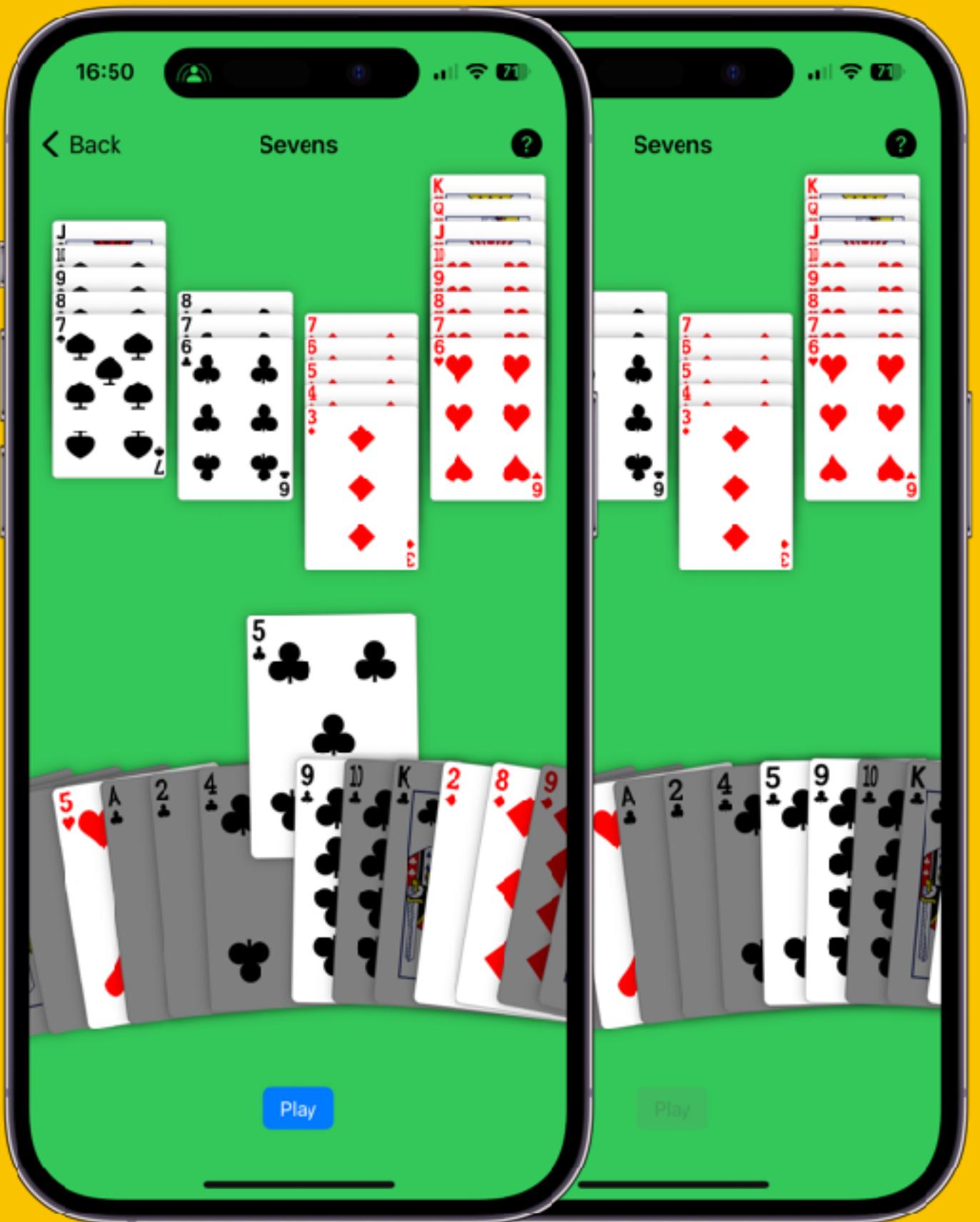


Turn-based games using SharePlay

iOSDevUK 11 : Sep. 2023



Today's Talk

SHAREPLAY AND GROUPACTIVITIES API

?

What is SharePlay?

?

How to use SharePlay in your apps

?

Deep dive on GroupActivities API

- Collaboration

- Streaming

- Turn-based games

?

Disclaimer...

?

GameCenter

?

GKMatchmaker
.startGroupActivity



About me

LEAD MOBILE DEVELOPER

 oliver_binns@mastodon.social

 ~~oliver_binns~~

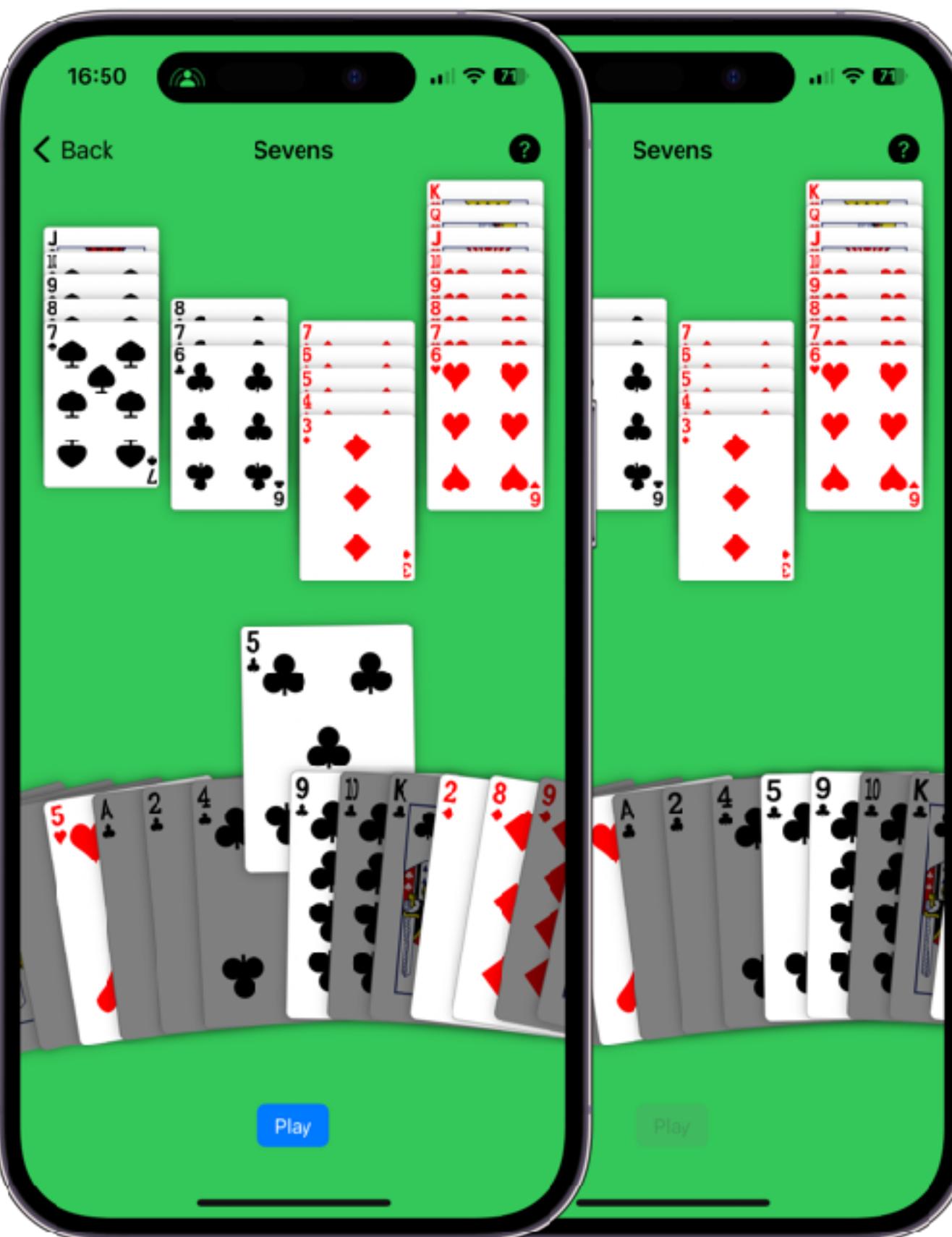
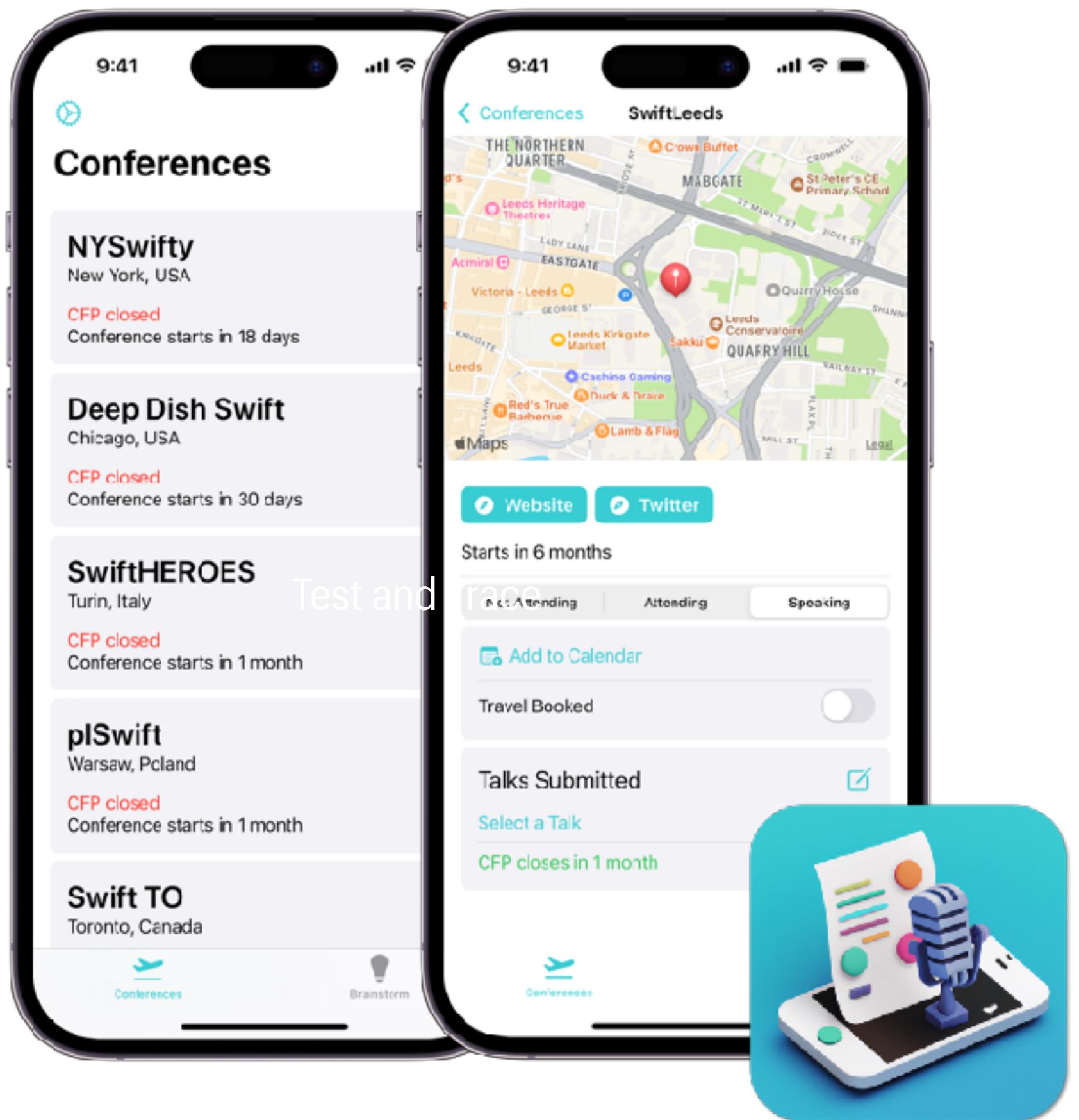
 oliver-binns

 oliverbinns.co.uk

 obinns



Some of my work



C?RD
S

What is SharePlay?

“GROUPACTIVITY API”

② Real-time shared experiences

Multiple devices – simultaneously ②

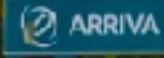
② Works across platforms

iOS ②, iPadOS ②, macOS ②, tvOS ②, web! ②

② Privacy-centric: Apple does not see any app data

LLANFAIRPWLLGWYNGYLLGOGERYCHWYRNDROBWLLLANTYSILIOGOGOGOCH

Llan-vire-pooll-guin-gill-go-ger-u-queern-drob-ooll-llandus-ilio-gogo-goch



Arriva Trains Wales / Trenau Arriva Cymru

Way Out ←

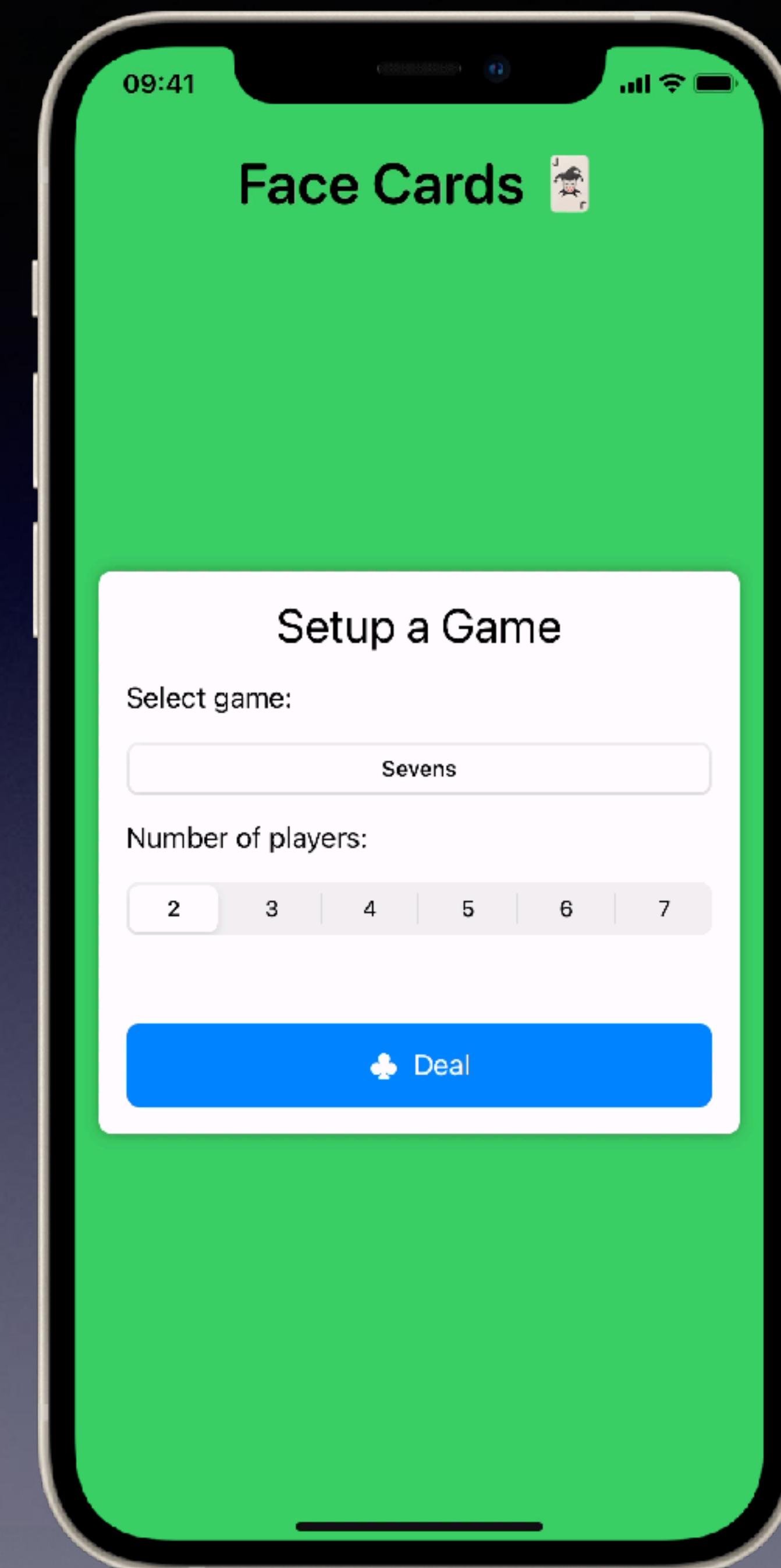
Holyhead →

Works with many apps

FIRST AND THIRD PARTY



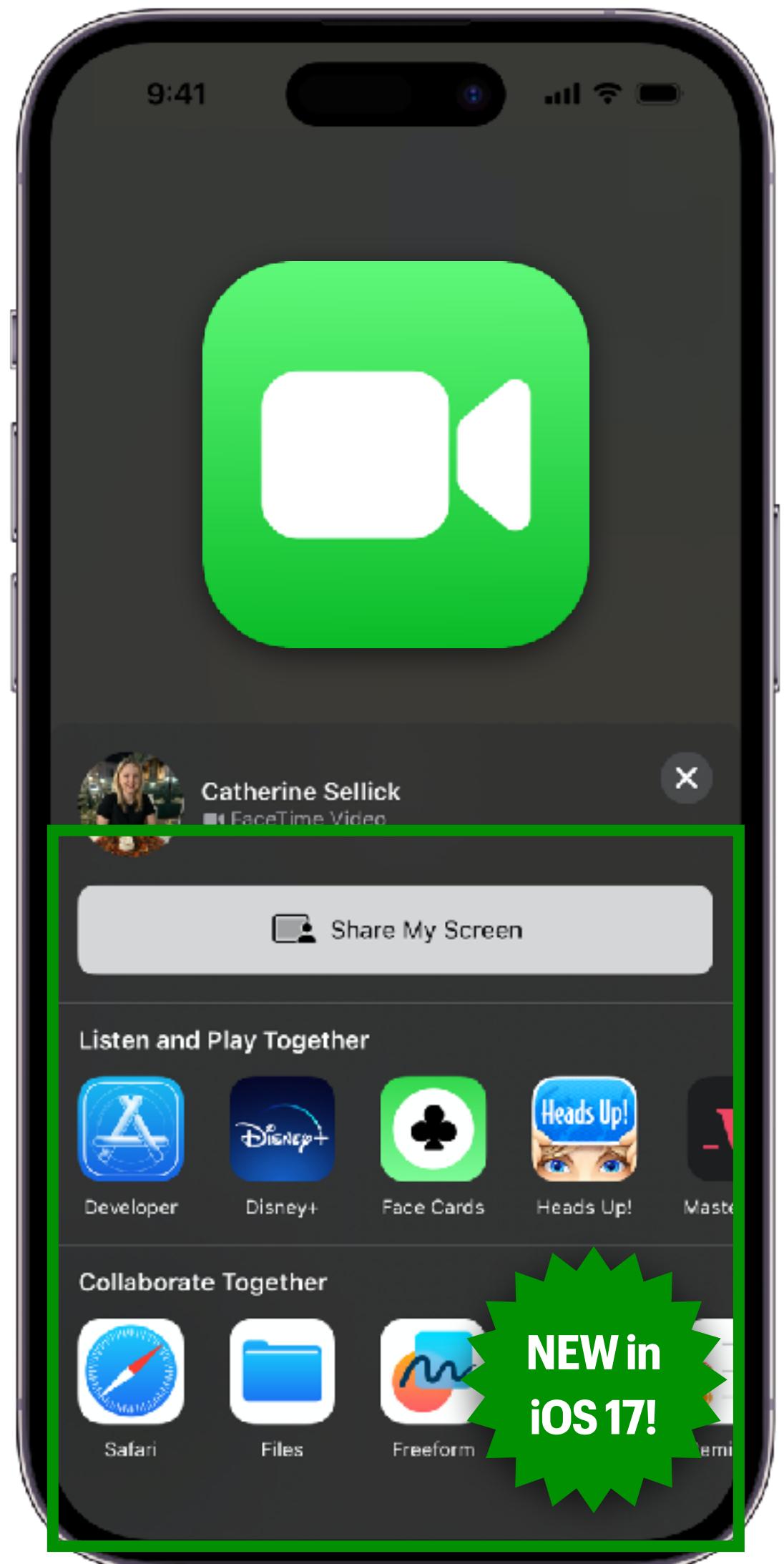
Demo



Use it in a FaceTime call

INITIATE WITHIN THE APP OR OVER A CALL

Great for activities where live participation is essential.



Use it in Messages

INITIATE WITHIN AN APP

Great for remote collaboration.

You easily build Google Docs / Mural / Figma style collaboration into your app, with Apple doing the heavy lifting.



Use it in person

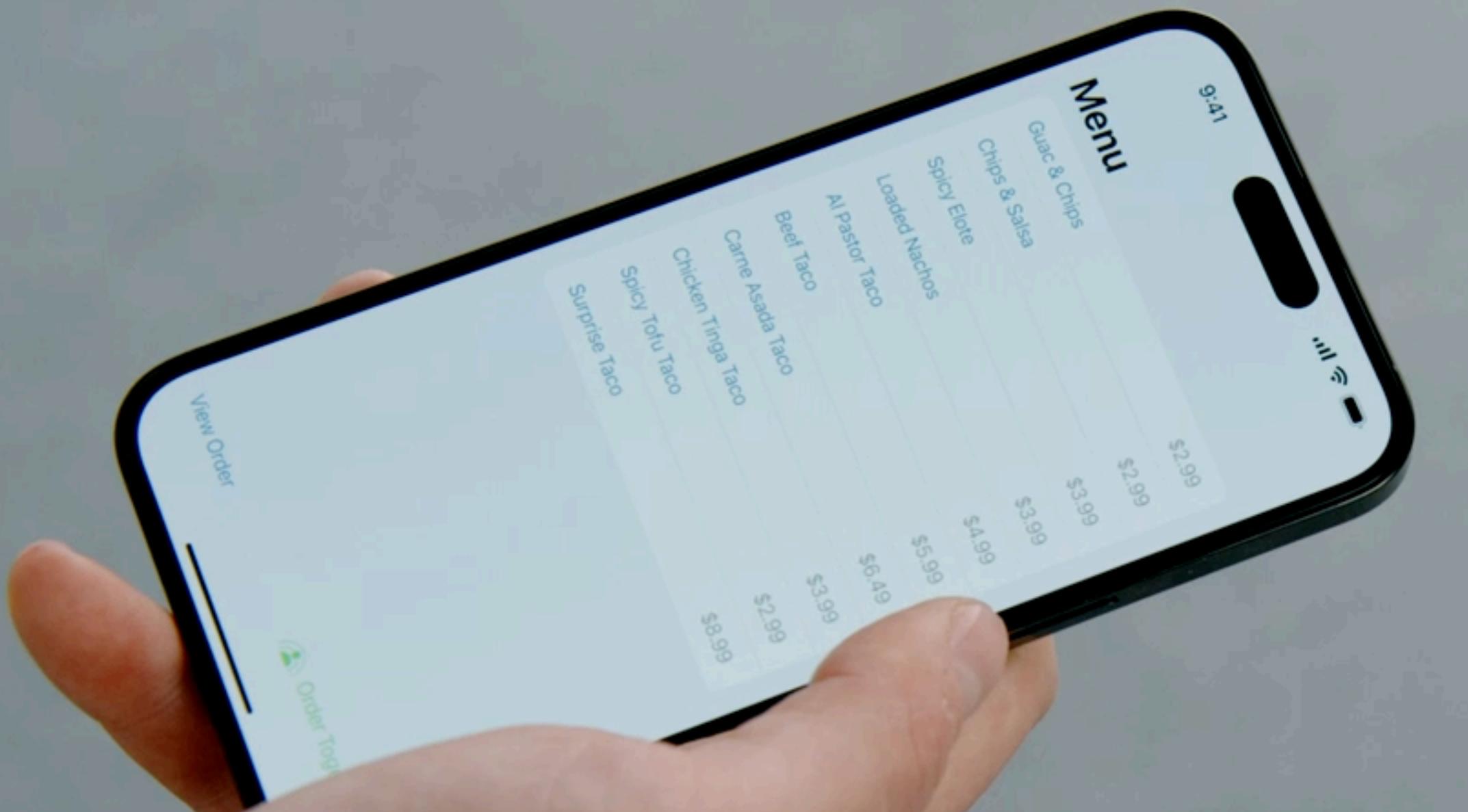
PLACE TWO DEVICES NEXT TO EACH OTHER

Apple are using this for their own features:

- SharePlay in the CarPlay Music app using QR

AirDrop to initiate a session





View Order

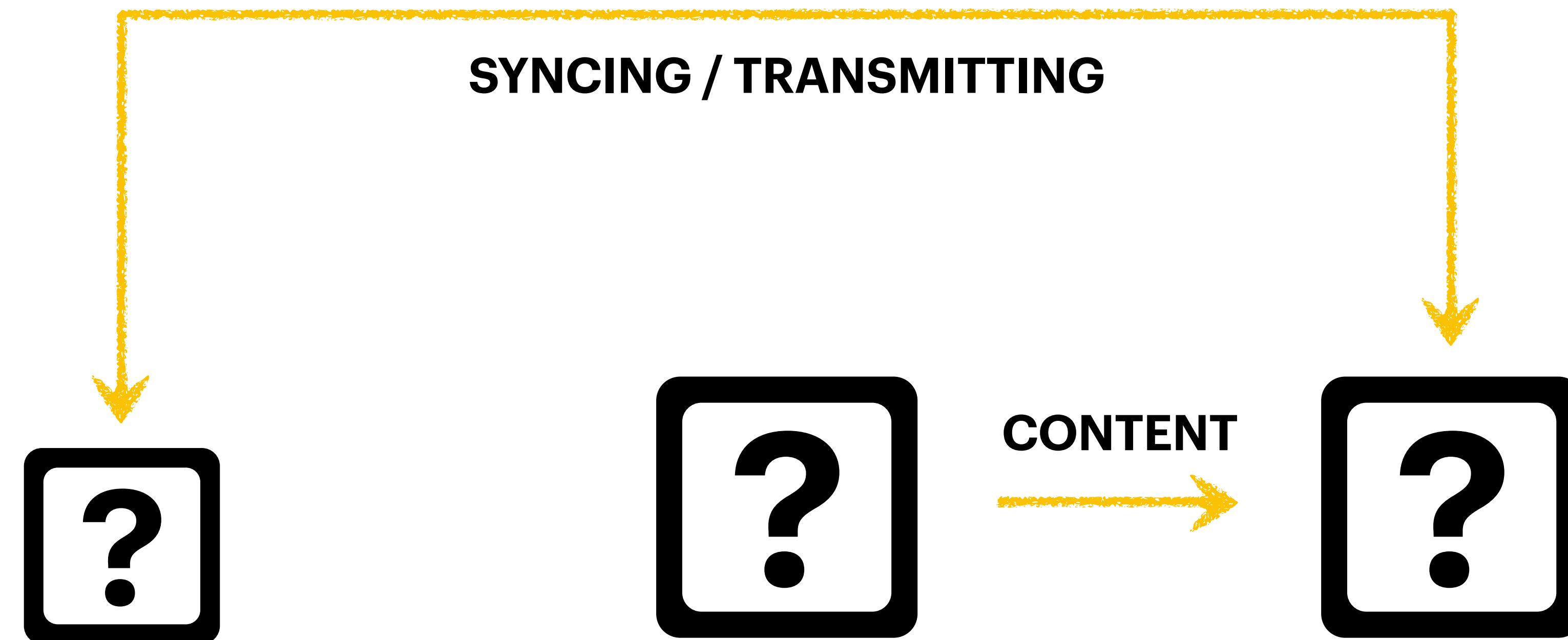


Order Today

Guac & Chips	\$2.99
Chips & Salsa	\$2.99
Spicy Elote	\$3.99
Loaded Nachos	\$3.99
Al Pastor Taco	\$4.99
Beef Taco	\$5.99
Carne Asada Taco	\$6.49
Chicken Tinga Taco	\$6.49
Spicy Tofu Taco	\$8.99
Surprise Taco	

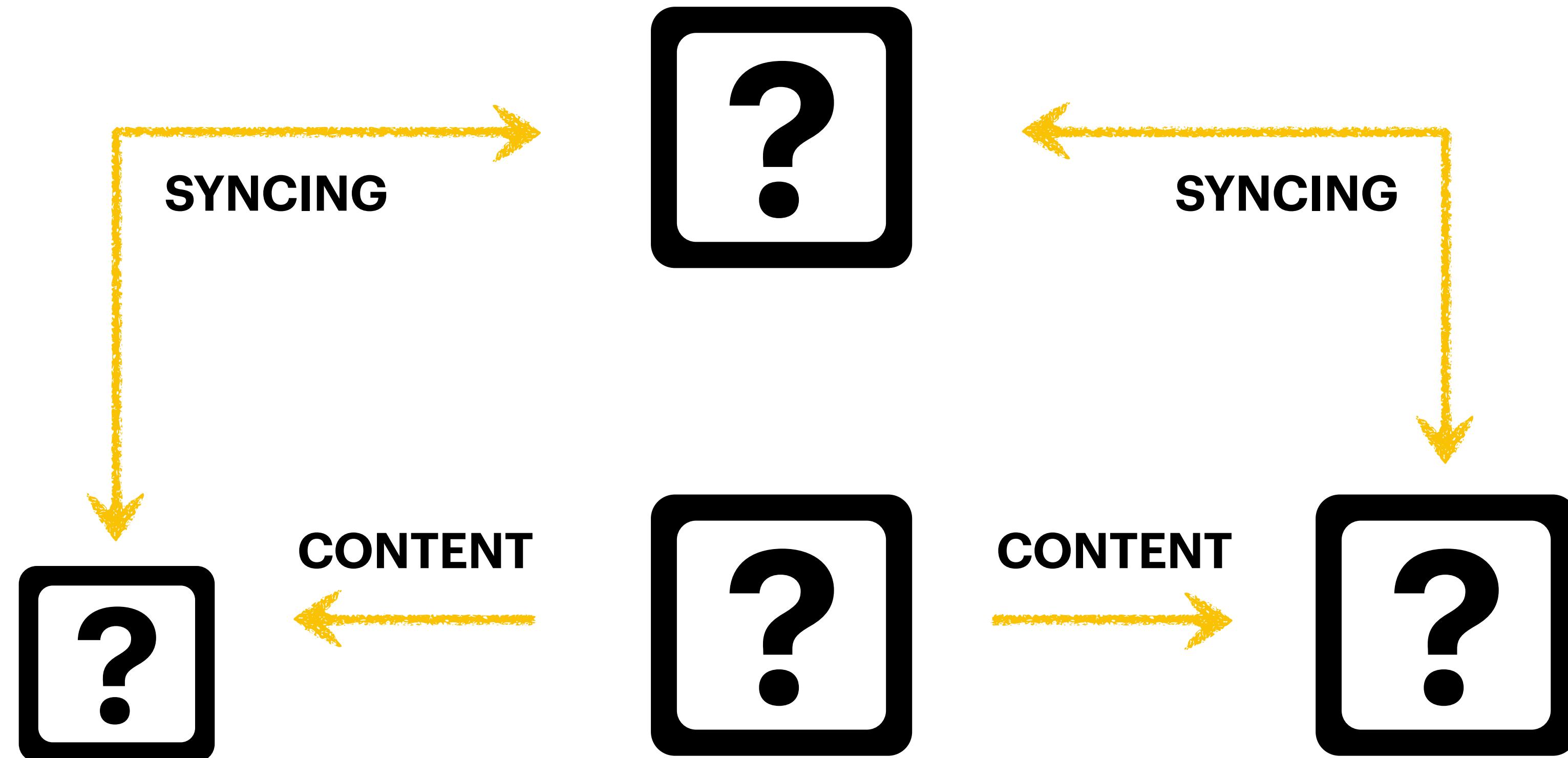
SharePlay is NOT screen sharing

SCREENSHARING SENDS CONTENT TO A DEVICE, THEN RETRANSMITS IT



SharePlay is NOT screen sharing

CONTENT STREAMS DIRECTLY FROM SERVER TO DEVICE

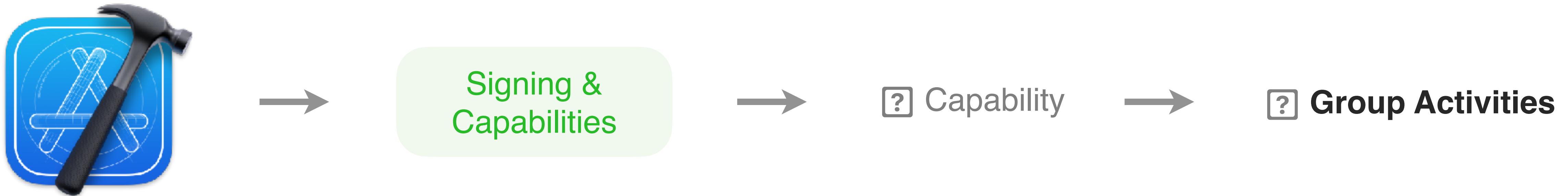




Oliver Binns

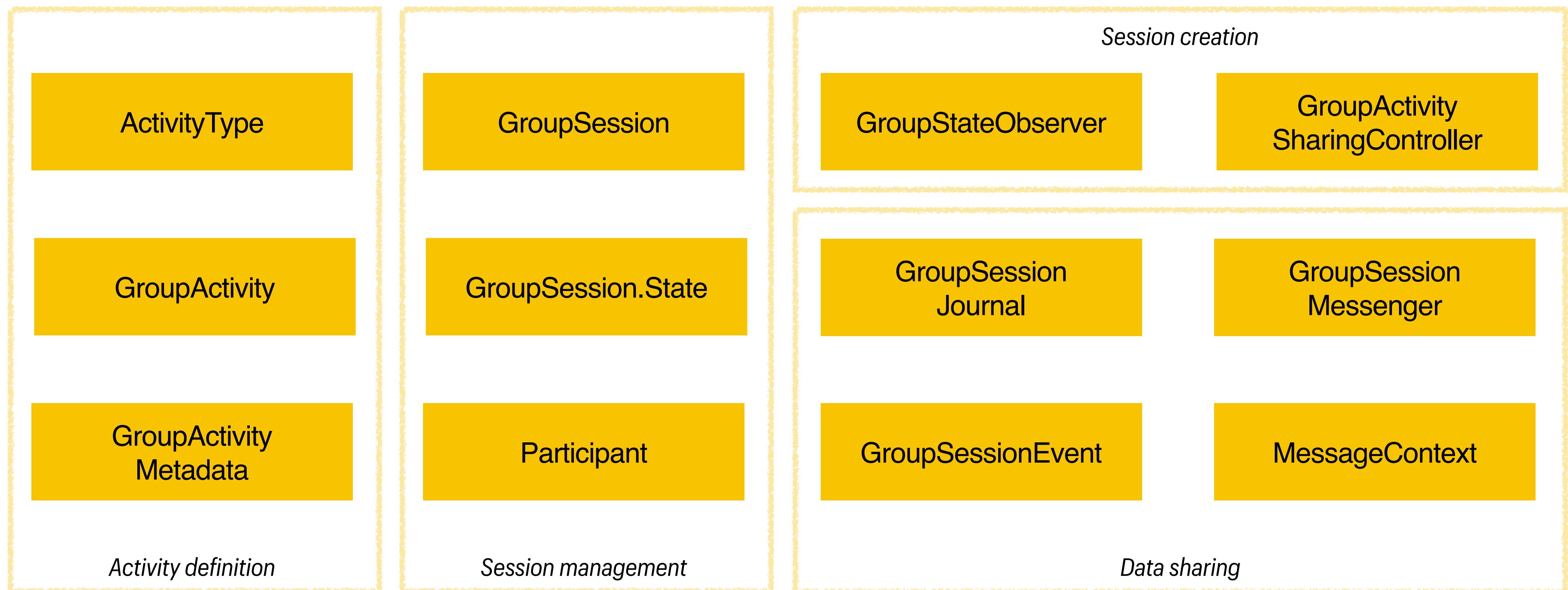
First – add the entitlement!

MAJOR HEALTH WARNING



GroupActivities API

HOW IT FITS TOGETHER



GroupActivities API

HOW IT FITS TOGETHER



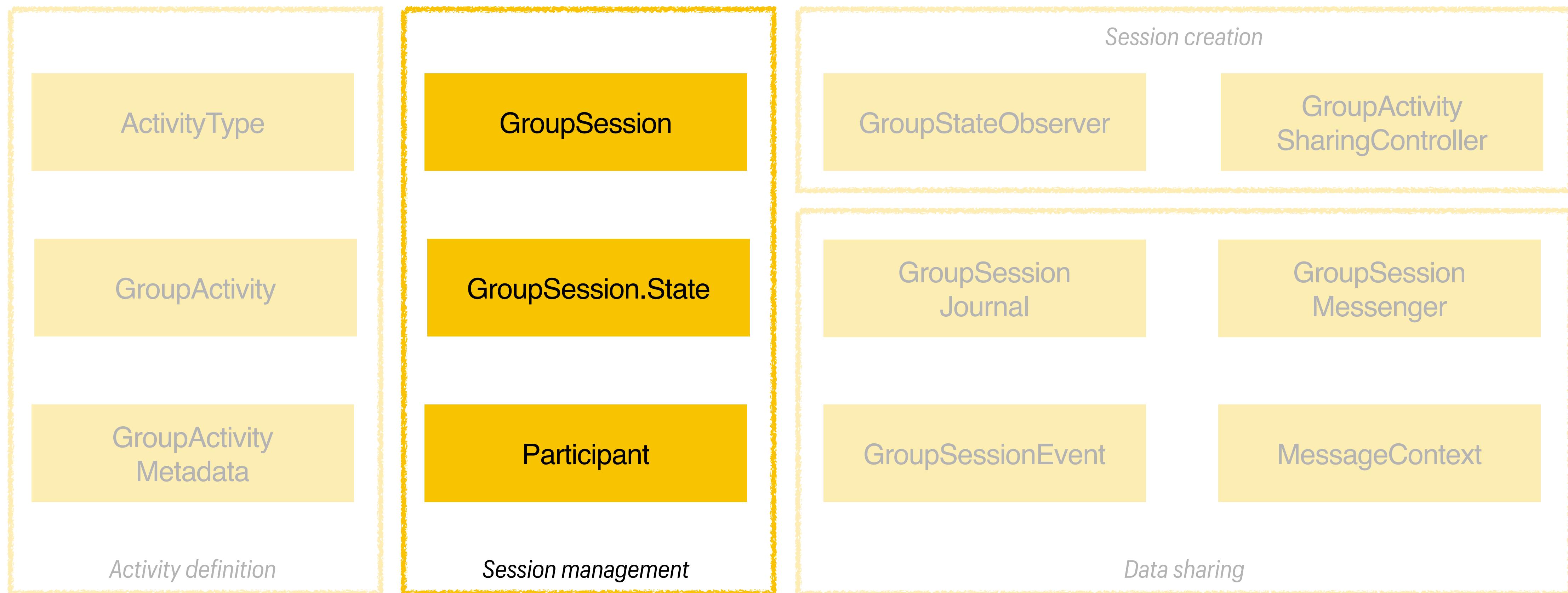
Joining a session...

“CAN YOU SEE MY SCREEN?”



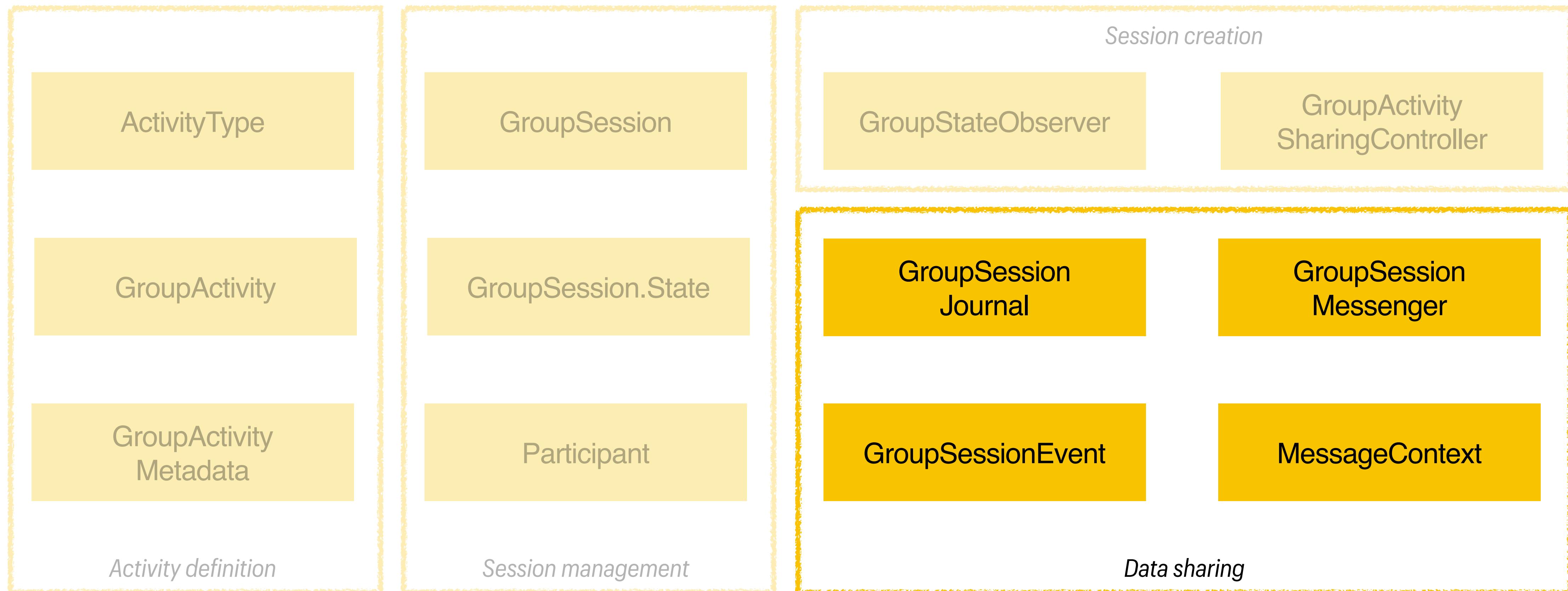
Looking after the session...

SESSION, STATE & OTHER PARTICIPANTS



Incoming...!

KEEPING STATE UP-TO-DATE: MESSAGES VS FILES*



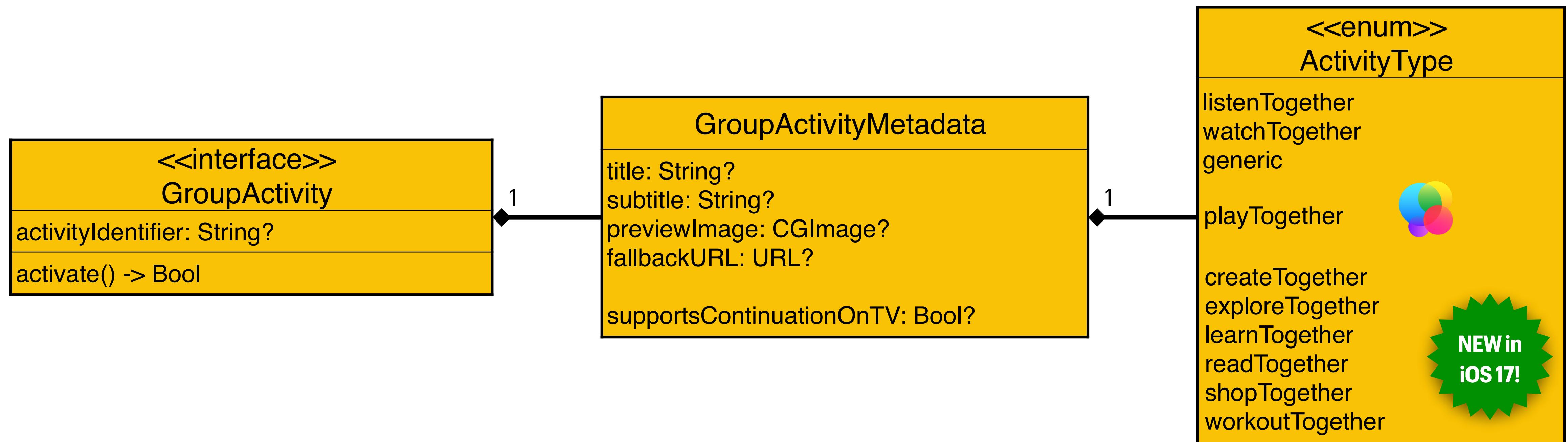
Activity definition

INVITE PARTICIPANTS TO SHARE AN ACTIVITY



Activity definition

INVITE PARTICIPANTS TO SHARE AN ACTIVITY



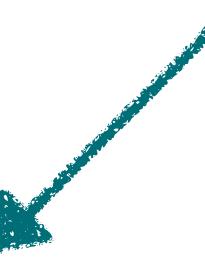
Activity definition

INVITE PARTICIPANTS TO SHARE AN ACTIVITY

Here's the `GroupActivity` protocol

```
protocol GroupActivity: Decodable, Encodable {  
    // An app-defined string that uniquely identifies the activity.  
    static var activityIdentifier: String { get set }  
  
    // A description of the activity, and optional image to display to the user.  
    var metadata: GroupActivityMetadata { get async }  
}  
  
struct GroupActivityMetadata: Decodable, Encodable {  
    var title: String?  
    var subtitle: String?  
    var previewImage: CGImage?  
    var fallbackURL: URL?  
}
```

Conforms to Codable



Activity definition

INVITE PARTICIPANTS TO SHARE AN ACTIVITY

Define a custom GroupActivity implementation

```
struct PlayTogether: GroupActivity {  
    // Store app-specific information about your activity  
    let game: Game  
}
```

```
extension PlayTogether {  
    // Provide information about the activity to the system  
    var metadata: GroupActivityMetadata {  
        ...  
    }  
}
```

Add your own custom properties

Activity definition

INVITE PARTICIPANTS TO SHARE AN ACTIVITY

Metadata is displayed as part of system UI when starting an activity

```
// Provide information about the activity to the system
var metadata: GroupActivityMetadata {
    var title: String? = "Face Cards 🎭"
    var subtitle: String? = "Let's play Sevens"
    var previewImage: CGImage?
    var fallbackURL: URL?

    var type: GroupActivityMetadata.ActivityType?
}
```



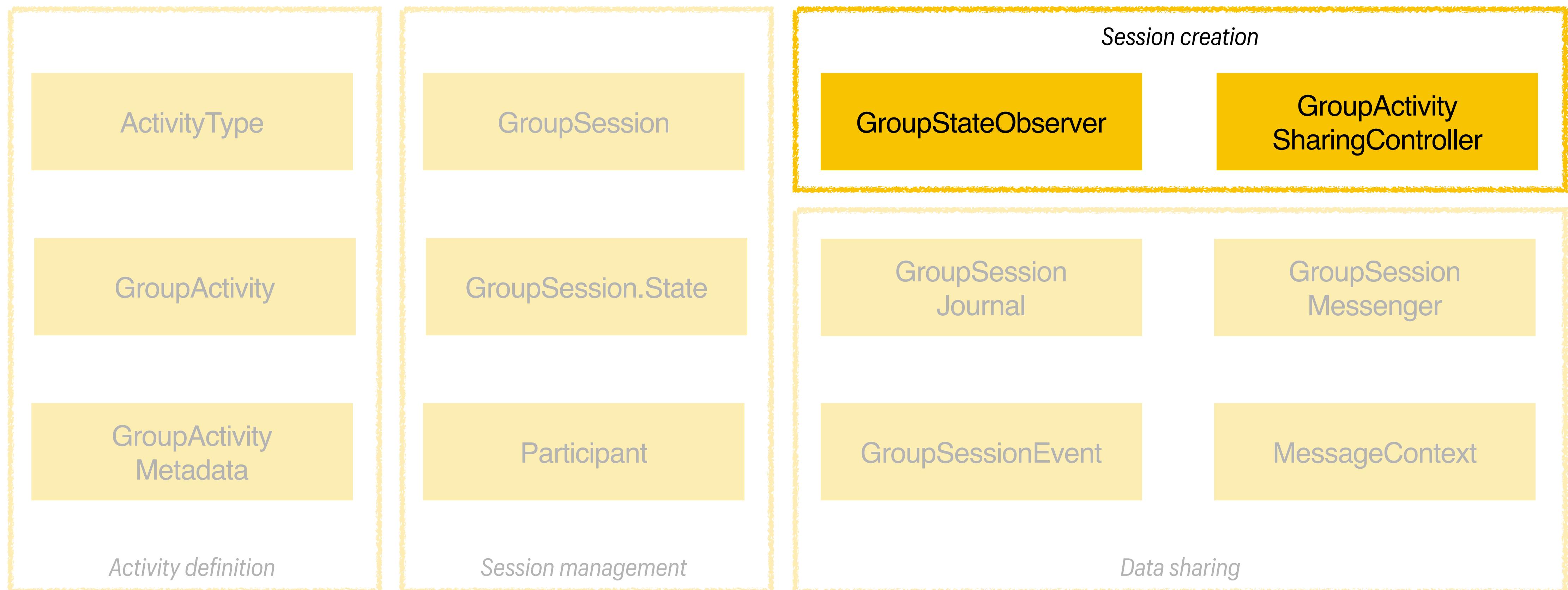
Joining a session...

“CAN YOU SEE MY SCREEN?”



Joining a session...

“CAN YOU SEE MY SCREEN?”



Joining a session...

ALREADY ON A FACETIME CALL? OR NEED TO CREATE A CONNECTION?

Check whether SharePlay is supported:

```
let observer = GroupStateObserver()  
// A Boolean value that indicates whether we can start a group session  
  
if observer.isEligibleForGroupSession {  
    // a connection is open — we can create a new GroupActivity  
  
} else {  
    // no active connection — we must start one to create a GroupActivity  
}
```

GroupActivities API

INVITE PARTICIPANTS TO SHARE AN ACTIVITY

Create an instance of the activity and activate it:

```
let activity = PlayTogether()  
let result = await activity.prepareForActivation()  
  
switch result {  
  case .activationPreferred: // the user wants to SharePlay  
    activity.activate() // - activate the activity  
  case .activationDisabled: // the user doesn't want to SharePlay  
    runActivityLocally() // - fallback to local, or show error  
  case .cancelled: // the user chose cancel  
    break // do nothing at all  
}
```

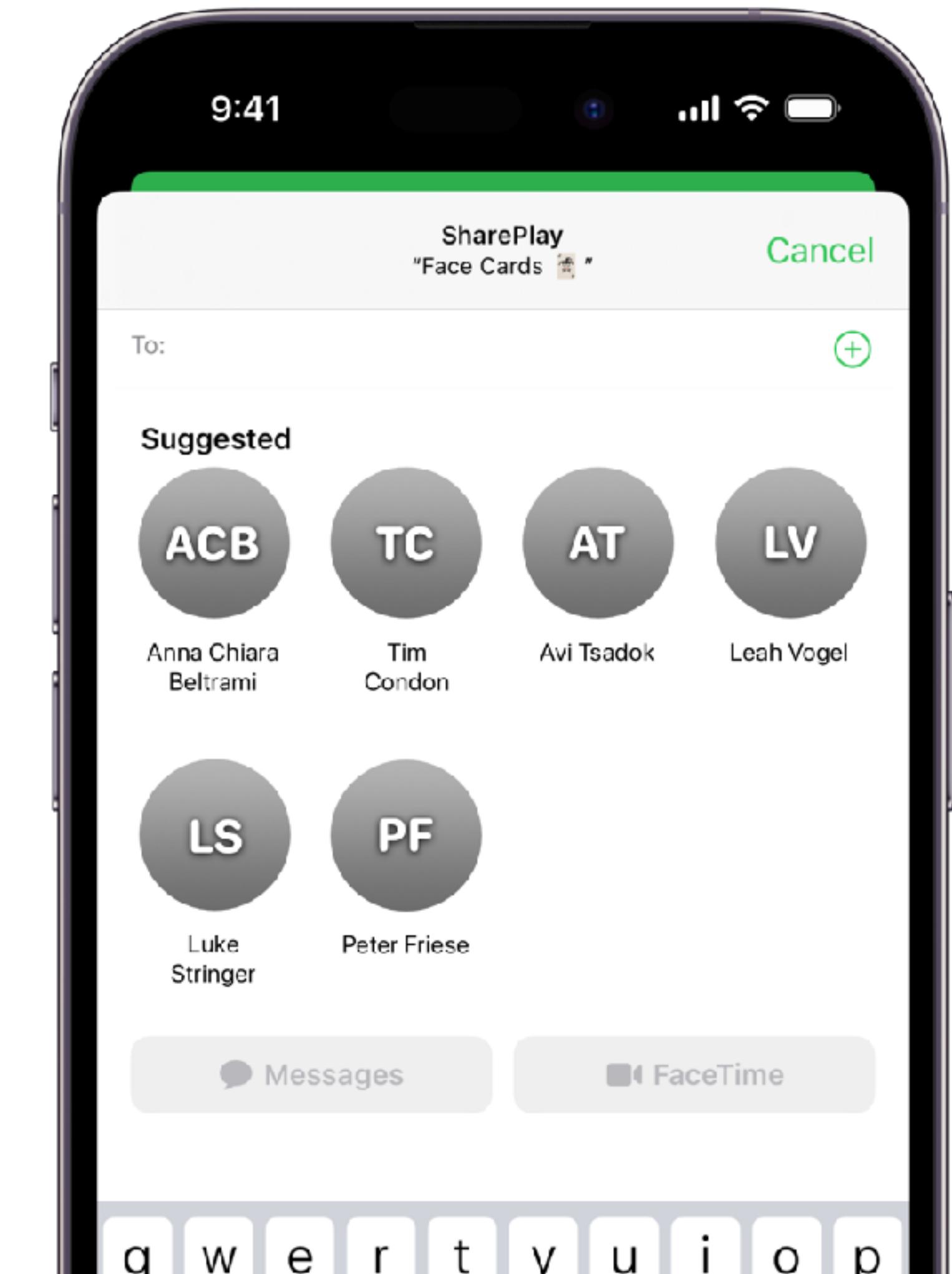
GroupActivitySharingController

START A SHAREPLAY SESSION WITHIN YOUR APP

Allow the user to start an Activity:

```
let activity = PlayTogether()
```

```
let vc = GroupActivitySharingController(activity)  
present(vc, animated: true)
```



Hold Devices together..!

CREATE A GROUPACTIVITYTRANSFERREPRESENTATION TYPE

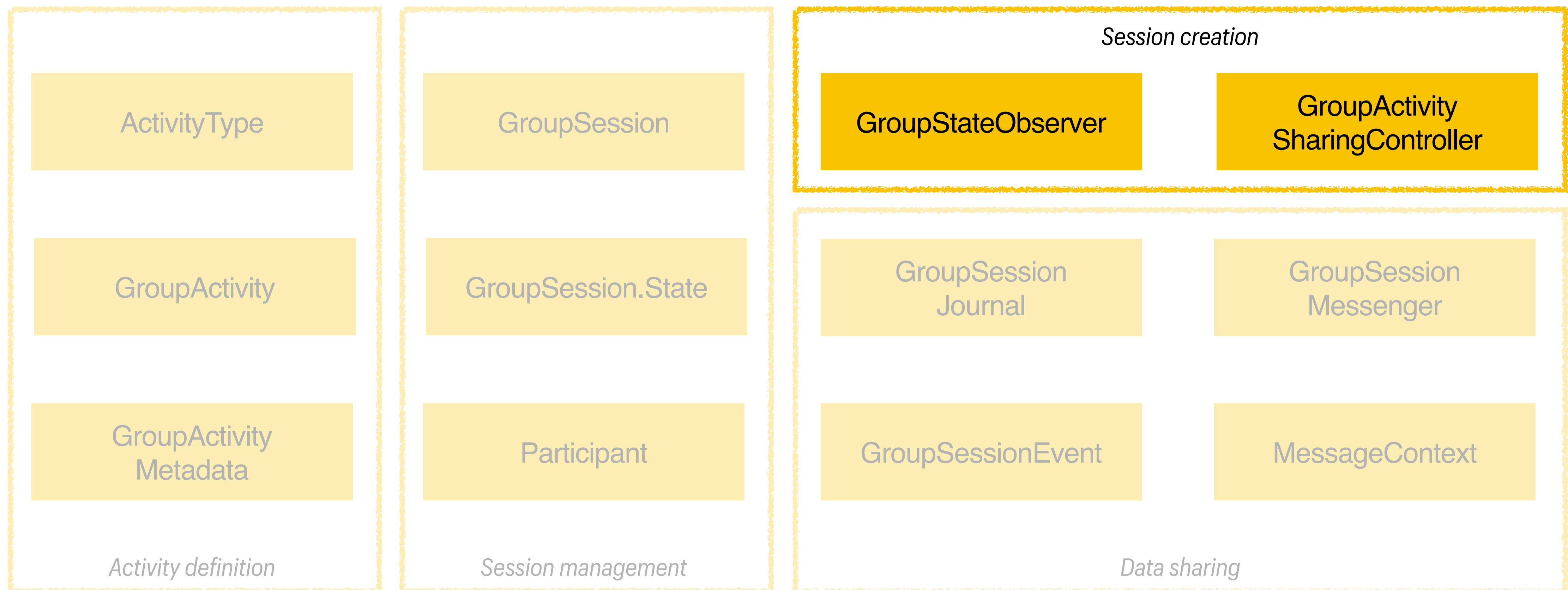
NEW in
iOS 17!

Hold devices together to start an Activity:

```
extension Sevens: Transferable {  
    public static var transferRepresentation: some TransferRepresentation {  
        GroupActivityTransferRepresentation { item in  
            PlayTogether(game: item)  
        }.visibility(.ownProcess)  
    }  
}  
  
// SwiftUI provides ShareLink which adds support for “touch to start”  
let preview = SharePreview("Sevens", image: Image("Preview Image"))  
ShareLink(item: game, preview: preview) .hidden()
```

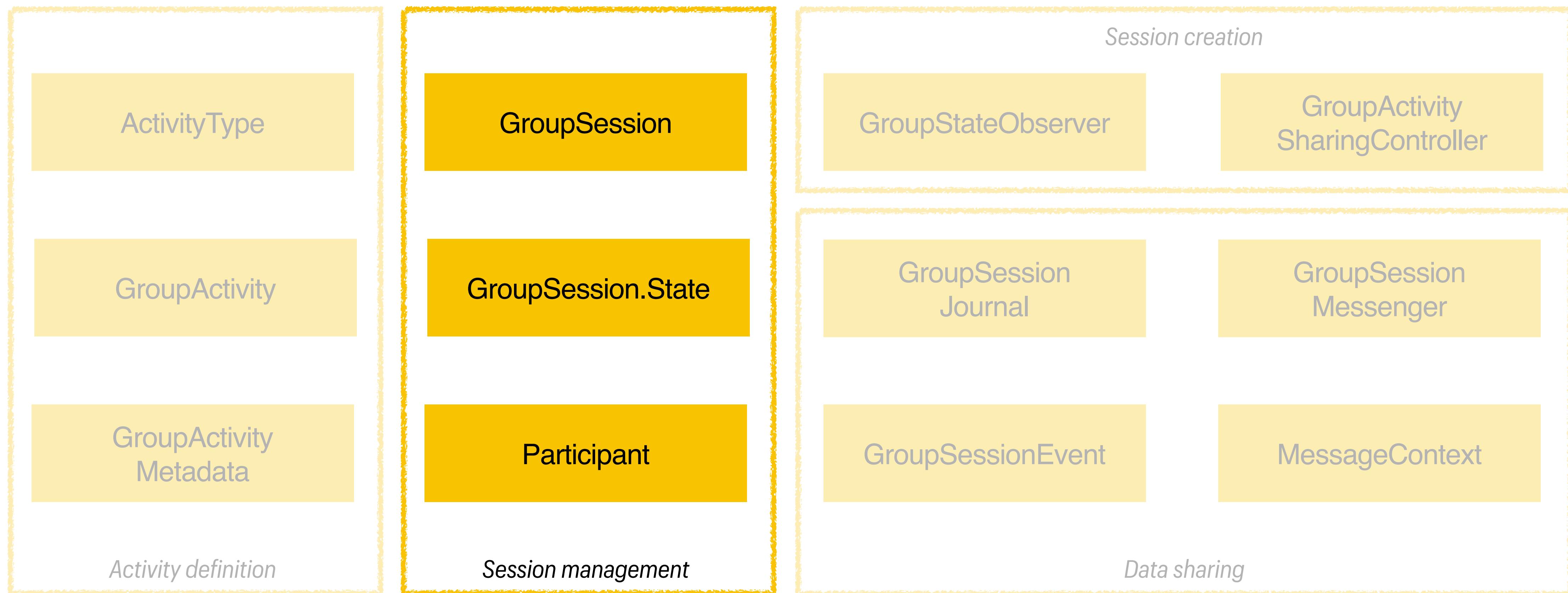
Looking after the session...

OTHER PARTICIPANTS, EVENTS AND STATE



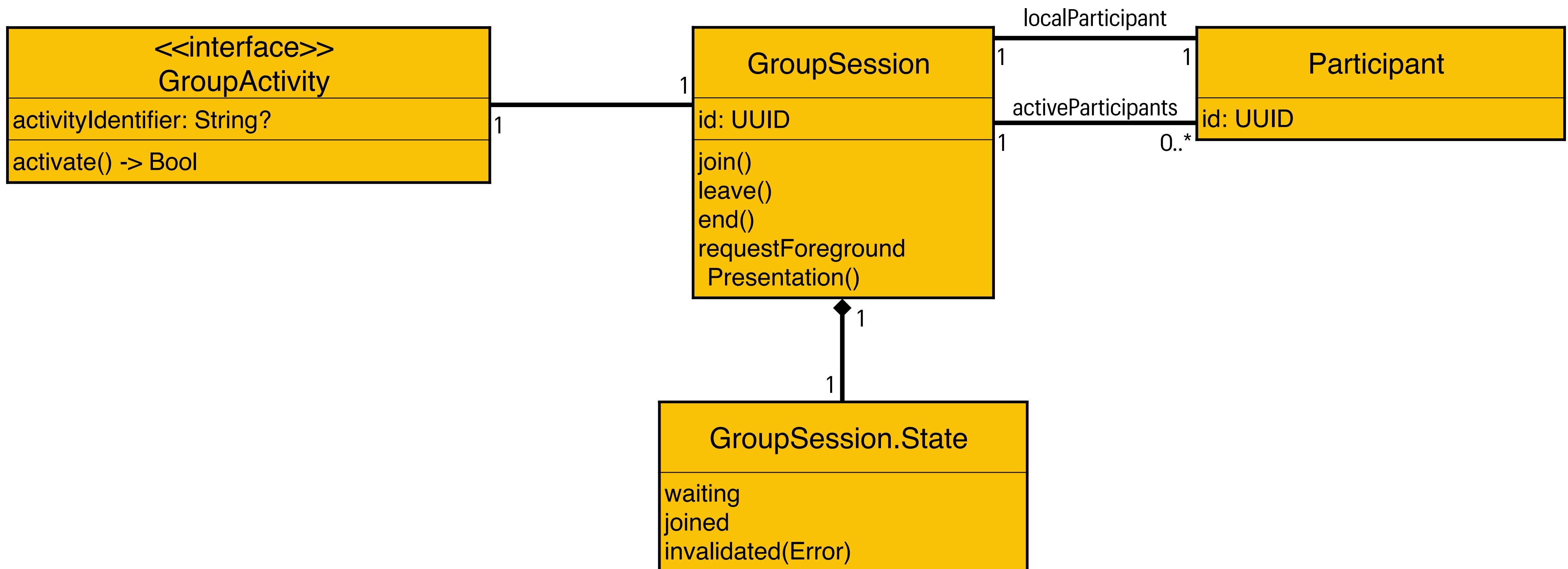
Looking after the session...

SESSION, STATE & OTHER PARTICIPANTS



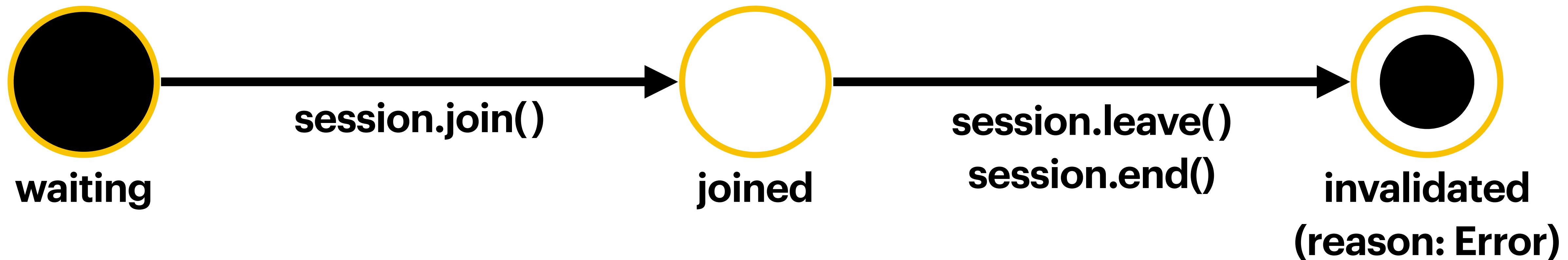
Looking after the session...

OTHER PARTICIPANTS, EVENTS AND STATE



GroupSession.State

HOW DOES STATE SESSION TRANSITION?



GroupSession

WHERE DOES IT COME FROM?

GroupSession is available as an AsyncStream on our GroupActivity type:

```
for await session in PlayTogether.sessions() {  
    await MainActor.run {  
        // a session has opened!  
        // let's store it so we can manage it  
        self.session = session  
  
        // if we're ready, we can join the session  
        session.join()  
    }  
}
```

Participant

HOW DO WE TRACK OUR USERS?



let localParticipant: Participant

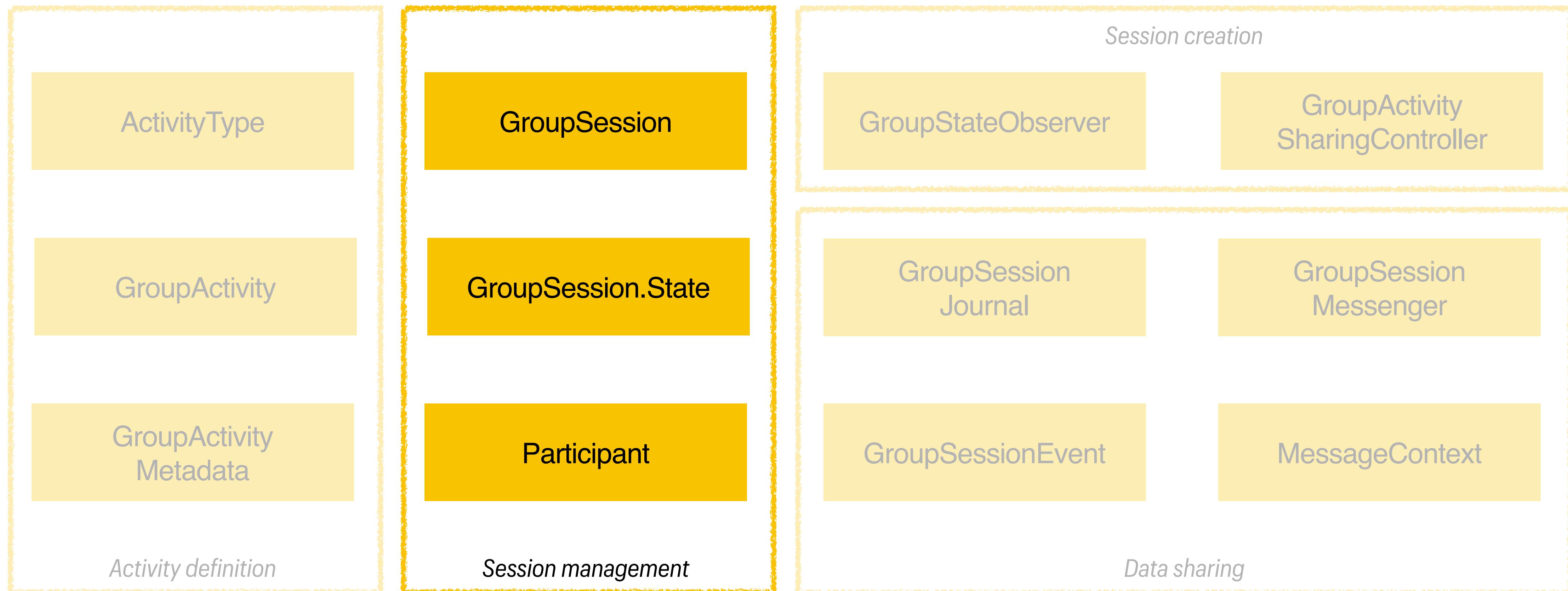


let activeParticipants: Set<Participant>

```
struct Participant: Identifiable, Hashable, Equatable {  
    let id: UUID  
}
```

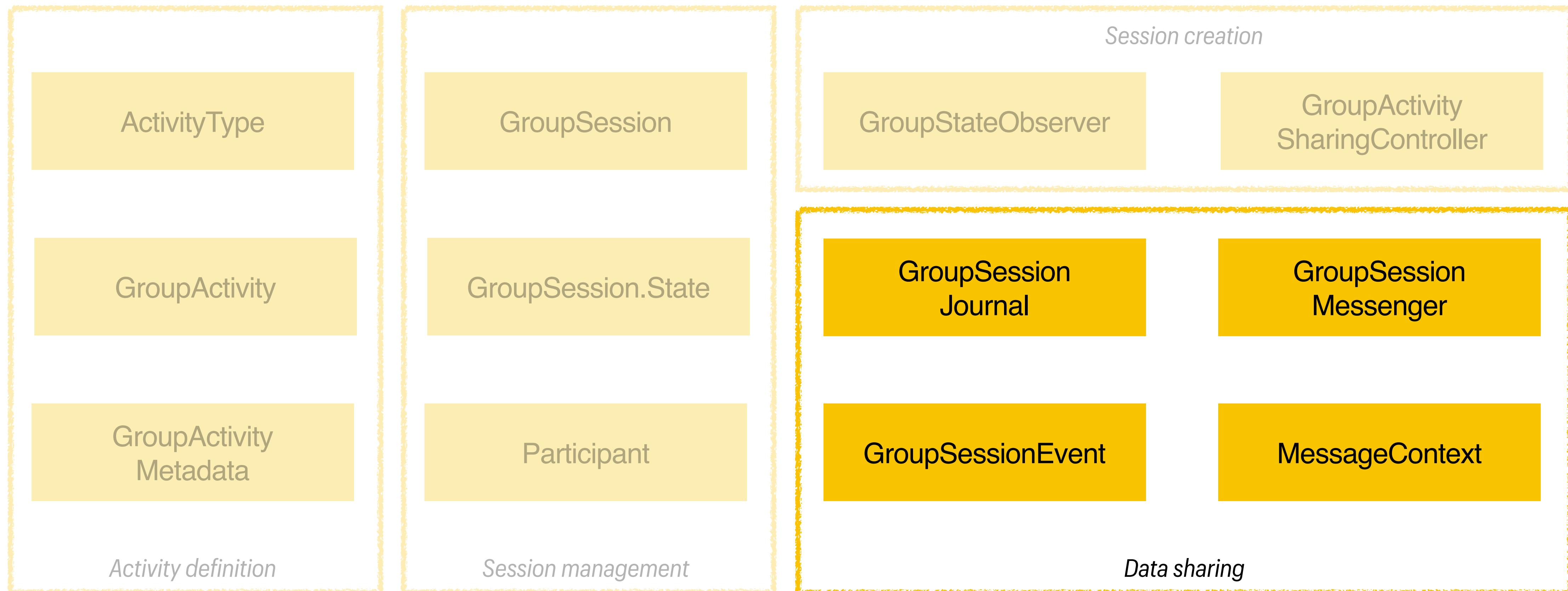
Incoming...!

KEEPING STATE UP-TO-DATE: MESSAGES VS FILES*



Incoming...!

KEEPING STATE UP-TO-DATE: MESSAGES VS FILES*



Sharing state

SHARE INFORMATION AND STATE BETWEEN DEVICES

GroupActivity

GroupSessionMessenger

GroupSessionJournal

NEW in
iOS 17!

?

initial state

e.g.

choice of game ?
drew a line ?

?

**small transient
messages**

e.g.

played a card ?
drew a line ?

?

**files and
larger data**

e.g.

images ?
audio ?

Playing a move

UPDATING STATE ACROSS ALL DEVICES WITH GROUPSESSIONMESSINGER

When a user plays a move, they can update other devices with a GroupMessage:

```
let card = PlayingCard()  
let messenger = GroupSessionMessenger(session: session)  
// anything Codable can be sent!  
try await messenger.send(card)  
// you can also specify which participants to send to:  
try await messenger.send(card, to: .all)
```

Receiving a move

UPDATING STATE ACROSS ALL DEVICES WITH GROUPSESSIONMESSINGER

You can receive a move using an `AsyncStream` on `GroupSessionMessenger`:

```
let messenger = GroupSessionMessenger(session: session)
```

```
// receives all moves in order...
```

```
for await move in messenger.messages(of: card) {  
    self.game.play(card: move)  
}
```

Playing catch-up!

WHAT IF A USER JOINS LATE?



Playing catch-up!

WHAT IF A USER JOINS LATE?

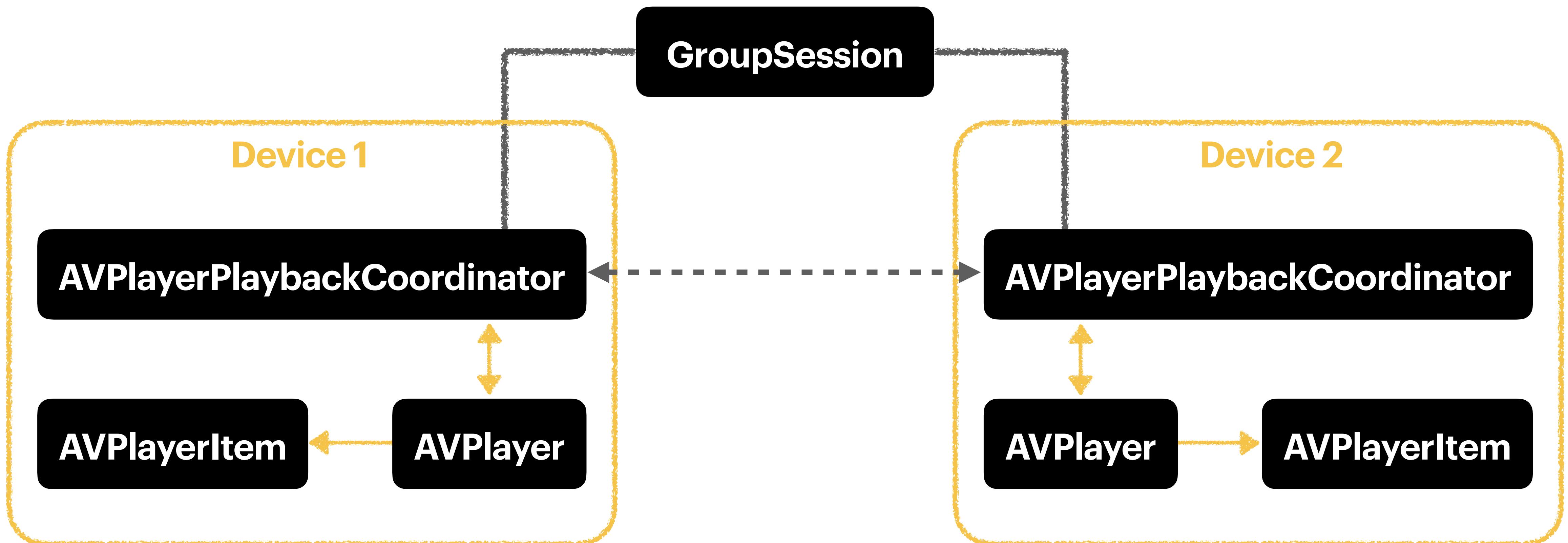
When a new device joins the session, the set of active participants is updated.
Other device can send a **Message** to pass along the current state:

```
groupSession.$activeParticipants
  .sink { activeParticipants in
    let newParticipants = activeParticipants
      .subtracting(groupSession.activeParticipants)
    Task {
      try await messenger.send(currentState, to: .only(newParticipants))
    }
  }
```

Apple Documentation: [Handling Late Joiners](#)

Media syncing

BUILT INTO AVFOUNDATION



AVPlayerPlaybackCoordinator

BUILT INTO AVFOUNDATION

Use the AVPlayerPlaybackCoordinator to sync video playback using the GroupSession:

```
func createPlayer(assetURL: URL, session: GroupSession) -> AVPlayer {
```

```
    let player = AVPlayer(url: assetURL)
```

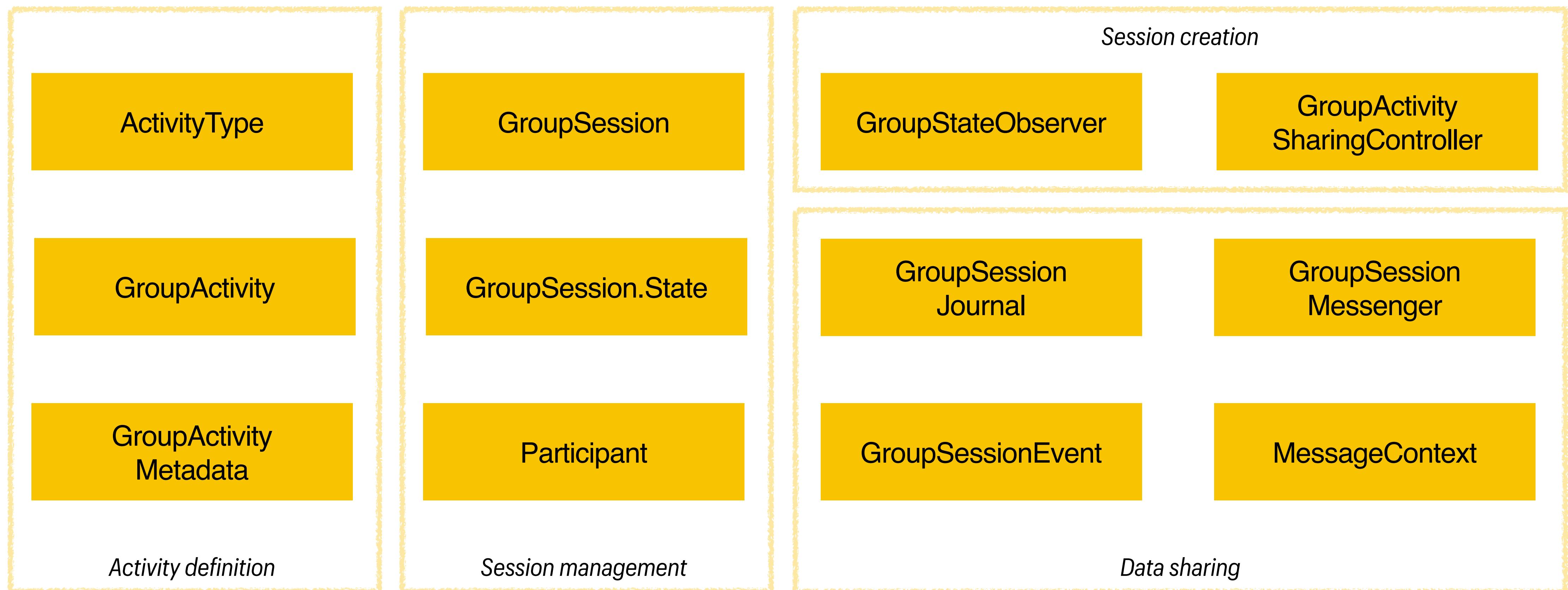
```
    player.playbackCoordinator.coordinateWithSession(session)
```

```
    return player
```

```
}
```

GroupActivities API

HOW IT FITS TOGETHER





“Sharing and collaboration are a central part of visionOS. And with SharePlay, next-generation shared experiences become possible.”

- APPLE







Architecture Design Build >



Shared >

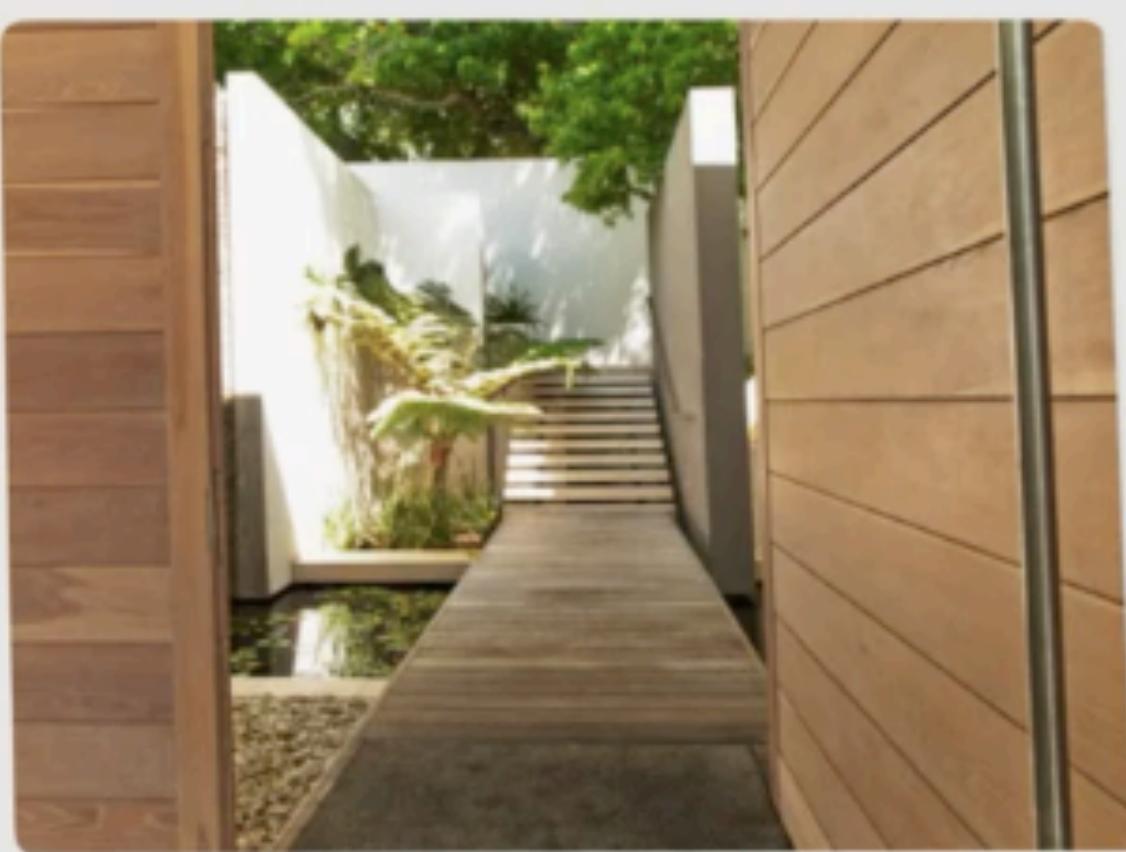


Living Office
Case Study



↑
NEUTRAL
PALETTE

TABER PL
SOUTH PARK ST



Living Office Case Study



Peperomia
Watermelon



Landscape Architects'
Work Samples:
See Keynote deck

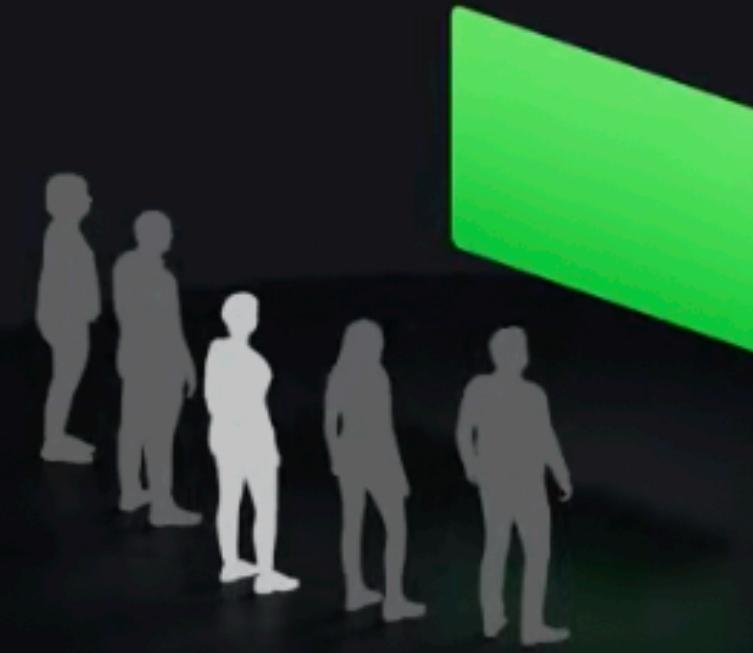
UISceneActivationConditions

SPECIFY WHICH SCENE IS SHARED WITH SHAREPLAY

Prefers / Allows

```
ContentView()  
.handlesExternalEvents {  
    preferring: ["com.example.facecards.play"],  
    allowing: ["com.example.facecards.play"]  
}
```

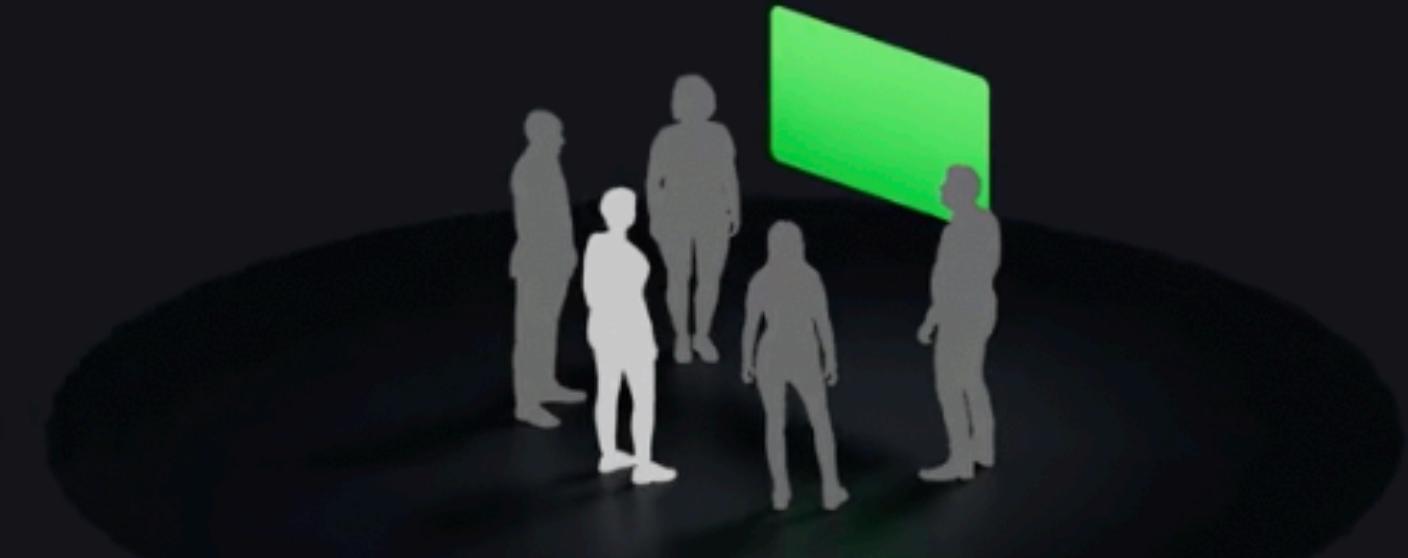
Spatial persona templates



Side-by-Side



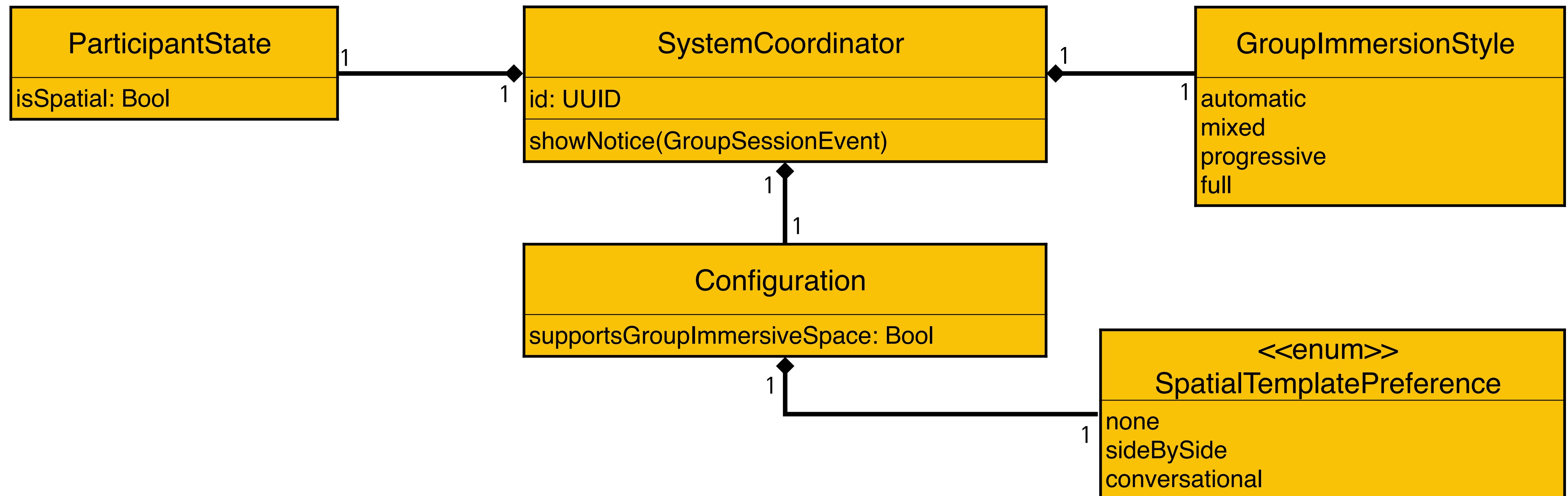
Surround



Conversational

SystemCoordinator

COORDINATE INTERFACE'S BEHAVIOUR WITH SPACIAL PLACEMENT OF CONTENT







... to wrap it up

What have we learnt?

... TO WRAP IT UP

- SharePlay enables shared experiences across devices and platforms
- It is powered by the [GroupActivities API](#)

What have we learnt?

... TO WRAP IT UP



What have we learnt?

... TO WRAP IT UP



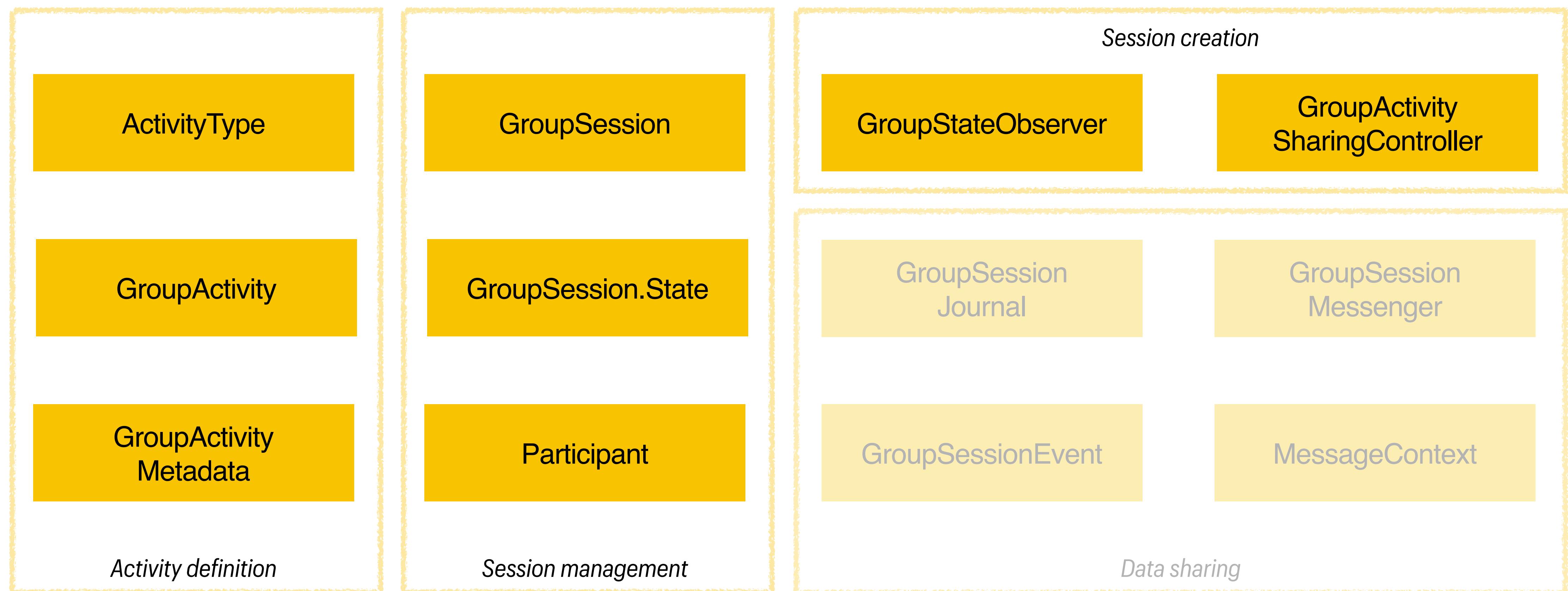
What have we learnt?

... TO WRAP IT UP



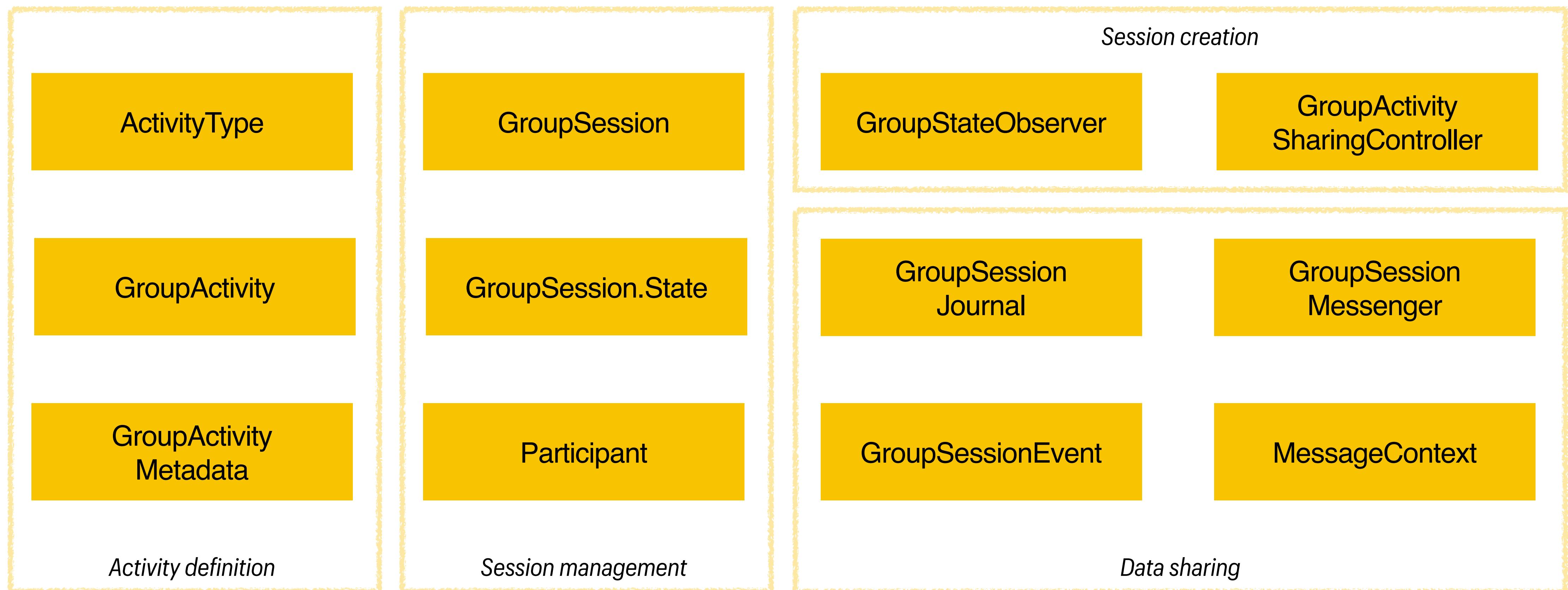
What have we learnt?

... TO WRAP IT UP



What have we learnt?

... TO WRAP IT UP



What have we learnt?

... TO WRAP IT UP

- SharePlay enables shared experiences across devices and platforms
- It is powered by the [GroupActivities API](#)
- Sharing and collaboration are a central part of visionOS
- Spatial shared experiences, including persona templates

Over to you...



Over to you...

WHAT ARE YOU GOING TO BUILD?

<https://developer.apple.com/shareplay/>

WWDC23:

- [Add SharePlay to your app](#)
- [Build spatial SharePlay experiences](#)

All my code is [open-source!](#)

The app is available on the [App Store](#).

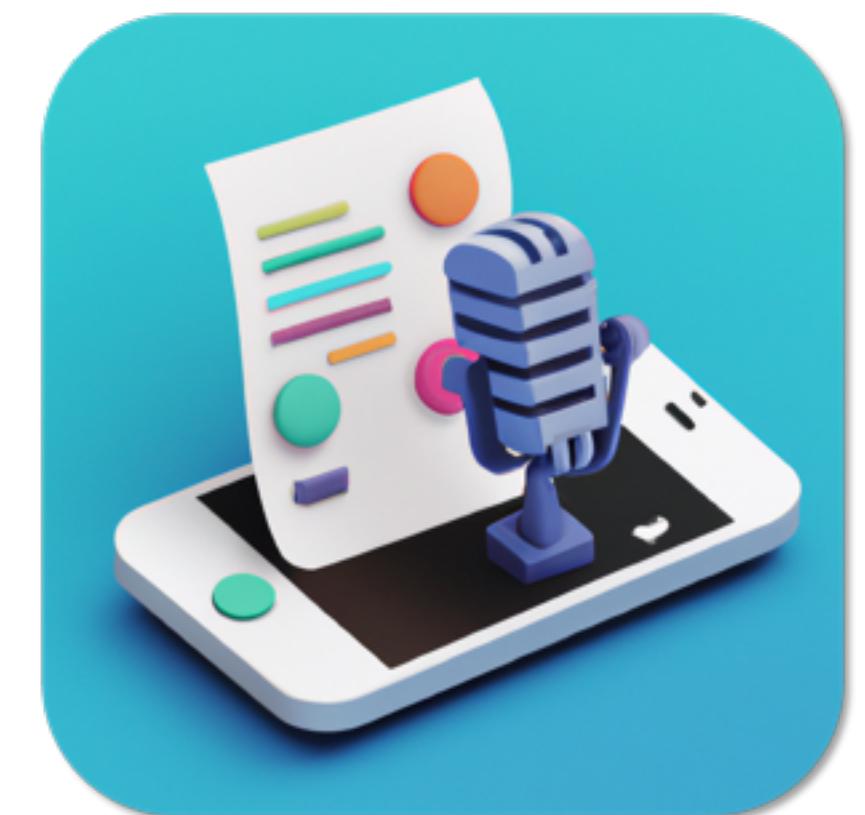
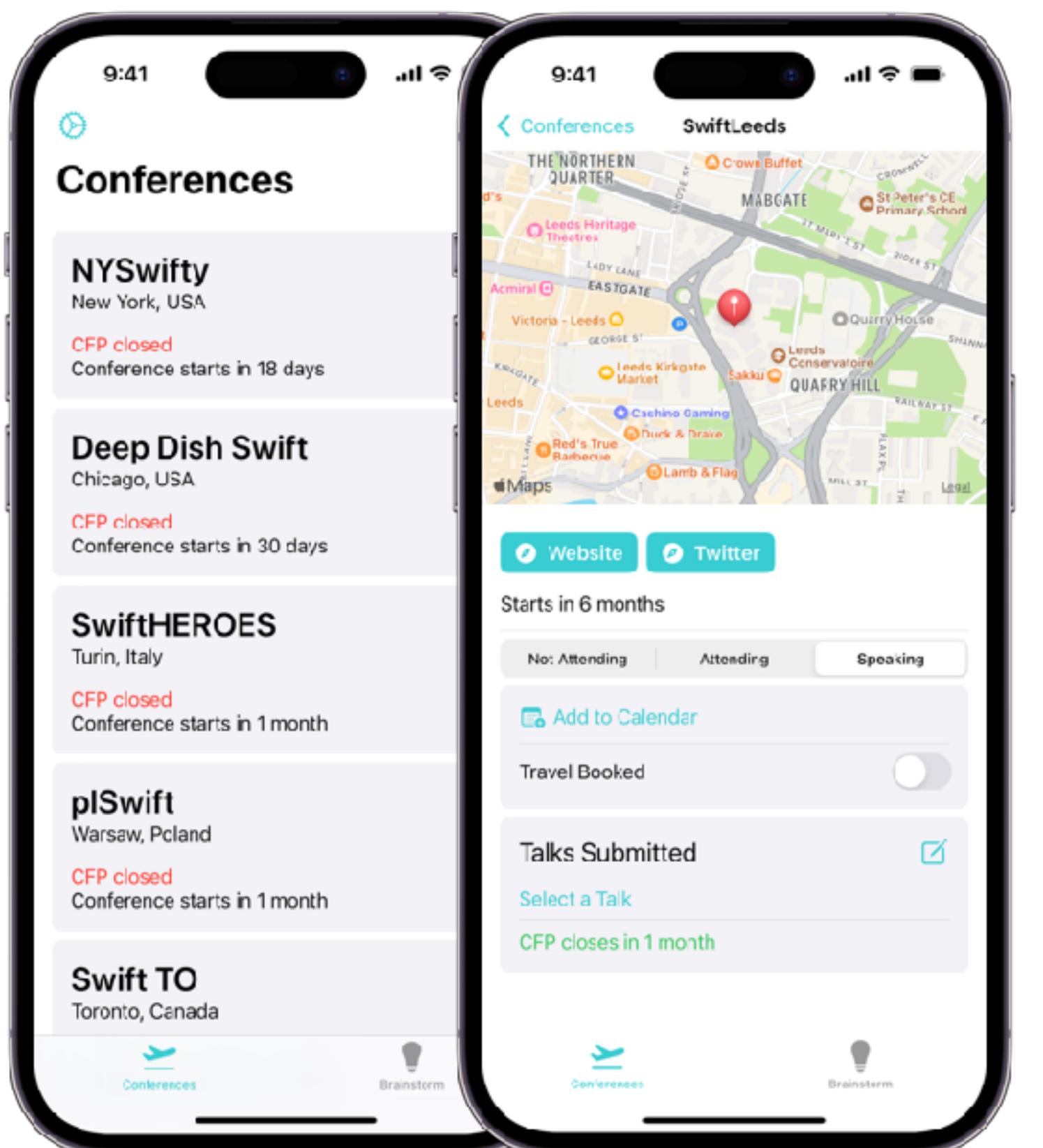


TRY IT OUT!

Shameless plug



CONFERENCES APP



THANK YOU

OLIVERBINNS.CO.UK



Oliver Binns