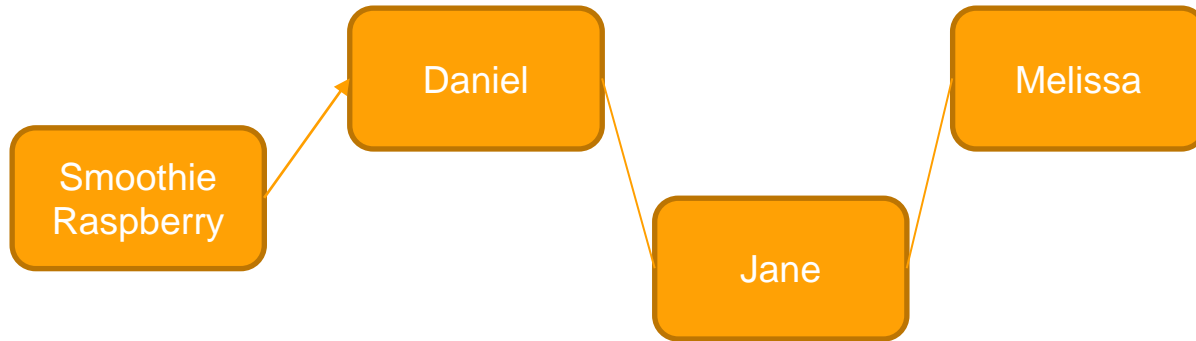
A smoothie

**People aka  
recievers**

Daniel,  
Jane,  
Melissa

**Output**

Does the  
smoothie has  
been choosen ?



# Another title really catchy, go !

## Input aka Sender

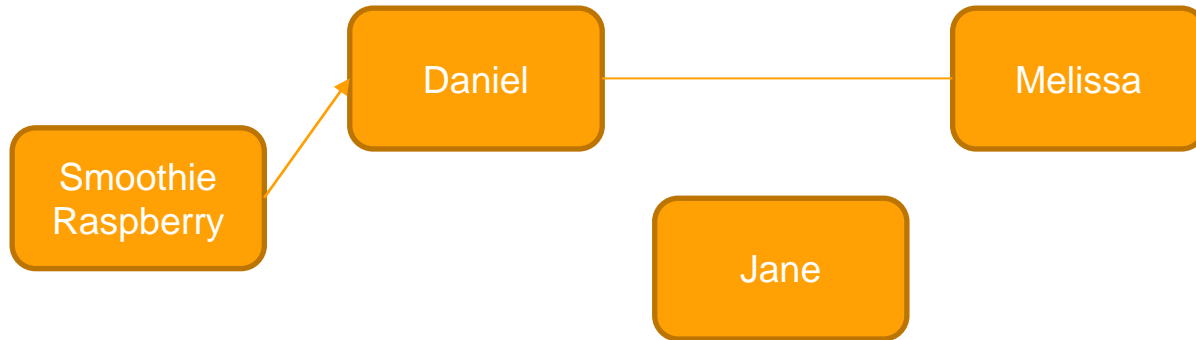
A smoothie

## People aka recievers

Daniel,  
Jane (died),  
Melissa

## Output

Does the  
smoothie has  
been choosen ?



Because she didn't take our smoothie...

The chain is  
dynamic

The slide is decorated with various hand-drawn illustrations of fruits and vegetables. At the top left is a blue bubble. Next to it is a slice of orange. To the right is a green vegetable, possibly a cucumber. Further right is a slice of watermelon. On the far right is a green lime. Below the orange slice is a green circle containing two black quotation marks. In the center, the text is written in a casual, handwritten style. To the left of the text is a yellow lemon. Below the lemon is a green leaf. At the bottom left is a strawberry. Next to it is a banana. In the bottom center is a green leaf. To the right of the leaf is a green vegetable, possibly a cucumber. At the bottom right is a cherry. Above the cherry is a small green leaf. To the left of the cherry is a slice of orange. Above the slice of orange is a green lime. At the top right is a slice of watermelon.

“

So you say...

You can **encapsulate** receivers  
behaviours, right ?

And change it at runtime, isn't it ?

The slide is decorated with various hand-drawn fruit illustrations in a whimsical style. At the top left is a blue bubble. Below it is a yellow lemon. To the right of the lemon is a blueberry. Further right is an orange slice. At the top center is a green lime. To the right of the lime is a watermelon slice. At the top right is a green lime. Below it is a green lime. To the right of the lime is a yellow lemon slice. At the bottom left is a strawberry. Below it is a banana. To the right of the banana is a green lime. At the bottom center is a green lime. To the right of the lime is a cherry. At the bottom right is an orange.

“

So it's like a pipeline ?

YES



So it like a "dynamic" pipeline ?

YES !

The slide is decorated with various hand-drawn illustrations of fruits and vegetables. At the top left is a blue bubble. Next to it is a slice of orange. To the right is a green vegetable, possibly a cucumber. Further right is a slice of watermelon. On the far right is a green lime. In the center, above the first text, is a green circle containing two black quotation marks. Below this, the text "Easy I got it !" is written in a casual, handwritten font. Underneath that, the text "Can you show me some code ?" is written in the same font. On the left side, there is a yellow lemon, a green leaf, and a strawberry. At the bottom left is a banana. In the bottom center is a green vegetable. To the right of the center is a slice of yellow fruit. At the bottom right is a cherry and an orange.

“

Easy I got it !

Can you show me some code ?





```
[TestMethod]
public void Melissa_Like_Raspberry()
{
    var smoothie = SmoothieFactory.Get("Raspberry");

    var smoothieLover = new ChainOfSmoothieLovers()
        .Of("Daniel")
        .Then("Jane")
        .Then("Melissa") //Melissa like Raspberry
        .GetHead();

    bool hasBeenChosen = smoothieLover.Recieve(smoothie);

    Assert.IsTrue(hasBeenChosen);
}
```

```
internal class SmoothieFactory
{
    internal static Smoothie Get(string name)
    {
        return new Smoothie(name);
    }
}
```



```
[TestMethod]
public void Melissa_Like_Raspberry()
{
    var smoothie = SmoothieFactory.Get("Raspberry");

    var smoothieLover = new ChainOfSmoothieLovers()
        .Of("Daniel")
        .Then("Jane")
        .Then("Melissa") //Melissa like Raspberry
        .GetHead();

    bool hasBeenChosen = smoothieLover.Recieve(smoothie);

    Assert.IsTrue(hasBeenChosen);
}
```



```
internal ChainOfSmoothieLovers Then(string name)
{
    Tail.SetNextSmoothieLover(smoothieLovers.Get(name));
}
```





```
internal class SmoothieLoversFactory
{
    Dictionary<string, SmoothieLover> dict = new Dictionary<string, SmoothieLover>();
    public SmoothieLoversFactory()
    {
        dict.Add("Melissa", new Melissa());
        dict.Add("Jane", new Jane());
        dict.Add("Daniel", new Daniel());
    }
    internal SmoothieLover Get(string name)
    {
        return dict[name];
    }
}
```



```
internal ChainOfSmoothieLovers Then(string name)
{
    Tail.SetNextSmoothieLover(smoothieLovers.Get(name));
}
```



```
public abstract class SmoothieLover
{
    protected SmoothieLover NextSmoothieLover;

    public void SetNextSmoothieLover(SmoothieLover next)
    {
        this.NextSmoothieLover = next;
    }

    protected abstract bool Check(Smoothie smoothie);
    protected abstract bool Excute(Smoothie smoothie);
    public virtual bool Recieve(Smoothie smoothie)
    {
        if (Check(smoothie))
            return Excute(smoothie);

        if (NextSmoothieLover != null)
            return NextSmoothieLover.Recieve(smoothie);

        return false;
    }
}
```





```
public class Melissa : SmoothieLover
{
    protected override bool Check(Smoothie smoothie)
    {
        return smoothie.Name == "Raspberry";
    }

    protected override bool Excute(Smoothie smoothie)
    {
        Console.WriteLine($"I'm Melissa and this smoothie is mine : {smoothie.Name}");
        return true;
    }
}
```





```
[TestMethod]
public void Melissa_Like_Raspberry()
{
    var smoothie = SmoothieFac
    var smoothieLover = new Ch
        .Of("Daniel")
        .Then("Jane")
        .Then("Melissa") //MeI
        .GetHead();

    bool hasBeenChosen = smoo

    Assert.IsTrue(hasBeenChosen
}
```

**NAILED IT**

berry

is mine : Raspberry

memegenerator.net

The chain  
is...

# Single responsability



The chain  
is...

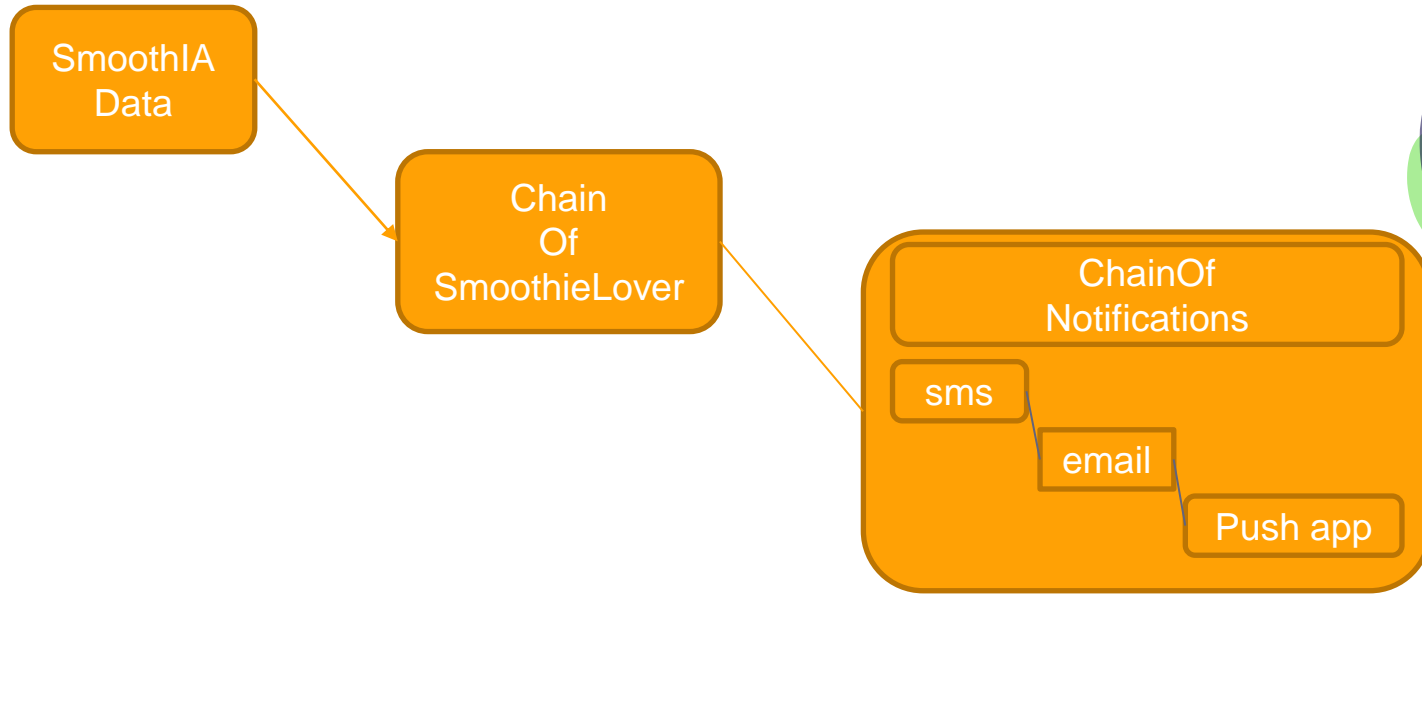
The background is white and decorated with various colorful, hand-drawn fruit illustrations. These include a blueberry cluster in the top left, an orange slice at the top center, a green kiwi slice at the top right, a watermelon slice in the upper right, a lime in the middle right, a lemon slice in the lower right, a cherry in the bottom right, an orange in the bottom right, a green kiwi slice in the bottom center, a banana in the bottom left, a strawberry in the lower left, a green leaf in the middle left, and a yellow lemon in the upper left.

# Open/Close

You can...

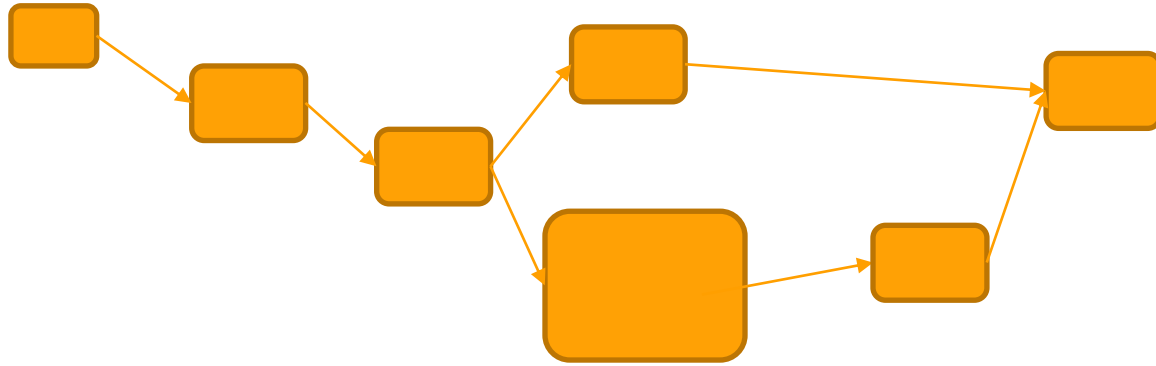
# Chainception

# Chain in a chain in a chain....

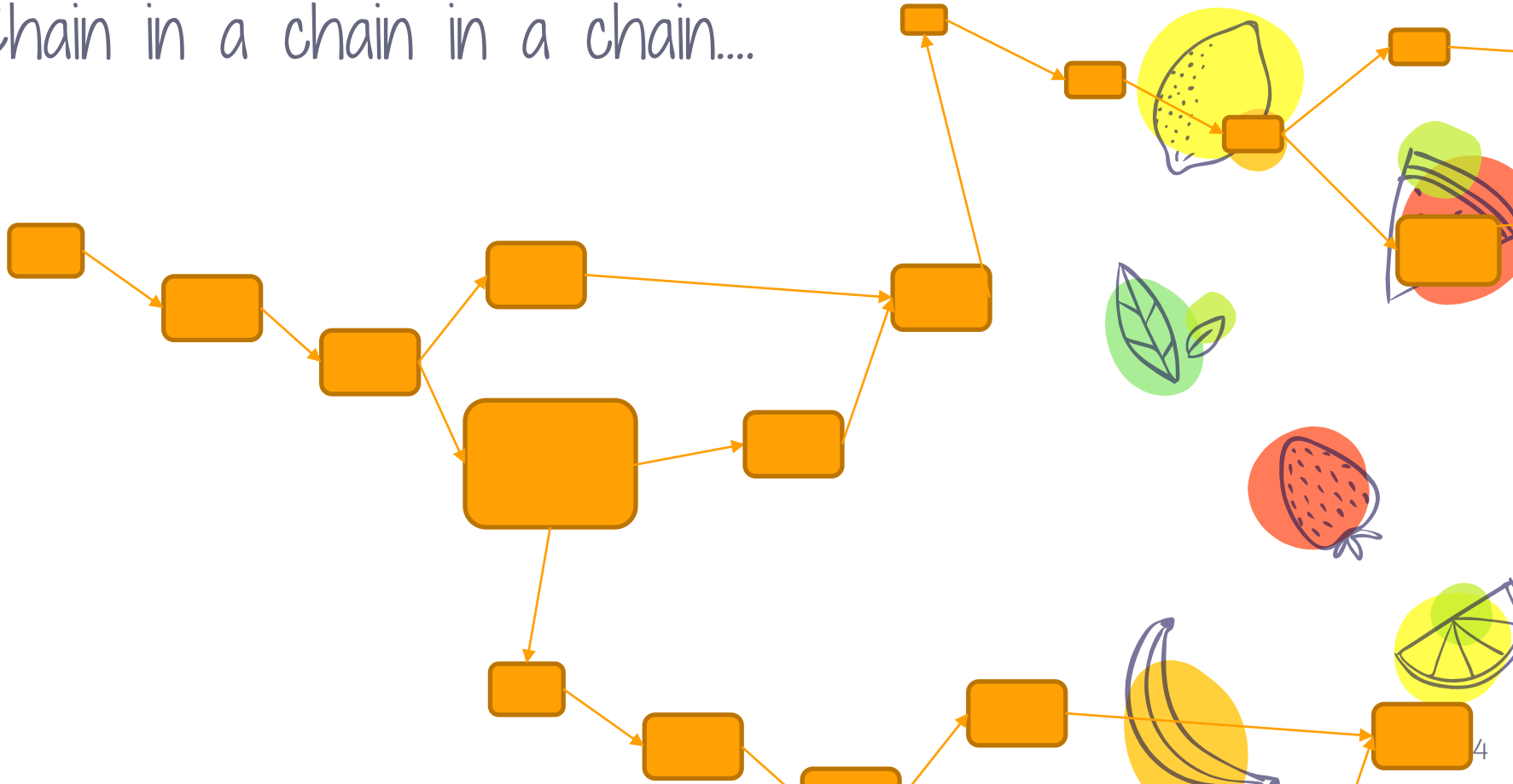




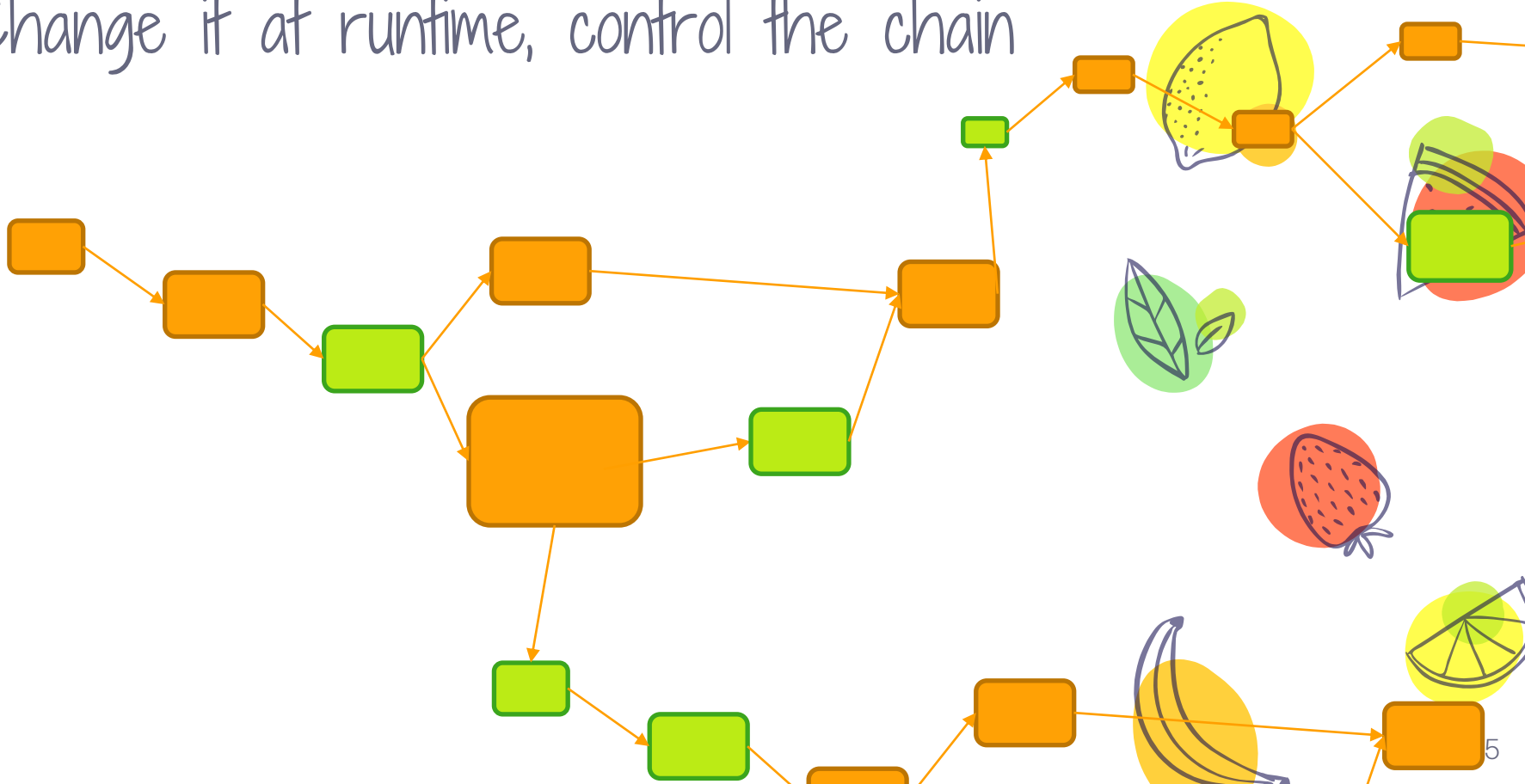
Chain in a chain in a chain....



Chain in a chain in a chain....



Change it at runtime, control the chain



The background of the slide is decorated with various colorful illustrations of fruits and vegetables. At the top left, there are blueberries and a slice of orange. To the right of the orange is a green kiwi. Further right is a slice of watermelon. On the far right is a green pear. Below the watermelon is a slice of lemon. On the left side, there is a whole yellow lemon. Below that is a green leaf. At the bottom left is a whole strawberry. In the bottom center is a banana. To the right of the banana is a green kiwi. At the bottom right is a whole orange and a small red cherry with a green leaf. In the center of the slide, above the text, is a green circle containing two black quotation marks.

What is the perfect smoothie for you ?

The slide features a white background decorated with various hand-drawn illustrations of fruits and leaves. These include a blueberry cluster in the top left, an orange slice at the top center, a green kiwi slice at the top right, a watermelon slice in the upper right, a lime in the middle right, a lemon slice in the lower right, a strawberry in the bottom left, a banana at the bottom center, a cherry in the bottom right, and several green leaves scattered throughout. The word 'Smoothie' is written in a large, orange, handwritten-style font in the center.

# Smoothie

**Find your smoothie.**

You can find me at:

Twitter : @manu\_the\_chene

Email : [emmanuel.conrardy@arolla.fr](mailto:emmanuel.conrardy@arolla.fr)

# Credits

Special thanks to all the people who made smoothIA:

- × Presentation template by [SlidesCarnival](#)
- × Photographs by [Unsplash](#)



# Links

My best smoothie :

<https://www.750g.com/smoothie-aux-framboises-r52312.htm>

Boring but not so bad :

<http://codenuclear.com/chain-of-responsibility-design-pattern-in-java/>

Response to the main title:

<https://www.wikihow.com/Make-a-Smoothie>

# How to make Smoothie !

