Peer-to-peer communication

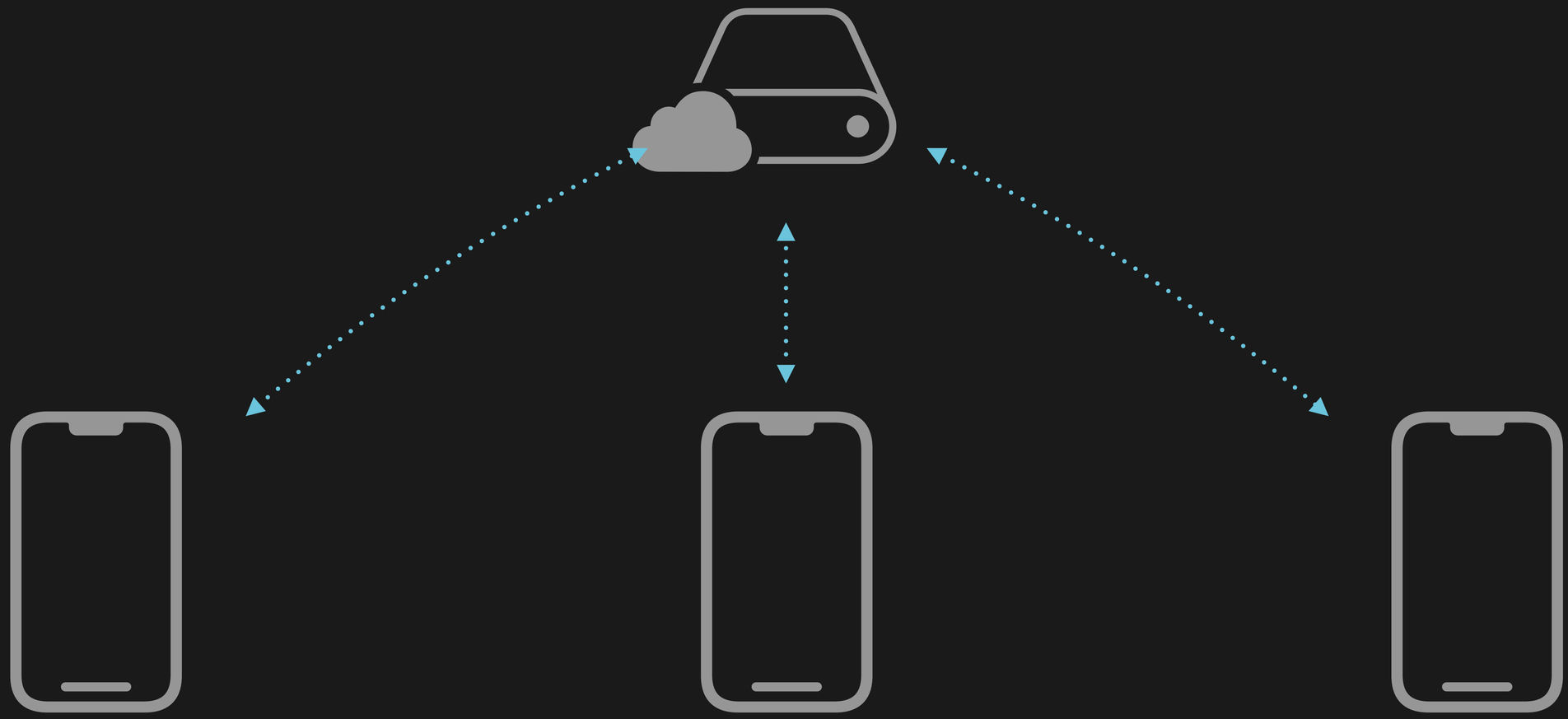
on iPhone

Ilian Konchev

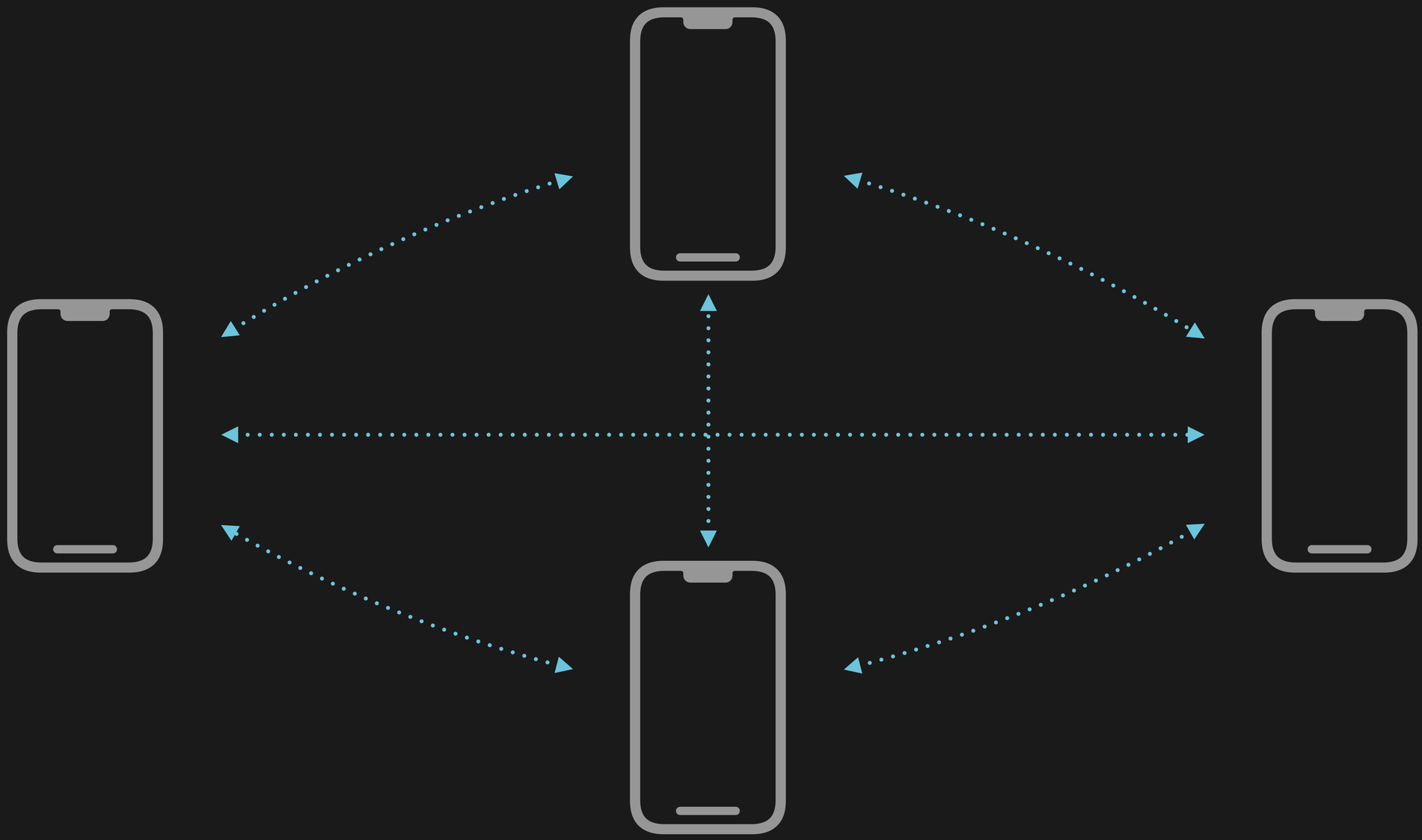
Snapp Mobile

About me

- Software engineer, father and husband
- 25+ years of experience in software development
- 10+ years of experience writing applications / SDKs for the ecosystem
- Associated partner & iOS team lead at SnappMobile Germany GmbH
- Swift language enthusiast
- Interested in security, performance, Formula 1, Arsenal FC and guitars



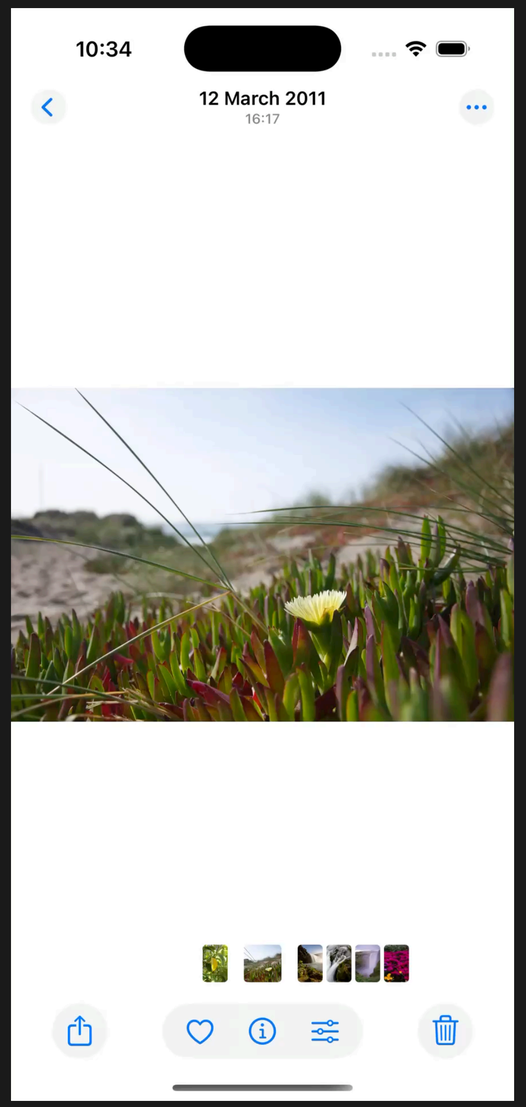
Client-Server communication model



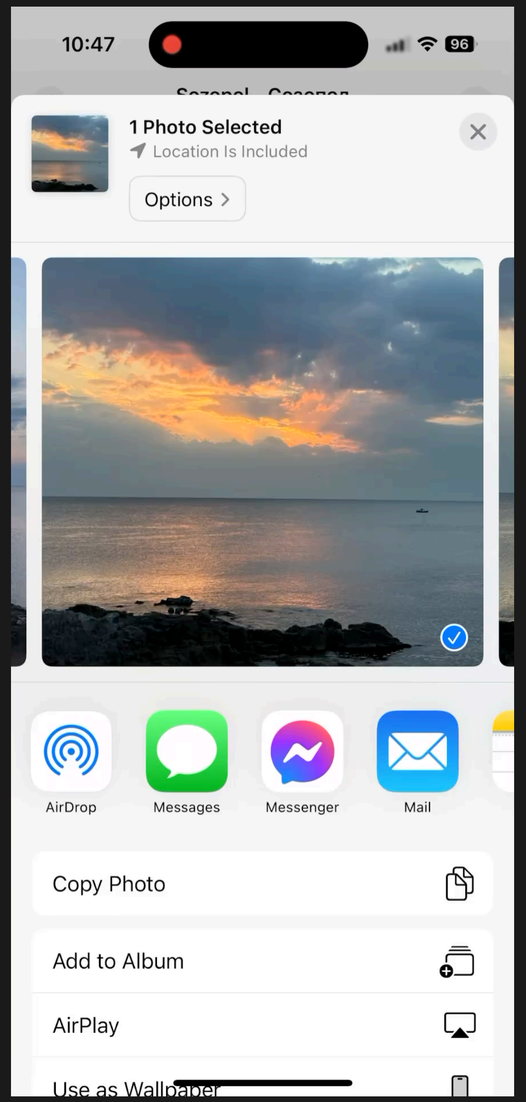
Peer-to-peer communication model

Peer-to-peer in action

iOS share sheet



AirDrop


10:47
96
1 Photo Selected
Location Is Included
Options
AirDrop
Messages
Messenger
Mail
Copy Photo
Add to Album
AirPlay
Use as Wallpaper

What's behind the AirDrop interaction?

...or how curiosity leads to learning

- AirDrop option only available when nearby devices are turned on
- Triggering the option does device discovery the first time, starts with a cached list on subsequent interactions, but still does the discovery when list gets loaded
- New devices pop-up and disappear as they are turned on and off
- Can send multiple file types, based on what the other side is capable of accepting

Multipeer Connectivity

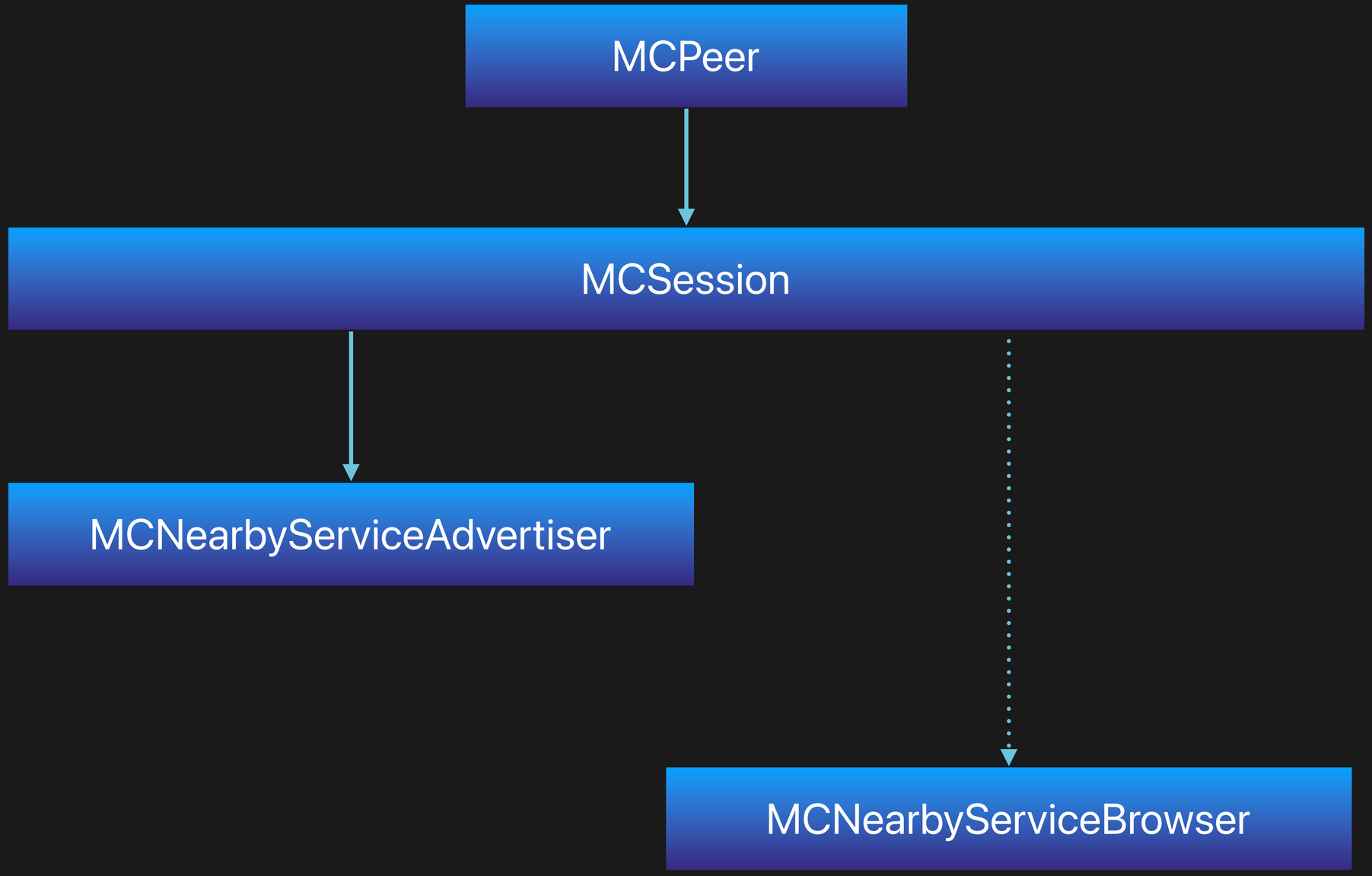
iOS / iPadOS 7.0+, macOS 10.10+, tvOS 10.0+, visionOS 1.0+

- Support peer-to-peer connectivity and the discovery of nearby devices.
- Allows the discovery of services provided by nearby devices and communicating with those services through message-based data, streaming data, and resources (such as files)
- In iOS, the framework uses infrastructure Wi-Fi networks, peer-to-peer Wi-Fi, and Bluetooth personal area networks for the underlying transport.
- In macOS and tvOS, it uses infrastructure Wi-Fi, peer-to-peer Wi-Fi, and Ethernet

Architecture

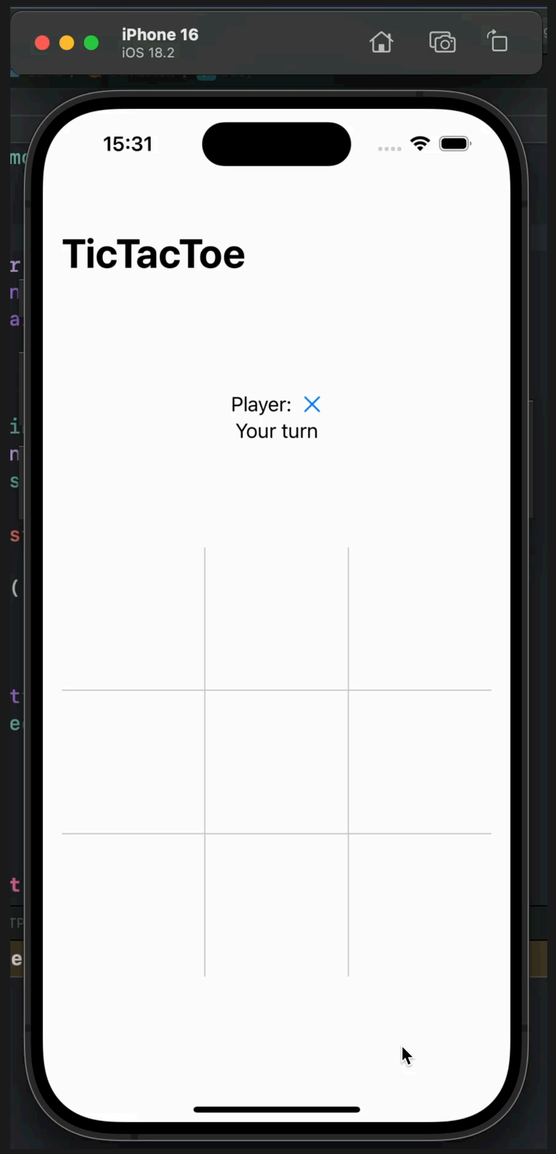
The building blocks

- Peer IDs (`MCPeerID`) uniquely identify an app running on a device to nearby peers
- Session objects (`MCSession`) support communication between connected peer devices.
- Advertiser objects (`MCNearbyServiceAdvertiser`) tell nearby peers that your app is willing to join sessions of a specified type
- Browser objects (`MCNearbyServiceBrowser`) let your app search programmatically for nearby devices with apps that support sessions of a particular type.


MCPeer
MCSession
MCNearbyServiceAdvertiser
MCNearbyServiceBrowser

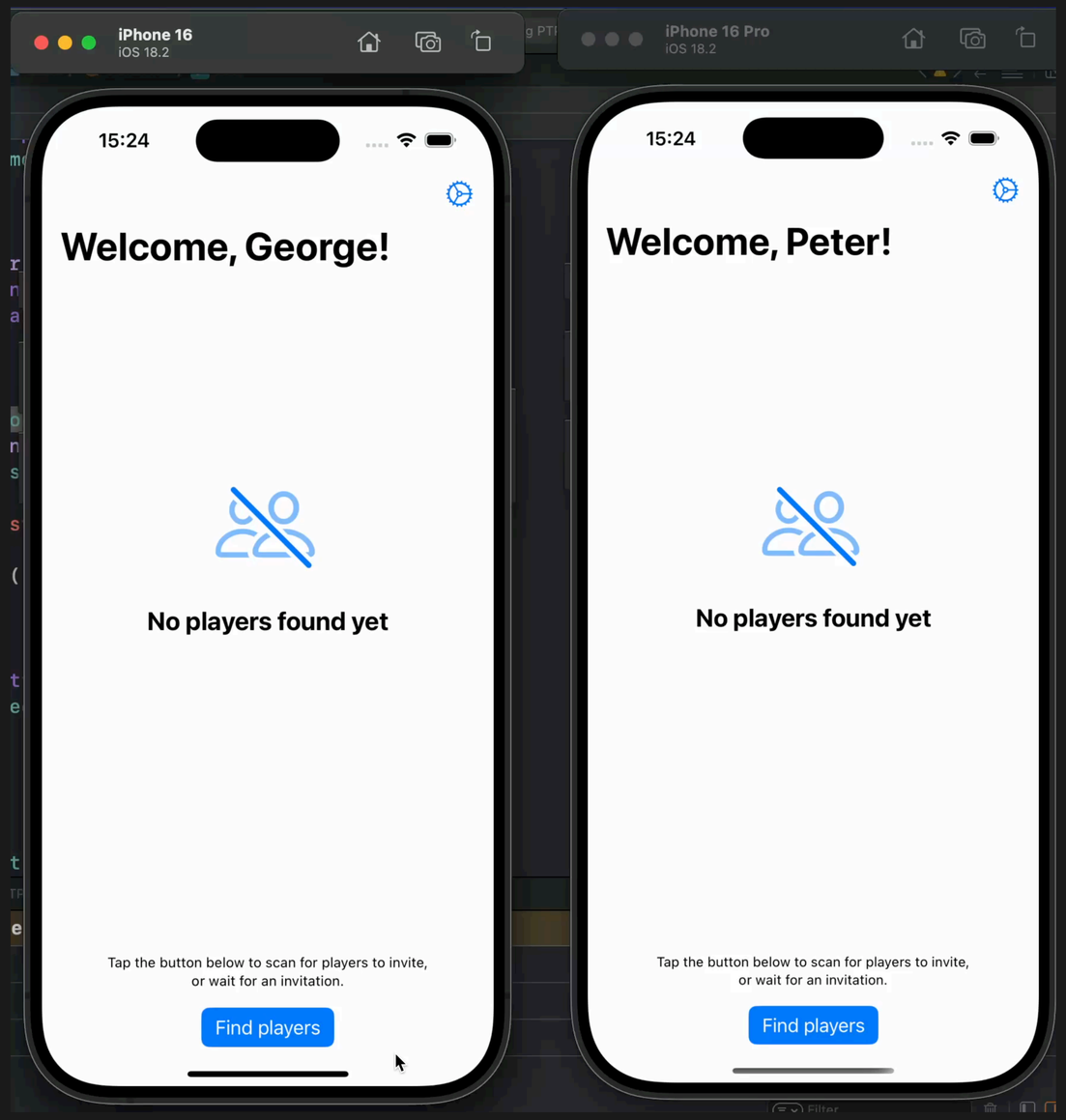
Today’s task

from single player...



Today's task

... to multiplayer



Let’s code!

A note on the repository

https://github.com/stinger/PTPDemo