

# Creating Complications with ClockKit

Session 209

Eliza Block watchOS Engineer

Paul Salzman watchOS Engineer















Extension




Extension





Extension

 11:00AM

Haight – Twin Peaks  
Group D



A horizontal timeline on a black background. A white line runs across the middle. Four vertical white tick marks are placed along this line. Below each tick mark is a text block. Each text block starts with a blue soccer ball icon, followed by a time in blue text, and then two lines of white text. The events are: 11:00AM Haight – Twin Peaks Group D, 1:30PM Bernal – Presidio Group C, 4:00PM SOMA – Castro Group A, and 6:00PM Sunset – Marina Group B.

⚽ 11:00AM  
Haight – Twin Peaks  
Group D

⚽ 1:30PM  
Bernal – Presidio  
Group C

⚽ 4:00PM  
SOMA – Castro  
Group A

⚽ 6:00PM  
Sunset – Marina  
Group B

⚽ 11:00AM  
Haight – Twin Peaks  
Group D



⚽ 4:00PM  
SOMA – Castro  
Group A

⚽ 6:00PM  
Sunset – Marina  
Group B

aks

1:30PM  
Bernal – Presidio  
Group C



6:00PM  
Sunset – Marina  
Group B



o

|

⚽ 4:00PM  
SOMA – Castro  
Group A



WED 10 6:15

⚽ 6:00PM  
Sunset – Marina  
Group B

8:27 🌑 68°



Time Travel



Time Travel

# Agenda

# Agenda

Getting Started



# Agenda

Getting Started

Complication Layout

# Agenda

Getting Started

Complication Layout

Providing Images and Text

# Agenda

Getting Started

Complication Layout

Providing Images and Text

Timeline

# Agenda

Getting Started

Complication Layout

Providing Images and Text

Timeline

Complication Data Source

# Agenda

Getting Started

Complication Layout

Providing Images and Text

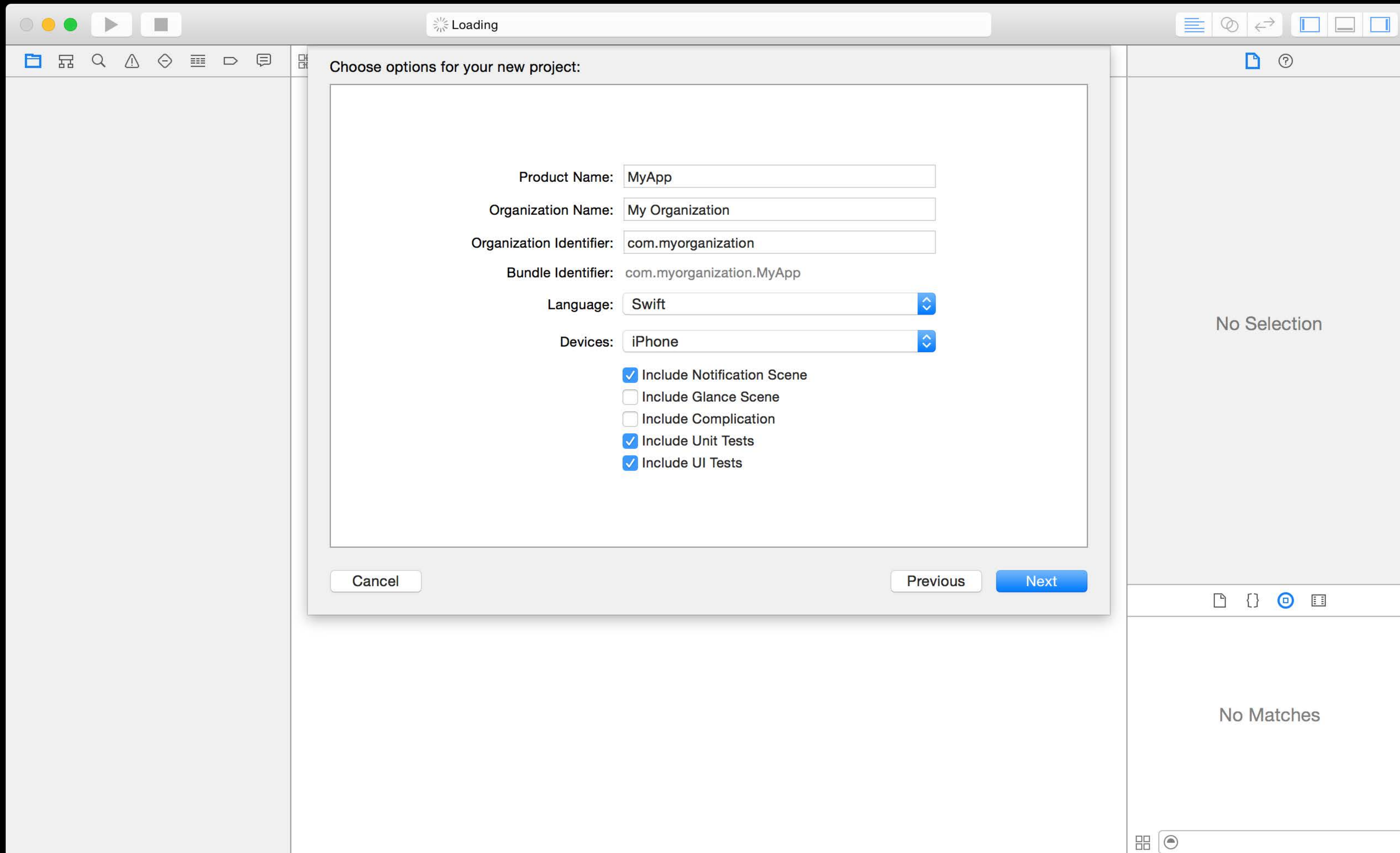
Timeline

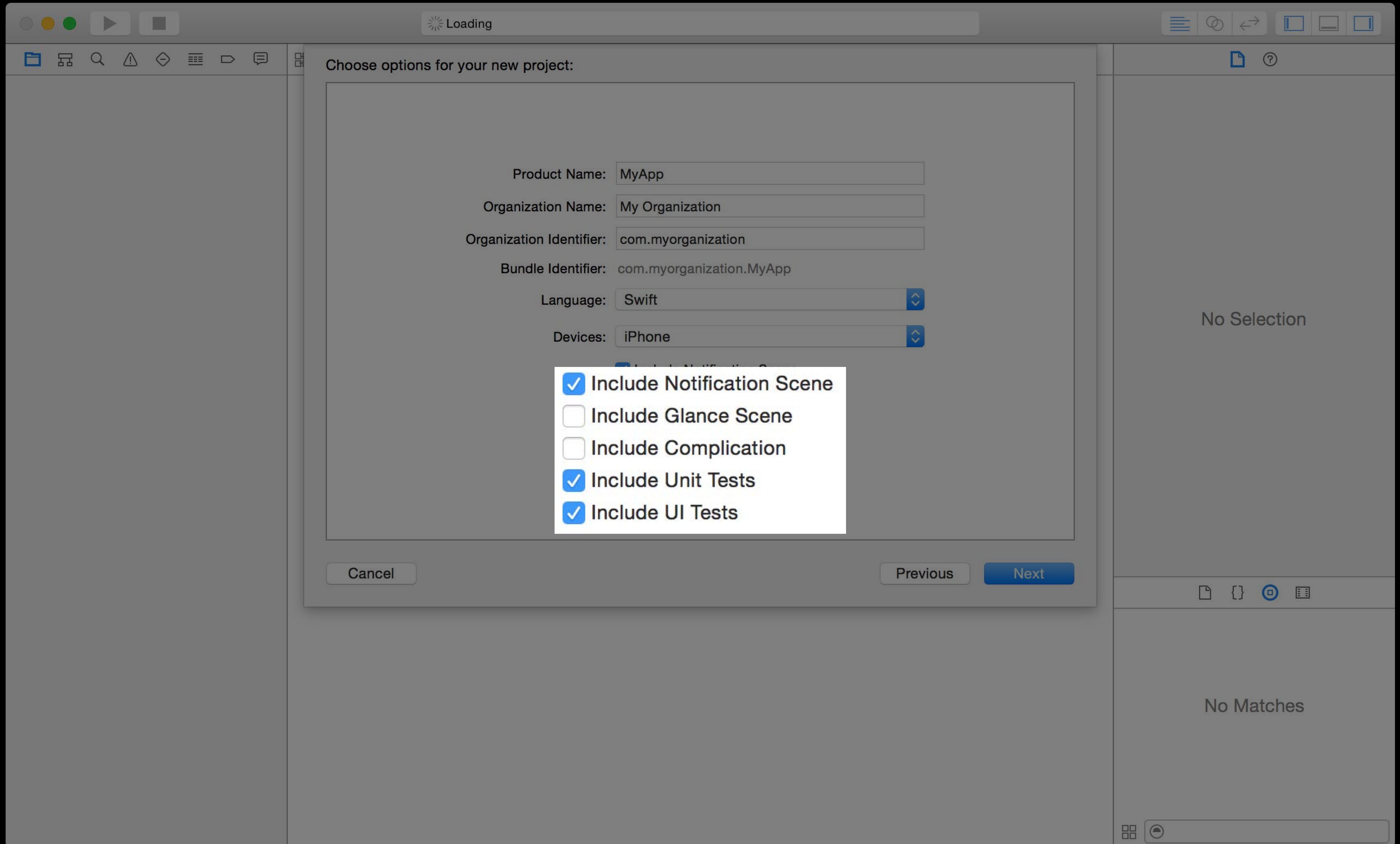
Complication Data Source

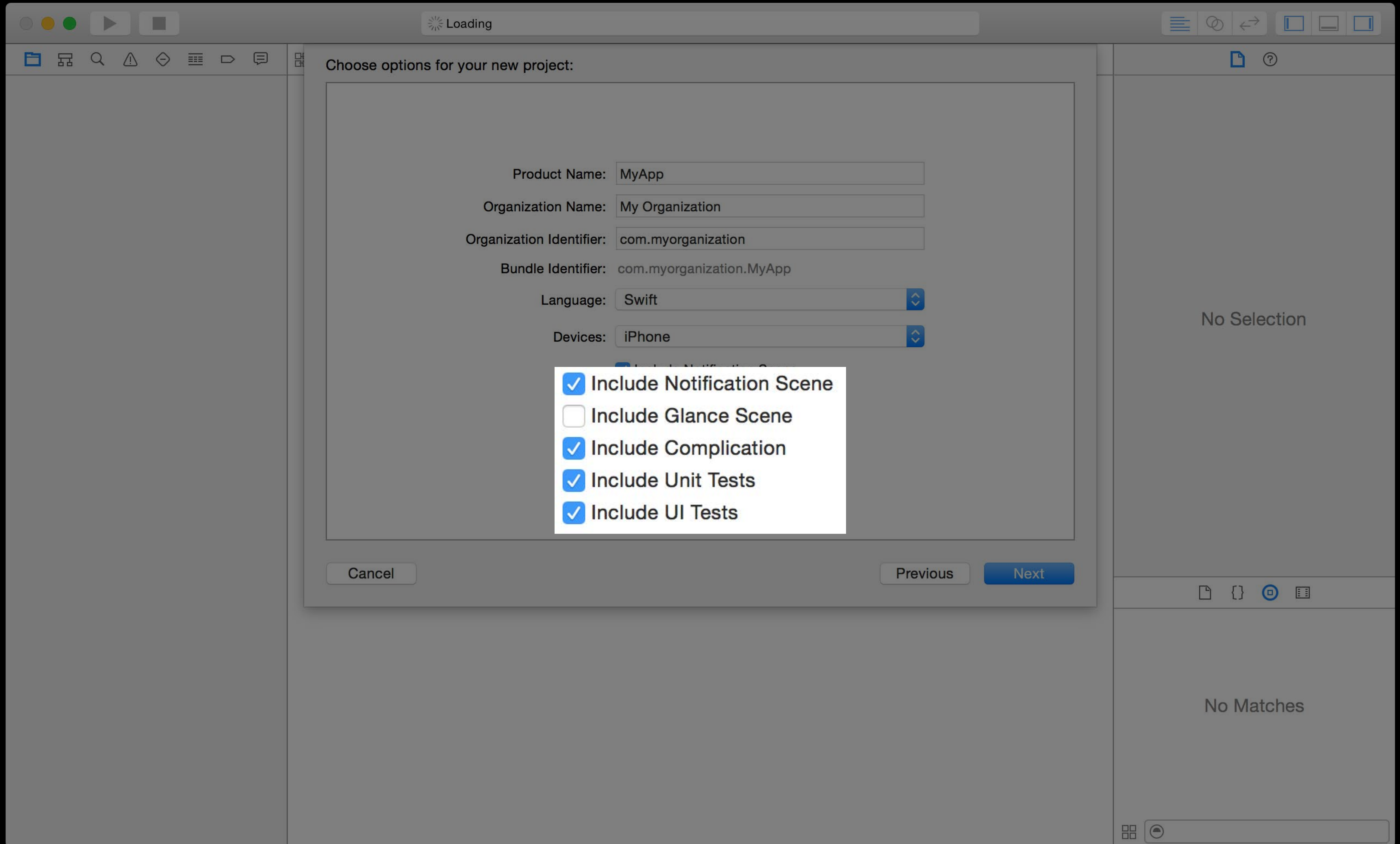
Keeping Your Data Fresh



# Getting Started









ⓧ MyApp > iPhone 6

MyApp: Ready | Today at 1:35 PM

☰

🔗

↔

🖼

📱

📺

📁 MyApp

📁 MyApp

📄 AppDelegate.swift

📄 ViewController.swift

📄 Main.storyboard

📁 Assets.xcassets

📄 LaunchScreen.storyboard

📄 Info.plist

📁 MyAppTests

📁 MyAppUITests

📁 MyApp WatchKit App

📁 MyApp WatchKit Extension

📁 Products

📁 MyApp

TARGETS

📱 MyApp

📁 MyAppTests

📁 MyAppUITests

📱 MyApp WatchKit...

📱 MyApp WatchKit...

General

Capabilities

Resource Tags

Info

Build Settings

Build Phases

Build Rules

▼ Identity

Bundle Identifier

com.myorganization.MyApp.watchkit

Version

1.0

Build

1

Team

None

▼ Complications Configuration

Data Source Class

ComplicationController

Supported

Data Source Class

ComplicationController

Supported Families

Modular Small

Modular Large

Utilitarian Small

Utilitarian Large

Circular Small

Complications Group

Complication

▼ Embedded Binaries

Add embedded binaries here

▼ Linked Frameworks and Libraries

Name

Status

Add frameworks & libraries here



MyApp iPhone 6

MyApp: Ready | Today at 1:35 PM

MyApp

MyApp

AppDelegate.swift

ViewController.swift

Main.storyboard

Assets.xcassets

LaunchScreen.storyboard

Info.plist

MyAppTests

MyAppUITests

MyApp WatchKit App

MyApp WatchKit Extension

Products

MyApp

MyApp

MyAppTests

MyAppUITests

MyApp WatchKit...

MyApp WatchKit...

General

Capabilities

Resource Tags

Info

Build Settings

Build Phases

Build Rules

PROJECT

MyApp

TARGETS

MyApp

MyAppTests

MyAppUITests

MyApp WatchKit...

MyApp WatchKit...

Identity

Bundle Identifier

com.myorganization.MyApp.watchkit

Version

1.0

Build

1

Team

None

Data Source Class

ComplicationController

Supported Families

Modular Small

Modular Large

Utilitarian Small

Utilitarian Large

Circular Small

Embedded Binaries

Add embedded binaries here

Linked Frameworks and Libraries

Name

Status

Add frameworks & libraries here

Data Source Class

ComplicationController

Supported Families

Modular Small

Modular Large

Utilitarian Small

Utilitarian Large

Circular Small

# Complication Families



# Complication Families

Modular



# Complication Families

Modular

Modular small





# Complication Families

Modular

Modular small

Modular large



# Complication Families

Utilitarian

Modular small

Modular large



# Complication Families

Utilitarian

Modular small

Modular large



# Complication Families

Utilitarian

Modular small

Modular large

Utilitarian small





# Complication Families

Utilitarian

Modular small

Modular large

Utilitarian small

Utilitarian large



# Complication Families

## Circular

Modular small

Modular large

Utilitarian small

Utilitarian large



# Complication Families

## Circular

Modular small

Modular large

Utilitarian small

Utilitarian large



# Complication Families

## Circular

Modular small

Modular large

Utilitarian small

Utilitarian large

Circular small



# Complication Families

Modular small

Modular large

Utilitarian small

Utilitarian large

Circular small


```
enum CLKComplicationFamily : Int {  
    case ModularSmall  
    case ModularLarge  
    case UtilitarianSmall  
    case UtilitarianLarge  
    case CircularSmall  
}
```

# Complication Layout


# CLKComplicationTemplate


## Modular Small


NYC  
5:54

  
8:21

75°







NY 2  
SF 0



# CLKComplicationTemplate

## Modular Small

NYC  
5:54

  
8:21

75°








NY 2  
SF 0

## Modular Large

12:00–2:00PM  
Lunch with Lucas  
Saru Sushi

 11:00AM  
Haight – Twin Peaks  
Group D

J inbound 1 min  
L outbound 4 min  
K inbound 11 min

Distant weekdays  
Dec 29, 0001 Fri  
Dec 31, 4000 Sun

Kitchen Timer  
5:17



# CLKComplicationTemplate

## Modular Small

NYC  
5:54

  
8:21

75°








NY 2  
SF 0

## Modular Large

12:00–2:00PM  
Lunch with Lucas  
Saru Sushi


 11:00AM  
Haight – Twin Peaks  
Group D


J inbound 1 min  
L outbound 4 min  
K inbound 11 min


Distant weekdays  
Dec 29, 0001 Fri  
Dec 31, 4000 Sun


Kitchen Timer  
5:17

## Utilitarian Small

 8:26PM









# CLKComplicationTemplate


## Modular Small


NYC  
5:54

  
8:21

75°








NY 2  
SF 0

## Modular Large

12:00–2:00PM  
Lunch with Lucas  
Saru Sushi


 11:00AM  
Haight – Twin Peaks  
Group D


J inbound 1 min  
L outbound 4 min  
K inbound 11 min


Distant weekdays  
Dec 29, 0001 Fri  
Dec 31, 4000 Sun


Kitchen Timer  
5:17

## Utilitarian Small


 8:26PM








## Utilitarian Large

 8:26PM 10HRS 8MINS


# CLKComplicationTemplate


Modular Small


NYC  
5:54

  
8:21

75°








NY 2  
SF 0

Modular Large

12:00–2:00PM  
Lunch with Lucas  
Saru Sushi


 11:00AM  
Haight – Twin Peaks  
Group D


J inbound 1 min  
L outbound 4 min  
K inbound 11 min


Distant weekdays  
Dec 29, 0001 Fri  
Dec 31, 4000 Sun


Kitchen Timer  
5:17

Utilitarian Small


 8:26PM







Utilitarian Large

 8:26PM 10HRS 8MINS

Circular Small

NYC  
1:47

  
8:26

65°







# CLKComplicationTemplate

⚽ 11:00AM

Haight – Twin Peaks

Group D

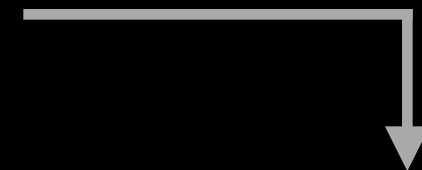
# CLKComplicationTemplate

⚽ 11:00AM

Haight – Twin Peaks  
Group D

# CLKComplicationTemplate

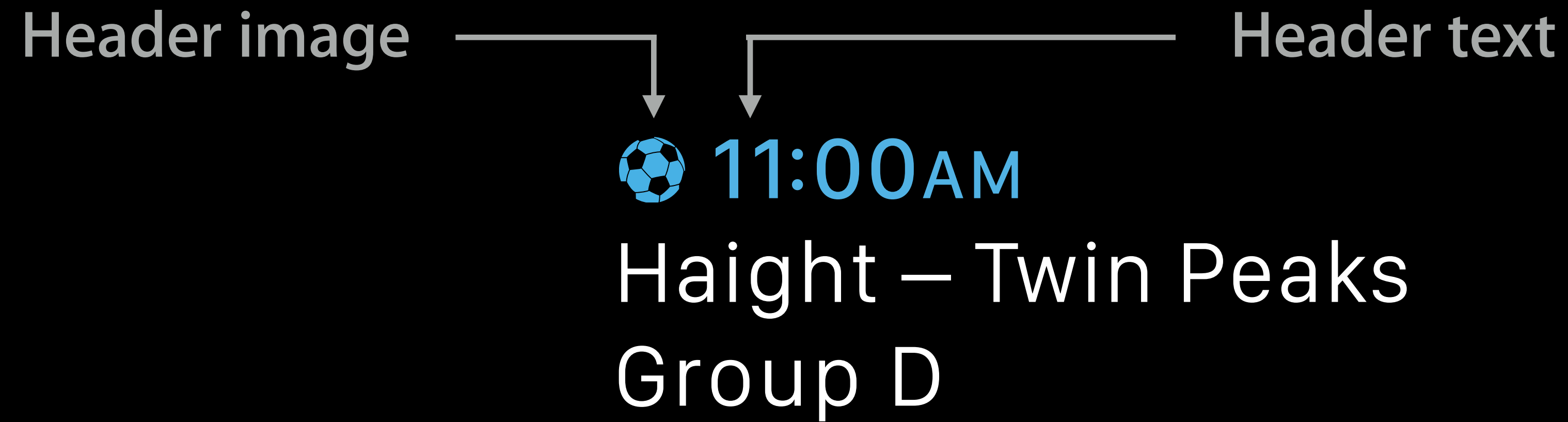
Header image



**11:00AM**

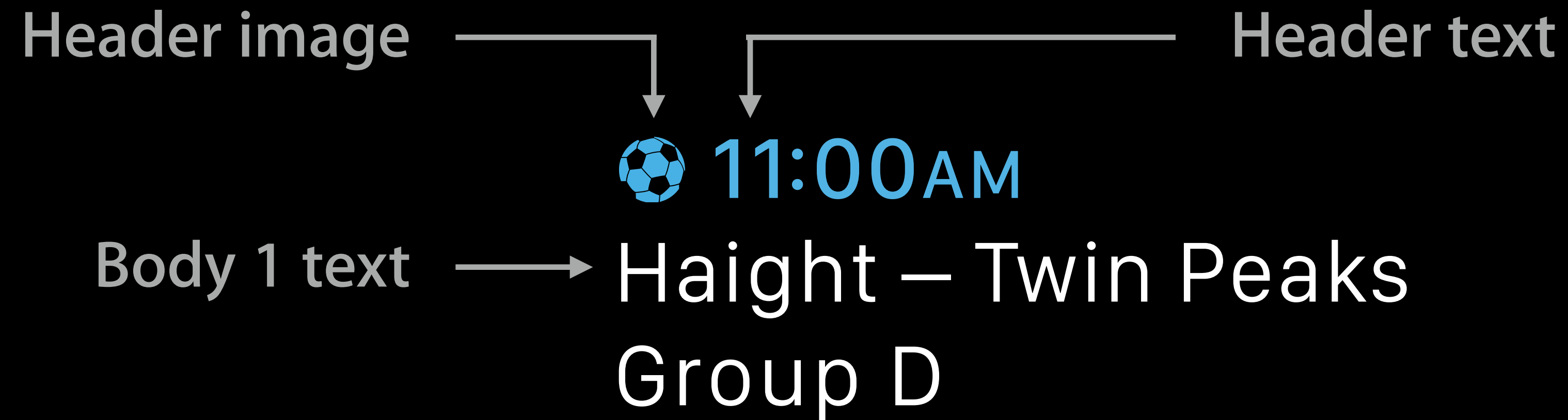
Haight – Twin Peaks  
Group D

# CLKComplicationTemplate



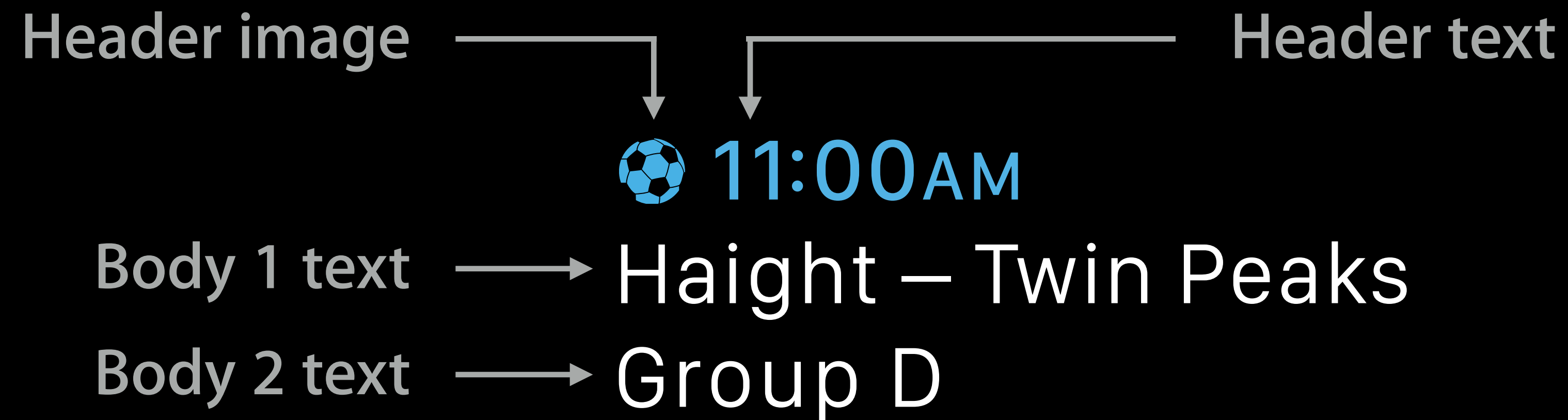


# CLKComplicationTemplate

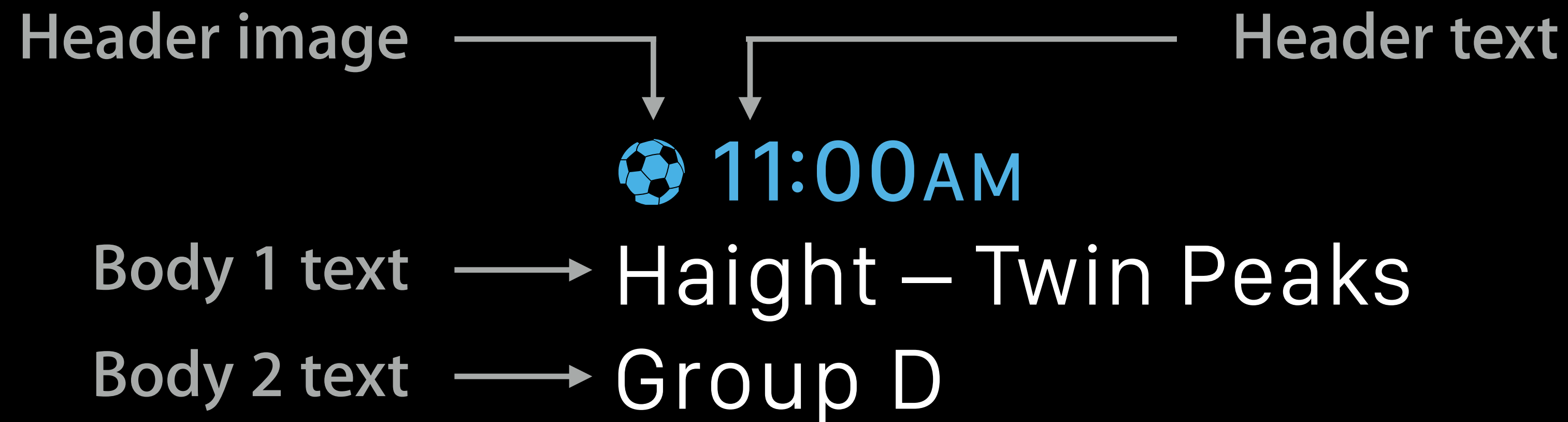




# CLKComplicationTemplate

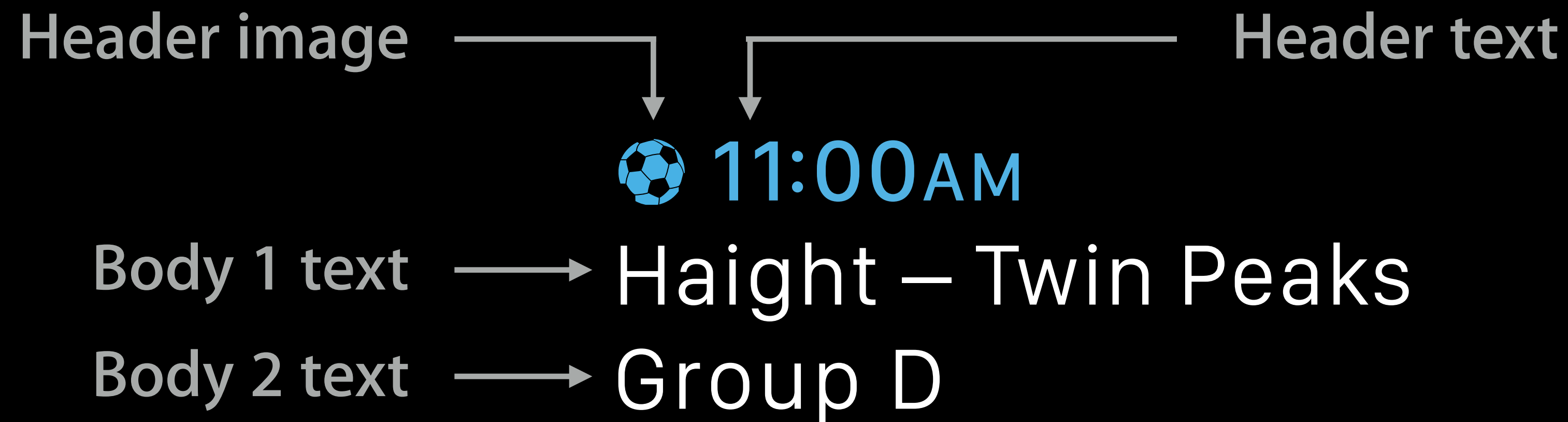


# CLKComplicationTemplate



```
class CLKComplicationTemplateModularLargeStandardBody {  
    var headerImageProvider: CLKImageProvider?  
    var headerTextProvider: CLKTextProvider  
    var body1TextProvider: CLKTextProvider  
    var body2TextProvider: CLKTextProvider?  
}
```

# CLKComplicationTemplate



```
class CLKComplicationTemplateModularLargeStandardBody {  
    var headerImageProvider: CLKImageProvider?  
    var headerTextProvider: CLKTextProvider  
    var body1TextProvider: CLKTextProvider  
    var body2TextProvider: CLKTextProvider?  
}
```

Providing Images and Text





# Image Providers

...not UIImages



# Image Providers

...not Ullimages



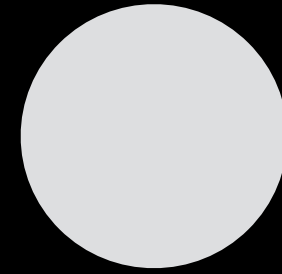
# Image Providers

...not Ullimages



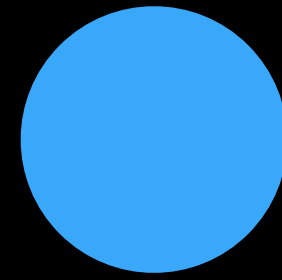
# Image Providers

...not Ullimages



# Image Providers

...not Ullimages



# Image Providers

...not Ullimages



# Image Providers

...not Ullimages



# Image Providers

...not UIImage



```
let bgImage = UIImage(named: "Background")
let fgImage = UIImage(named: "Foreground")

let imageProvider = CLKImageProvider(backgroundImage: bgImage,
                                     backgroundColor: aColor,
                                     foregroundImage: fgImage,
                                     foregroundColor: CLKImageProviderForegroundColor.White)
```



# Text Providers

...not NSStrings

# Text Providers

...not NSStrings

Space is very constrained

# Text Providers

...not NSStrings

Space is very constrained

Declare your intentions

# Text Providers

...not NSStrings

Space is very constrained

Declare your intentions

Let us handle formatting and fitting

# Example: Formatting Dates

CLKDateTextProvider



# Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23



# Example: Formatting Dates

CLKDateTextProvider

Wednesday, Se...





# Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23



# Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23

Wednesday, Sep 23



# Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23

Wednesday, Sep 23

Wed, Sep 23



# Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23

Wednesday, Sep 23

Wed, Sep 23

Sep 23



# Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23

Wednesday, Sep 23

Wed, Sep 23

Sep 23

23



# Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23

Wednesday, Sep 23

Wed, Sep 23

Sep 23

23

```
let sep23: NSDate = ...
```

```
let units = [NSCalendarUnit.Weekday,  
             NSCalendarUnit.Month,  
             NSCalendarUnit.Day]
```

```
textProvider = CLKDateTextProvider(sep23, units)
```





# Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23

Wednesday, Sep 23

Wed, Sep 23

Sep 23

23

```
let sep23: NSDate = ...
```

```
let units = [NSCalendarUnit.Weekday,  
             NSCalendarUnit.Month,  
             NSCalendarUnit.Day]
```

```
textProvider = CLKDateTextProvider(sep23, units)
```





# Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23

Wednesday, Sep 23

Wed, Sep 23

Sep 23

23

```
let sep23: NSDate = ...
```

```
let units = [NSCalendarUnit.Weekday,  
             NSCalendarUnit.Month,  
             NSCalendarUnit.Day]
```

```
textProvider = CLKDateTextProvider(sep23, units)
```



# Example: Formatting Dates

CLKDateTextProvider

Wednesday, September 23

Wednesday, Sep 23

Wed, Sep 23

Sep 23

23

```
let sep23: NSDate = ...
```

```
let units = [NSCalendarUnit.Weekday,  
             NSCalendarUnit.Month,  
             NSCalendarUnit.Day]
```

```
textProvider = CLKDateTextProvider(sep23, units)
```



# More Text Providers



# More Text Providers

Simple text provider



# More Text Providers

Simple text provider

Time text provider





# More Text Providers

Simple text provider

Time text provider



# More Text Providers

Simple text provider

Time text provider

Time interval text provider





# More Text Providers

Simple text provider

Time text provider

Time interval text provider



# Relative Dates



# Relative Dates



# Relative Dates



# Relative Dates



# Relative Dates





# Relative Dates



# Relative Dates





# Relative Date Text Provider

Waning crescent

Moonset 2:19PM

2HRS 56MINS

# Relative Date Text Provider

Waning crescent

Moonset 2:19PM

2HRS 56MINS

```
let moonset : NSDate = ... // 2:19pm
```

# Relative Date Text Provider

Waning crescent

Moonset 2:19PM

2HRS 56MINS

```
let moonset : NSDate = ... // 2:19pm
```

```
let units : NSCalendarUnit = [.Hour, .Minute]
```

# Relative Date Text Provider

Waning crescent

Moonset 2:19PM

2HRS 56MINS

```
let moonset : NSDate = ... // 2:19pm
```

```
let units : NSCalendarUnit = [.Hour, .Minute]
```

```
let style : CLKRelativeDateStyle = .Natural
```

# Relative Date Text Provider

Waning crescent  
Moonset 2:19PM  
2HRS 56MINS

[illegible]

# Relative Date Text Provider

Waning crescent  
Moonset 2:19PM  
2HRS 56MINS

```
let moonset : NSDate = ... // 2:19pm
let units : NSCalendarUnit = [.Hour, .Minute]
let style : CLKRelativeDateStyle = .Natural
let textProvider = CLKRelativeDateTextProvider(date: moonset,
                                                style: style,
                                                units: units)

template.body2TextProvider = textProvider
```

# Relative Date Text Provider

## Styles

Natural

2HRS 7MINS

13MINS 5SECS

3WKS 4DAYS

2YRS 3MOS

Offset

+14 SECONDS

-59 MINUTES

+2 HOURS

-6 DAYS

+12 WEEKS

Timer

02:32

21:15:59





NY 2  
SF 0



🌅 8:26PM

⚽ 11:00AM  
Haight – Twin Peaks  
Group D

CLKComplicationTemplate

NY 2  
SF 0



🌅 8:26PM

⚽ 11:00AM

Haight – Twin Peaks  
Group D

CLKComplicationTemplate



CLKImageProvider

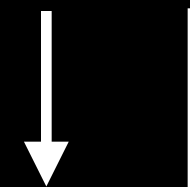
NY 2  
SF 0



🌅 8:26PM

⚽ 11:00AM  
Haight – Twin Peaks  
Group D

CLKComplicationTemplate



CLKImageProvider

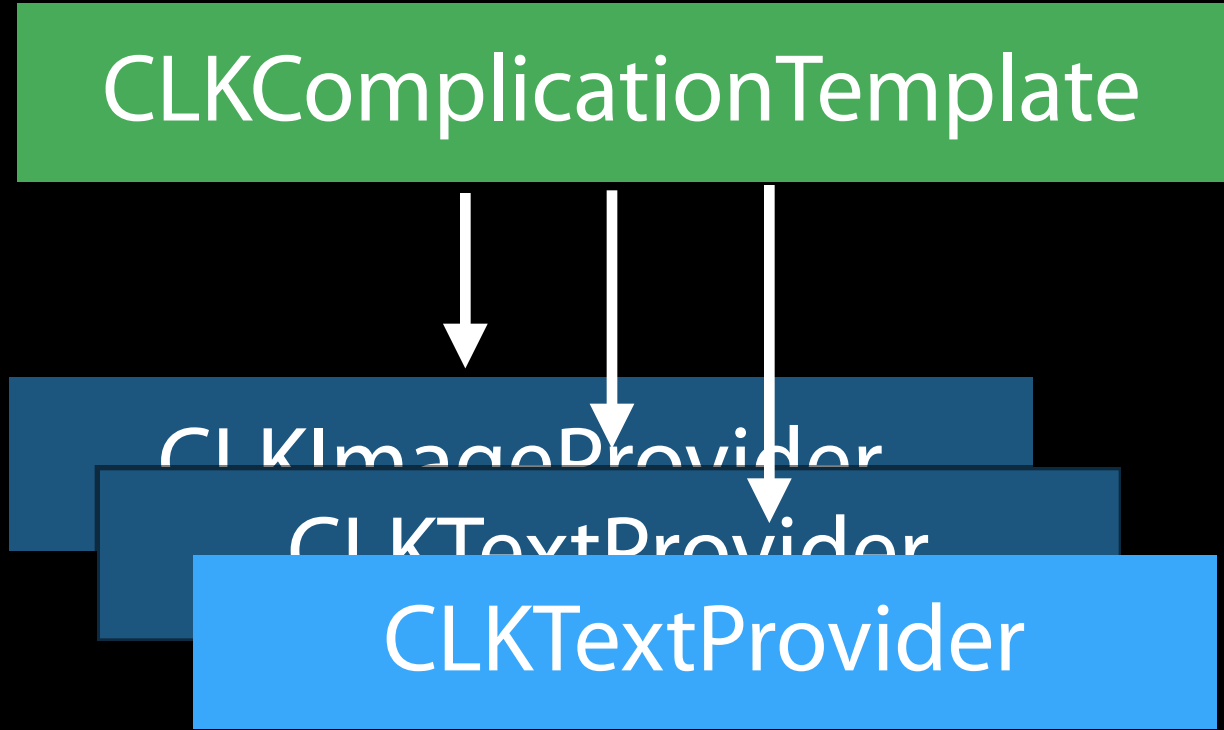
CLKTextProvider

NY 2  
SF 0



🌅 8:26PM

⚽ 11:00AM  
Haight – Twin Peaks  
Group D



# Timeline

Paul Salzman watchOS Engineer

# Timeline



# Timeline



57°

# Timeline

57°

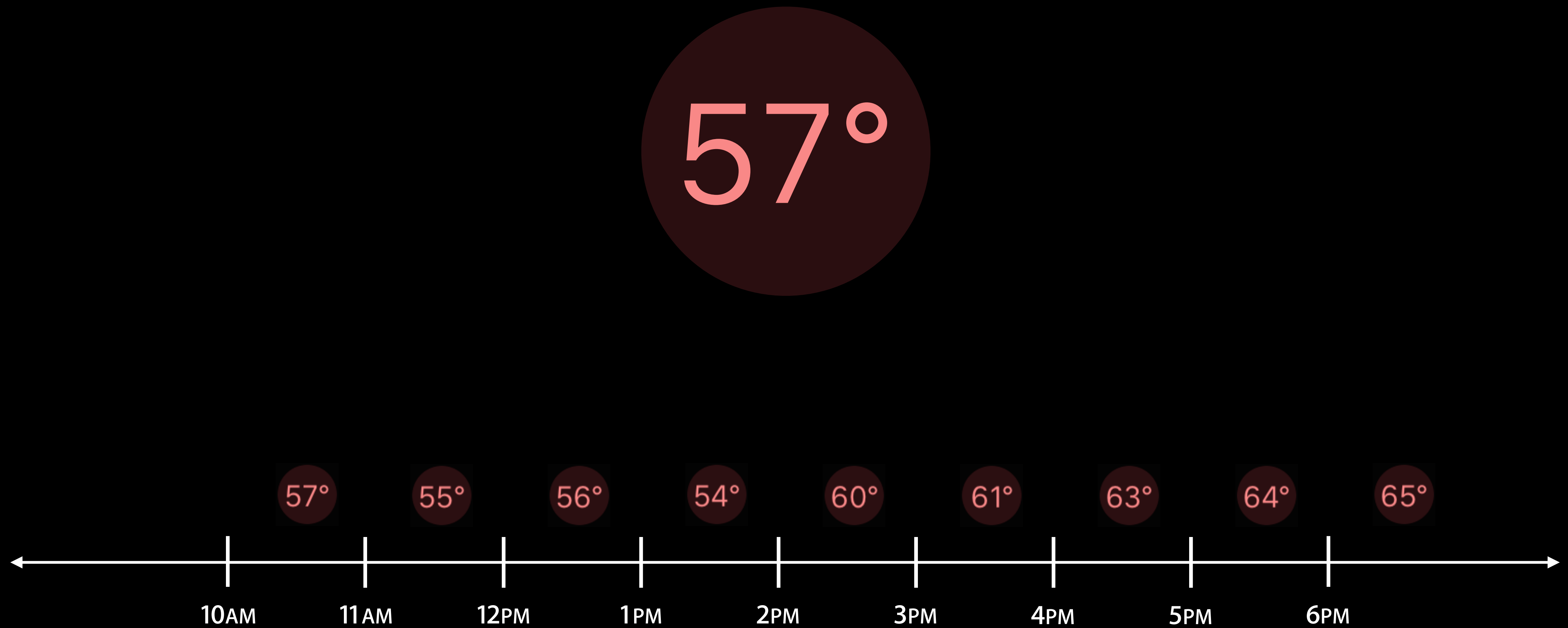
57°

10AM

11AM

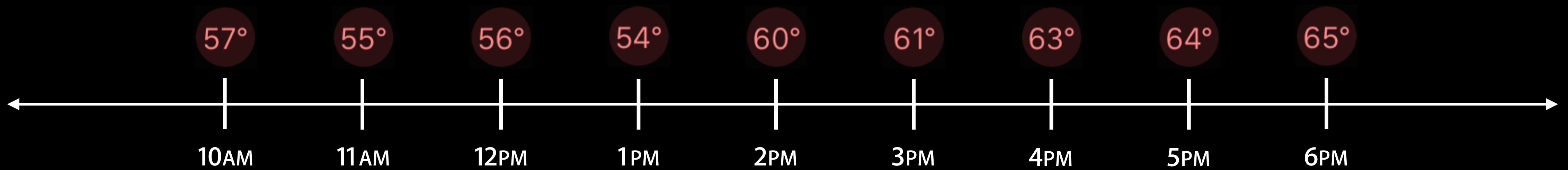


# Timeline



# Timeline

57°



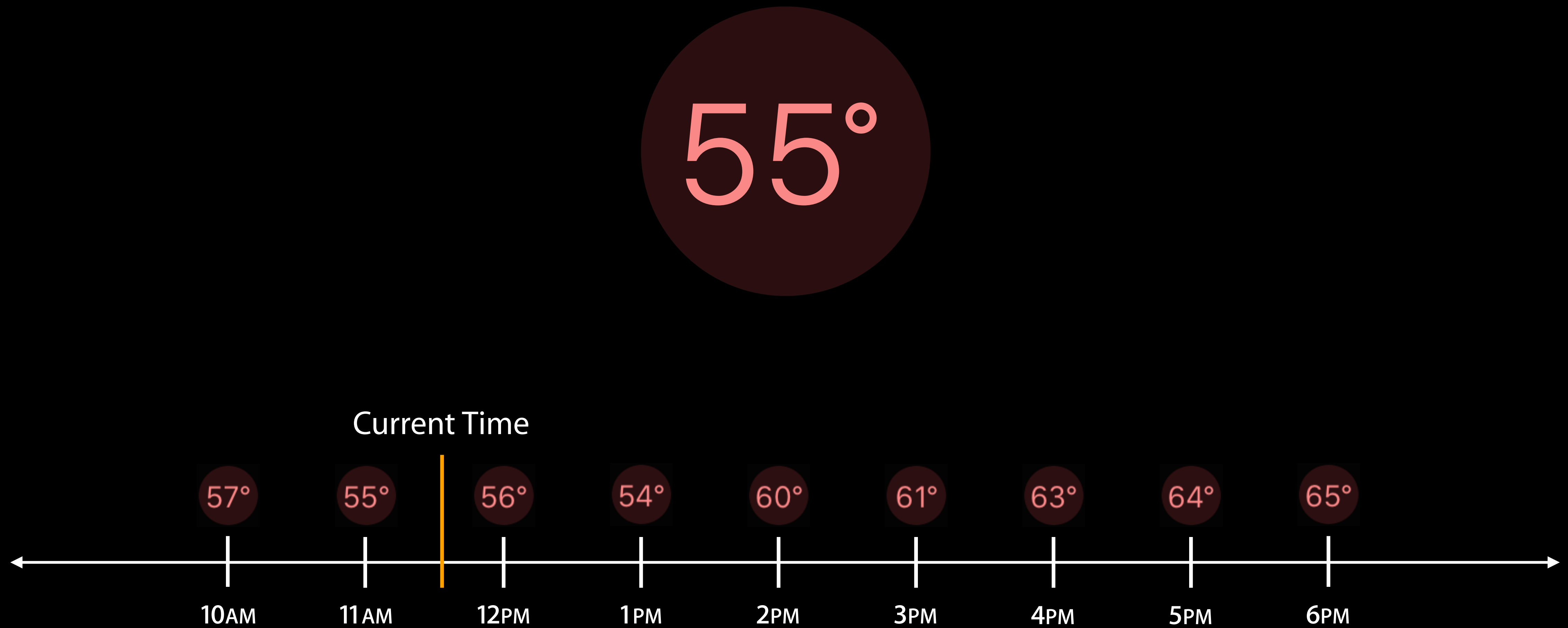
# Timeline

57°

Current Time



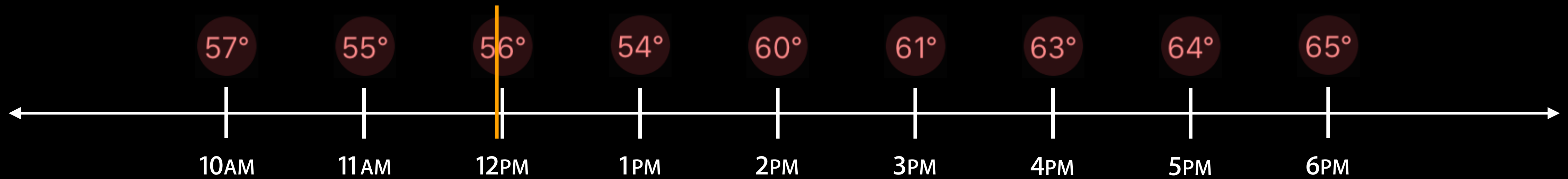
# Timeline



# Timeline

55°

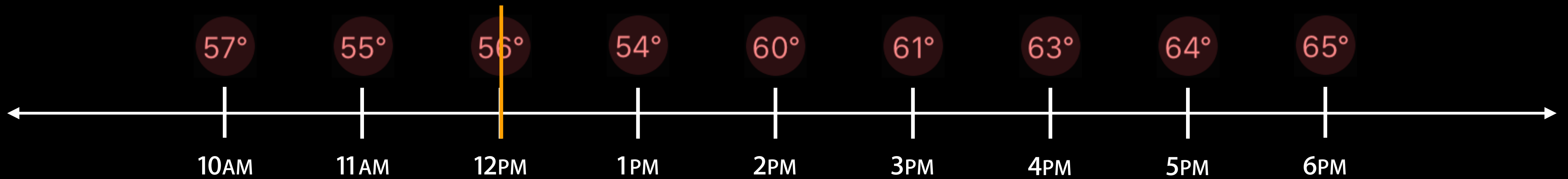
Current Time



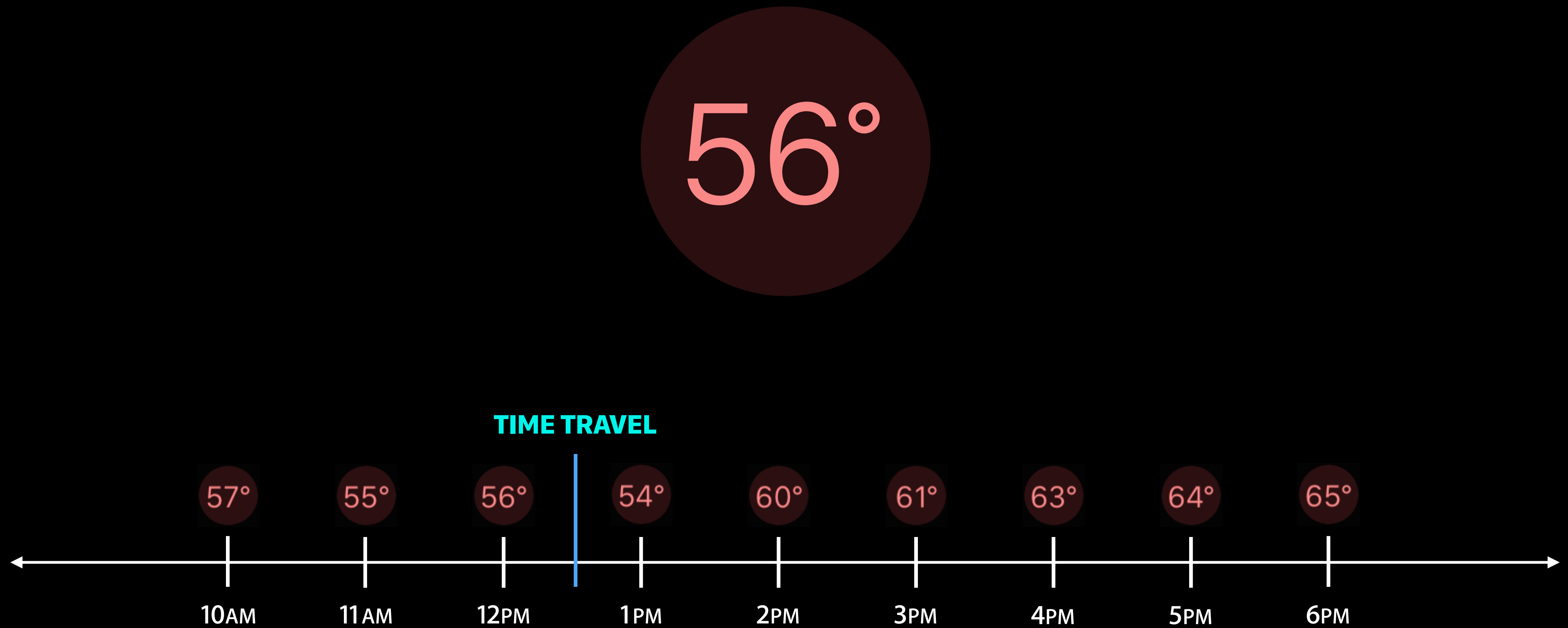
# Timeline

56°

Current Time



# Timeline

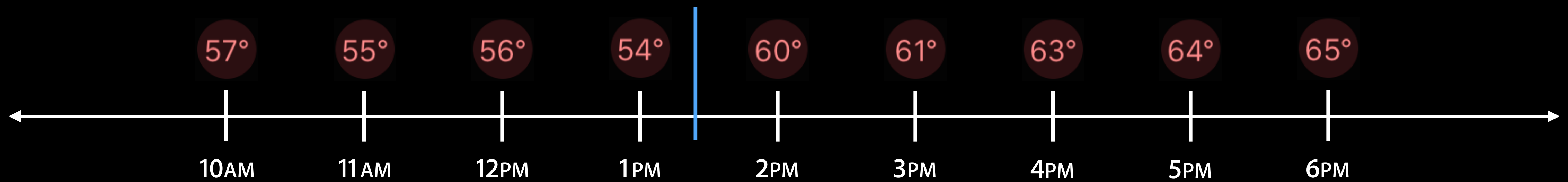




# Timeline

54°

TIME TRAVEL




# Timeline



# Timeline


**11:00AM—1:00PM**  
Brunch with K.K.  
*Joanie's Café*



11AM 1PM

A horizontal blue bar representing the time interval from 11:00 AM to 1:00 PM. The bar is slightly wider at the ends and tapers in the middle. It is flanked by vertical white lines at each end.


**4:00—5:00PM**  
Haircut  
*Nancy Bee*



4PM 5PM

A horizontal blue bar representing the time interval from 4:00 PM to 5:00 PM. The bar is slightly wider at the ends and tapers in the middle. It is flanked by vertical white lines at each end.

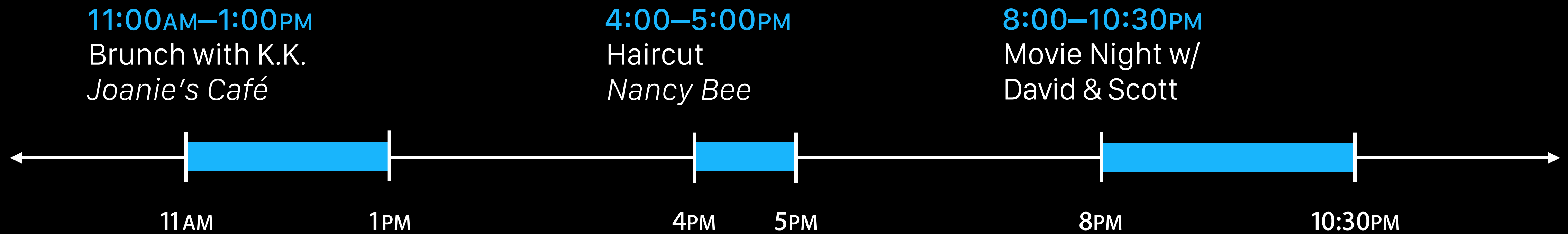
**8:00—10:30PM**  
Movie Night w/  
David & Scott



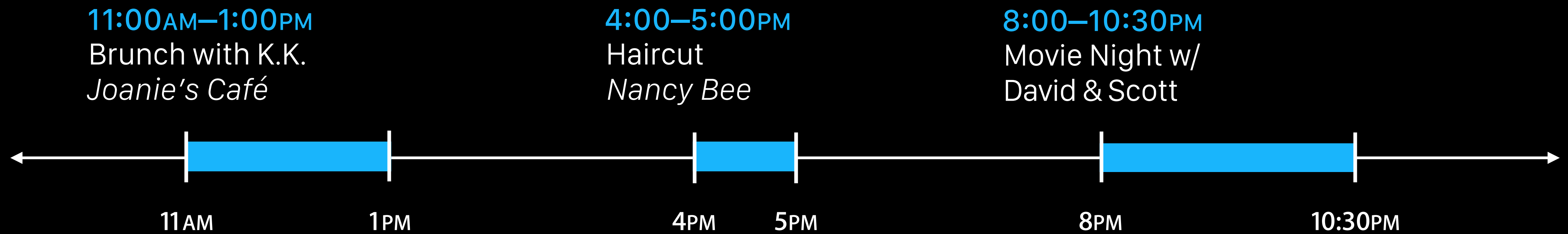
8PM 10:30PM

A horizontal blue bar representing the time interval from 8:00 PM to 10:30 PM. The bar is wider at the ends and tapers in the middle. It is flanked by vertical white lines at each end.

# Timeline

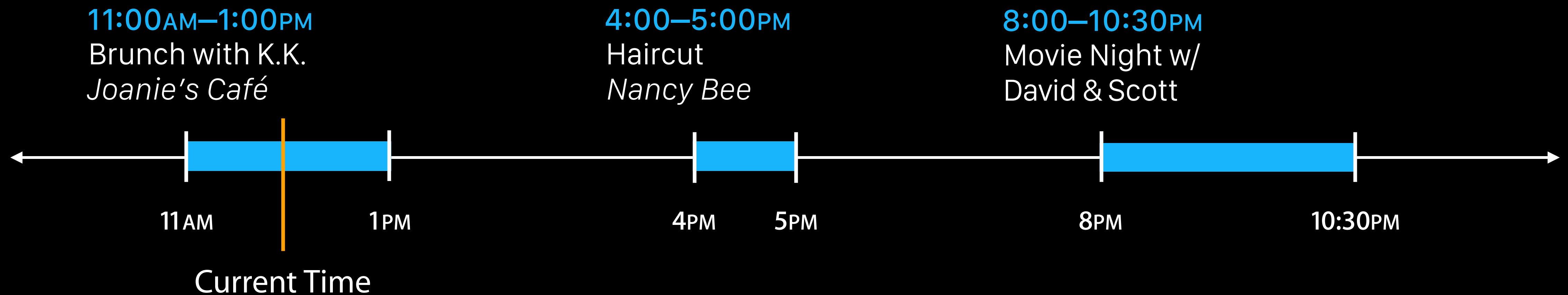


# Timeline

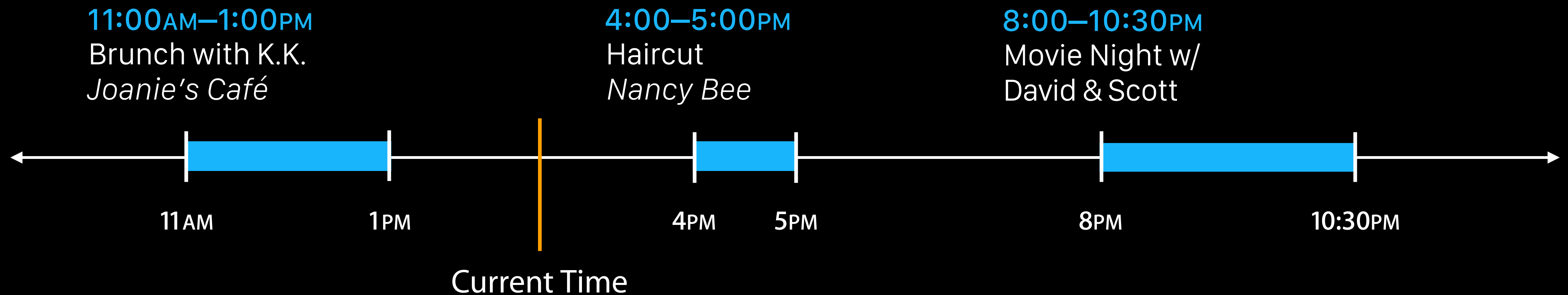
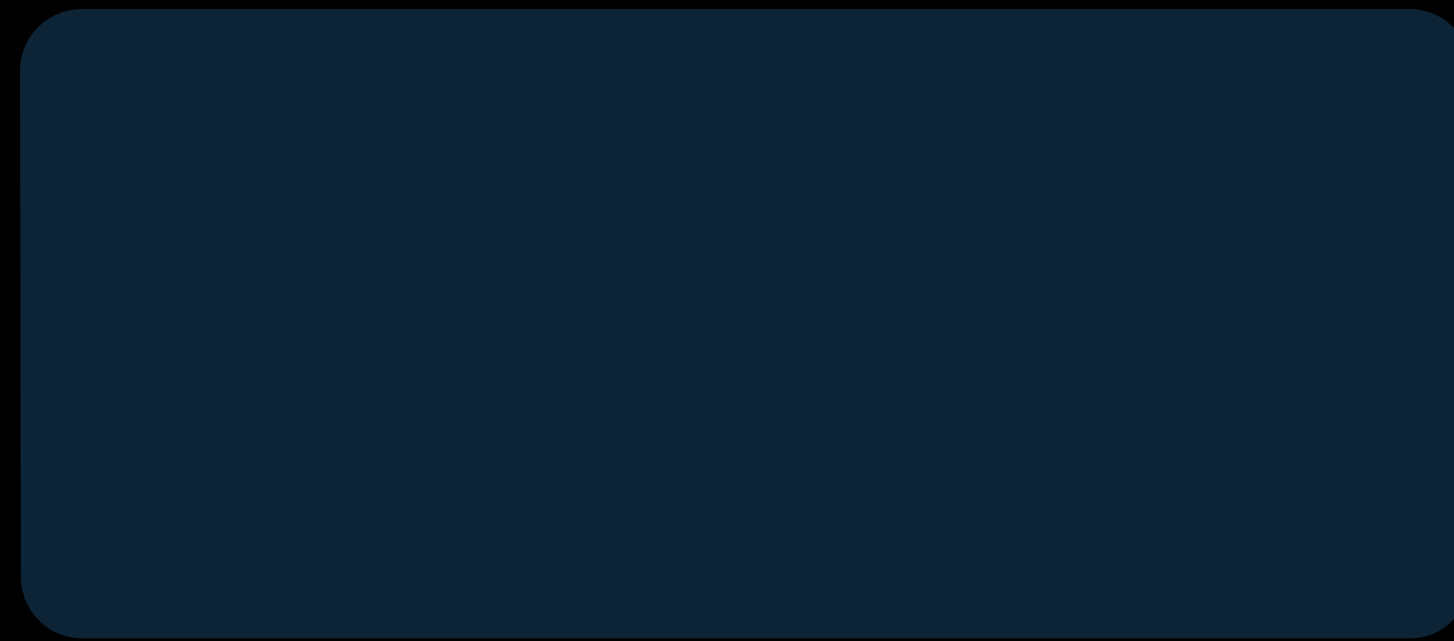


# Timeline

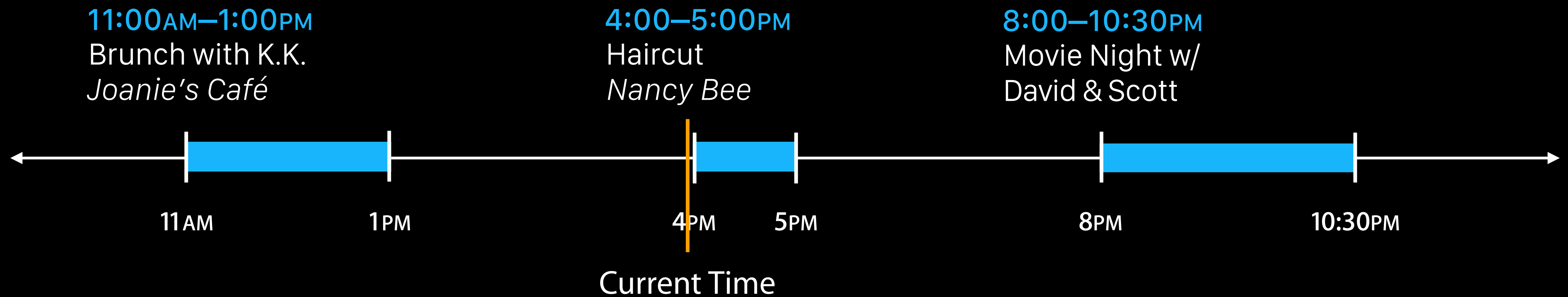
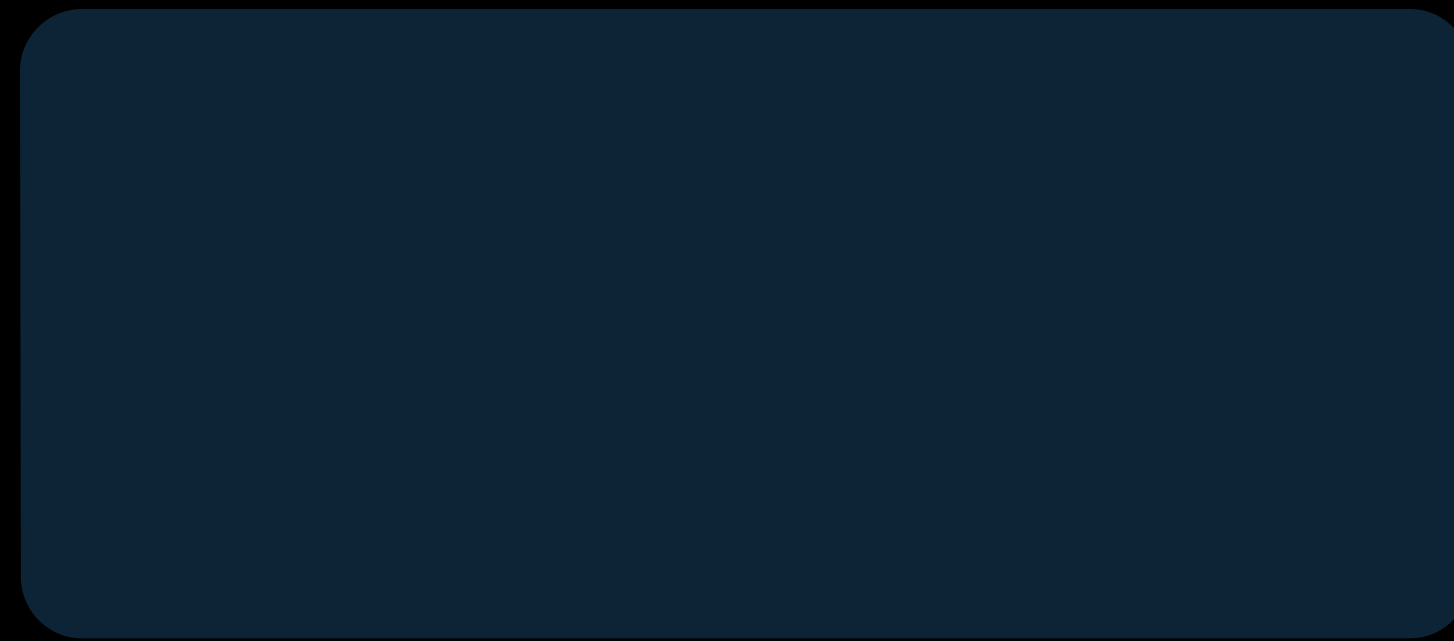
11:00AM—1:00PM  
Brunch with K.K.  
*Joanie's Café*



# Timeline



# Timeline





# Timeline

4:00–5:00PM

Haircut

*Nancy Bee*

11:00AM–1:00PM

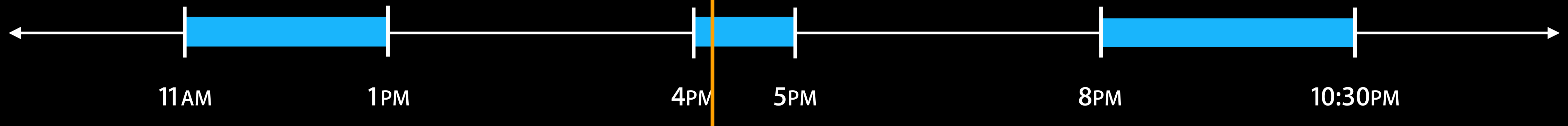
Brunch with K.K.  
*Joanie's Café*

4:00–5:00PM

Haircut  
*Nancy Bee*

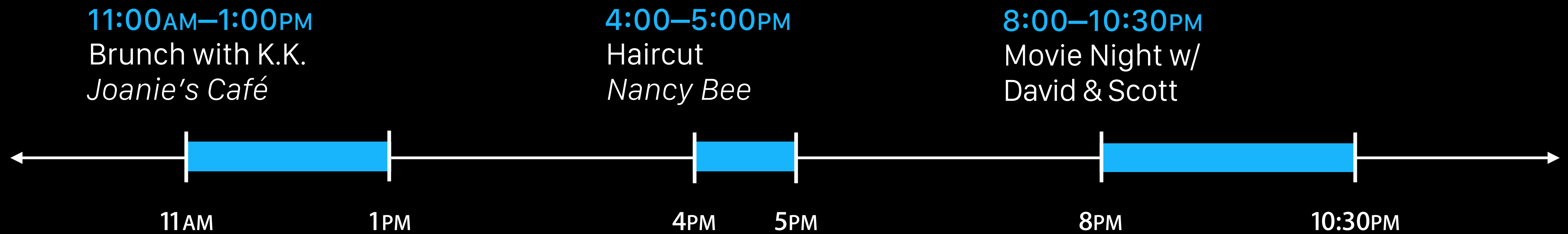
8:00–10:30PM

Movie Night w/  
David & Scott

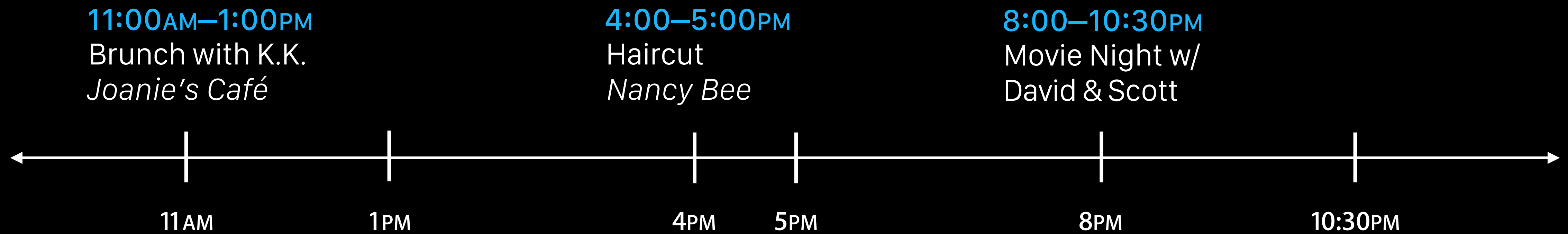


Current Time

# Timeline



# Timeline

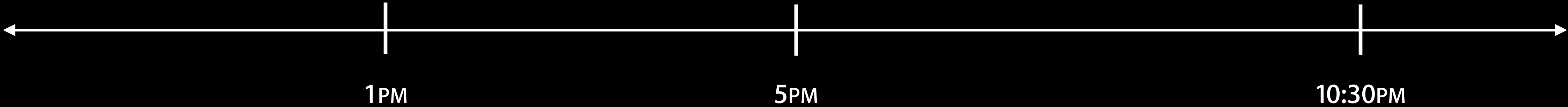


# Timeline

11:00AM—1:00PM  
Brunch with K.K.  
*Joanie's Café*

4:00—5:00PM  
Haircut  
*Nancy Bee*

8:00—10:30PM  
Movie Night w/  
David & Scott

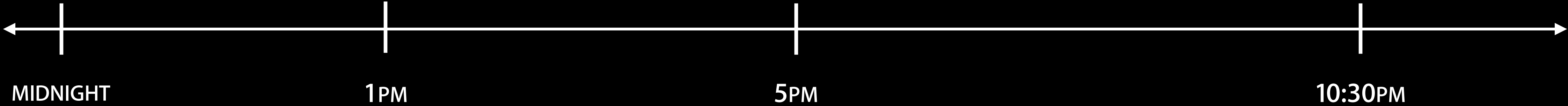


# Timeline

11:00AM—1:00PM  
Brunch with K.K.  
*Joanie's Café*

4:00—5:00PM  
Haircut  
*Nancy Bee*

8:00—10:30PM  
Movie Night w/  
David & Scott



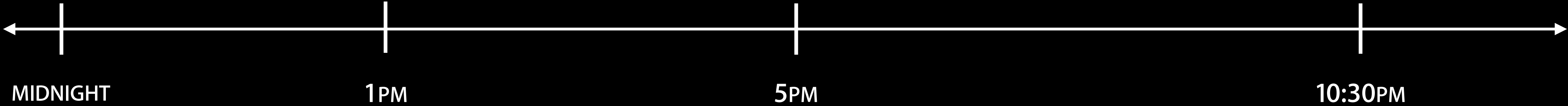
# Timeline

11:00AM—1:00PM  
Brunch with K.K.  
*Joanie's Café*

4:00—5:00PM  
Haircut  
*Nancy Bee*

8:00—10:30PM  
Movie Night w/  
David & Scott

Calendar  
No more events  
today



# Timeline

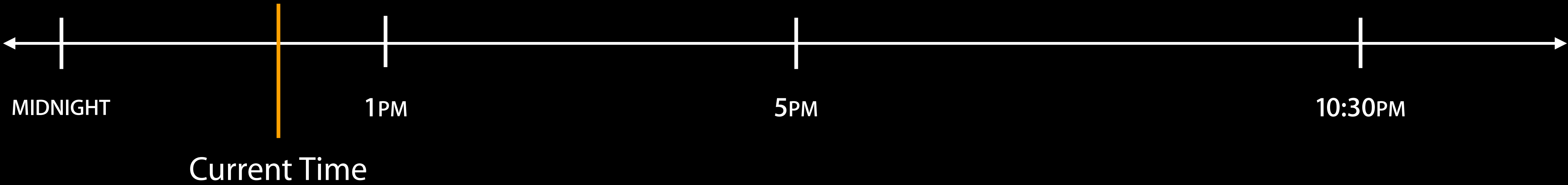
11:00AM—1:00PM  
Brunch with K.K.  
*Joanie's Café*

11:00AM—1:00PM  
Brunch with K.K.  
*Joanie's Café*

4:00—5:00PM  
Haircut  
*Nancy Bee*

8:00—10:30PM  
Movie Night w/  
David & Scott

Calendar  
No more events  
today



# Timeline

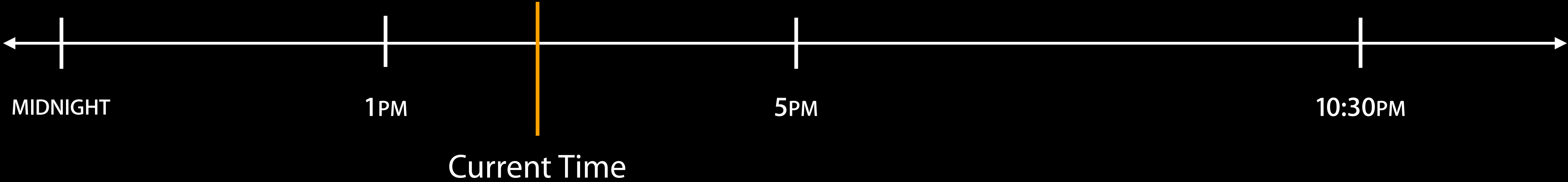
4:00–5:00PM  
Haircut  
*Nancy Bee*

11:00AM–1:00PM  
Brunch with K.K.  
*Joanie's Café*

4:00–5:00PM  
Haircut  
*Nancy Bee*

8:00–10:30PM  
Movie Night w/  
David & Scott

Calendar  
No more events  
today





# Timeline

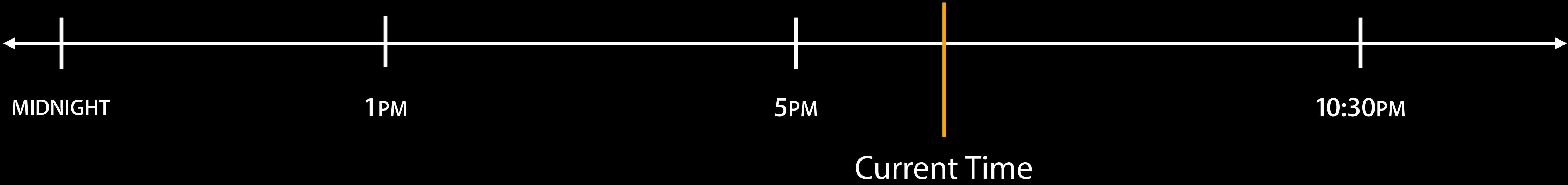
8:00–10:30PM  
Movie Night w/  
David & Scott

11:00AM–1:00PM  
Brunch with K.K.  
*Joanie's Café*

4:00–5:00PM  
Haircut  
*Nancy Bee*

8:00–10:30PM  
Movie Night w/  
David & Scott

Calendar  
No more events  
today



# Timeline

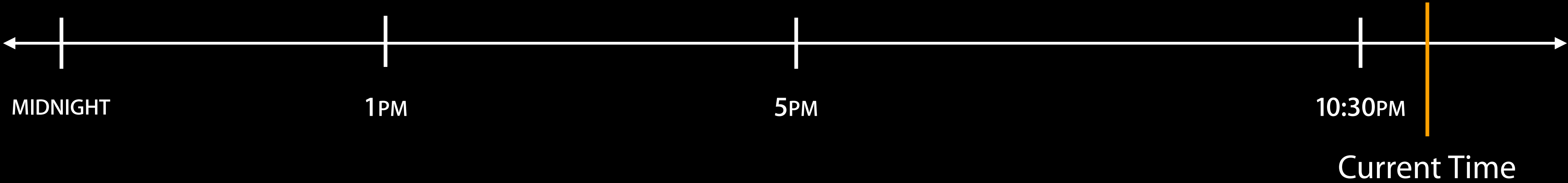
Calendar  
No more events  
today

11:00AM—1:00PM  
Brunch with K.K.  
*Joanie's Café*

4:00—5:00PM  
Haircut  
*Nancy Bee*

8:00—10:30PM  
Movie Night w/  
David & Scott

Calendar  
No more events  
today

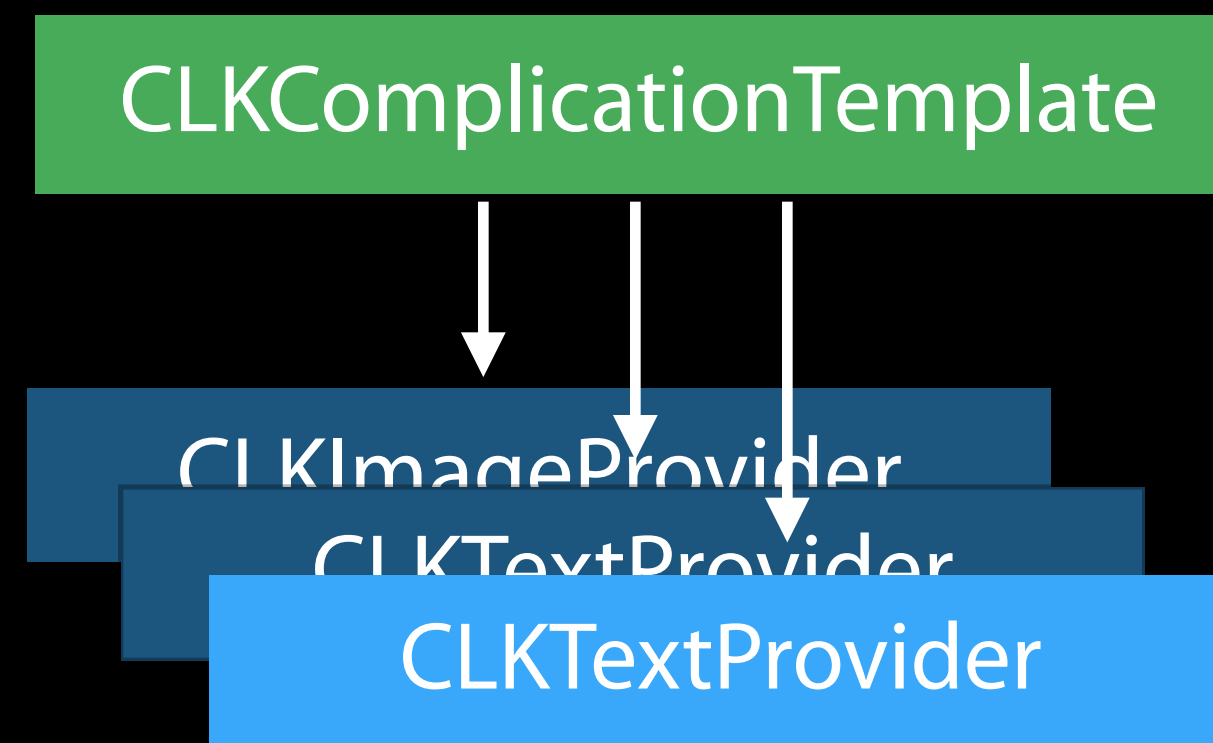


# Timeline

CLKComplicationTimelineEntry

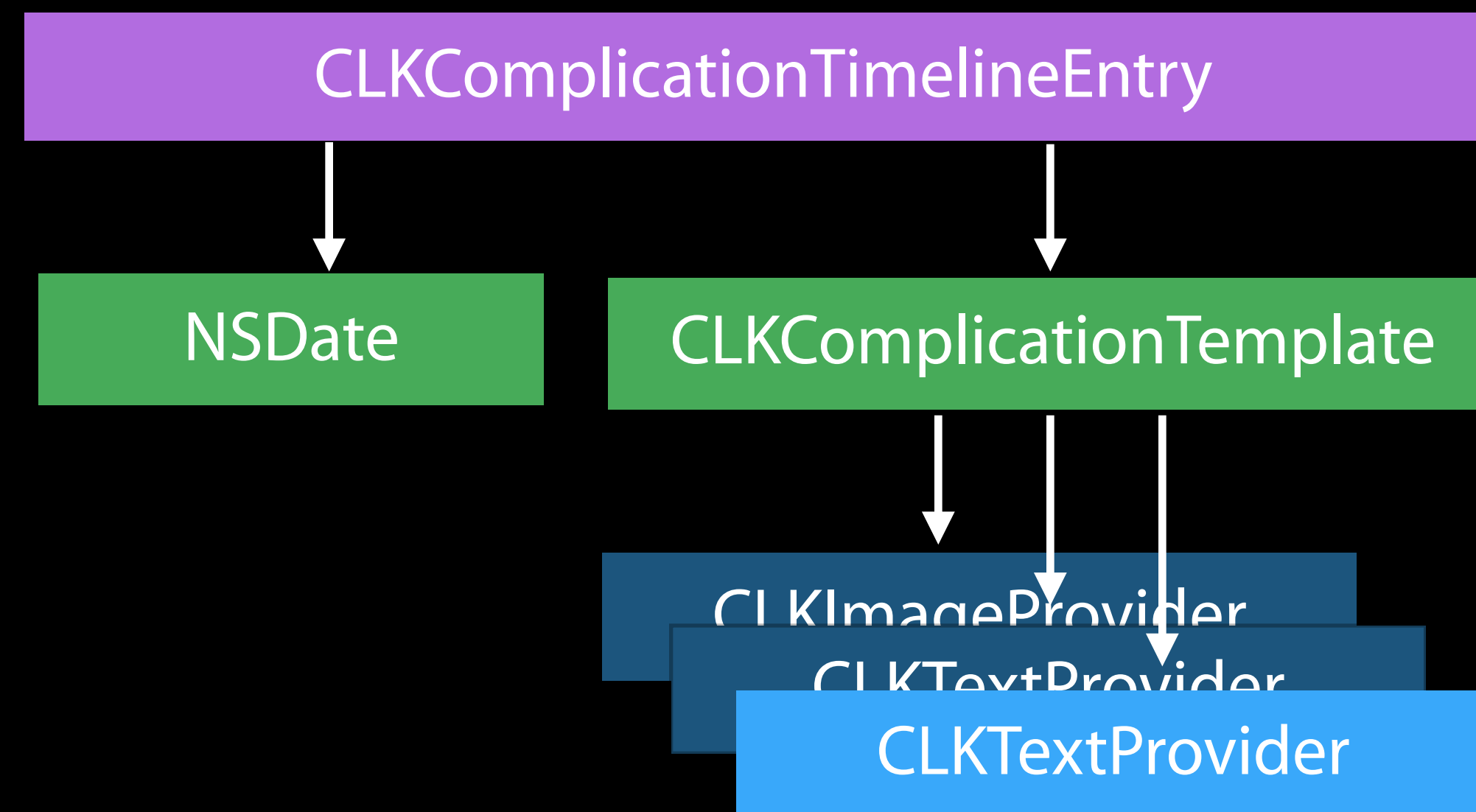
# Timeline

CLKComplicationTimelineEntry



# Timeline

## CLKComplicationTimelineEntry



# Timeline

## CLKComplicationTimelineEntry

```
class CLKComplicationTimelineEntry : NSObject {  
    var date: NSDate  
    var complicationTemplate: CLKComplicationTemplate  
}
```



Template

Complication Data Source

# Complication Data Source

CLKComplicationDataSource protocol



# Complication Data Source

CLKComplicationDataSource protocol

Object annotated in extension target settings

# Complication Data Source

CLKComplicationDataSource protocol

Object annotated in extension target settings

Callbacks per complication

# Complication Data Source

CLKComplicationDataSource protocol

Object annotated in extension target settings

Callbacks per complication

- Check CLKComplication's family property

# Complication Data Source

CLKComplicationDataSource protocol

Object annotated in extension target settings

Callbacks per complication

- Check CLKComplication's family property

Handler parameter for completion and passing data

# Complication Data Source

Current timeline entry



Clock  
Face

Extension



Now

# Complication Data Source

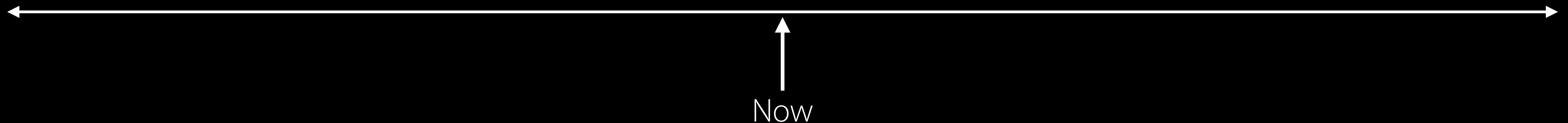
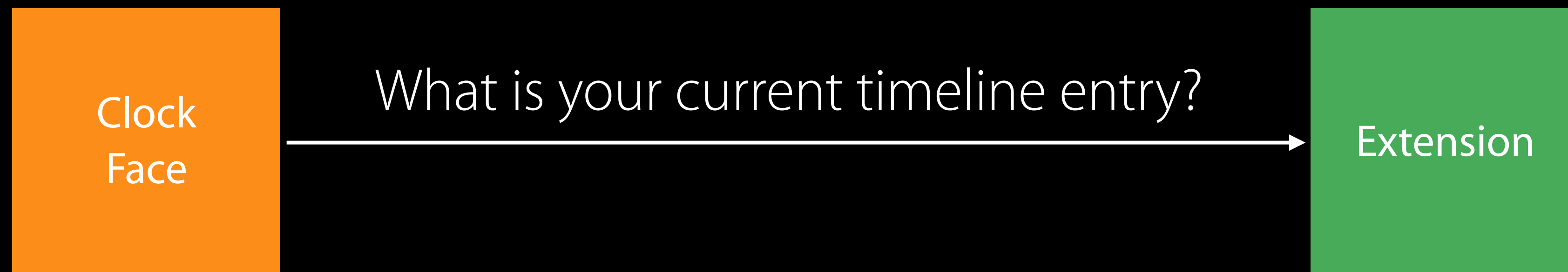
Current timeline entry



Now

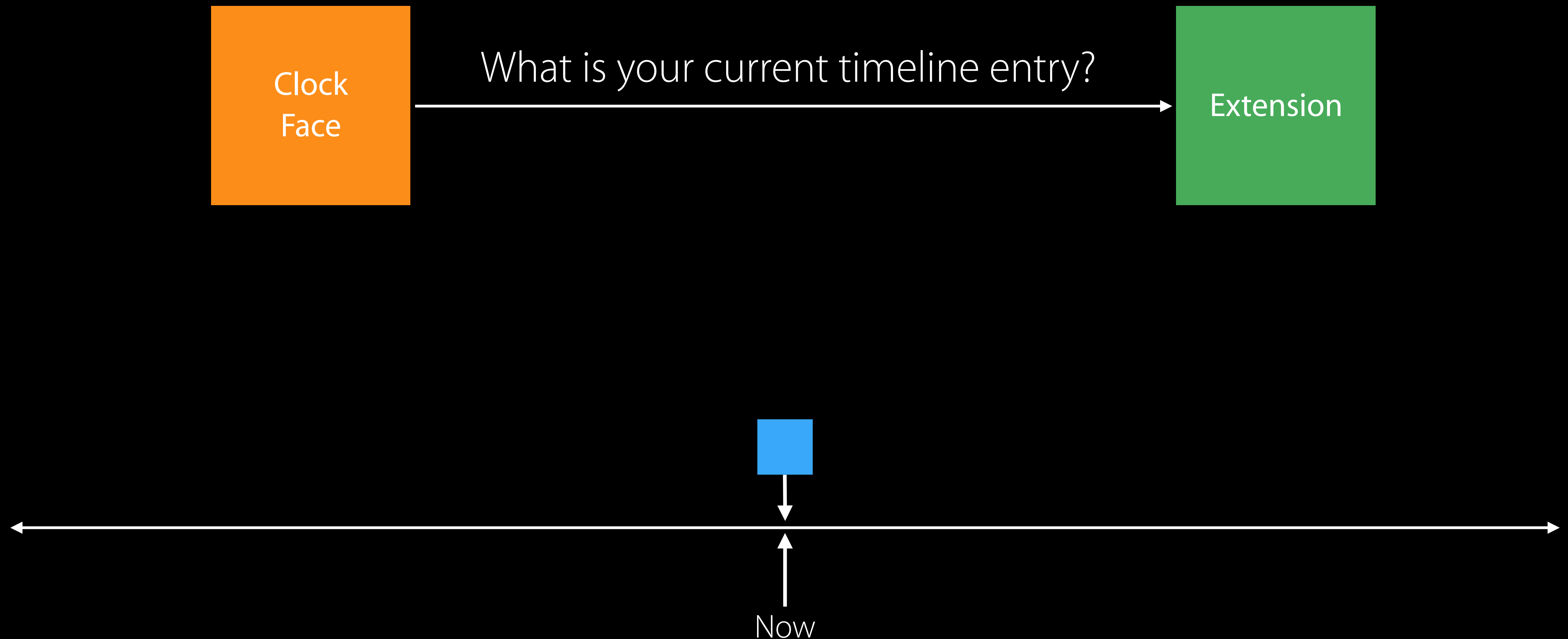
# Complication Data Source

Current timeline entry



# Complication Data Source

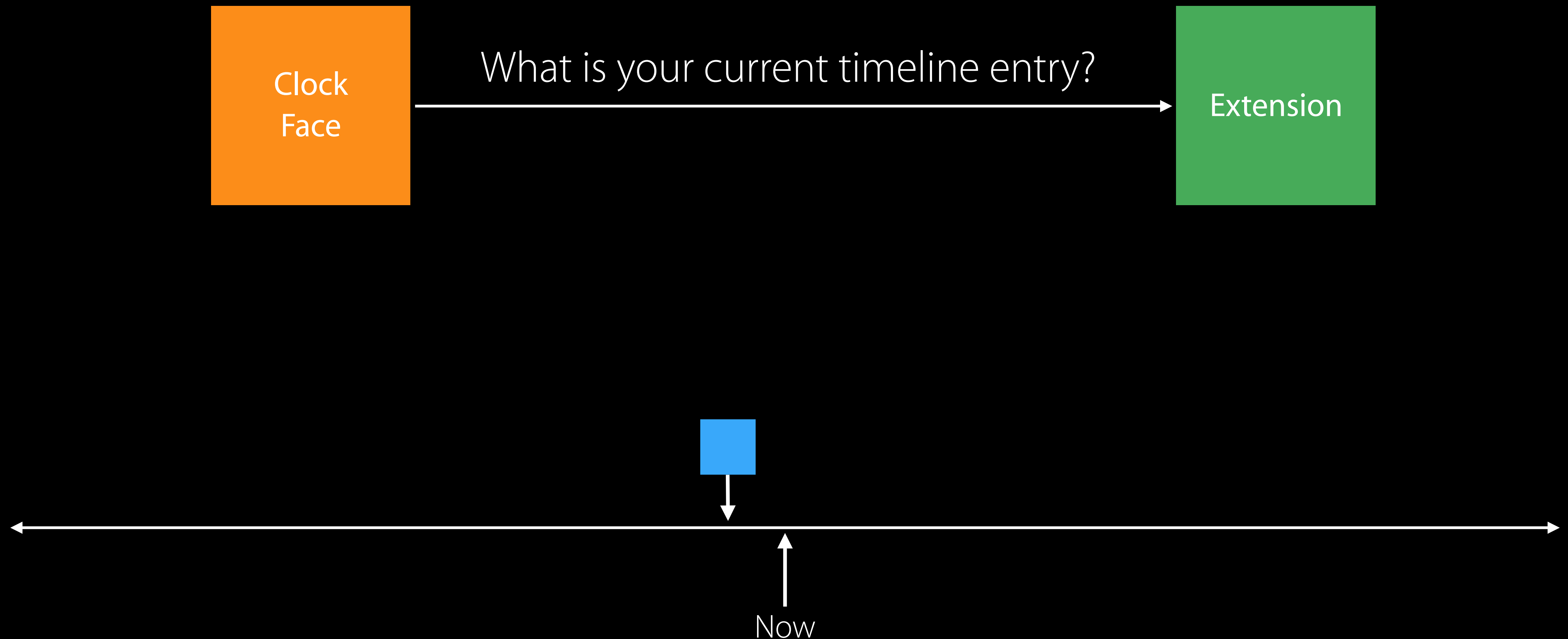
Current timeline entry





# Complication Data Source

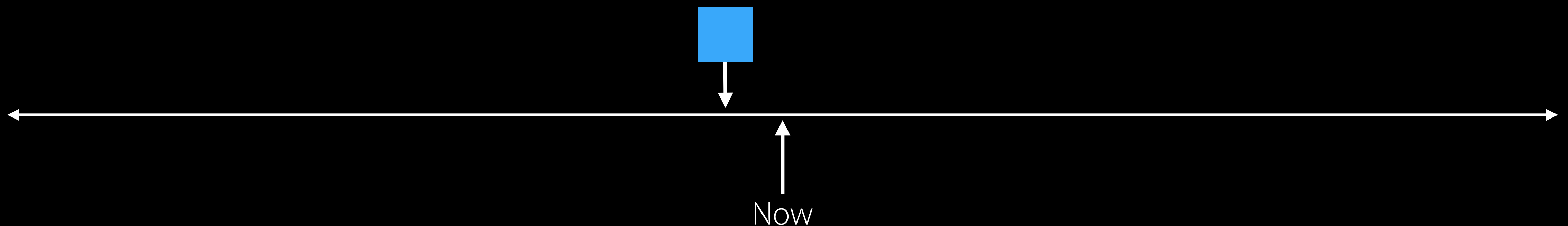
Current timeline entry



# Complication Data Source

Current timeline entry

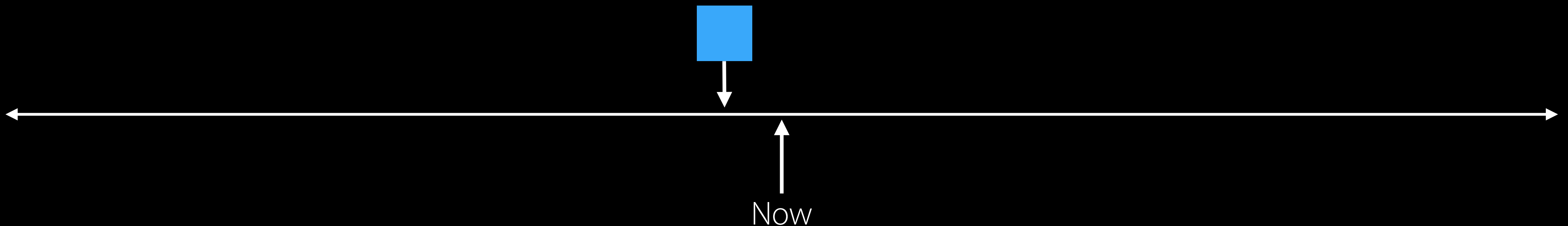
```
func getCurrentTimelineEntryForComplication(  
    complication: CLKComplication,  
    withHandler handler: (CLKComplicationTimelineEntry?) -> Void)
```



# Complication Data Source

Current timeline entry

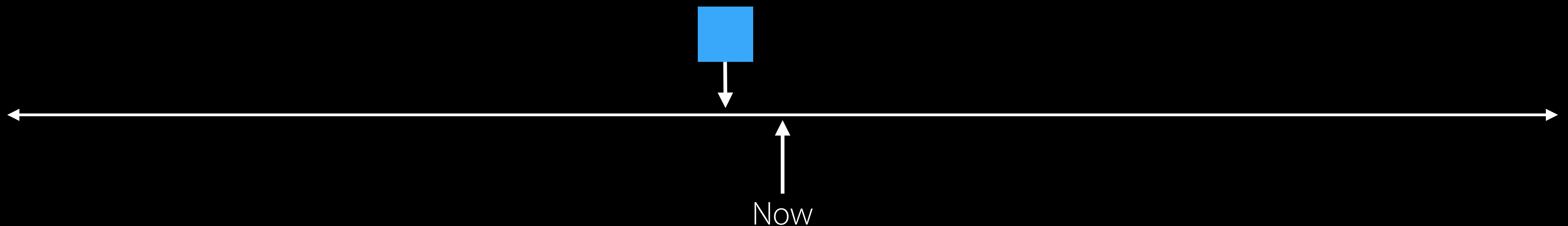
```
func getCurrentTimelineEntryForComplication(  
    complication: CLKComplication,  
    withHandler handler: (CLKComplicationTimelineEntry?) -> Void)
```



# Complication Data Source

Current timeline entry

```
func getCurrentTimelineEntryForComplication(  
    complication: CLKComplication,  
    withHandler handler: (CLKComplicationTimelineEntry?) -> Void)
```



# Complication Data Source

Extend timeline

Clock  
Face

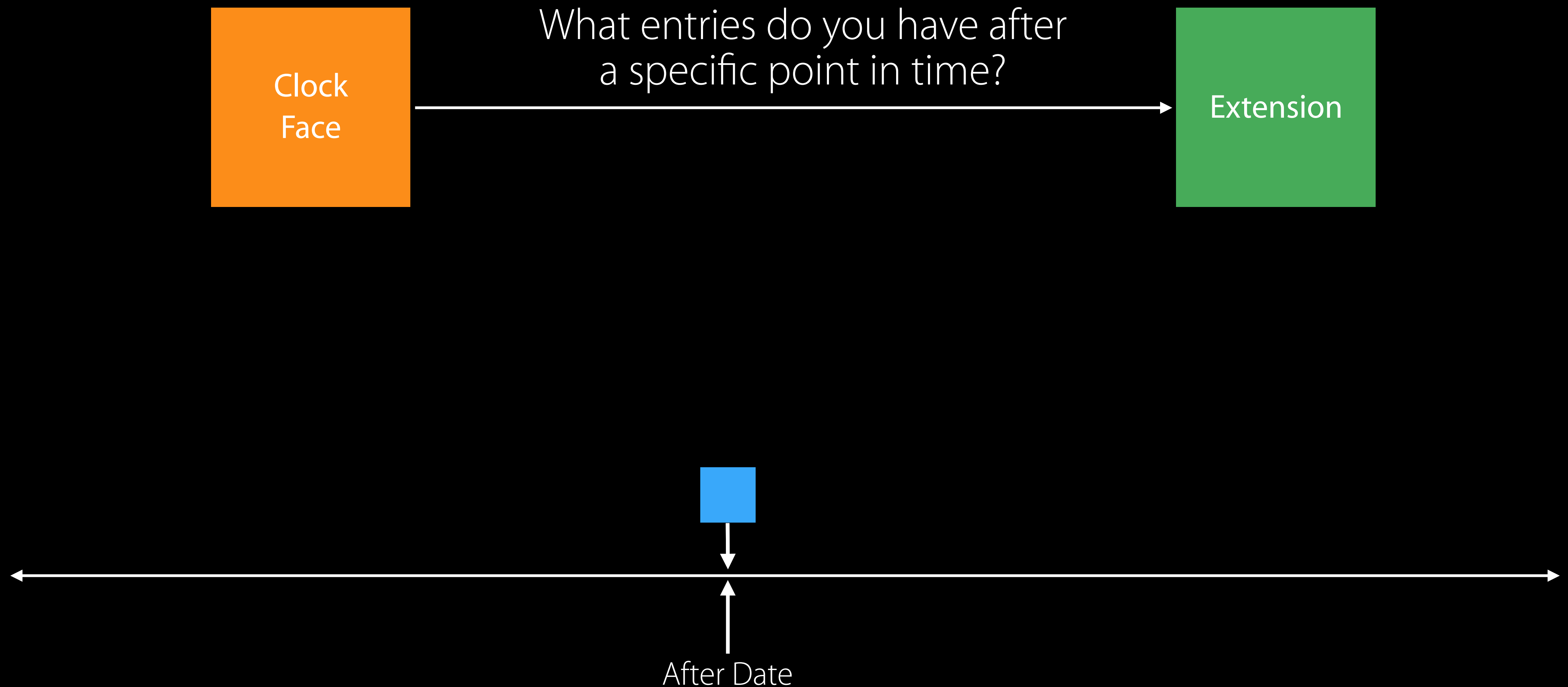
Extension



Now

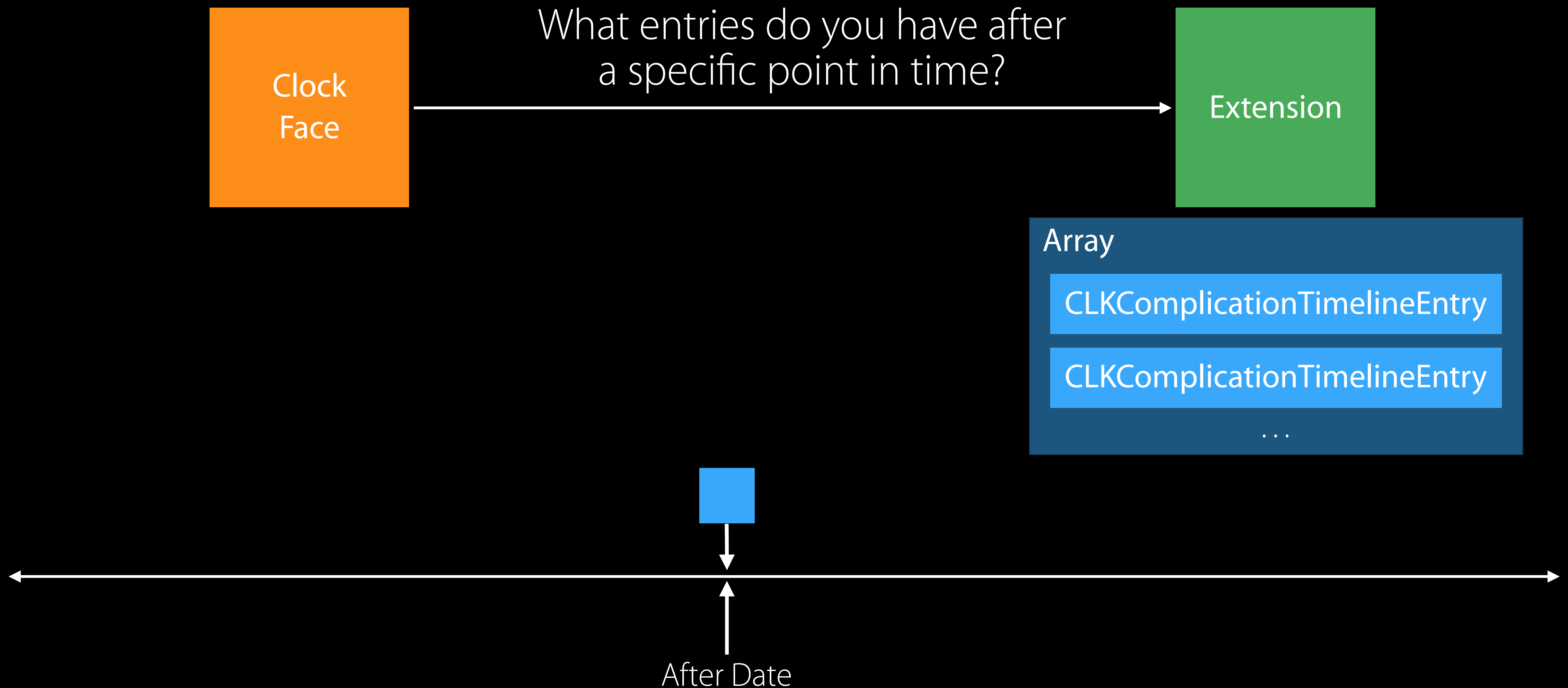
# Complication Data Source

Extend timeline



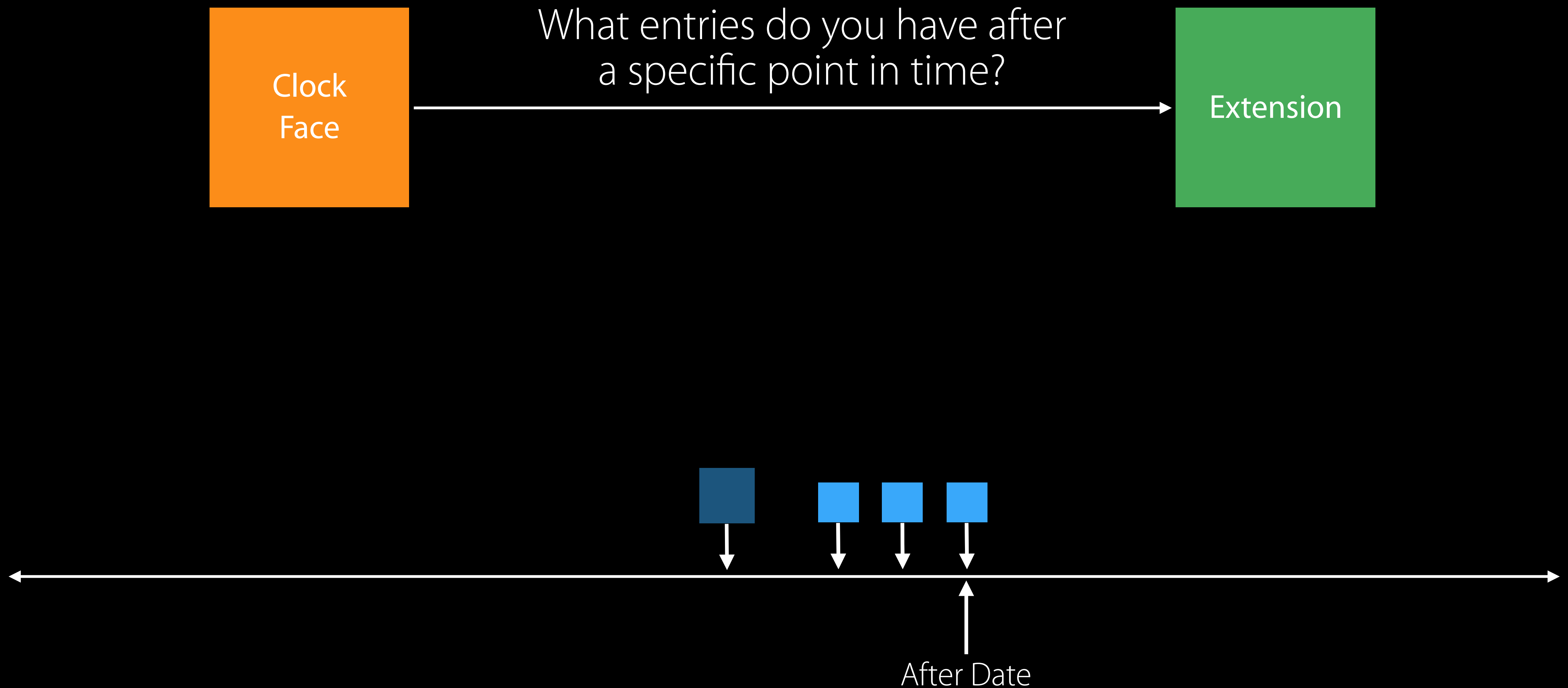
# Complication Data Source

Extend timeline



# Complication Data Source

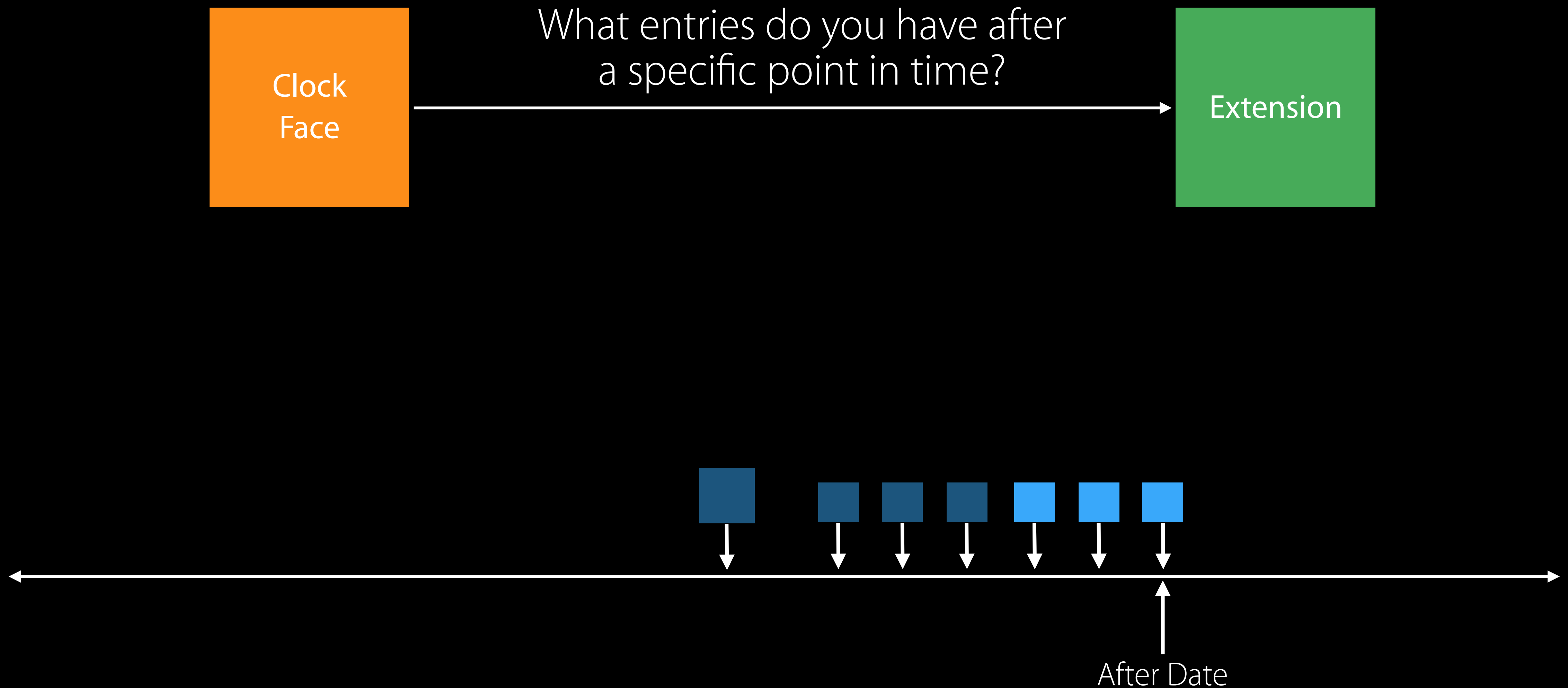
Extend timeline





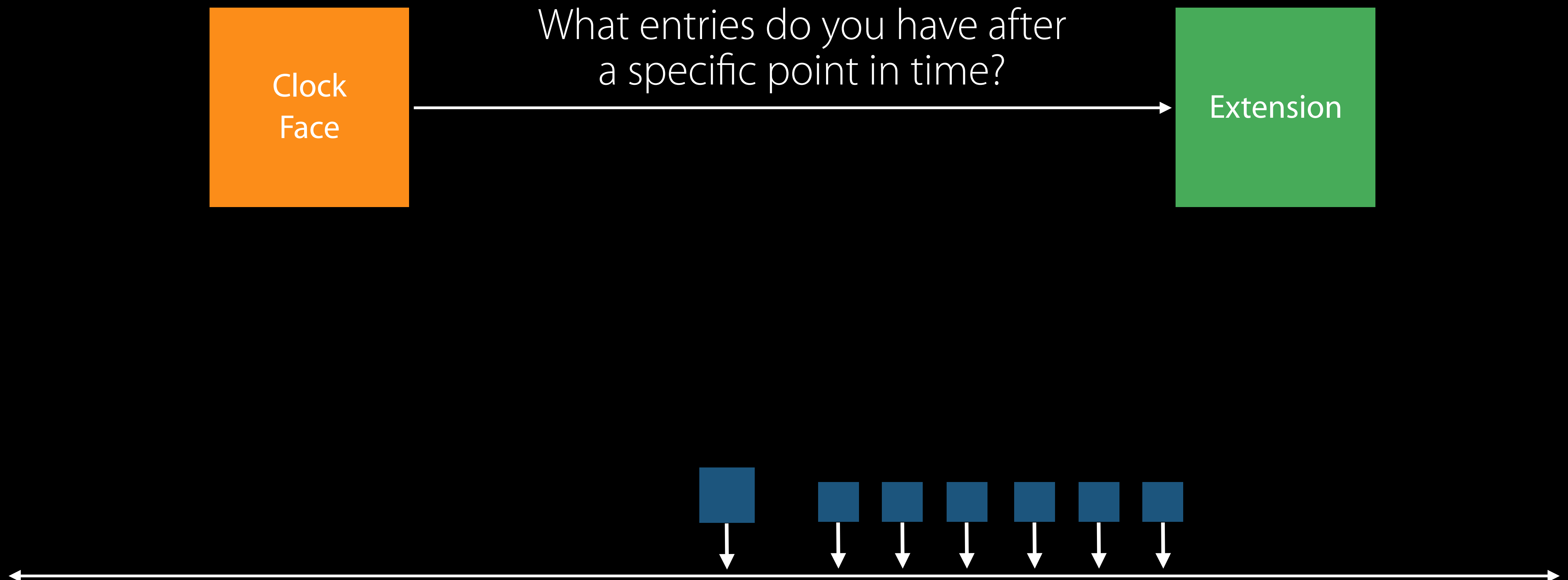
# Complication Data Source

Extend timeline



# Complication Data Source

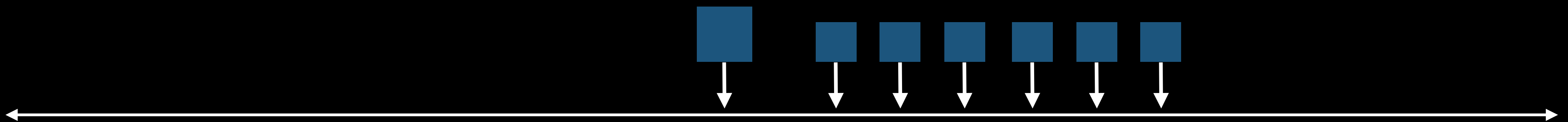
Extend timeline



# Complication Data Source

Extend timeline

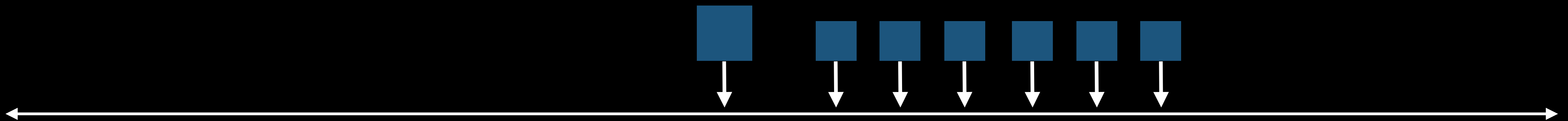
```
func getTimelineEntriesForComplication(  
    complication: CLKComplication,  
    afterDate date: NSDate,  
    limit: Int,  
    withHandler handler: ([CLKComplicationTimelineEntry]?) -> Void)
```



# Complication Data Source

Extend timeline

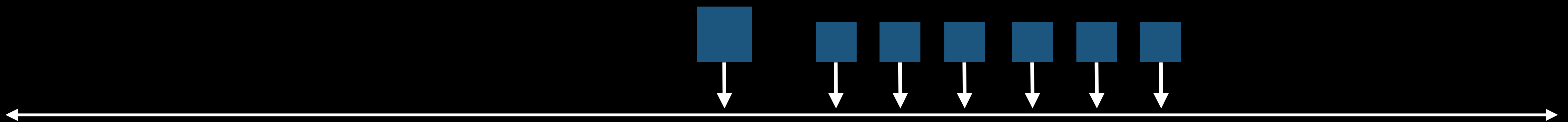
```
func getTimelineEntriesForComplication(  
    complication: CLKComplication,  
    afterDate date: NSDate,  
    limit: Int,  
    withHandler handler: ([CLKComplicationTimelineEntry]? ) -> Void)
```



# Complication Data Source

Extend timeline

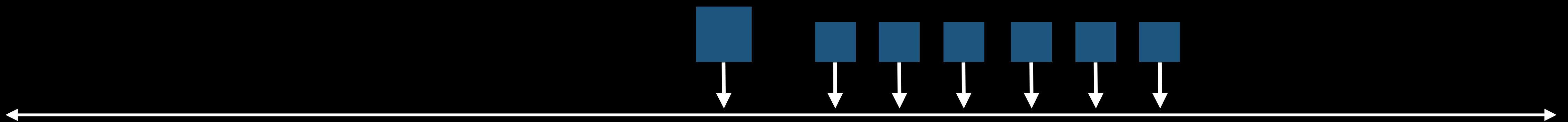
```
func getTimelineEntriesForComplication(  
    complication: CLKComplication,  
    afterDate date: NSDate,  
    limit: Int,  
    withHandler handler: ([CLKComplicationTimelineEntry]?) -> Void)
```



# Complication Data Source

Extend timeline

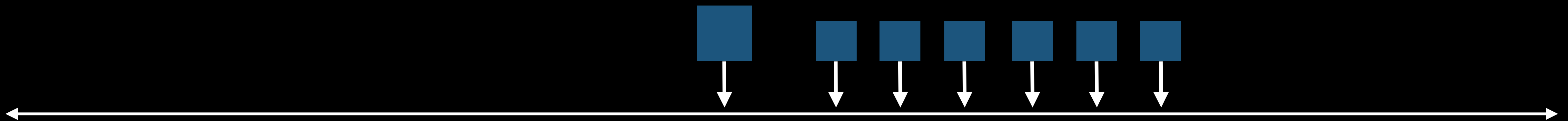
```
func getTimelineEntriesForComplication(  
    complication: CLKComplication,  
    afterDate date: NSDate,  
    limit: Int,  
    withHandler handler: ([CLKComplicationTimelineEntry]? ) -> Void)
```



# Complication Data Source

Extend timeline

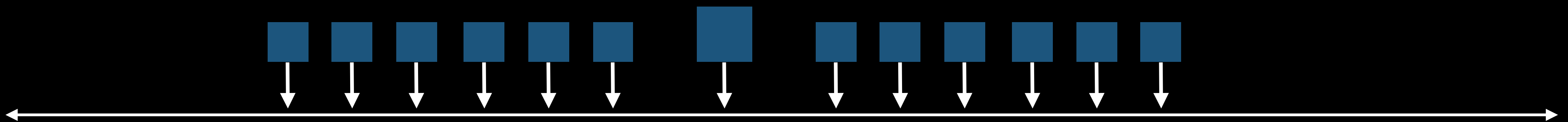
```
func getTimelineEntriesForComplication(  
    complication: CLKComplication,  
    afterDate date: NSDate,  
    limit: Int,  
    withHandler handler: ([CLKComplicationTimelineEntry]?) -> Void)
```



# Complication Data Source

Extend timeline

```
func getTimelineEntriesForComplication(  
    complication: CLKComplication,  
    beforeDate date: NSDate,  
    limit: Int,  
    withHandler handler: ([CLKComplicationTimelineEntry]? ) -> Void)
```





# Complication Data Source

Time Travel directions



Clock  
Face

Extension

Now

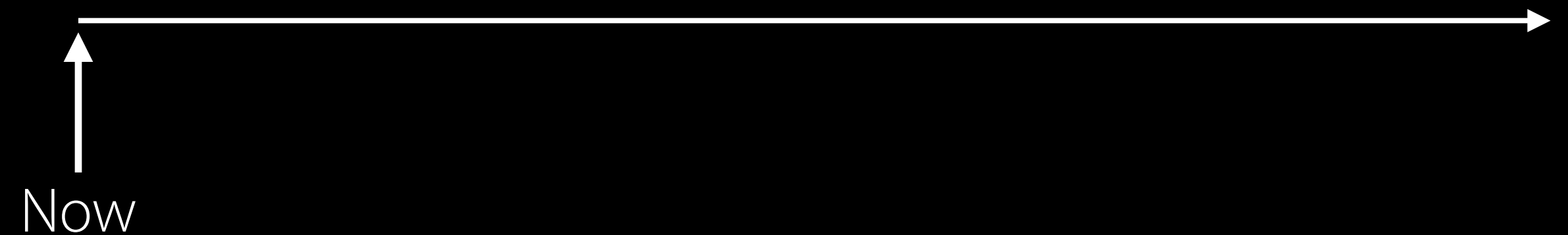
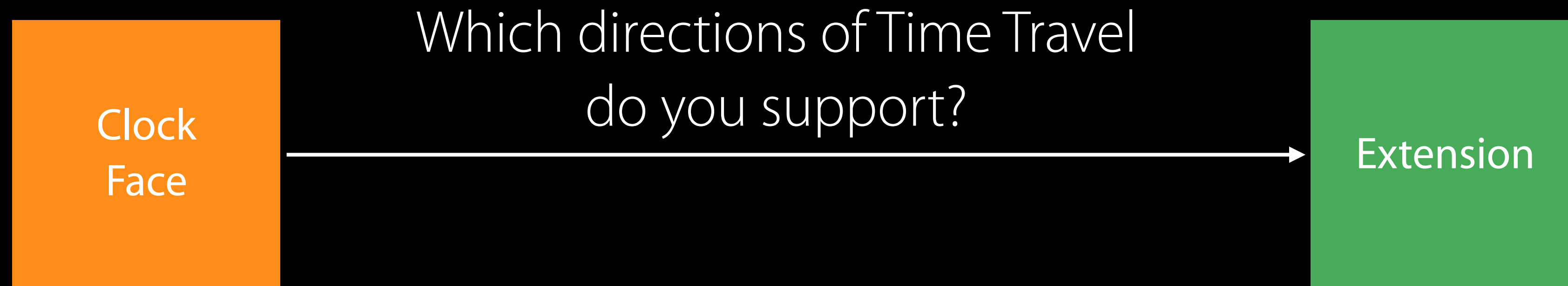
# Complication Data Source

Time Travel directions



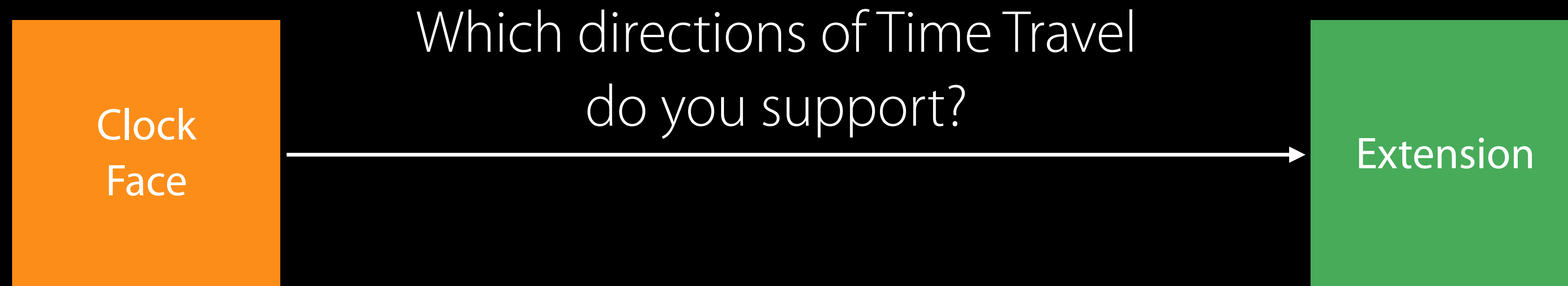
# Complication Data Source

Time Travel directions



# Complication Data Source

Time Travel directions



# Complication Data Source

Time Travel directions



↑  
Now

# Complication Data Source

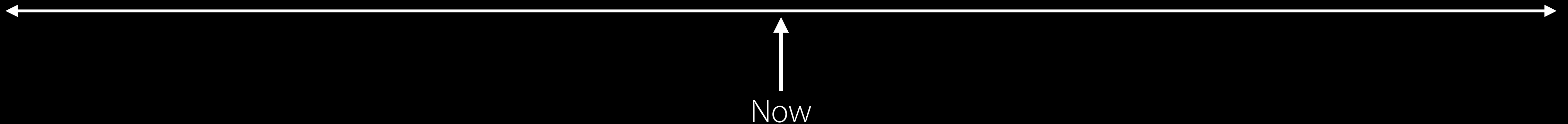
Time Travel directions



# Complication Data Source

Time Travel directions

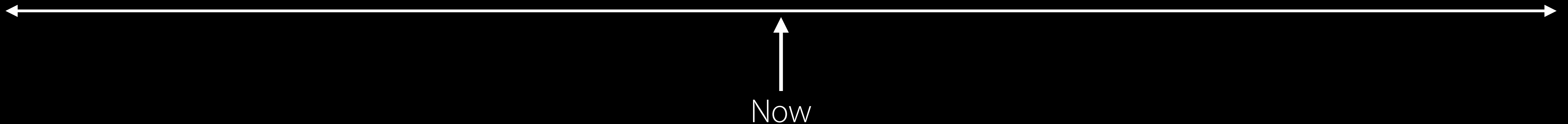
```
func getSupportedTimeTravelDirectionsForComplication(  
    complication: CLKComplication,  
    withHandler handler: (CLKComplicationTimeTravelDirections) -> Void)
```



# Complication Data Source

Time Travel directions

```
func getSupportedTimeTravelDirectionsForComplication(  
    complication: CLKComplication,  
    withHandler handler: (CLKComplicationTimeTravelDirections) -> Void)
```

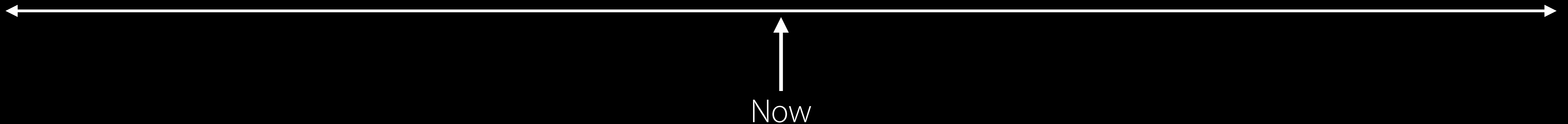




# Complication Data Source

Time Travel directions

```
func getSupportedTimeTravelDirectionsForComplication(  
    complication: CLKComplication,  
    withHandler handler: (CLKComplicationTimeTravelDirections) -> Void)
```





SUN  
31

10:09

11:00AM—1:00PM

Brunch with K.K.

*Joanie's Café*

8:21

65°

TOK  
2:09



TIME TRAVEL

+6:00

SUN  
31

4:09

4:00-5:00PM

Haircut

*Nancy Bee*

8:21

75°

TOK  
8:09



# Complication Data Source

Timeline boundaries



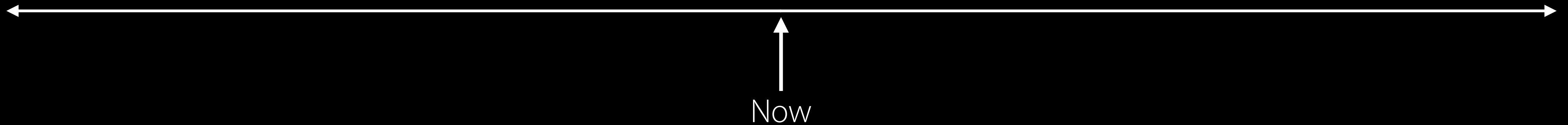
Clock  
Face

Extension

Now

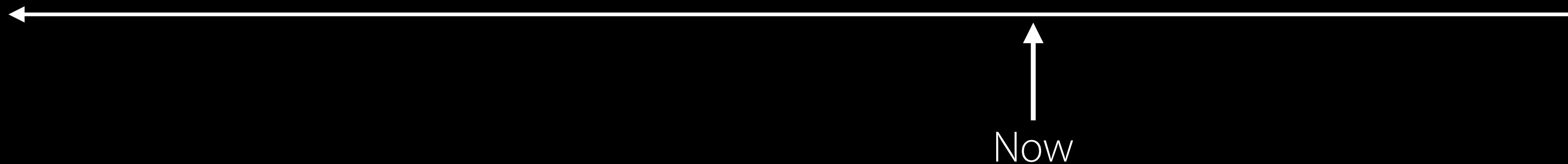
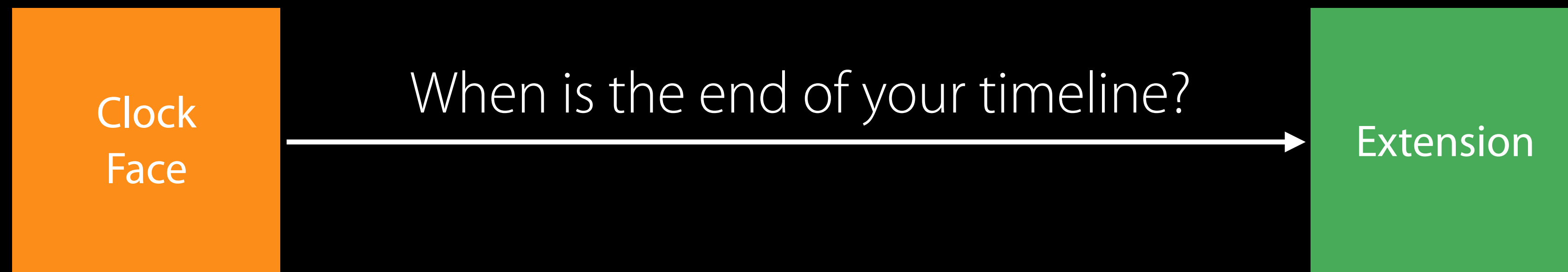
# Complication Data Source

Timeline boundaries



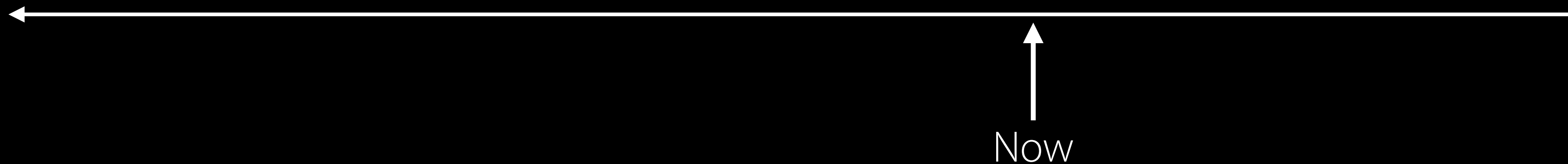
# Complication Data Source

Timeline boundaries



# Complication Data Source

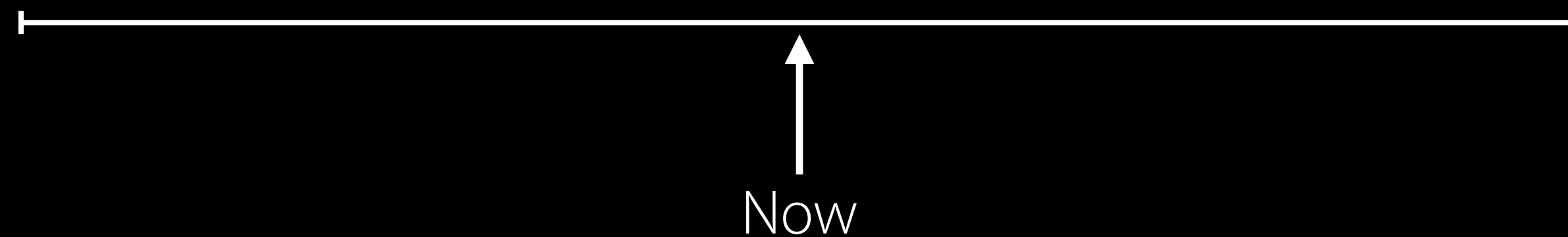
Timeline boundaries





# Complication Data Source

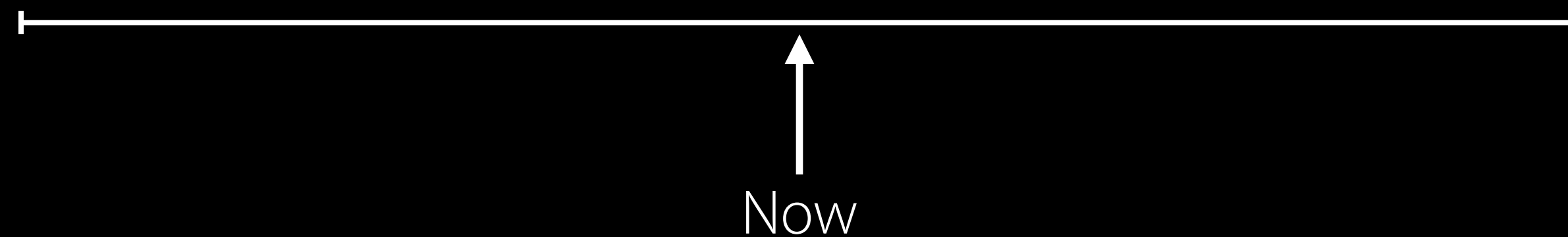
Timeline boundaries



# Complication Data Source

Timeline boundaries

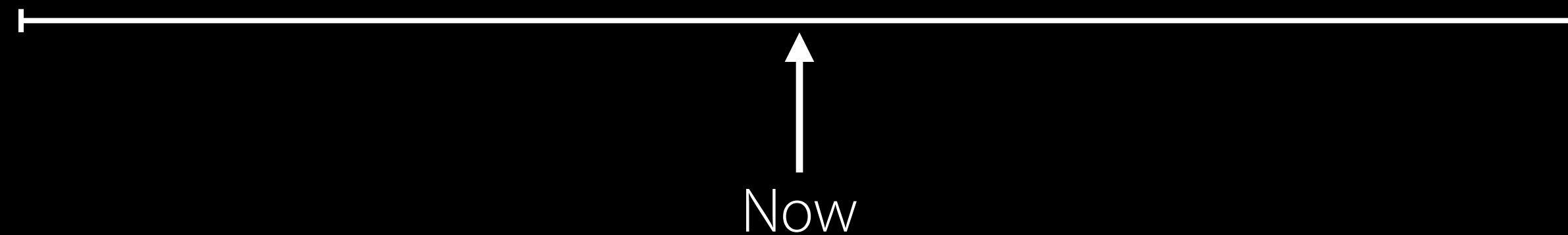
```
func getTimelineEndDateForComplication(  
    complication: CLKComplication,  
    withHandler handler: (NSDate?) -> Void)
```



# Complication Data Source

Timeline boundaries

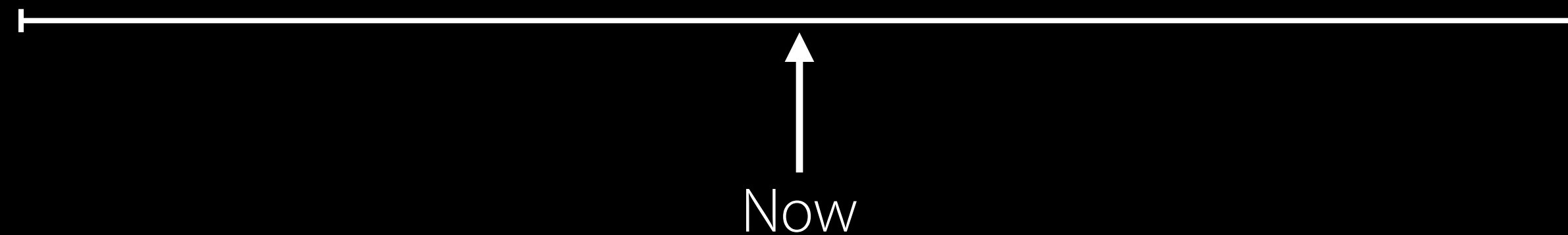
```
func getTimelineEndDateForComplication(  
    complication: CLKComplication,  
    withHandler handler: (NSDate?) -> Void)
```



# Complication Data Source

Timeline boundaries

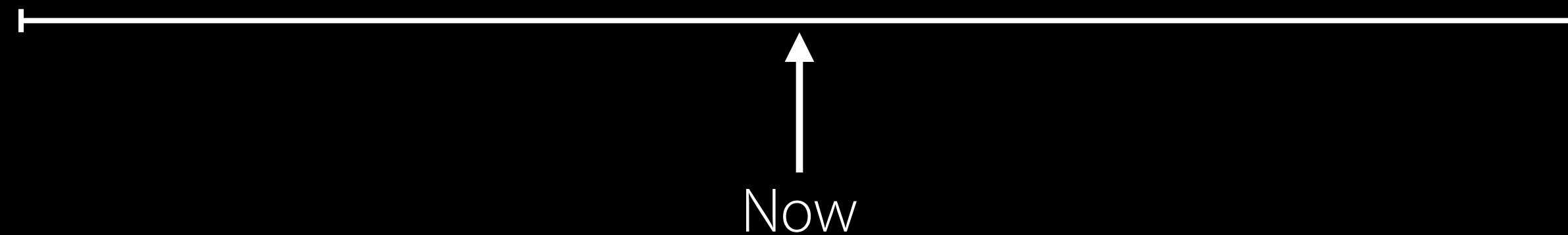
```
func getTimelineEndDateForComplication(  
    complication: CLKComplication,  
    withHandler handler: (NSDate?) -> Void)
```



# Complication Data Source

Timeline boundaries

```
func getTimelineStartDateForComplication(  
    complication: CLKComplication,  
    withHandler handler: (NSDate?) -> Void)
```











# Complication Data Source

Placeholder templates



Clock  
Face



Extension

# Complication Data Source

Placeholder templates



# Complication Data Source

Placeholder templates



# Complication Data Source

Placeholder templates

# Complication Data Source

## Placeholder templates

```
func getPlaceholderTemplateForComplication(  
    complication: CLKComplication,  
    withHandler handler: (CLKComplicationTemplate?) -> Void)
```

# Complication Data Source

## Placeholder templates

```
func getPlaceholderTemplateForComplication(  
    complication: CLKComplication,  
    withHandler handler: (CLKComplicationTemplate?) -> Void)
```

# Complication Data Source

## Placeholder templates

```
func getPlaceholderTemplateForComplication(  
    complication: CLKComplication,  
    withHandler handler: (CLKComplicationTemplate?) -> Void)
```

*Demo*

Building a complication



Keeping Your Data Fresh

# Data Lifecycle



# Data Lifecycle



WatchConnectivity

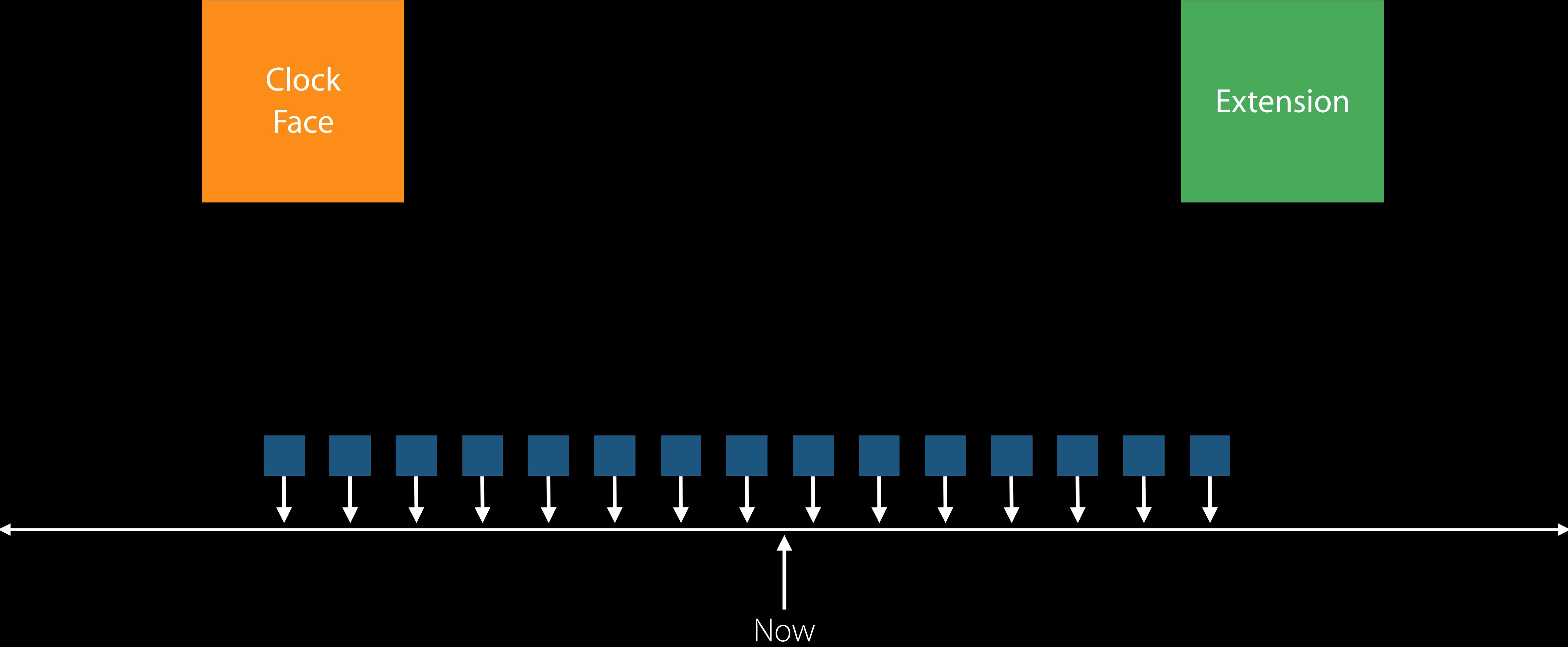




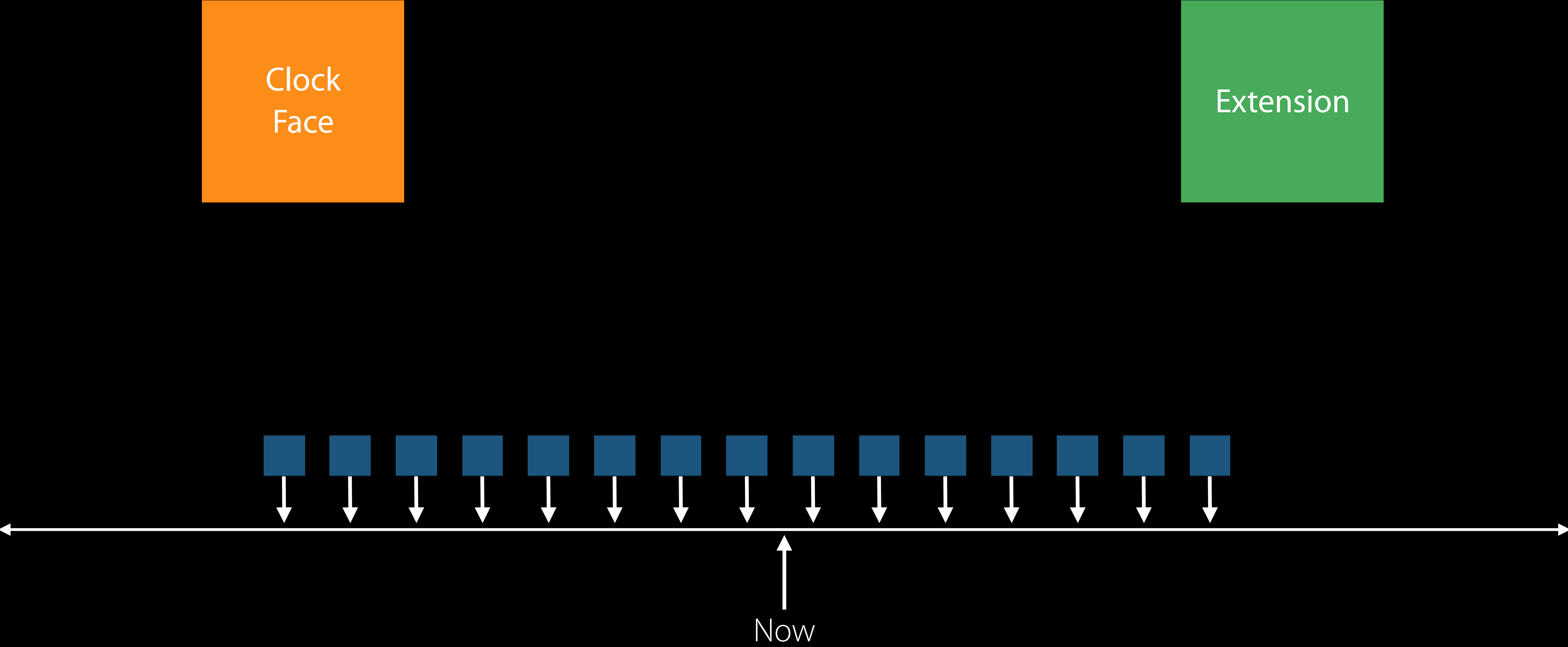
# Data Lifecycle



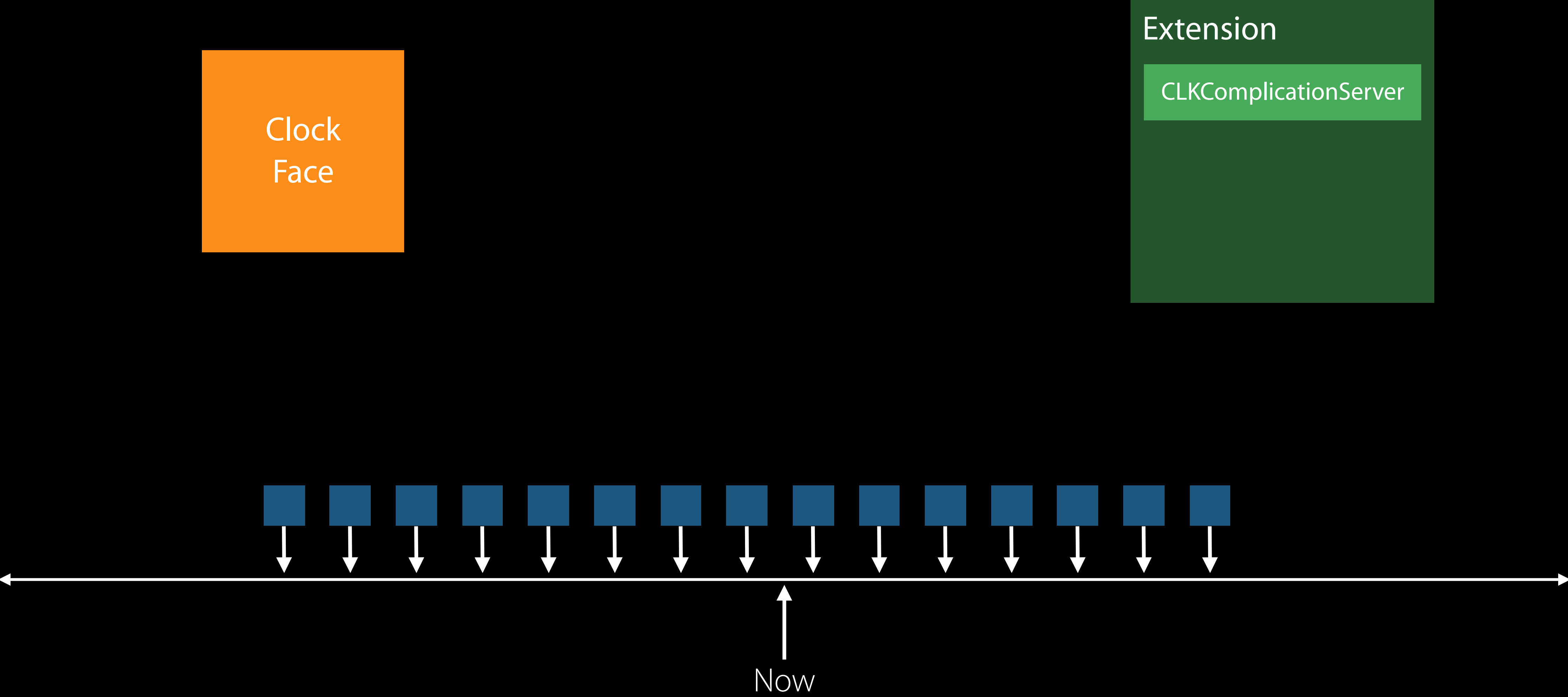
# Reload Timeline



# Reload Timeline



# Reload Timeline

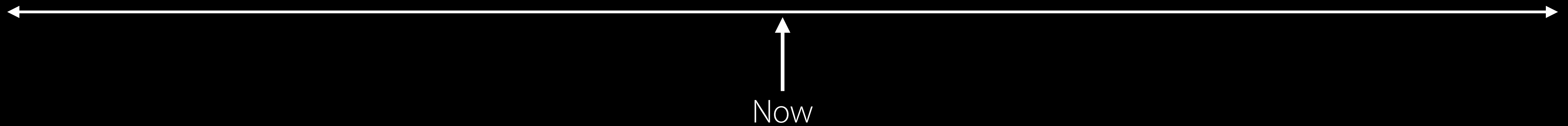
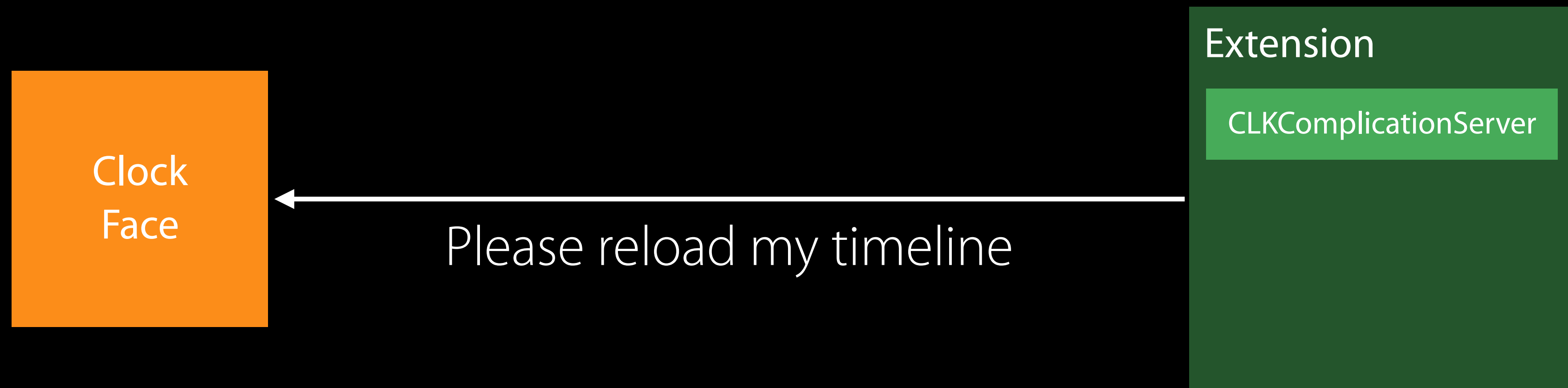


# Reload Timeline

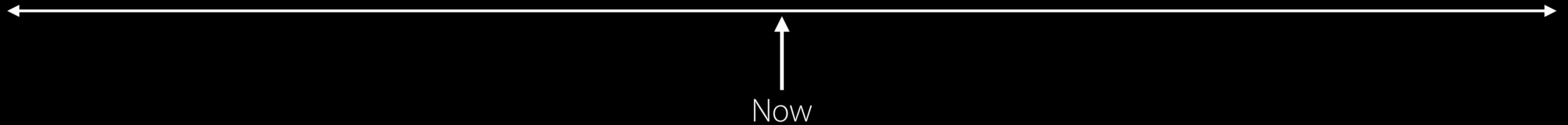
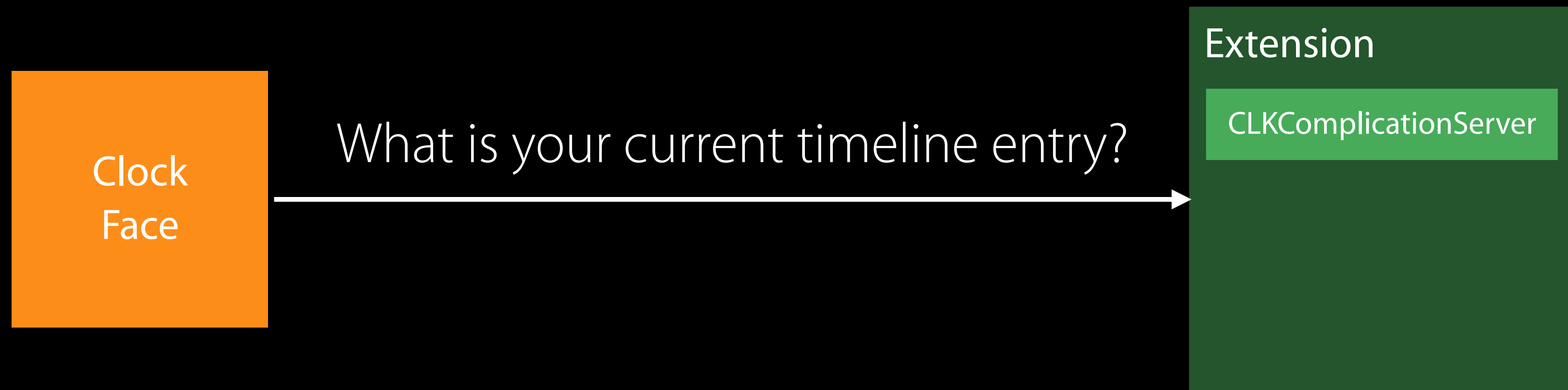




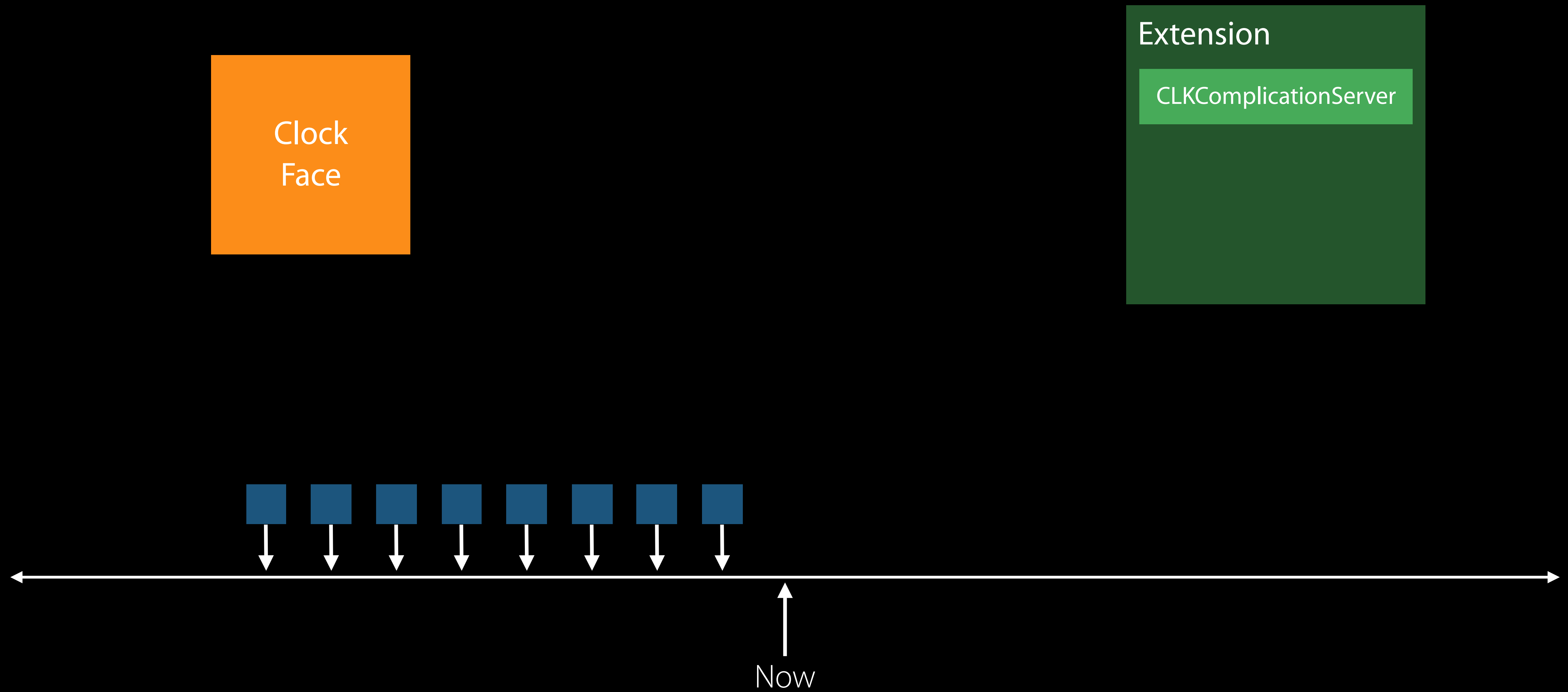
# Reload Timeline



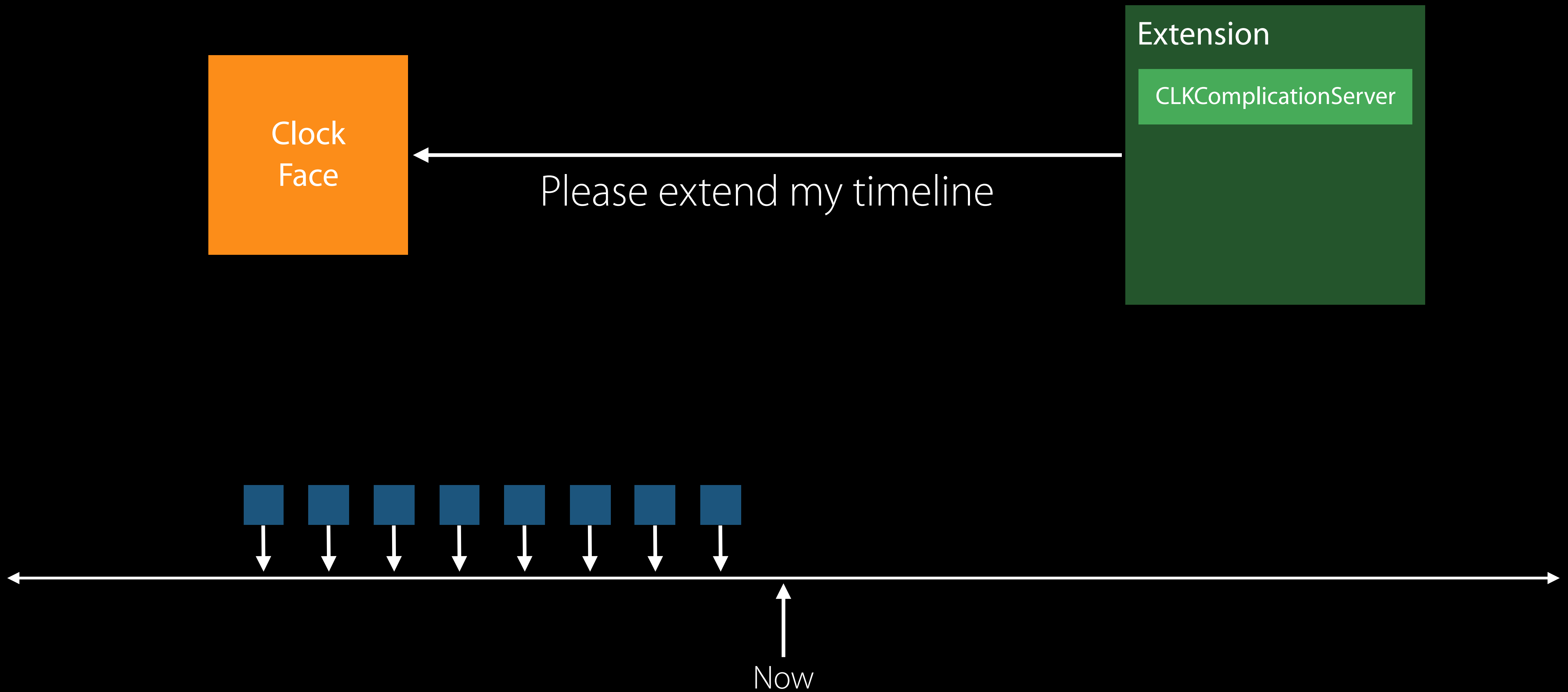
# Reload Timeline



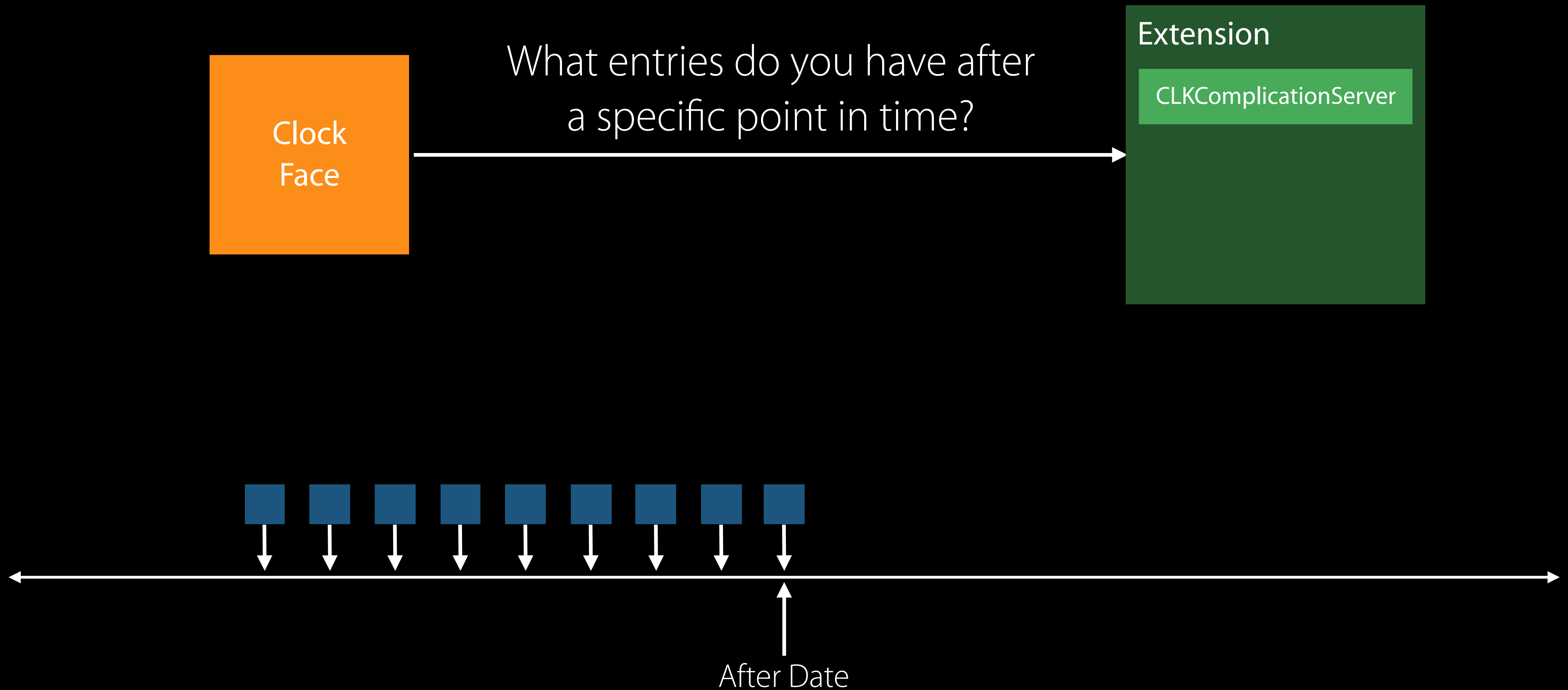
# Extend Timeline



# Extend Timeline



# Extend Timeline



# Reload or Extend

```
// get the shared instance
```

```
let server = CLKComplicationServer.sharedInstance()
```

# Reload or Extend

```
// get the shared instance
let server = CLKComplicationServer.sharedInstance()

// extend the timeline for all complications
for complication in server.activeComplications {

}
```

# Reload or Extend

```
// get the shared instance
let server = CLKComplicationServer.sharedInstance()

// extend the timeline for all complications
for complication in server.activeComplications {
    server.extendTimelineForComplication(complication)
}
```



# Reload or Extend

```
// get the shared instance
```

```
let server = CLKComplicationServer.sharedInstance()
```

```
// extend the timeline for all complications
```

```
for complication in server.activeComplications {  
    server.extendTimelineForComplication(complication)  
}
```

---

```
// reload the timeline for all complications
```

```
for complication in server.activeComplications {  
    server.reloadTimelineForComplication(complication)  
}
```

# When Can You Update?

# When Can You Update?

Whenever extension runs

# When Can You Update?

Whenever extension runs

- When Watch app runs

# When Can You Update?

Whenever extension runs

- When Watch app runs
- Locally scheduled wake

# When Can You Update?

Whenever extension runs

- When Watch app runs
- Locally scheduled wake
- iPhone initiated wake

# When Can You Update?

Whenever extension runs

- When Watch app runs
- Locally scheduled wake Budgeted
- iPhone initiated wake Budgeted

# When Can You Update?

Whenever extension runs

- When Watch app runs
- Locally scheduled wake Budgeted
- iPhone initiated wake Budgeted



# CLKComplicationDataSource

One more call

```
func getNextRequestedUpdateDateWithHandler(  
    handler: (NSDate?) -> Void)
```

# CLKComplicationDataSource

One more call

```
func getNextRequestedUpdateDateWithHandler(  
    handler: (NSDate?) -> Void)
```

One date returned across all active complications

# CLKComplicationDataSource

One more call

```
func getNextRequestedUpdateDateWithHandler(  
    handler: (NSDate?) -> Void)
```

One date returned across all active complications

Hint for budgeted background launch of extension

# CLKComplicationDataSource

One more call

```
func getNextRequestedUpdateDateWithHandler(  
    handler: (NSDate?) -> Void)
```

One date returned across all active complications

Hint for budgeted background launch of extension

Allows data delegate callbacks

# Summary

# Summary

Build timeline of templates and providers

# Summary

Build timeline of templates and providers

Take advantage of formatting and fitting

# Summary

Build timeline of templates and providers

Take advantage of formatting and fitting

Refresh data using delegate events



# Summary

Build timeline of templates and providers

Take advantage of formatting and fitting

Refresh data using delegate events

Do less work to run more

# More Information

## Documentation

watchOS 2 Transition Guide

WatchKit Programming Guide

## Sample Code

Lister

WatchKit Catalog

<http://developer.apple.com/watchOS>

## Technical Support

Apple Developer Forums

Developer Technical Support

## General Inquiries

Jake Behrens, watchOS Frameworks Evangelist

[behrens@apple.com](mailto:behrens@apple.com)

# Related Sessions

Introducing Watch Connectivity	Pacific Heights	Thursday 11:00AM
WatchKit Tips and Tricks	Presidio	Friday 10:00AM
Introducing WatchKit for watchOS 2	Presidio	Tuesday 10:00AM
Building Watch Apps	Pacific Heights	Tuesday 4:30PM
WatchKit In-Depth, Part 1	Pacific Heights	Wednesday 9:00AM
WatchKit In-Depth, Part 2	Pacific Heights	Wednesday 10:00AM

# Related Labs

---

ClockKit Complications Lab

Frameworks Lab D   Wednesday 1:30PM

---

Watch Connectivity Lab

Frameworks Lab B   Thursday 1:30PM

---

WatchKit and ClockKit Complications Lab

Frameworks Lab A   Friday 1:30PM

---

