

WidgetKit

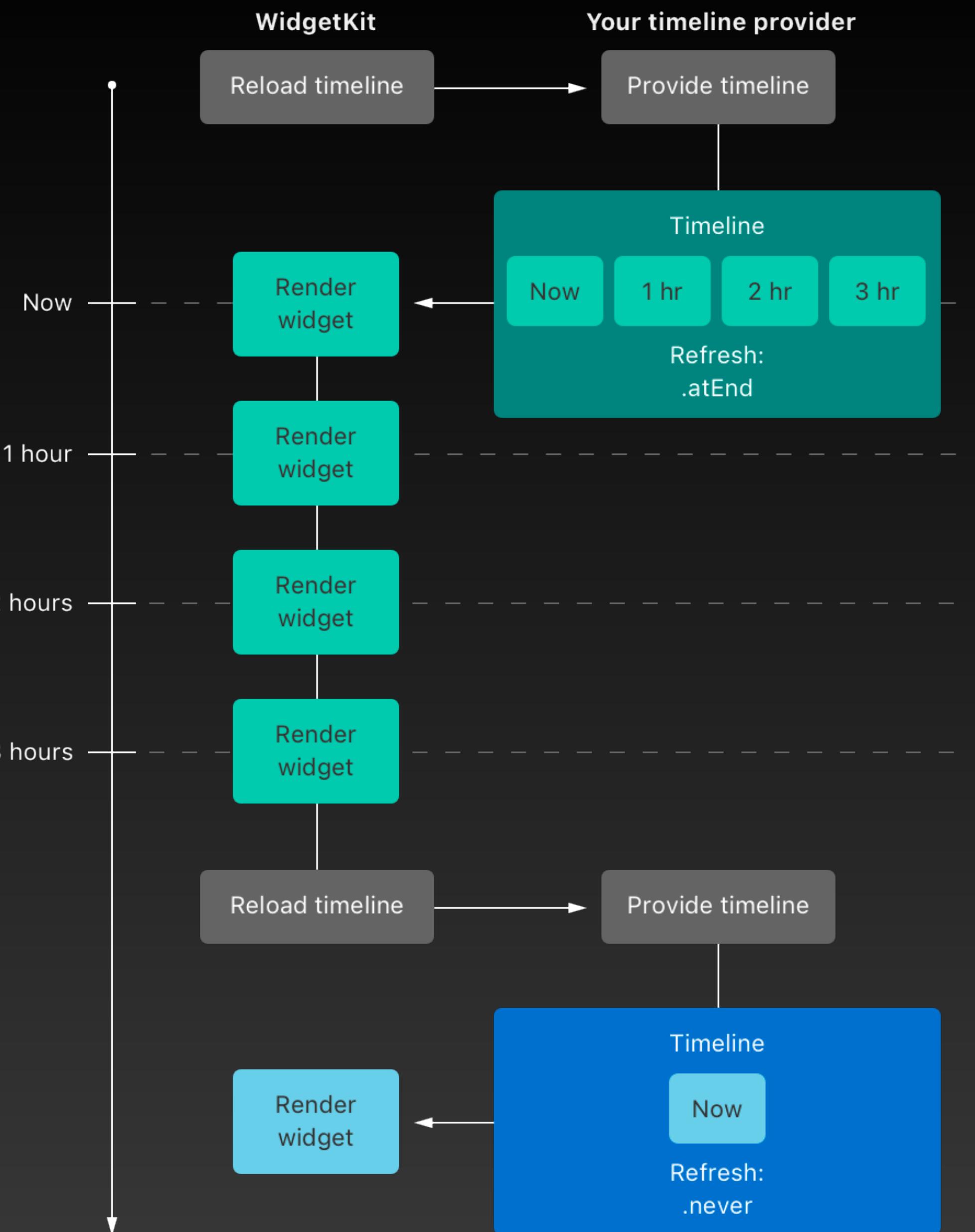
Exploring Interactive Widget

Bibek Timalsina

Agenda

- Widget recap
- Interactive widget
- Demo





Widget Recap

- WidgetKit extension are Background extension
- Timeline allows views to be ready up front
 - Can refresh from main app
 - Schedule updates from extension
- StatelessWidget
- Deeplinks/Intents

Interactive Widget

- iOS 17+ & Xcode 15.0 beta
- Only Button and Toggle
- AppIntent
- StatelessWidget
- Preview widgets with #Preview macro
- New widget Family

Demo

Q & A

```
struct SwiftUIWidgetExtension: Widget {
    let kind: String = "SwiftUIWidgetExtension"

    var body: some WidgetConfiguration {
        StaticConfiguration(
            kind: kind,
            provider: Provider()
        ) { entry in
            SwiftUIWidgetExtensionEntryView(entry: entry)
                .containerBackground(.fill.tertiary, for: .widget)
        }.supportedFamilies([
            .accessoryCircular,
            .accessoryInline,
            .accessoryRectangular,
            .systemSmall,
            .systemMedium,
            .systemLarge
        ])
    }
}
```

```
struct SwiftUIWidgetExtensionEntryView : View {
    var entry: Provider.Entry

    @Environment(\.widgetFamily) var family

    var body: some View {
        switch family {
            case .accessoryInline:
                Text(entry.text)
                    .font(.footnote)
                    .widgetAccentable()
            case .systemMedium:
                VStack {
                    VStack {
                        Spacer()
                        Text(entry.text)
                        Spacer()
                    }
                    if entry.type != .none {
                        ZStack {
                            ActionButtonsView(
                                isFavourite: entry.type == .favourite,
                                quote: entry.text
                            )
                            HStack {
                                Spacer()
                                Button(intent: EmptyIntent() ) {
                                    Image(systemName: "shuffle")
                                }
                            }
                        }
                    }
                }
            default :
                Text(entry.text)
                    .widgetAccentable()
```

```
struct LikeIntent: AppIntent {
    static var title: LocalizedStringResource = "Like the quote"
    static var description = IntentDescription("Use this to denote you like the quote")

    @Parameter(title: "quote")
    var quote: String

    init(quote: String) {
        self.quote = quote
    }

    init() {}

    func perform() async throws -> some IntentResult {
        guard let userDefault = UserDefaults(suiteName: "group.widgetdatacache.com"),
              var quotes = userDefault.dictionary(forKey: "quotes") as? [String: Int],
              let weight = quotes[quote] else {
            return .result()
        }
        quotes[quote] = weight + 1
        userDefault.setValue(quotes, forKey: "quotes")
        return .result()
    }
}
```

```
ZStack {
    HStack {
        if entry.type != .must {
            Button(intent: LikeIntent(quote: entry.text)) {
                Text("👍")
            }
            Button(intent: DislikeIntent(quote: entry.text)) {
                Text("👎")
            }
        }
        Toggle("❤️", isOn: entry.type == .must, intent: MustShowIntent(quote: entry.text))
    }
    HStack {
        Spacer()
        Button(intent: EmptyIntent()) {
            Image(systemName: "shuffle")
        }
    }
}
```

```
176 #Preview(as: .systemMedium, widget: {
177     SwiftUIWidgetExtension()
178 }, timeline: {
179     previewEntries[0]
180     previewEntries[1]
181     previewEntries[2]
182 })
183
184 #Preview(as: .systemMedium) {
185     SwiftUIWidgetExtension()
186 } timeline: {
187     previewEntries[0]
188     previewEntries[1]
189     previewEntries[2]
190 }
191
192 #Preview(as: .accessoryInline) {
193     SwiftUIWidgetExtension()
194 } timeline: {
195     previewEntries[0]
196     previewEntries[1]
197     previewEntries[2]
198 }
199
200 #Preview(as: .accessoryRectangular) {
201     SwiftUIWidgetExtension()
202 } timeline: {
203     previewEntries[0]
204     previewEntries[1]
205     previewEntries[2]
206 }
207
208 #Preview(as: .systemLarge) {
209     SwiftUIWidgetExtension()
210 } timeline: {
211     previewEntries[0]
212     previewEntries[1]
213     previewEntries[2]
214 }
215
```

The screenshot shows the Xcode interface with a code editor on the left and a preview area on the right. The preview area displays a floating accessory view with a white background and rounded corners. Inside the view, the text "Life itself is the most wonderful fairy tale." is centered. Below the text are four small circular icons: a thumbs up, a thumbs down, a heart, and a share symbol. At the bottom of the screen, there are three smaller preview cards labeled "Entry 1", "Entry 2", and "Entry 3". Each card contains a snippet of text and a set of interaction icons. The top of the screen shows a toolbar with various preview options and a device selection dropdown set to "Automatic - iPhone 14".