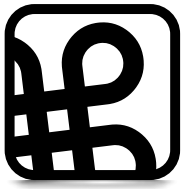
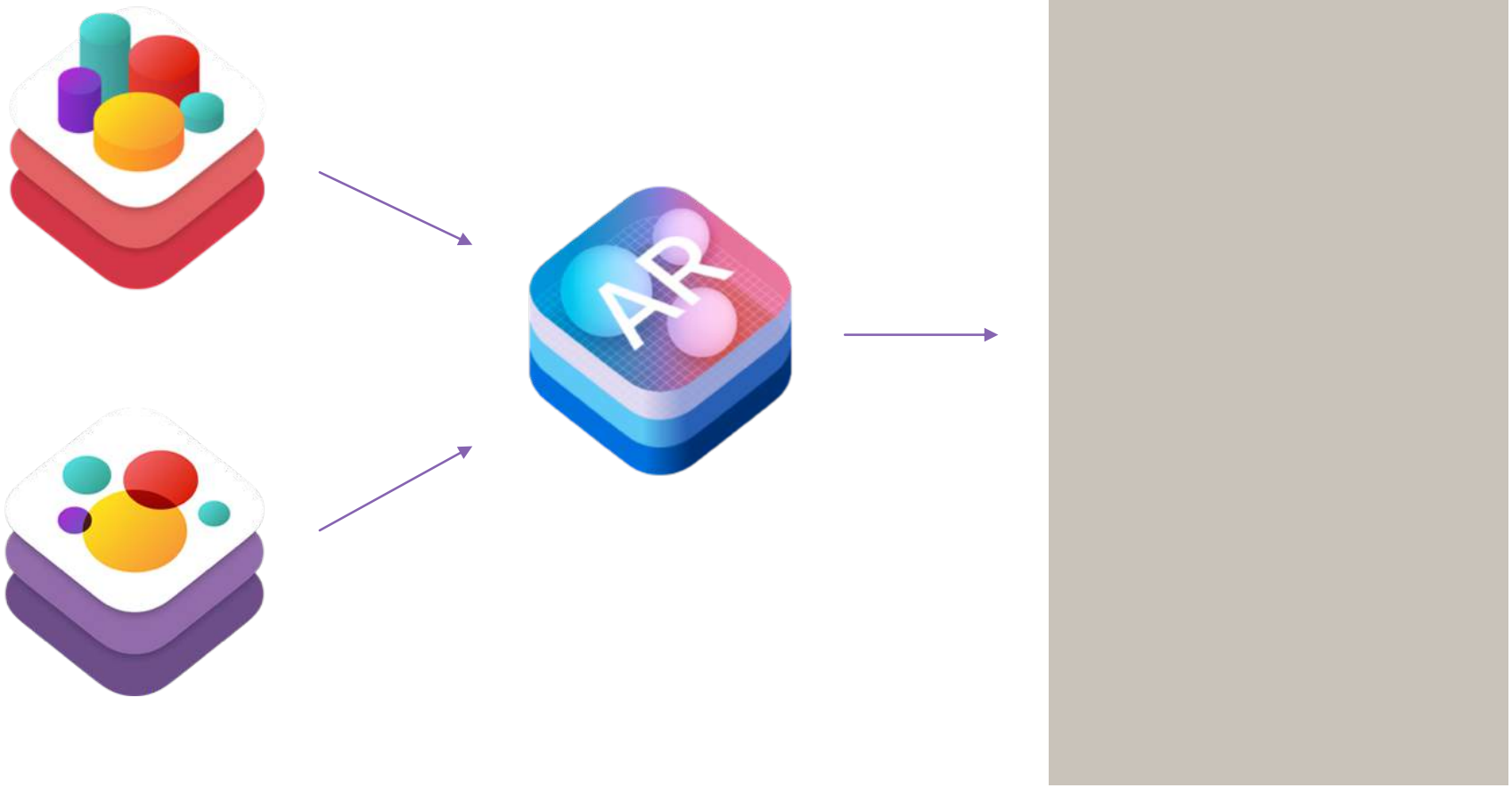


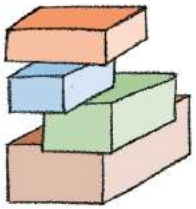
Macoun

Neu und heiß!

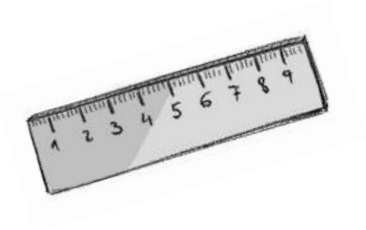
ARKit is heiß - ARKit2 ist heißer

von Le Ortwin und La Manu


AR



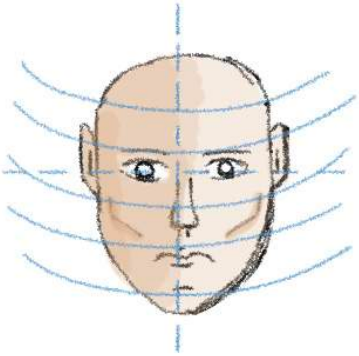
Blöcke stapeln



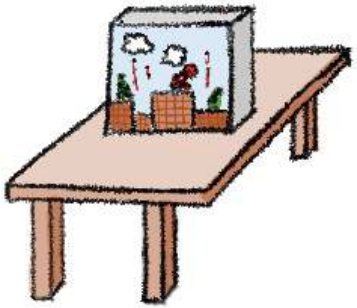
Messen

Ikea Places

draußen $Irgendwo
etwas hinmalen



Face Tracking



2D Games $irgendwo
im Raum spielen

AR als nächste Innovation?

Was geht denn… technisch?

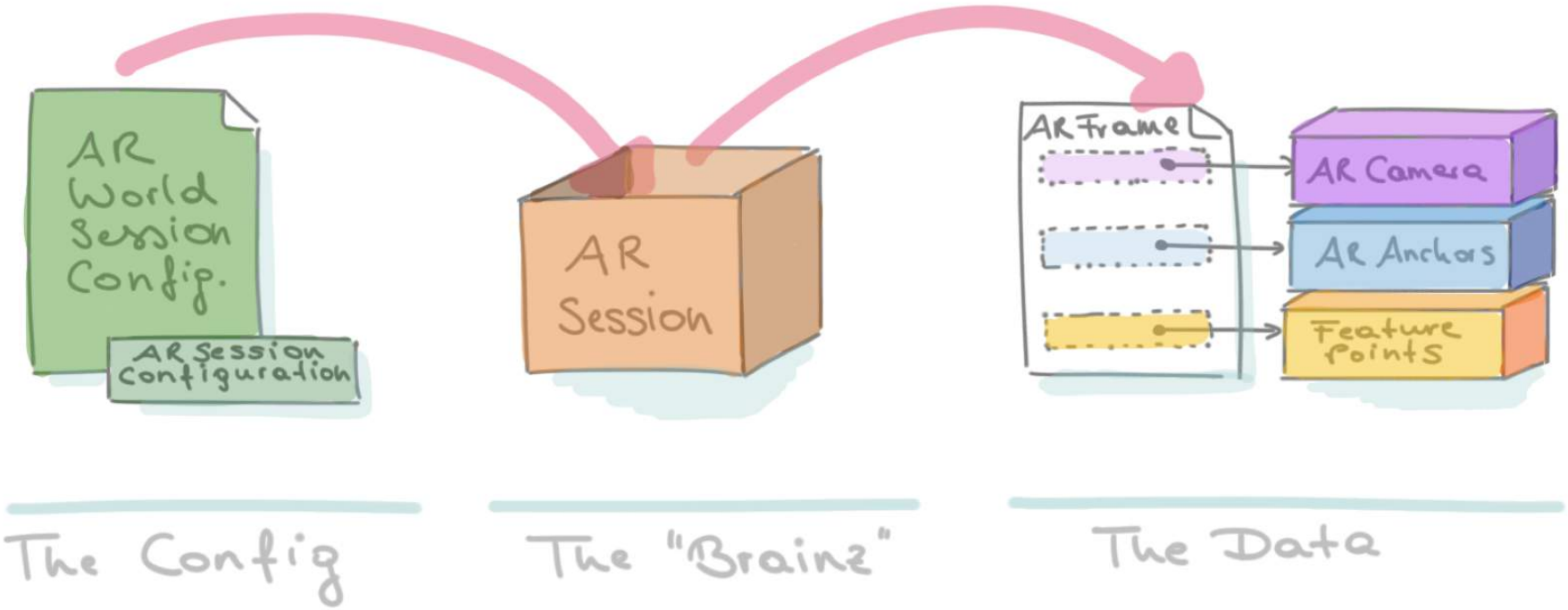
Echt viel!

ARKit

- Detect planes
- Get light estimates
- Place virtual content on planes
- Detect 2D images

ARKit 2

- Track 2D images
- Detect 3D objects
- Classify plane types (wall, table, seat, …)
- Save & Load world maps
- Enjoy multi user experiences

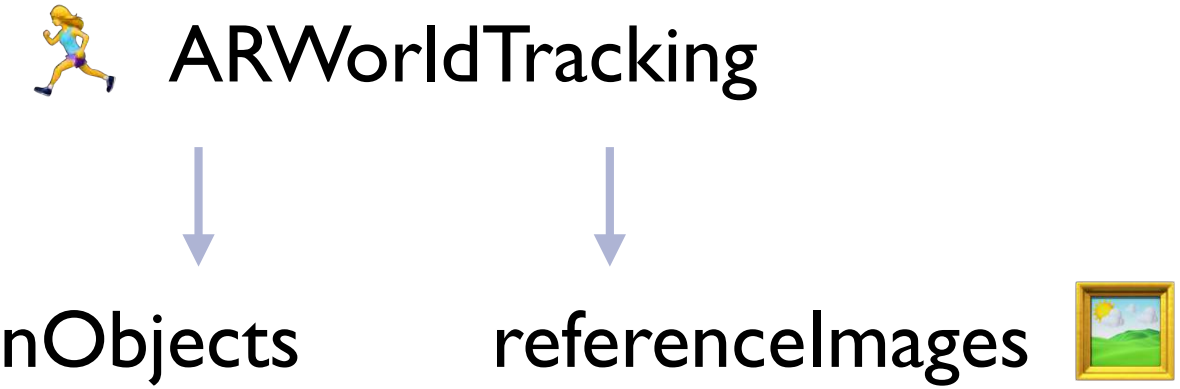

AR World Session Config.
AR Session Configuration
AR Session
ARFrame
AR Camera
AR Anchors
Feature Points
The Config
The "Brainz"
The Data

ARWorldTracking

ARImageTracking

ARObjectScanning

ARFaceTracking


ARWorldTracking
detectionObjects
referenceImages

Plane Detection

ARWorldTrackingConfiguration

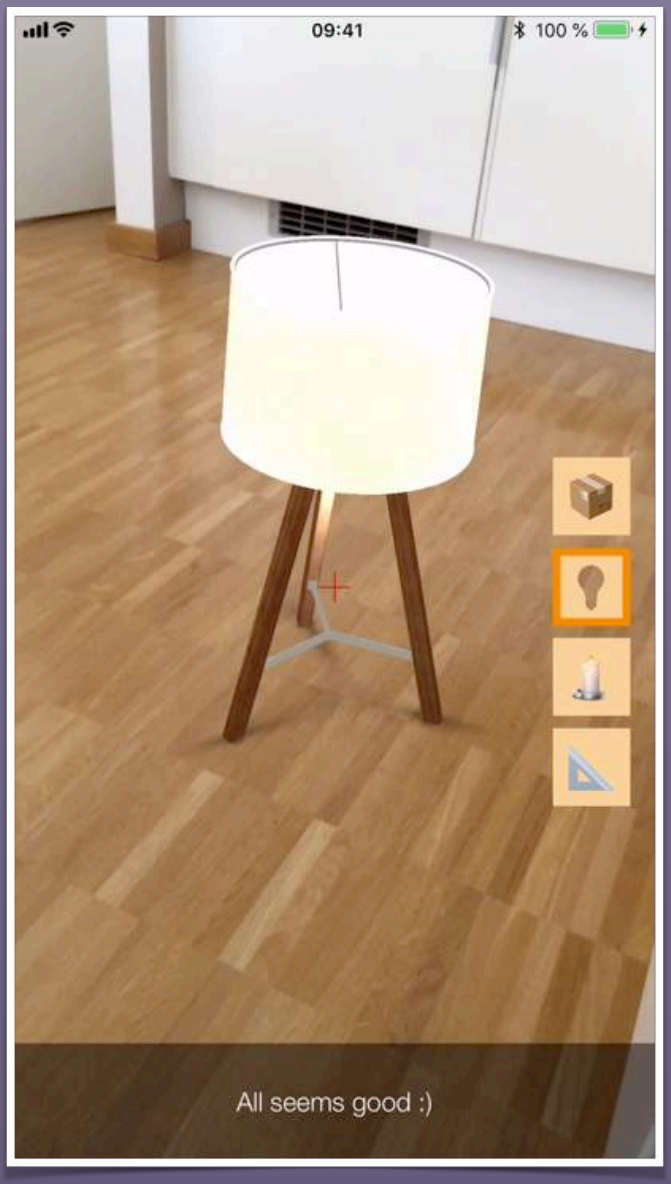
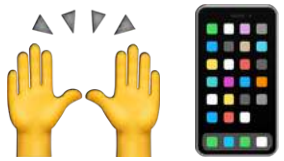


Image Tracking

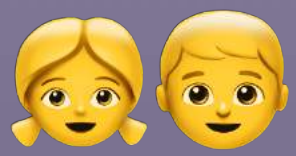
ARImageTrackingConfiguration



Endlich ‘ne Live-Demo

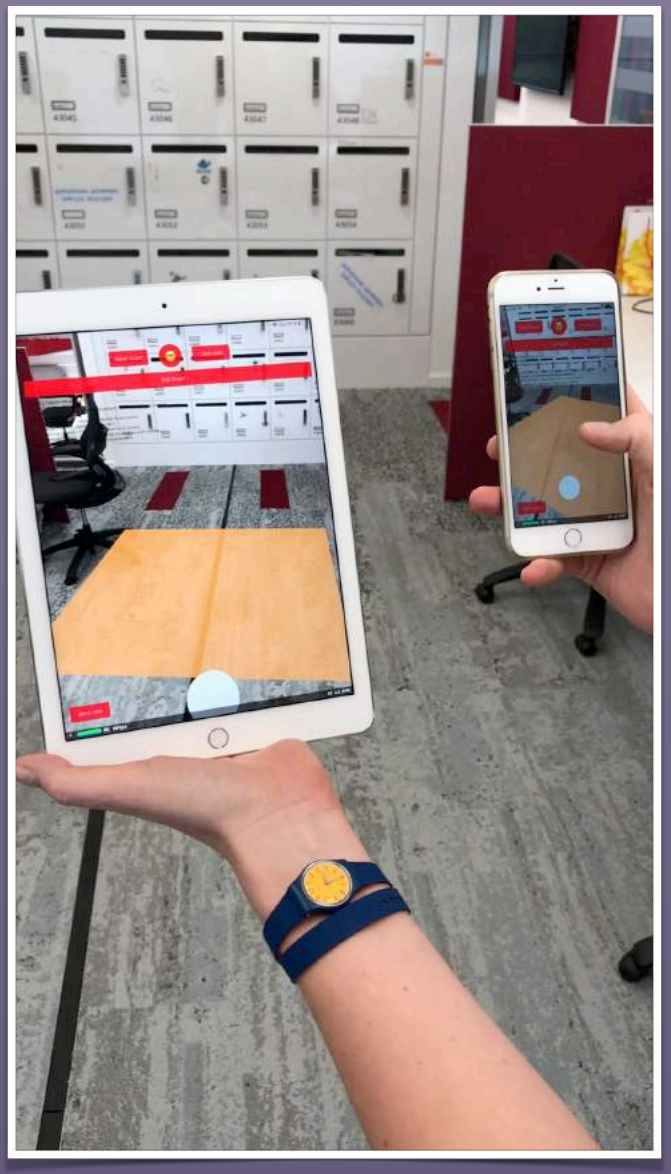
Save & Load World Maps

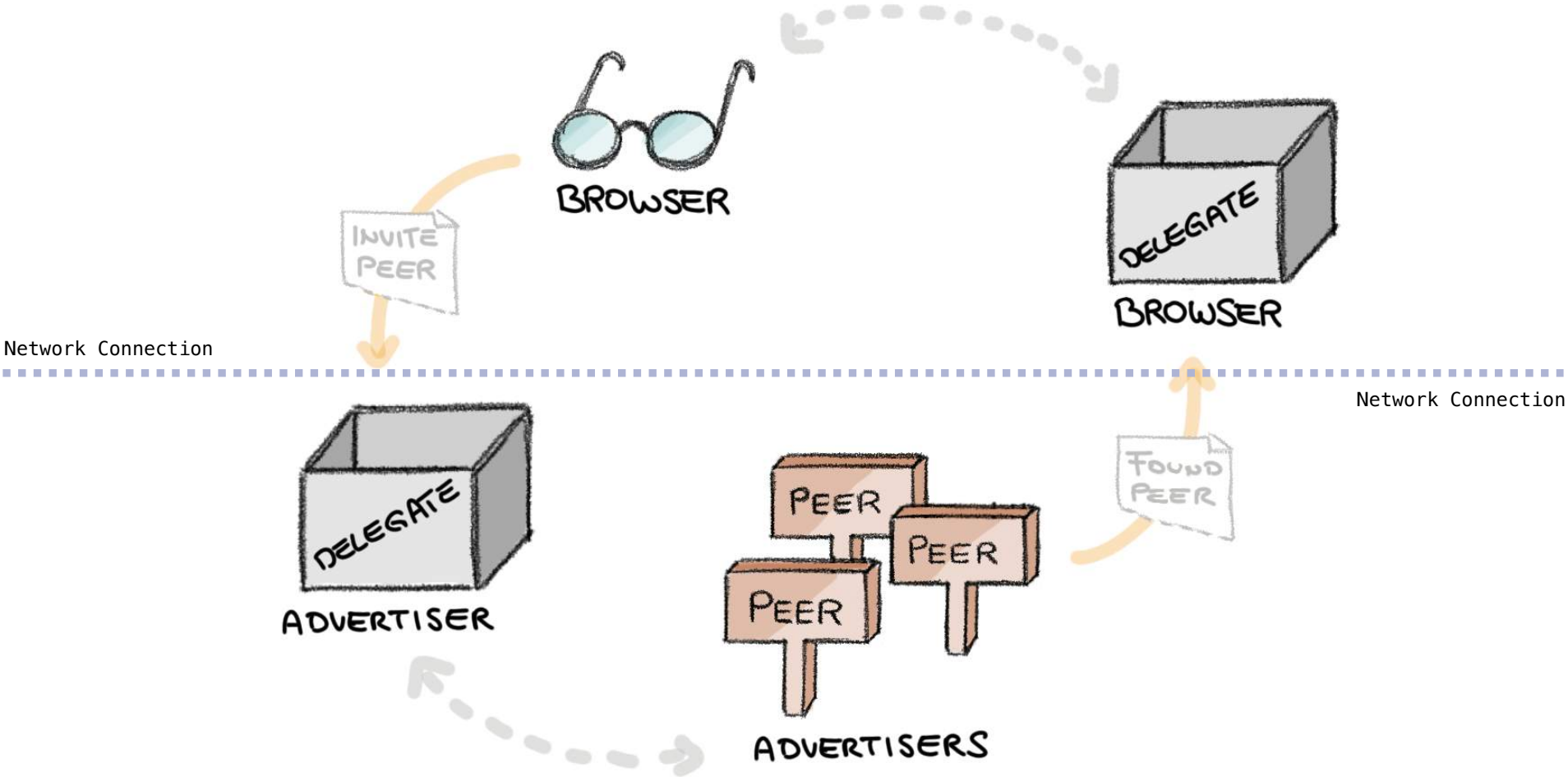
ARWorldTrackingConfiguration

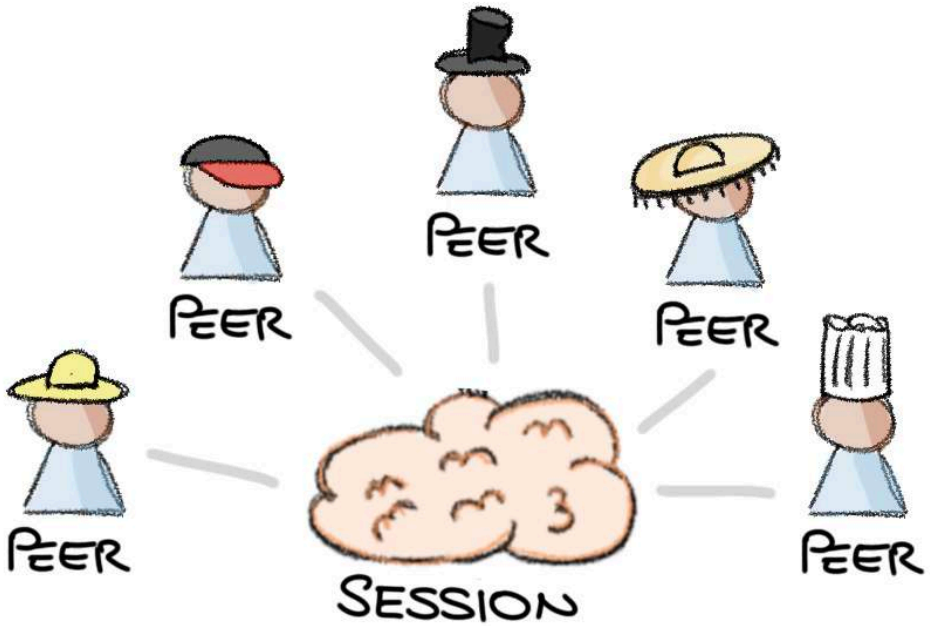


Multiuser Experience

Multipeer Connectivity Framework




BROWSER
INVITE PEER
DELEGATE
BROWSER
Network Connection
Network Connection
DELEGATE
ADVERTISER
PEER
PEER
PEER
ADVERTISERS
FOUND PEER


PEER
PEER
PEER
PEER
PEER
SESSION

```
func createSession () {
    session = MCSession(peer: myPeerID, securityIdentity: nil, encryptionPreference: .required)
}

func startBrowsing () {
    serviceBrowser = MCNearbyServiceBrowser(peer: myPeerID, serviceType: "beerpong")
    serviceBrowser.delegate = self
    serviceBrowser.startBrowsingForPeers()
}

extension MultipeerSession: MCNearbyServiceBrowserDelegate {
    func browser(_ browser: MCNearbyServiceBrowser,
                 foundPeer peerID: MCPeerID,
                 withDiscoveryInfo info: [String: String]?)
    {
        browser.invitePeer(peerID, to: session, withContext: nil, timeout: 10)
    }
}
```

```
func startAdvertising() {
    advertiser = MCNearbyServiceAdvertiser(peer: myPeerID,
                                           discoveryInfo: nil,
                                           serviceType: "beerpong")
    advertiser.delegate = self
    advertiser.startAdvertisingPeer()
}

extension MultipeerSession: MCNearbyServiceAdvertiserDelegate {
    func advertiser(_ advertiser: MCNearbyServiceAdvertiser,
                    didReceiveInvitationFromPeer peerID: MCPeerID,
                    withContext context: Data?,
                    invitationHandler: @escaping (Bool, MCSession?) -> Void)
    {
        invitationHandler(true, session)
     }
}
```

```swift
class Transmission: NSObject, NSCoding {

    var userName: String? = nil
    var worldMap: ARWorldMap? = nil
    var tableNode: SCNNode? = nil

    init(userName: String?, worldMap: ARWorldMap?, tableNode: SCNNode?) {
        self.userName = userName
        self.worldMap = worldMap
        self.tableNode = tableNode
    }

    required init?(coder aDecoder: NSCoder) {
        self.userName = aDecoder.decodeObject(forKey: "userName") as? String
        self.worldMap = aDecoder.decodeObject(forKey: "worldMap") as? ARWorldMap
        self.tableNode = aDecoder.decodeObject(forKey: "tableNode") as? SCNNode
    }

    func encode(with aCoder: NSCoder) {
        aCoder.encode(userName, forKey: "userName")
        aCoder.encode(worldMap, forKey: "worldMap")
        aCoder.encode(tableNode, forKey: "tableNode")
    }
}
```

NSCodable ☝️

✅ SCNNode
🚫 SCNMatrix4

Summary

The transform applied to the node relative to its parent. Animatable.

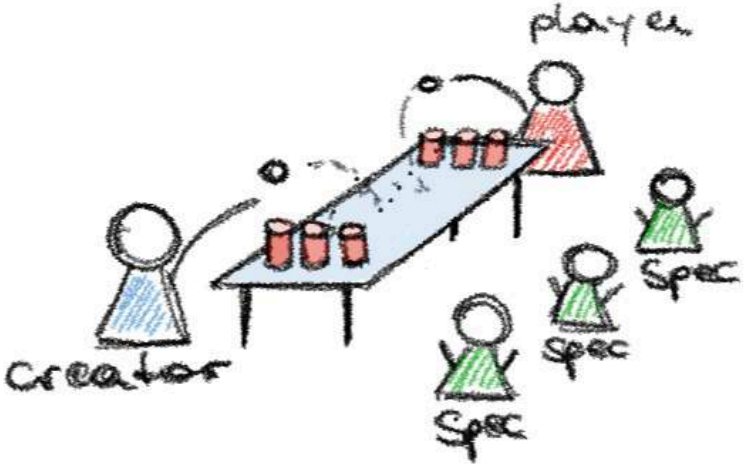
Declaration

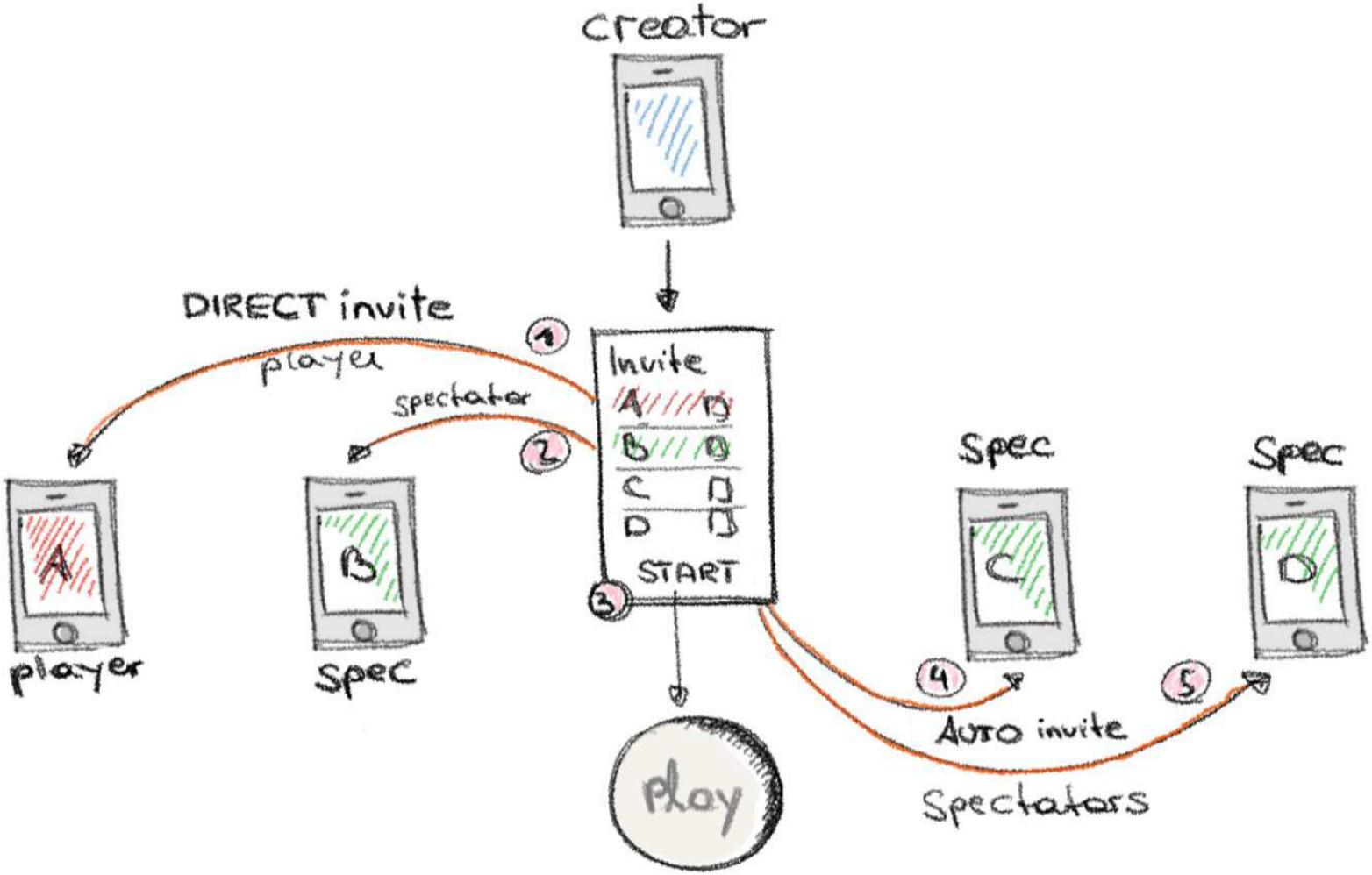
```
var transform: SCNMatrix4 { get set }
```

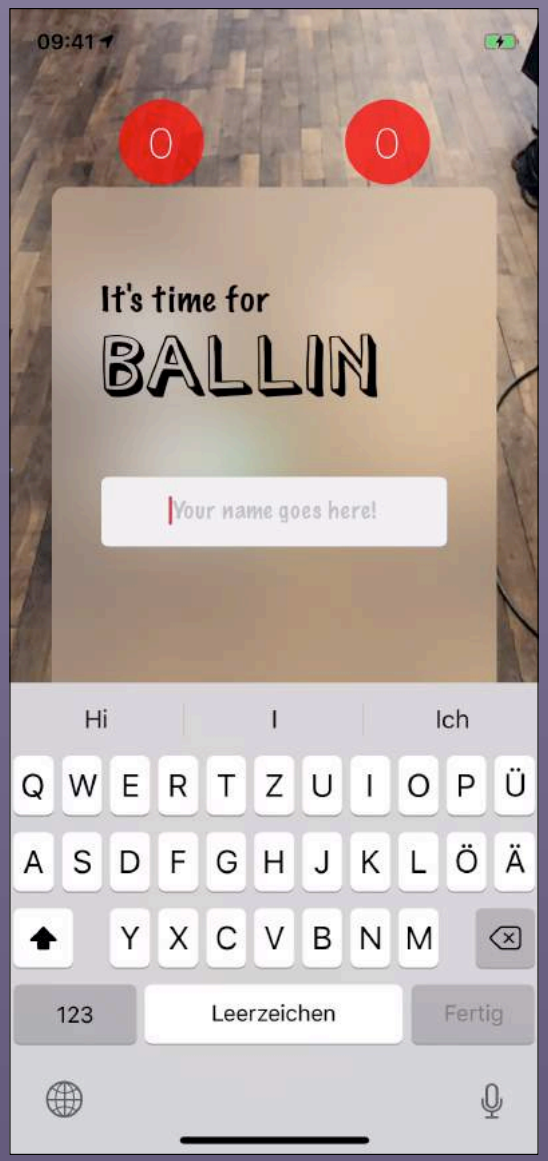
```
func shareWorldMapWithPeers() {
    sceneView.session.getCurrentWorldMap { worldMap, error in

        let transmission = Transmission(userName:   myPlayerName,
                                        worldMap:   worldMap,
                                        tableNode:  tableNode)

        guard let data = try? NSKeyedArchiver.archivedData(
                                    withRootObject: transmission,
                                    requiringSecureCoding: false) else
        {
            fatalError("can't encode table transmission")
        }

        do {
            try session.send(data, toPeers: session.connectedPeers, with: .reliable)
        } catch {
            print("error sending data to peers: \(error.localizedDescription)")
        }
    }
}
```

```
func receivedData(_ data: Data, from peer: MCPeerID) {
    DispatchQueue.main.async {
        if let unarchived = try? NSKeyedUnarchiver.unarchiveTopLevelObjectWithData(data),
            let transmission = unarchived as? Transmission,
            let worldMap = transmission.worldMap
        {
            let configuration = ARWorldTrackingConfiguration()
            configuration.initialWorldMap = worldMap
            sceneView.session.run(configuration, options: [.resetTracking, .removeExistingAnchors])
        }
    }
}
```

How-to Multiplayer




creator
DIRECT invite
player
spectator
1
2
Invite
A
B
C
D
START
3
player
Spec
Spec
Spec
4
5
AUTO invite
Spectators
play


09:41
0
0
It's time for
BALLIN
Your name goes here!
Hi
I
Ich
Q W E R T Z U I O P Ü
A S D F G H J K L Ö Ä
Y X C V B N M
123
Leerzeichen
Fertig

Happy AR Coding ❤️

Manu Rink
Software Engineer - Microsoft
@codeprincess

Ortwin Gentz
CEO & Engineer - FutureTap
@ortwingentz