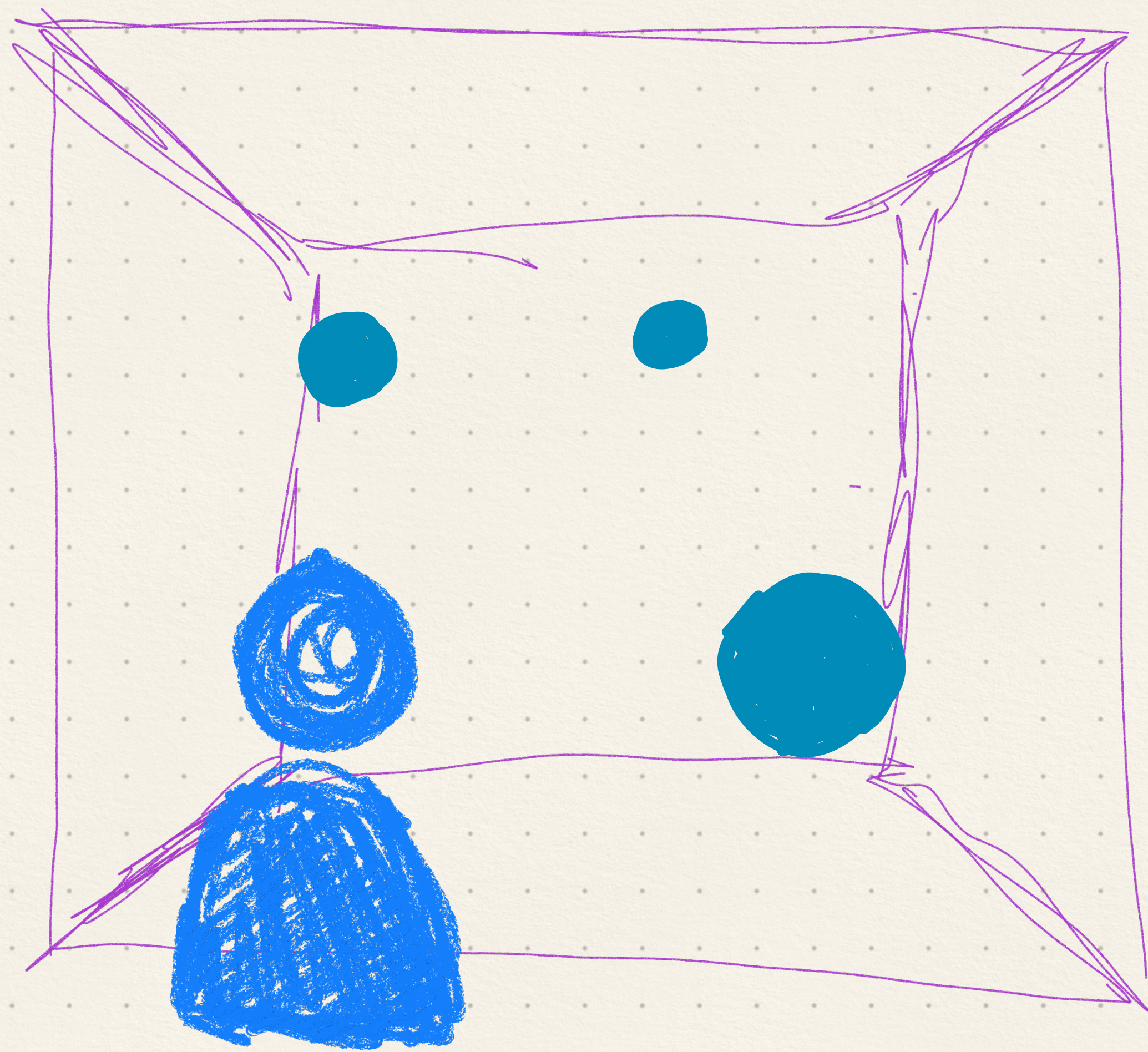
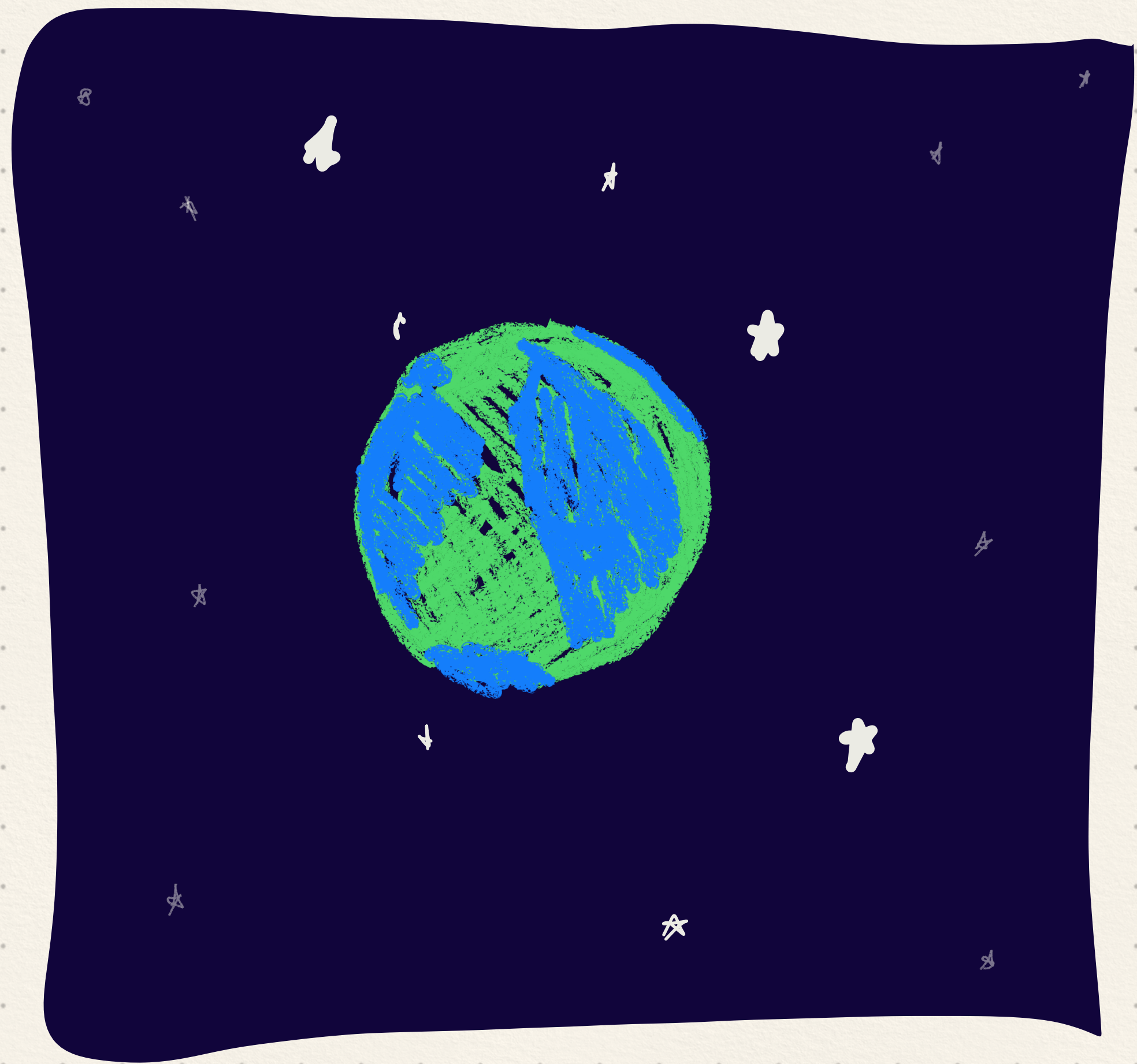




Everything's Cooler in Space!

Lena Mattea Stöxen



Agenda

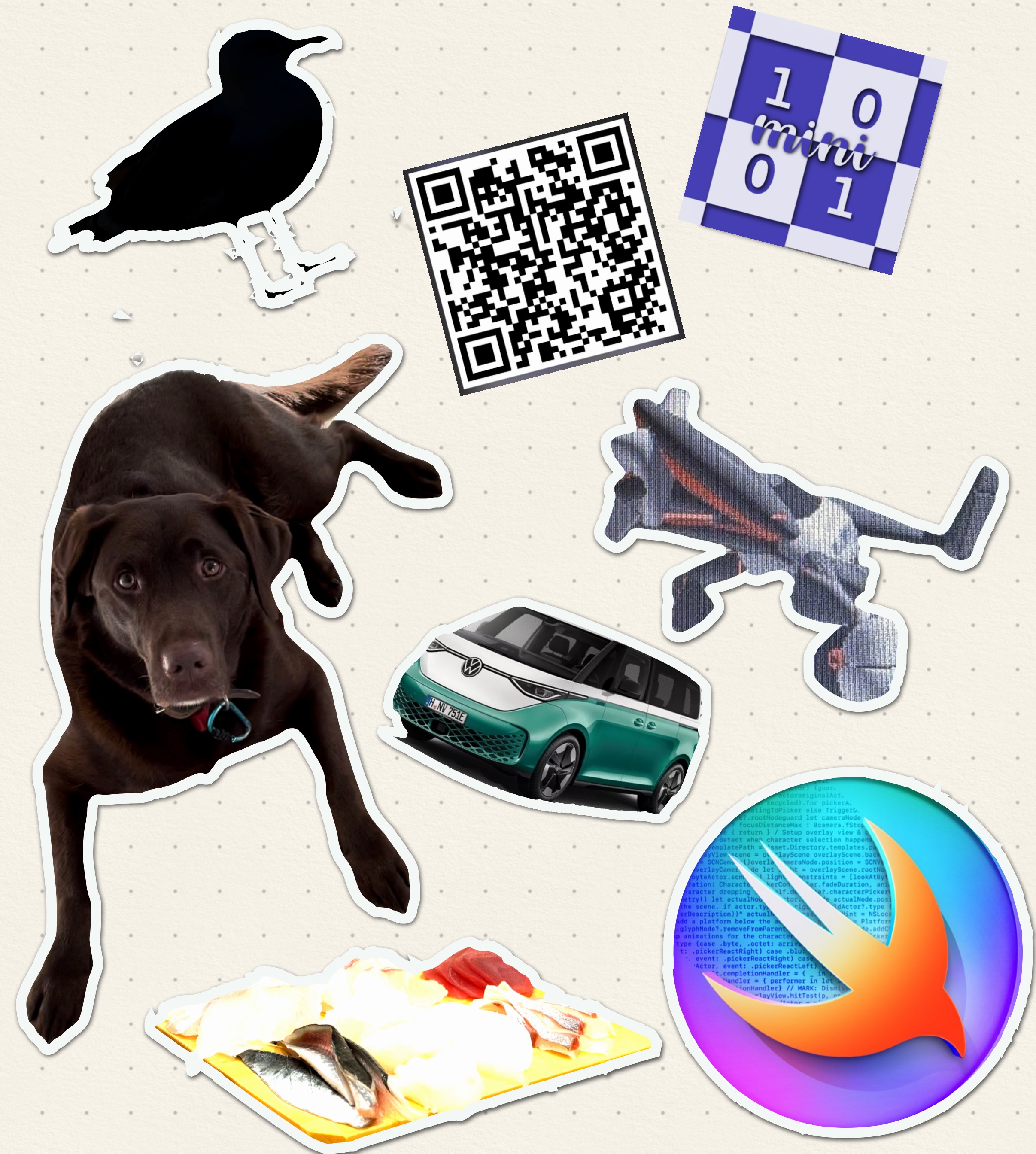
- Who am I
- What do I want
- AR == AR?
- iPad vs. Apple Vision Pro
- ARAnchor vs. Anchor
- 3DModels
- ECS
- Closing arguments

A collage on a light-colored, dotted background. The top half features a landscape with a rainbow arching over a field under a dark sky. On the right side, there is a black and white photograph of a smiling woman with her hand near her face. A purple rectangular banner with a torn edge is positioned across the middle, containing the text "Who am I?".

Who am I?

Meeeeeee

- Lena Mattea Stöxen
- Live near Hannover in Germany
- Student at Volkswagen Commercial Vehicles
- 3 x Swift Student Challenge Winner
- FancyGoose.dev
- The Assembler



What do I want?

Friendsbook Freunde buch

- Size = relevancy
↳ private?
- Color = emotions sliders
* then create color for emotion
- Shape = person who is sharing
- transparency = how detailed & memorable
- distance = time
distance from now

- shared with Cloudkit . shared DB
- Friendsbook , social media but friendship memories
- Text only • Date (approx.)
- Title • feeling in words/sliders
- Add people who don't have the app
↳ Contacts needs to be private
entity that links to shared with people

why?

why would you want that?
why not just remember things?
selfish? being told that you're great

Title

Thu/22/12/2023

Desc.

Feeling

calm

excited

good

bad

detiled

vague

important

irrelevant

Parts:

- iCloud
- contacts
- AR
- color

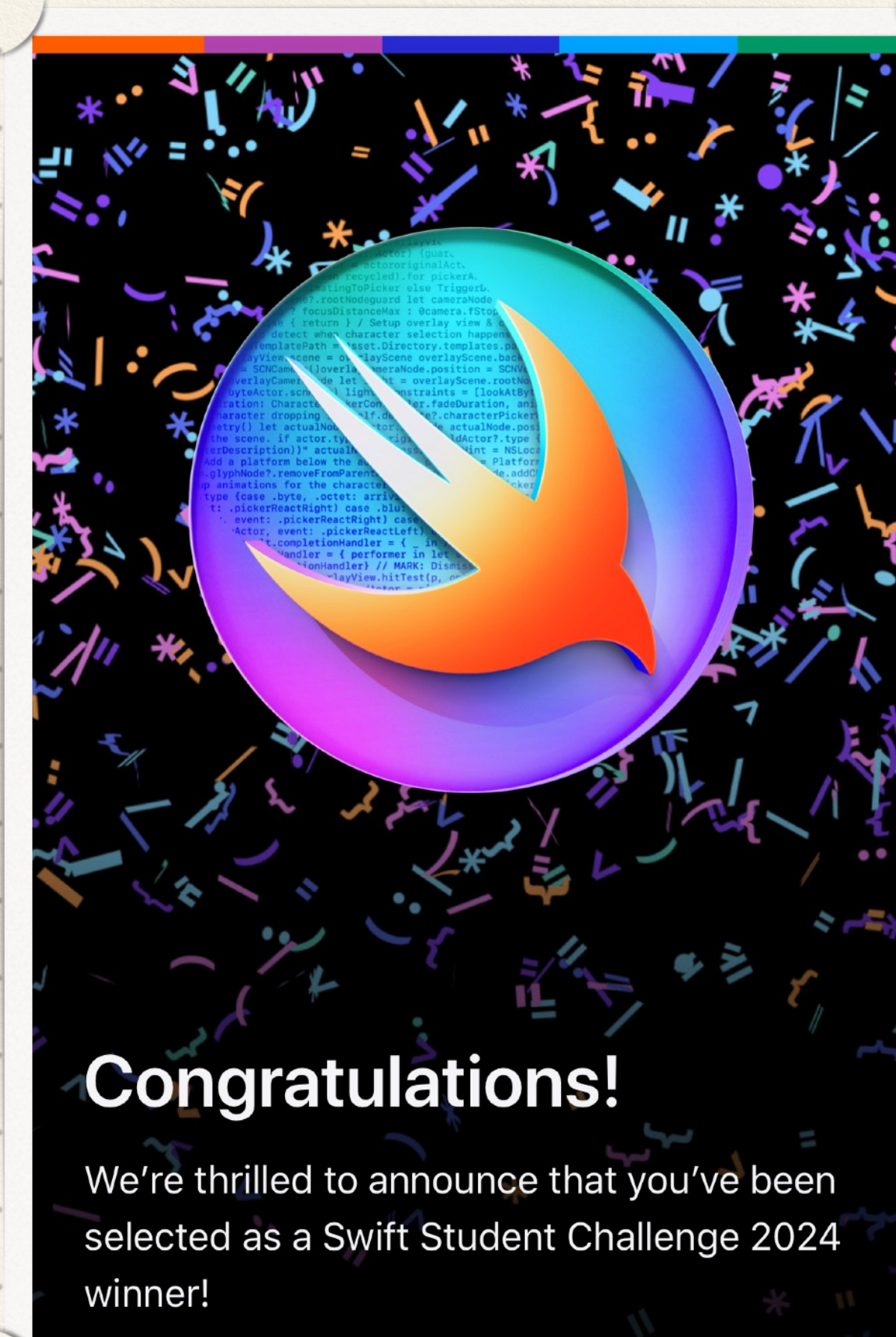
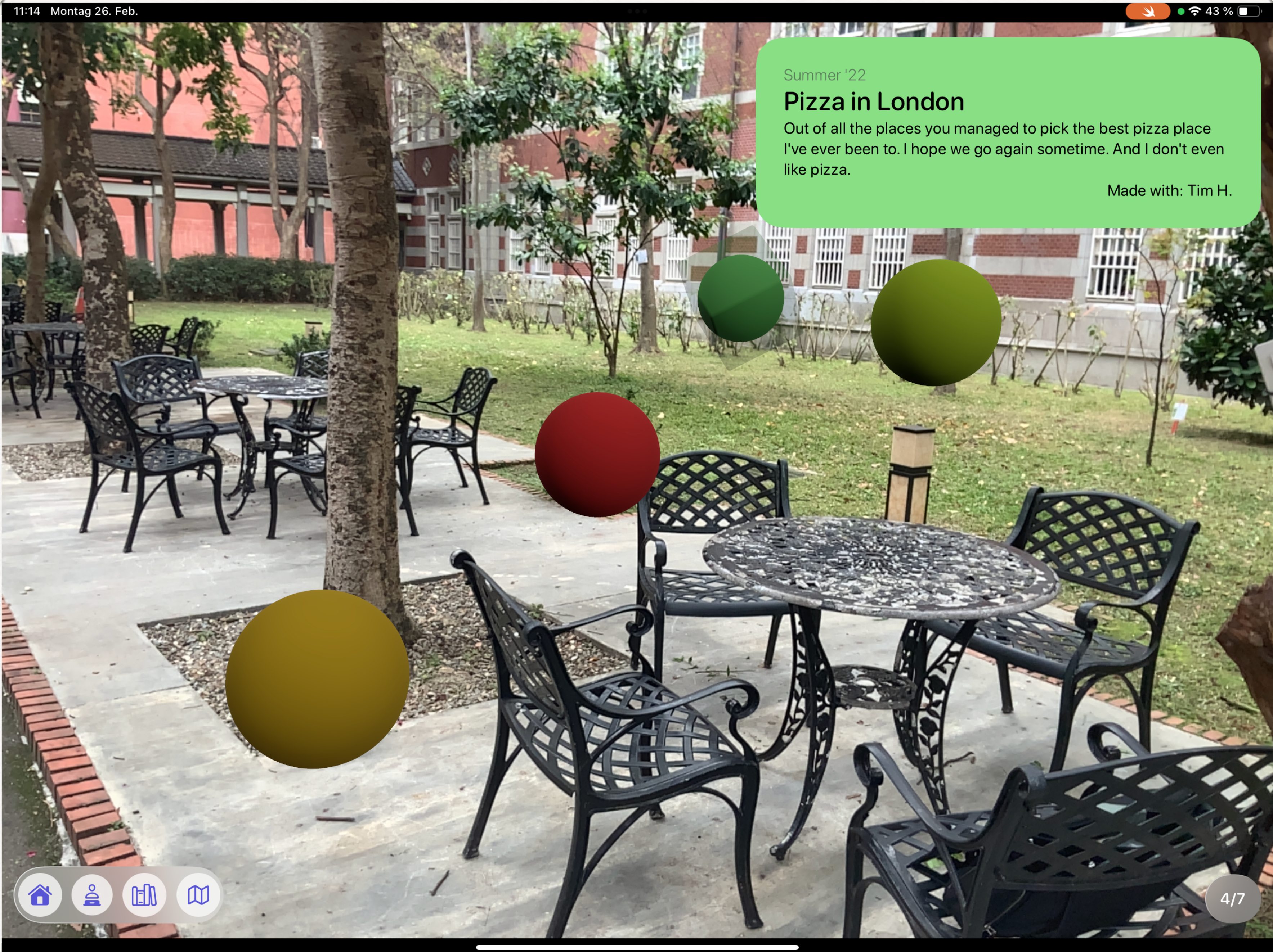
can I use AI to get a color
from emotion strings?

⇒ yes!

https://developer.apple.com/documentation/naturallanguage/creating_a_text_classifier_model

classify text by its
emotion, or the tags
to a value and do RGB?

"romantic" → 'red'

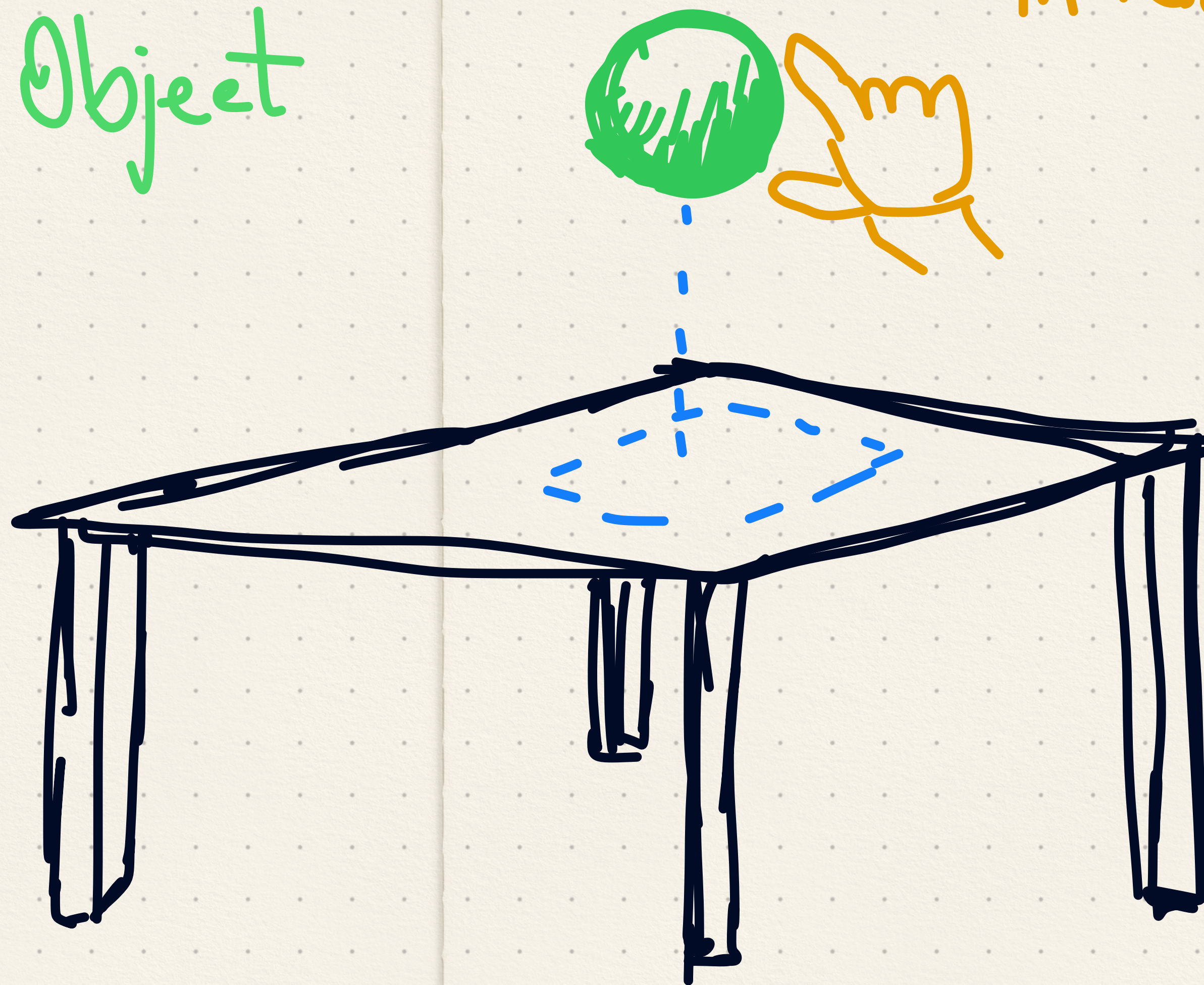


Virtual Object

interaction

Anchor

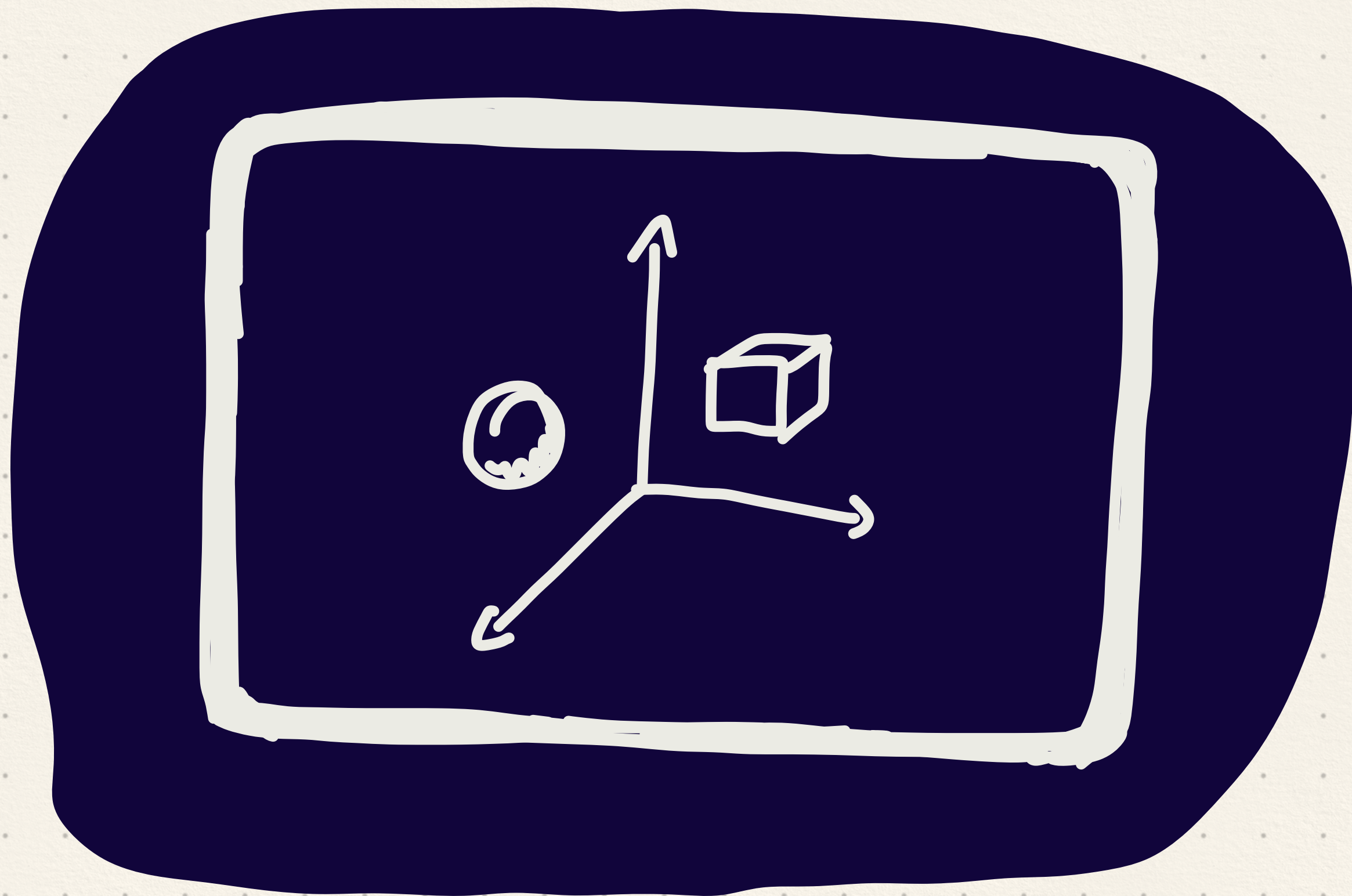
Real
World



„I want this to run on Vision Pro and iPad without changing all my code or having two codebases. And I want to test on my iPad.“

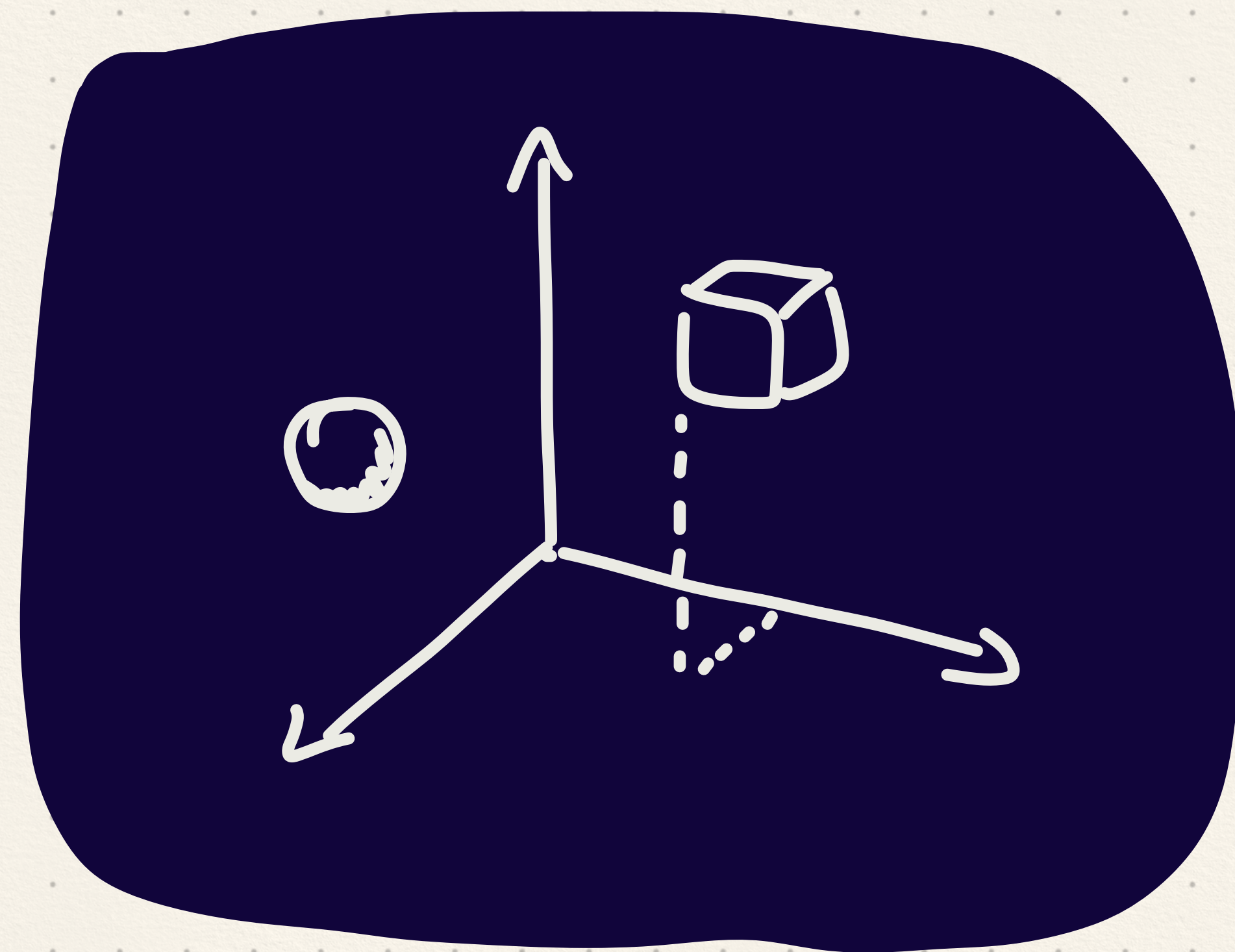
AR == AR?

iPad



This is AR

Vision Pro



This is also AR

„This should be easy!“

Getting Started

1. Start on iPad, works
2. Click 'Vision Pro' on Targets
3. 200 errors in Xcode

„So... this isn't easy?“

iPad vs. AVP

AR is different on Apple Vision Pro
and iPad

Concepts

iPad

UIKit

Camera Permission

ARView

ARKit Session

RealityKit Models

Concepts

iPad

UIKit

Camera Permission

ARView

ARKit Session

RealityKit Models

Apple Vision Pro

Swift UI

Tracking Permission

RealityView

ARKit Session Provider

RealityKit Models

Concepts

iPad

~~UIKit~~ SwiftUI



Camera Permission

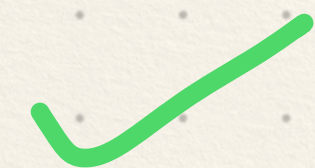
ARView

ARKit Session

RealityKit Models

Apple Vision Pro

SwiftUI



Tracking Permission

RealityView

ARKit Session Provider

RealityKit Models

Concepts

iPad

~~UIKit~~ SwiftUI ✓

Camera Permission ✓

ARView

ARKit Session

RealityKit Models

Apple Vision Pro

SwiftUI ✓

Tracking Permission ✓

RealityView

ARKit Session Provider

RealityKit Models

[SwiftUI](#) / [UIKit integration](#) / UINavigationControllerRepresentable

Protocol

UINavigationControllerRepresentable

A view that represents a UIKit view controller.

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst 13.0+ | tvOS 13.0+ | visionOS 1.0+

```
@MainActor @preconcurrency
```

```
protocol UINavigationControllerRepresentable : View where Self.Body == Never
```


[RealityKit](#) / [Views and attachments](#) / RealityView

Structure

RealityView

A view that contains RealityKit content.

RealityKit | SwiftUI | iOS 18.0+ | iPadOS 18.0+ | Mac Catalyst 18.0+ | macOS 15.0+ | visionOS 1.0+

```
@MainActor @preconcurrency
```

```
struct RealityView<Content> where Content : View
```


Concepts

iPad

~~UIKit~~ SwiftUI ✓

Camera Permission ✓

~~ARView~~ RealityView ✓

ARKit Session

RealityKit Models

Apple Vision Pro

SwiftUI ✓

Tracking Permission ✓

RealityView ✓

ARKit Session Provider

RealityKit Models

View

UIViewControllerRepresentable

UIViewController

Create 3D Content

ARKit Session Functions
for
AR Anchors

View

UIViewControllerRepresentable

UIViewController

Create 3D Content

ARKit Session Functions
for
AR Anchors

View

Reality View

Create 3D Content

Anchoring? 222
...o

Anchor vs. ARAnchor

ARKit / Anchor

Protocol

Anchor

The identity, location, and orientation of an object in world space.

visionOS 1.0+ | Xcode 15.0+

protocol Anchor

ARKit / ARAnchor

Class

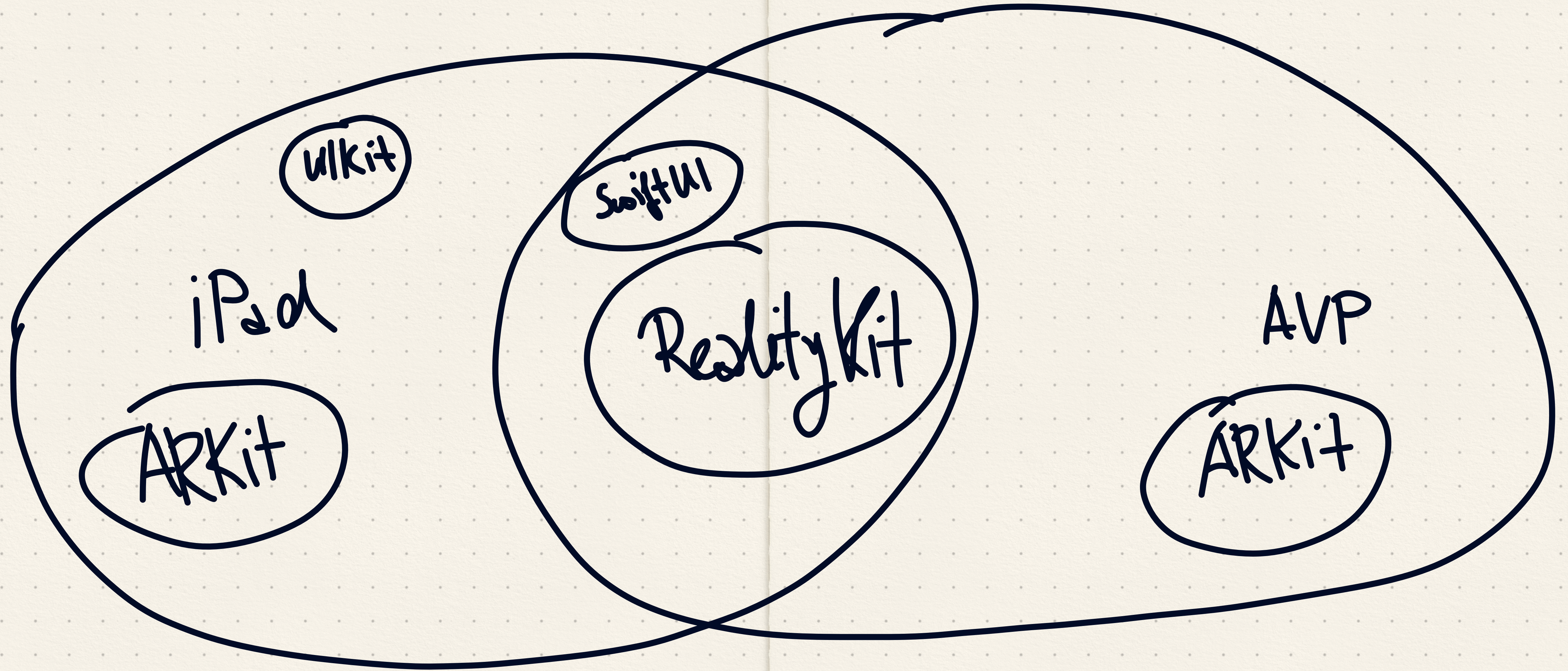
ARAnchor

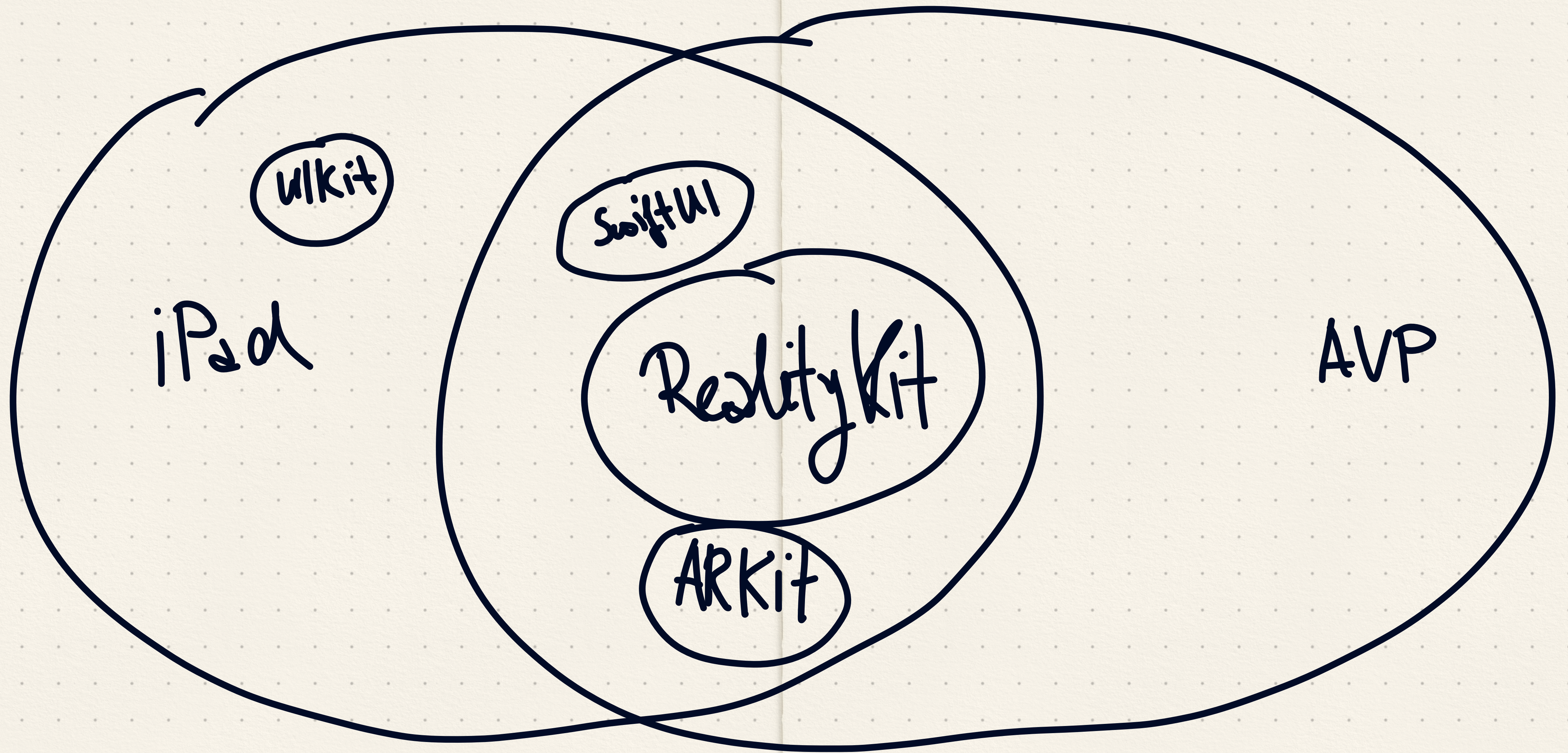
An object that specifies the position and orientation of an item in the physical environment.

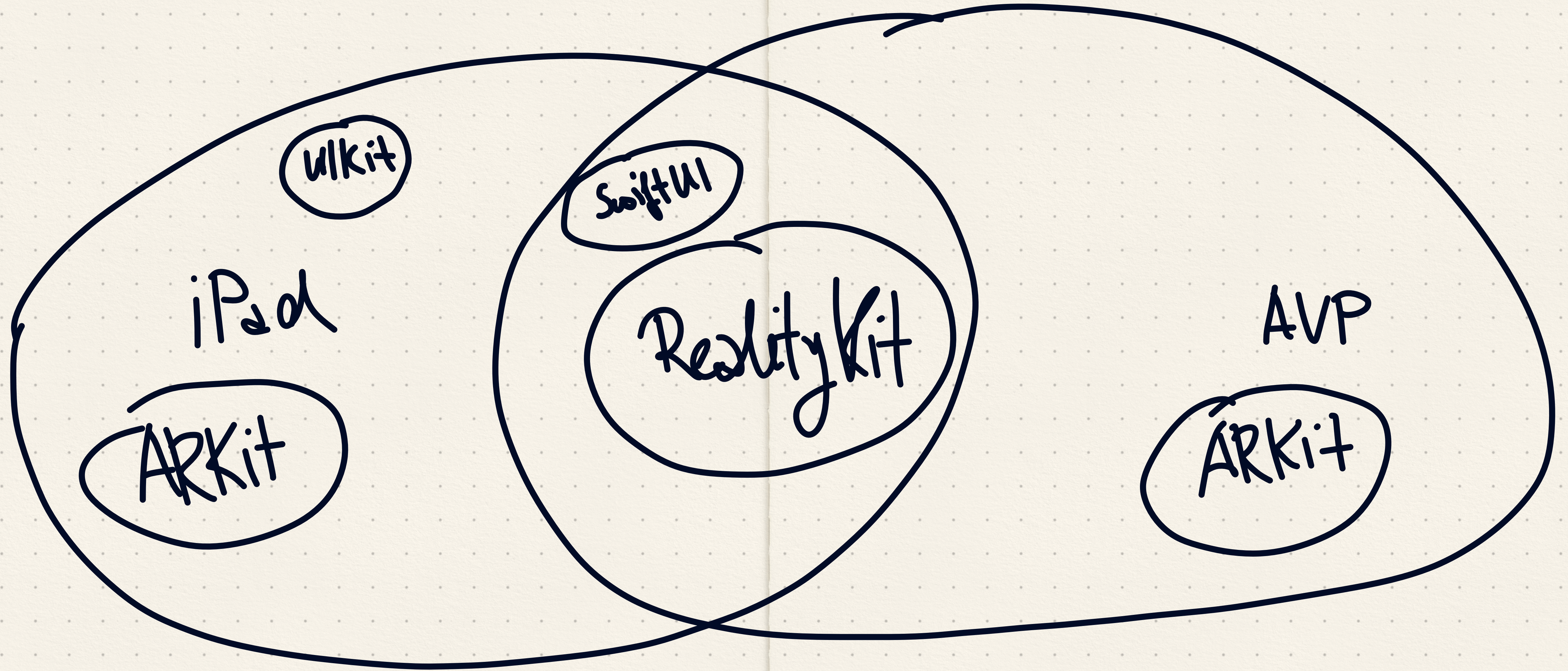
iOS 11.0+ | iPadOS 11.0+ | Mac Catalyst 13.1+

```
class ARAnchor : NSObject
```


The anchors aren't compatible.







Two Options

- Different code for anchoring
- Conditional compiling for incompatible code
- Or
- RealityKit's AnchorEntities for both devices

```
#if os(visionOS)  
#endif
```

AnchorEntities

ARAnchor & Anchor

- Handed to you async as they are 'discovered'
- Classification as table, wall, etc.
- Good tracking
- I like them
- Only testable on device

AnchorEntity

- Kinda random
- Less control
- Ok tracking
- I don't like them
- Testable in Vision Pro Simulator

Concepts

iPad

~~UIKit~~ SwiftUI ✓

Camera Permission ✓

~~ARView~~ RealityView

ARKit Session #

AnchorEntity ✓

RealityKit Models

Apple Vision Pro

SwiftUI ✓

Tracking Permission ✓

RealityView ✓

ARKit Session Provider #

AnchorEntity ✓

RealityKit Models

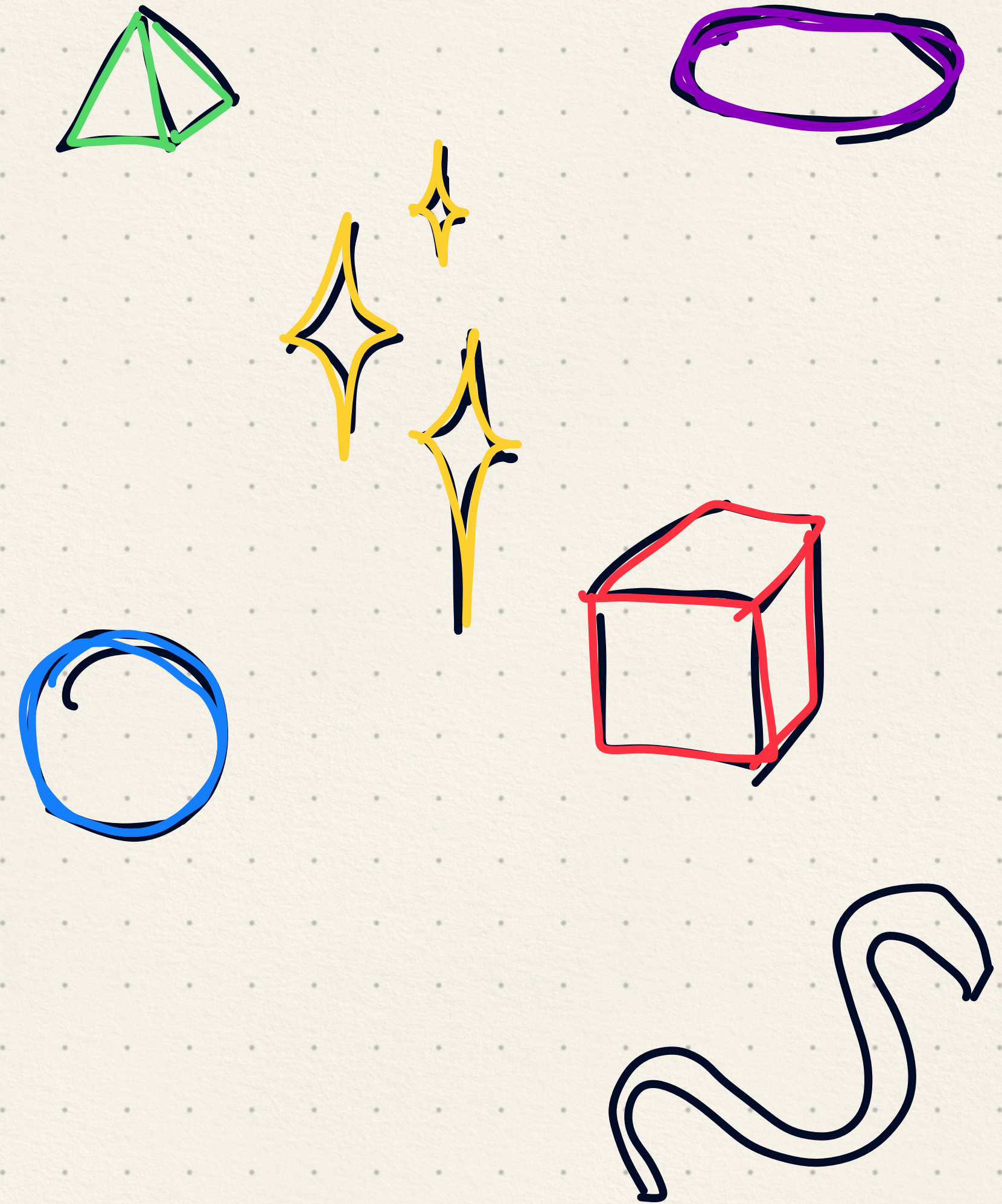
3D Models

3D Models

- Import

or

- Create




```
// creating an orange sphere
```

```
let mesh = MeshResource.generateSphere(radius: 0.1) // in meters
```

```
let orangeMaterial = SimpleMaterial(color: .orange, isMetallic: false)
```

```
let entity = ModelEntity(mesh: mesh, materials: [orangeMaterial])
```


„Why is that called ‘ModelEntity’?“

ECS

Entity

Component

System

Everything's

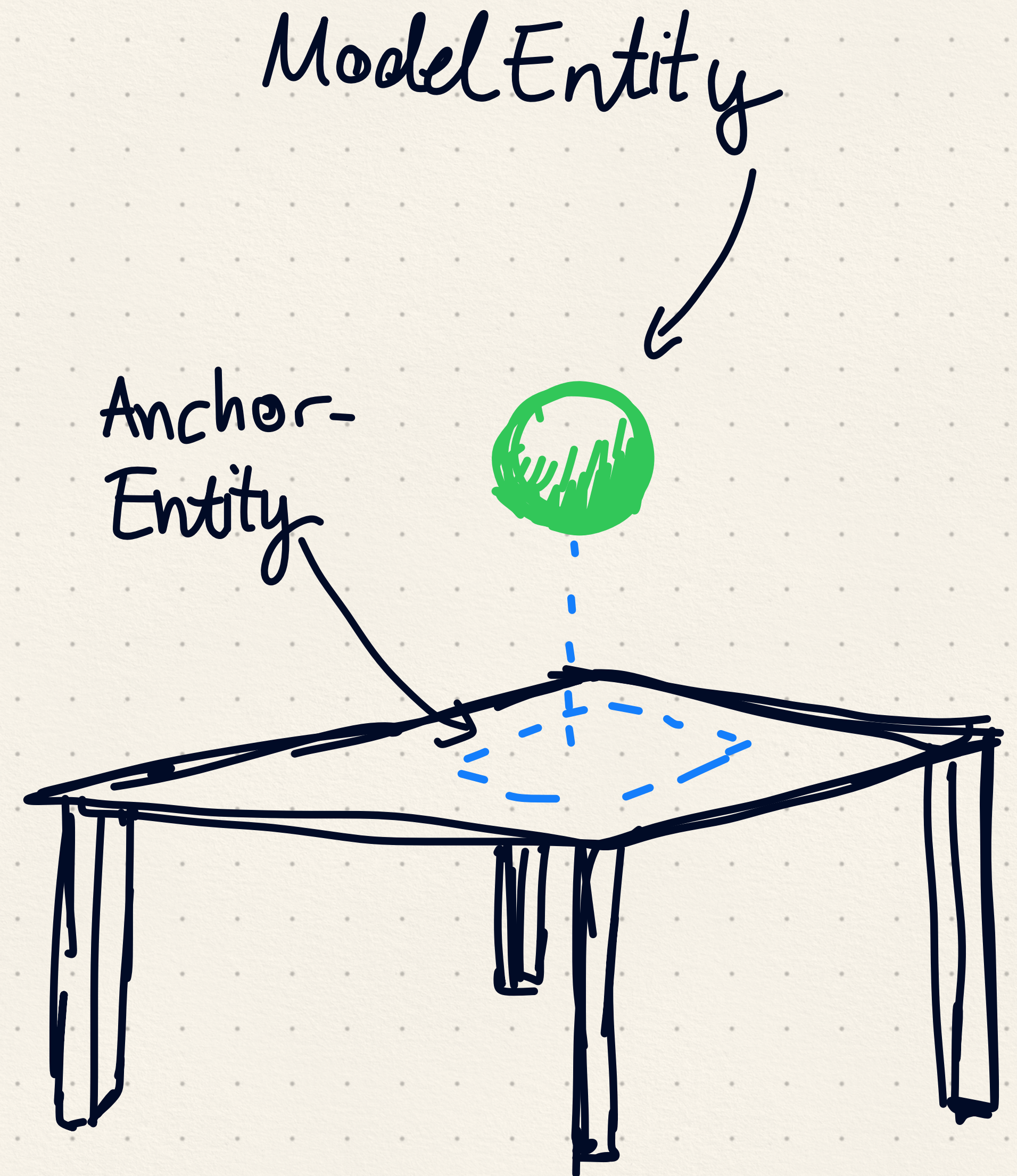
Cooler in

Space

„What’s an ‘Entity’?“

Everything's an Entity

- Building blocks of experience
- Types:
 - ModelEntity
 - AnchorEntity
 - Entity



Entities are Hulls

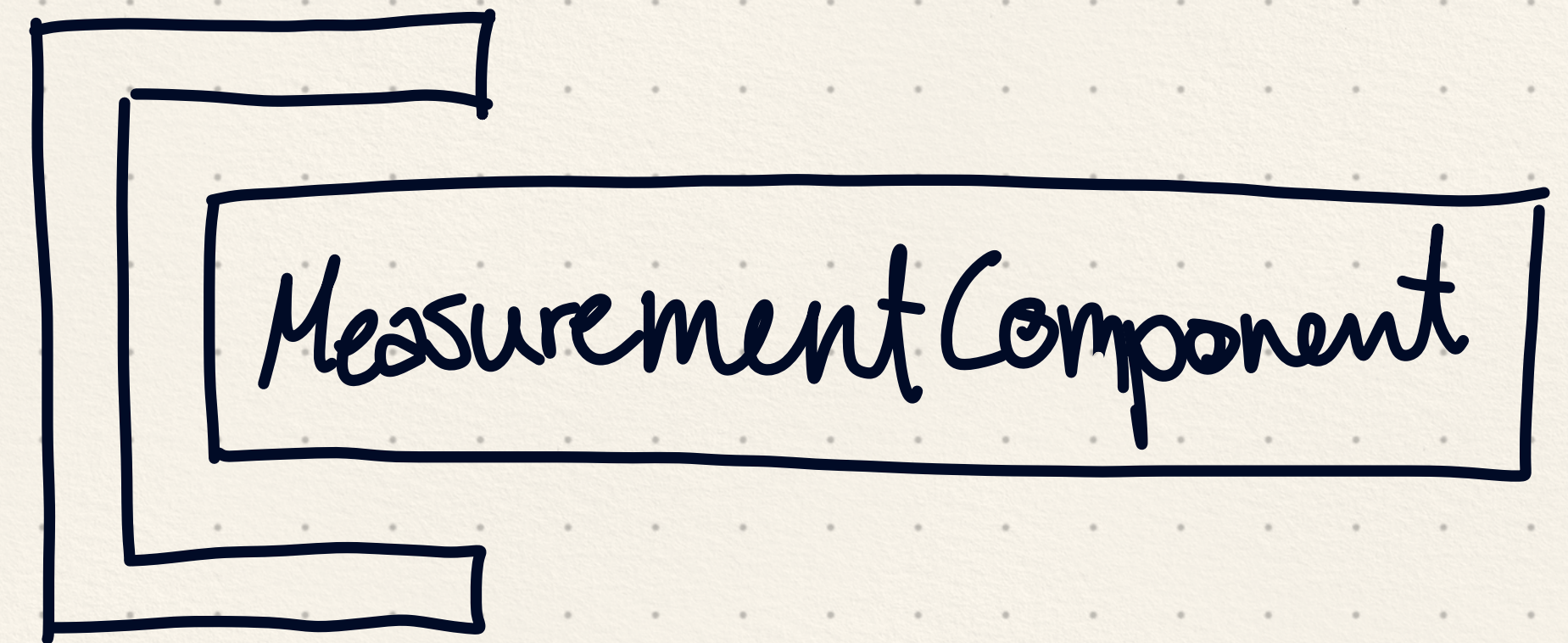
- A bit like an Instance of an empty class
- In classes, properties live on the class
- Entities use components to save properties



Components

- Entities can be filled with Components

Entity



```
let entity = Entity()  
entity.components.set(MyComponent())
```


Components

- Components can have Properties
- Components can be empty

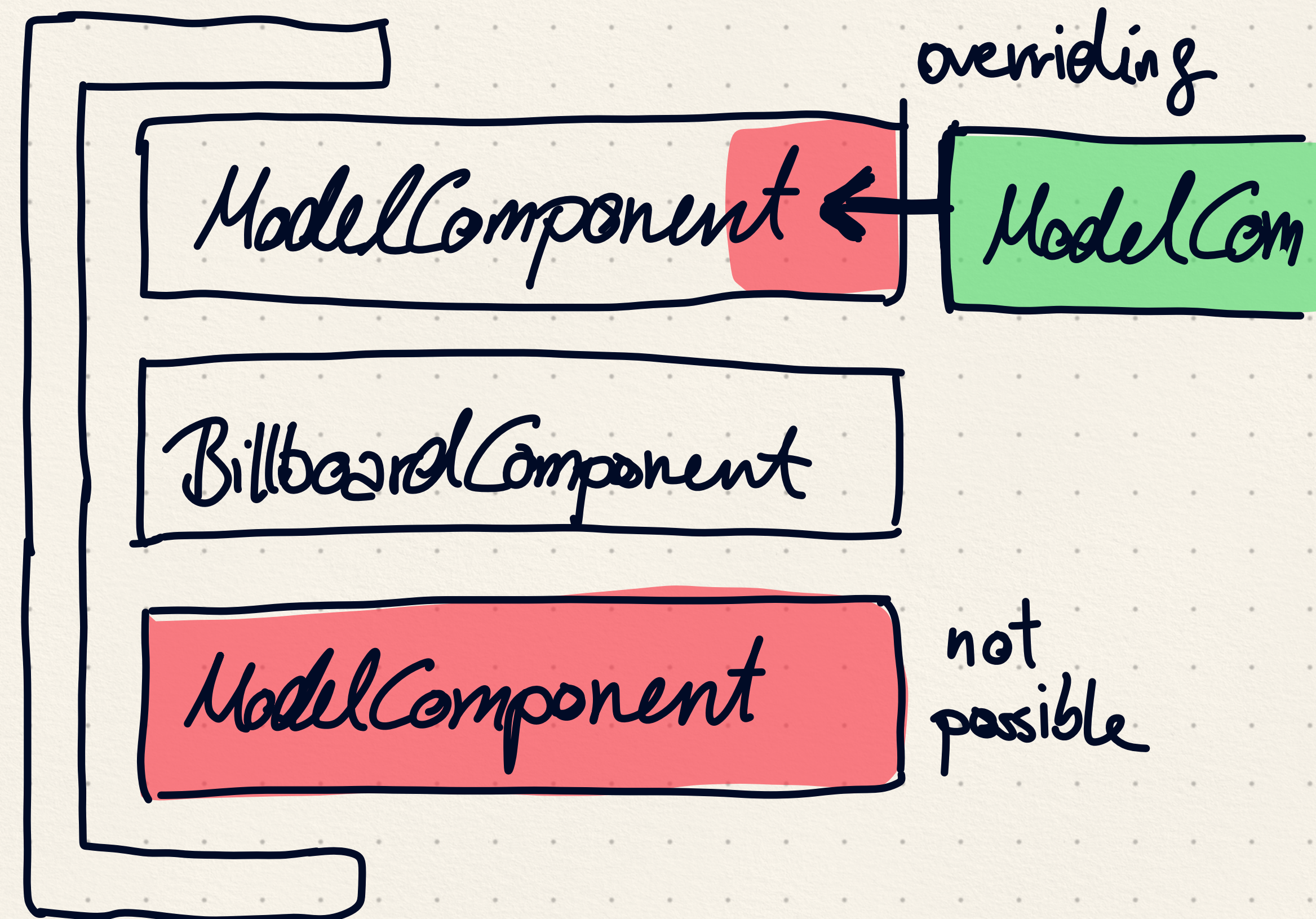
```
// components do not need to have a body  
struct SelectedComponent: Component {}
```

```
enum RotationState{  
    case none, x, y, z  
}  
  
// Component to power the rotation state  
struct RotationComponent: Component {  
    var state: RotationState  
}
```


Set Components

- Entities can only have one of each component
- If components are added twice, they override each other

Entity



Components

- Components used for customisation
- There are standard components
- You can make custom components

ModelComponent

PhysicsBodyComponent

BillboardComponent

CollisionComponent

PortalComponent

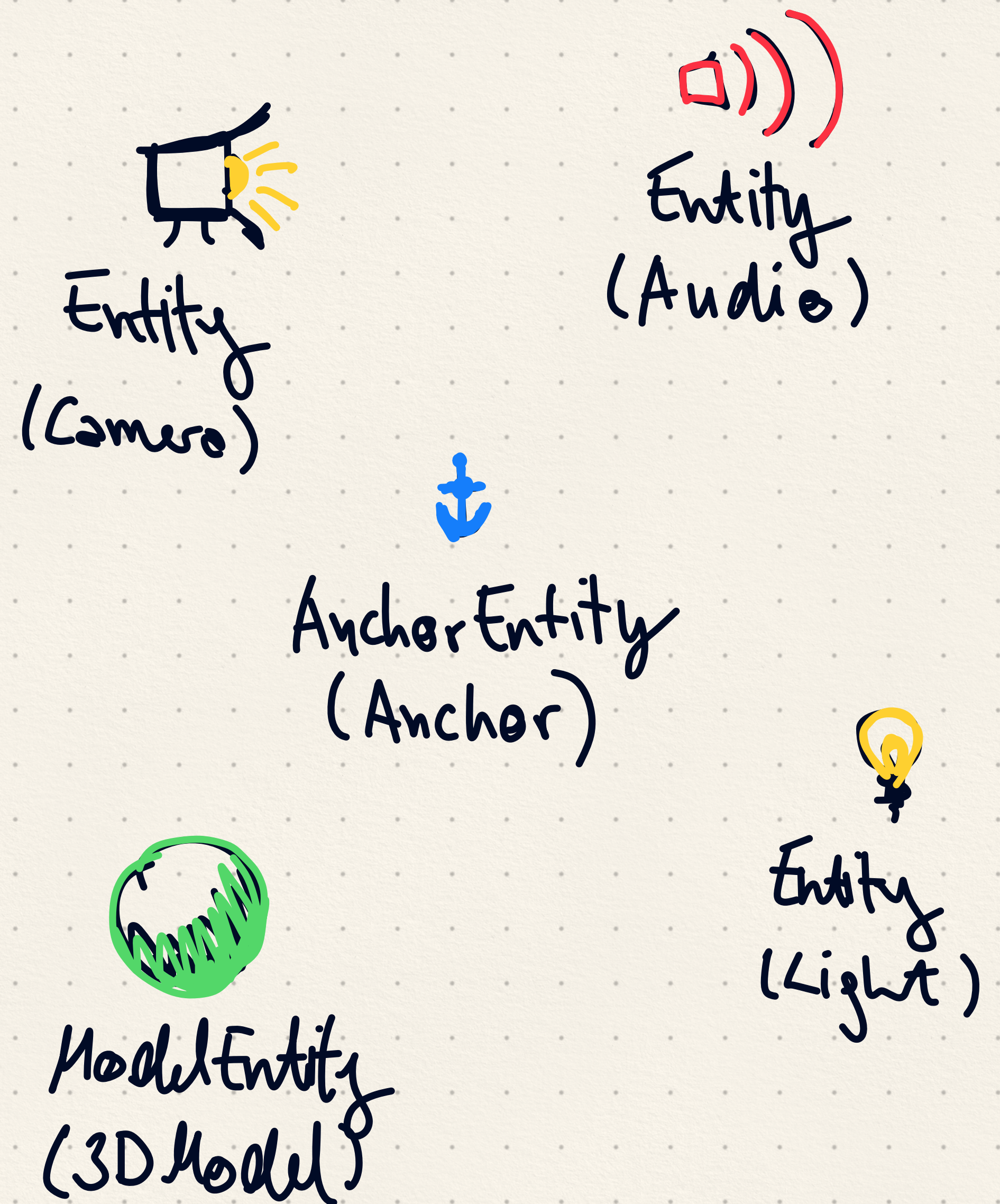
AccessibilityComponent

HoverEffectComponent

PointLightComponent

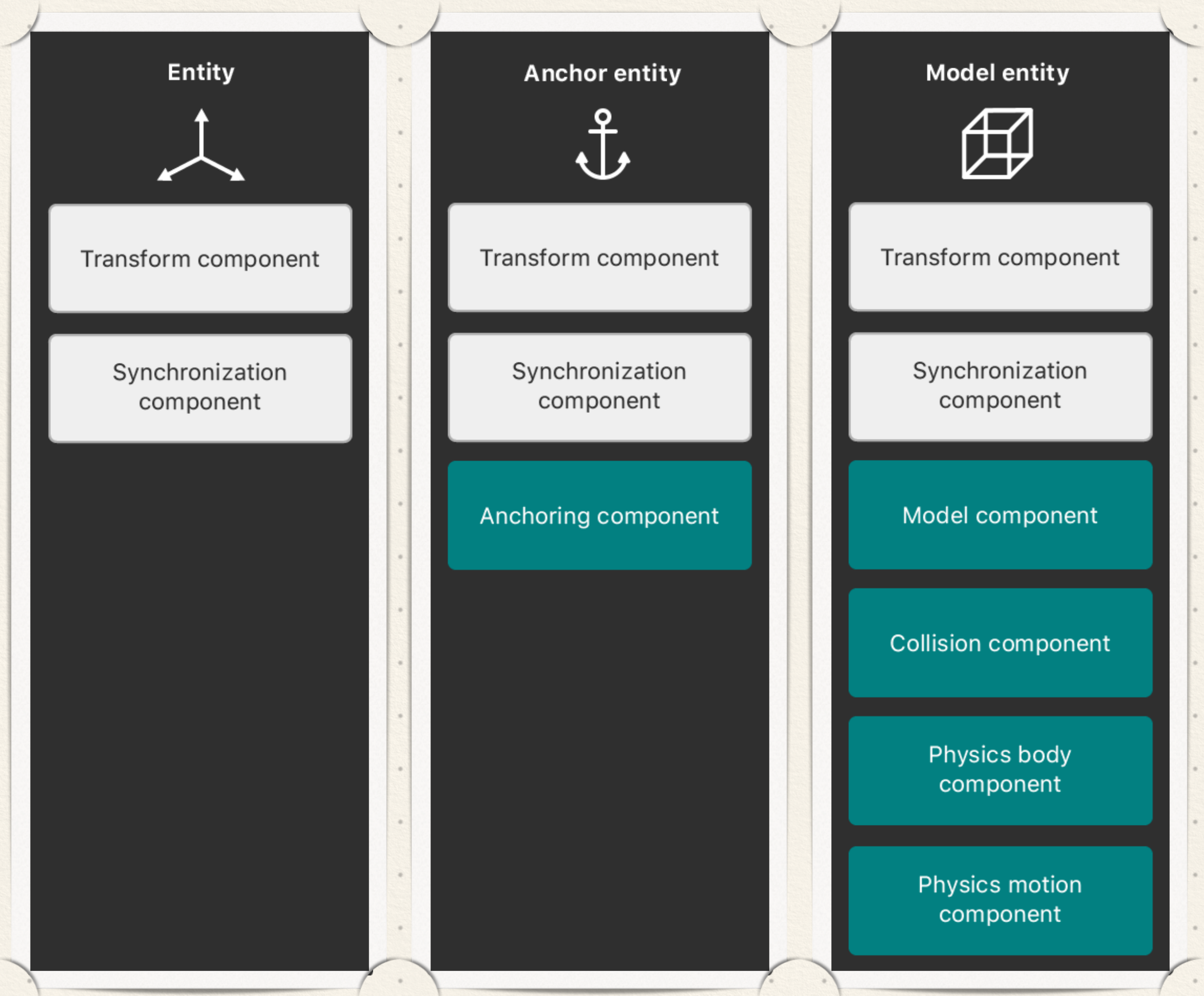
Entities are Anything

- Entities can be:
 - Anchors
 - Models
 - Lights
 - Audio
 - Cameras



Anchor-, Model-, Entity

- Presets guarantee certain components
- You can build presets from Entities yourself



[RealityKit](#) / [Presentation UI](#) / BillboardComponent

Structure

BillboardComponent

A component that orients an entity instance so that it continuously points toward the active camera.

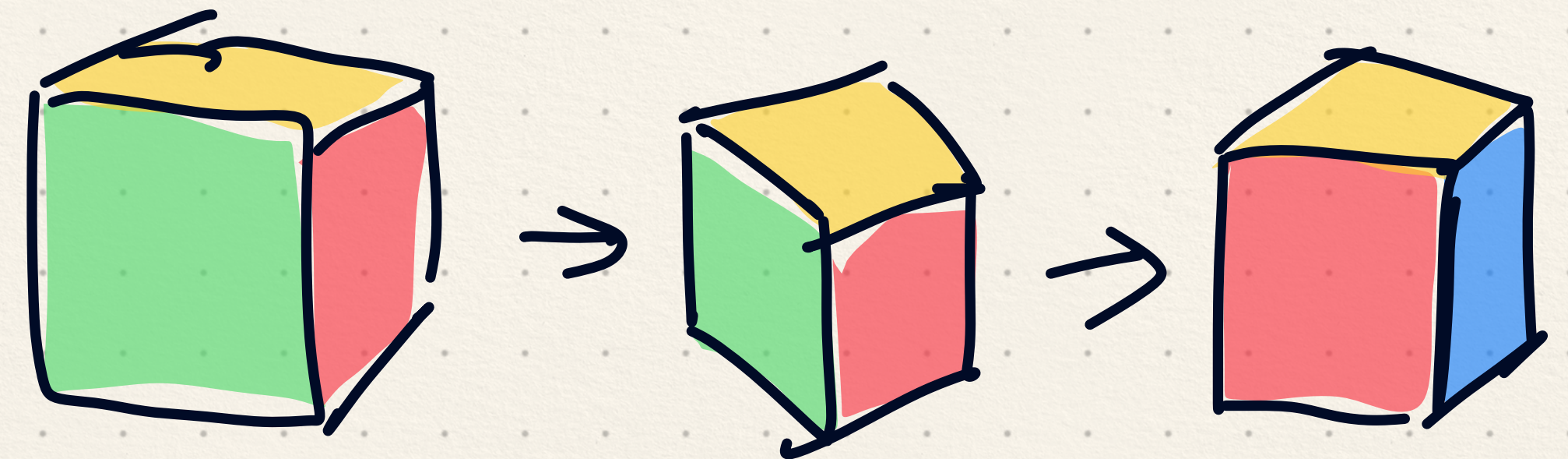
iOS 18.0+ | iPadOS 18.0+ | Mac Catalyst 18.0+ | macOS 15.0+ | visionOS 2.0+

```
struct BillboardComponent
```

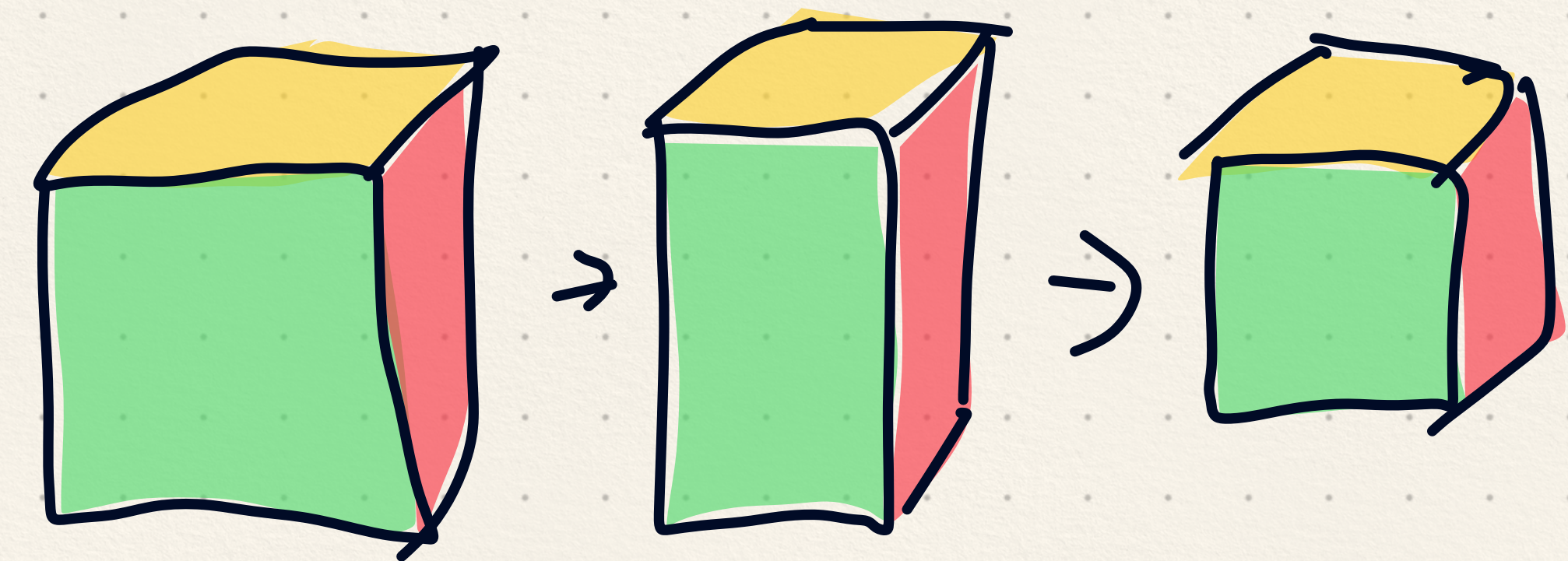

Billboards

- Billboards turn to always face you
- User can't "walk around" a model
- Used for signs or text

without:



with:



„So, the BillboardComponent turns
the object?“

Systems

- Components often have corresponding Systems
- Systems perform an action every rendering
- Custom Components and Systems have to be registered before use

Component	System
Properties	Functions

```
RotationComponent.registerComponent()  
RotationSystem.registerSystem()
```

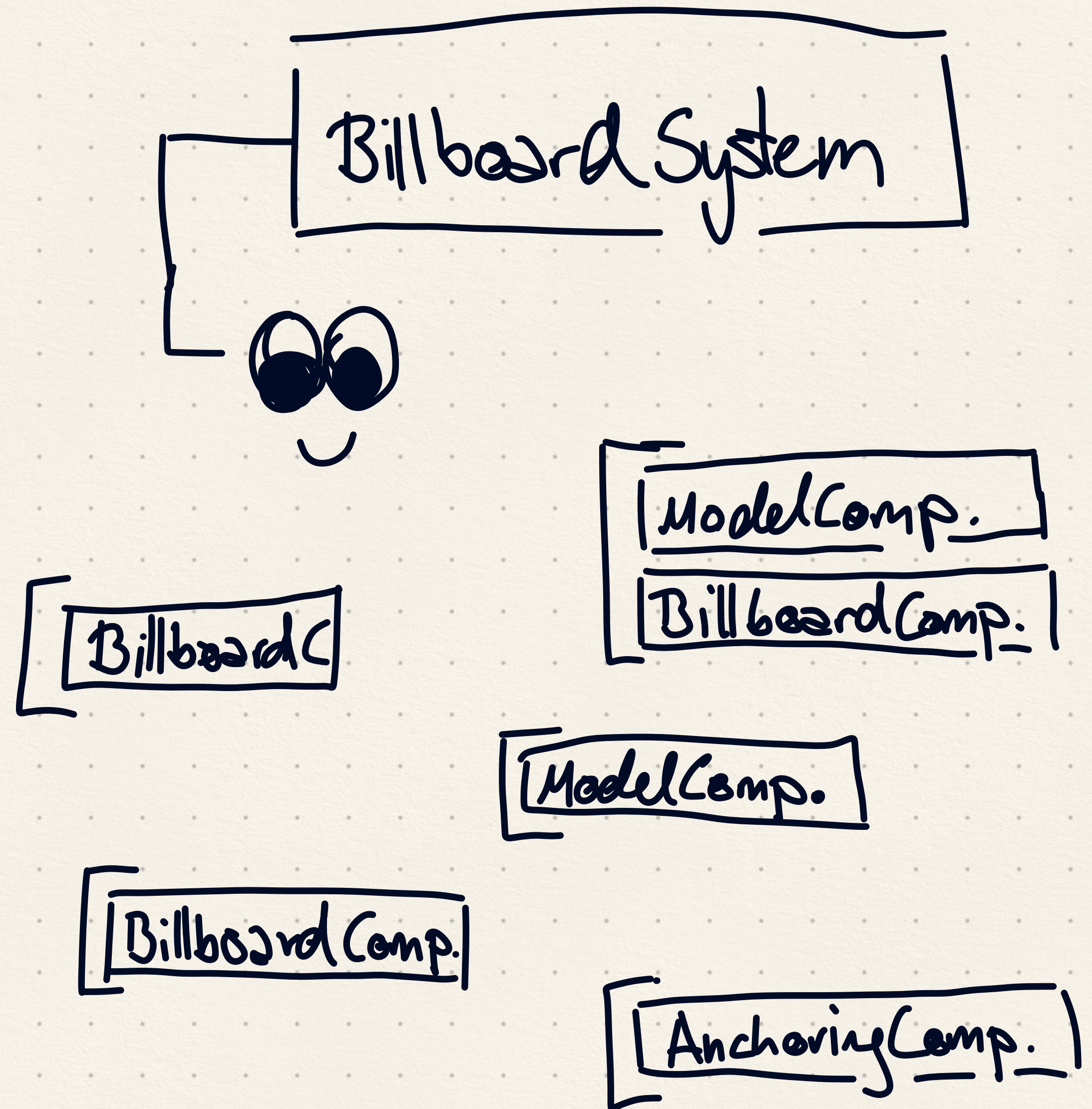

Components do not perform
actions.

Systems do.

Components supply information.

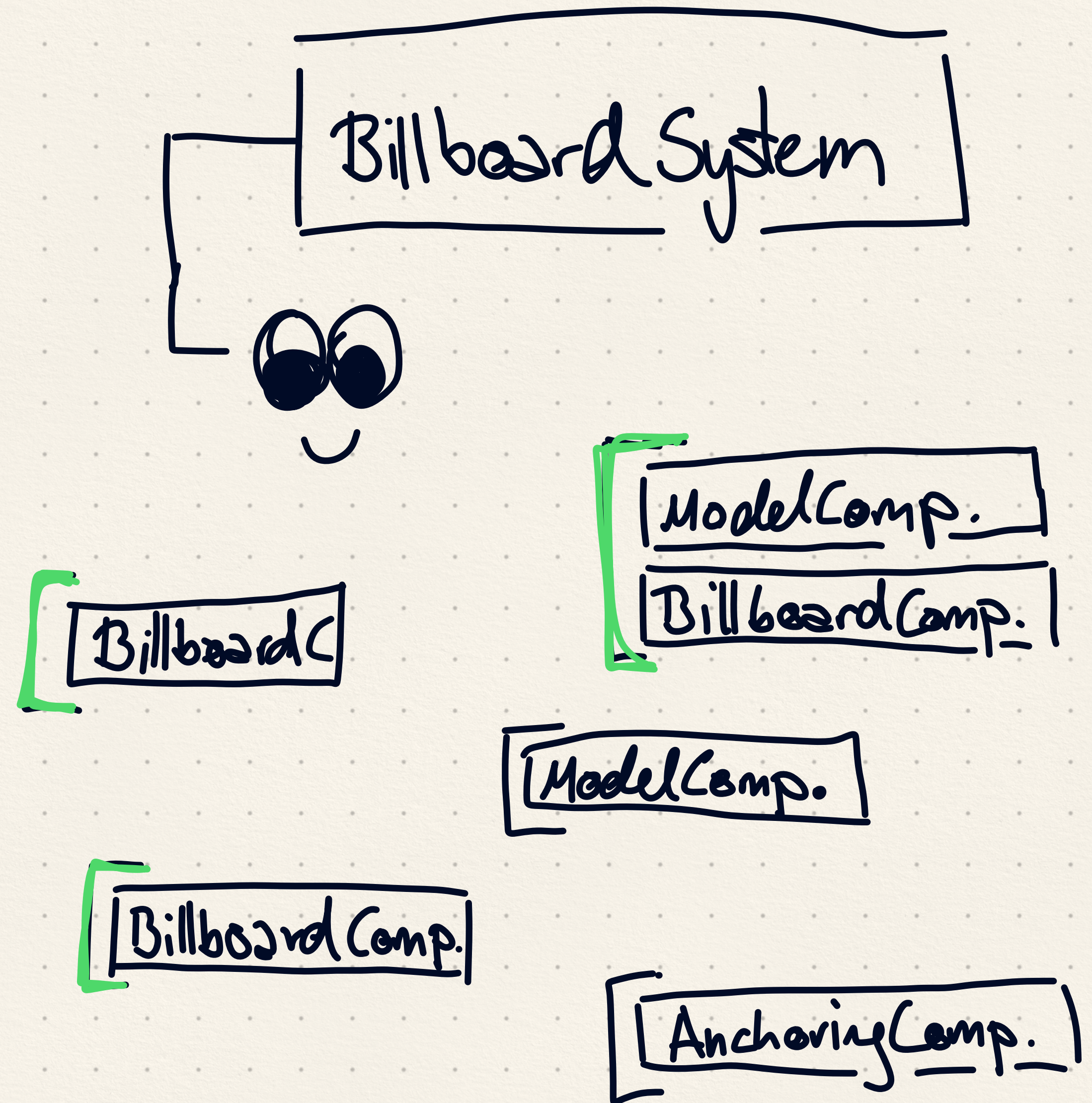
Systems

- Systems can filter which entities they act upon
- Generally, they filter by available components
- Iterate over all applicable entities



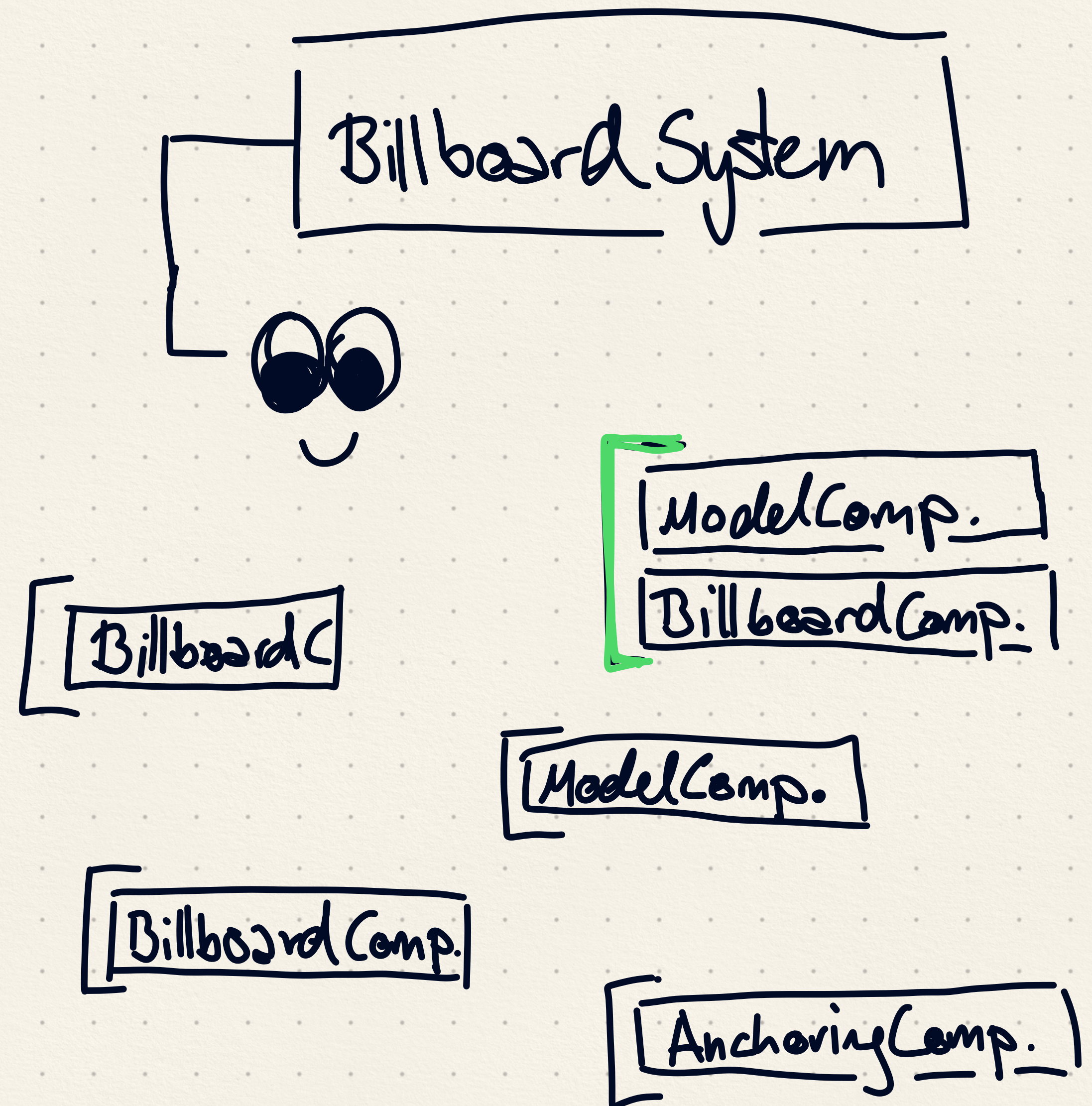
Systems

- Systems can filter which entities they act upon
- Generally, they filter by available components
- Iterate over all applicable entities



Systems

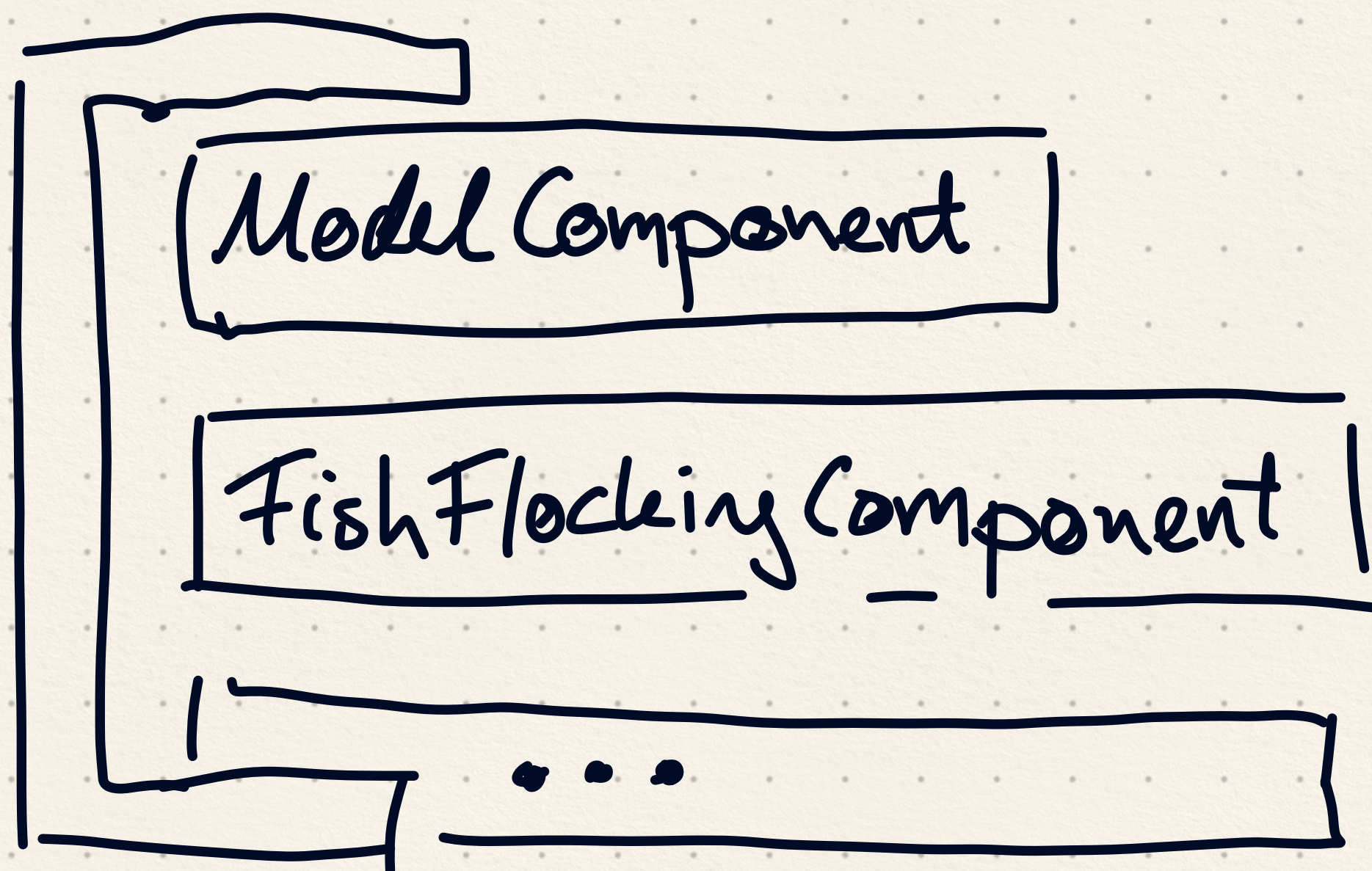
- Systems can filter which entities they act upon
- Generally, they filter by available components
- Iterate over all applicable entities




```
// System to add a rotation to objects with model and rotation component
class RotationSystem : RealityKit.System {
    // Define a query to return all entities with certain Components
    private static let query = EntityQuery(where: .has(RotationComponent.self) && .has(ModelComponent.self))
    // Initializer is required
    required init(scene: RealityKit.Scene) { print("Rotation System! Wohoooo!") }

    // Check all entities with rotation and model component
    func update(context: SceneUpdateContext) {
        context.scene.performQuery(Self.query).forEach { entity in
            guard let entity = entity as? ModelEntity else { return }
            // do something
        }
    }
}
```


Entities



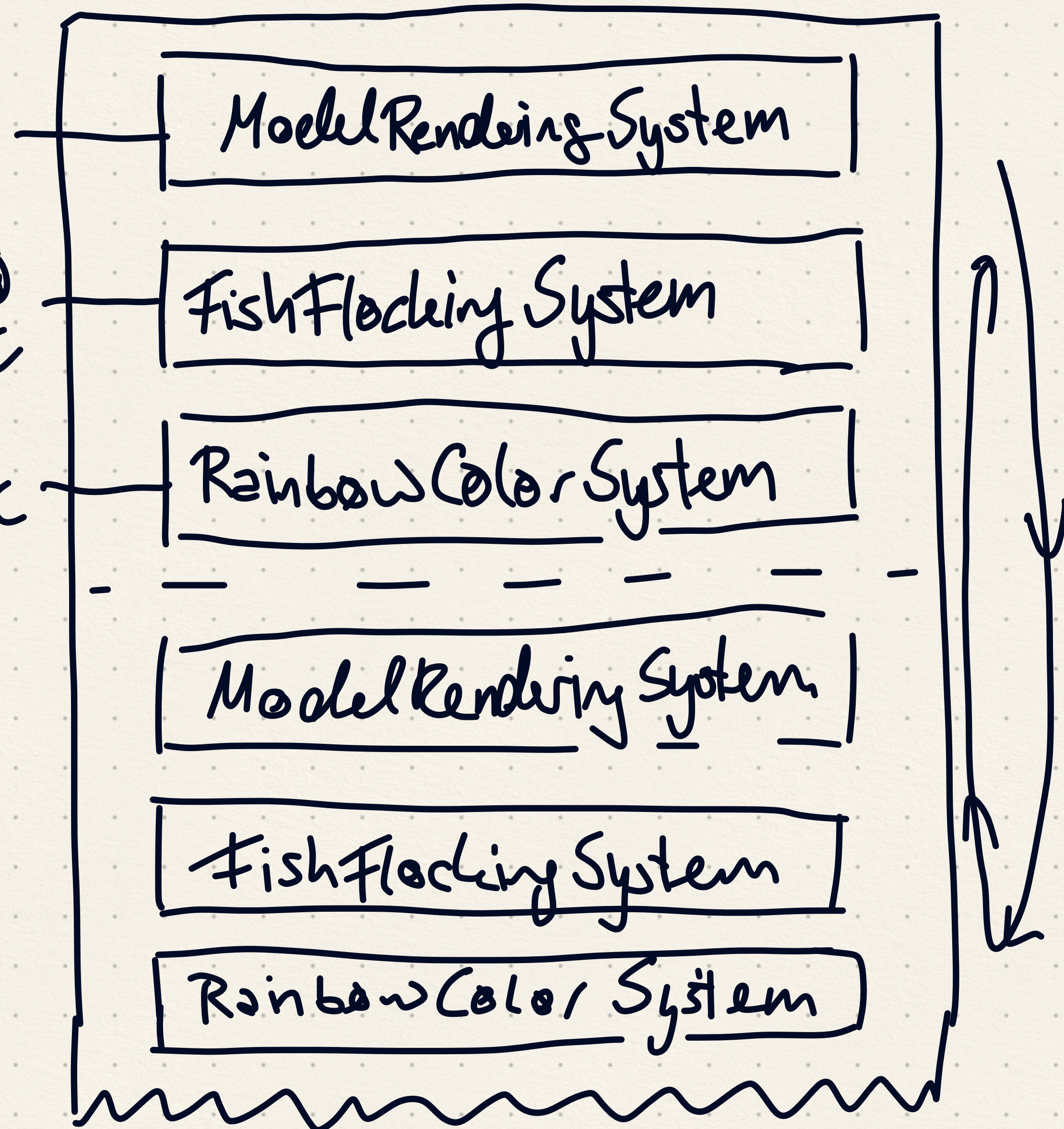
MC



FFC



RCC



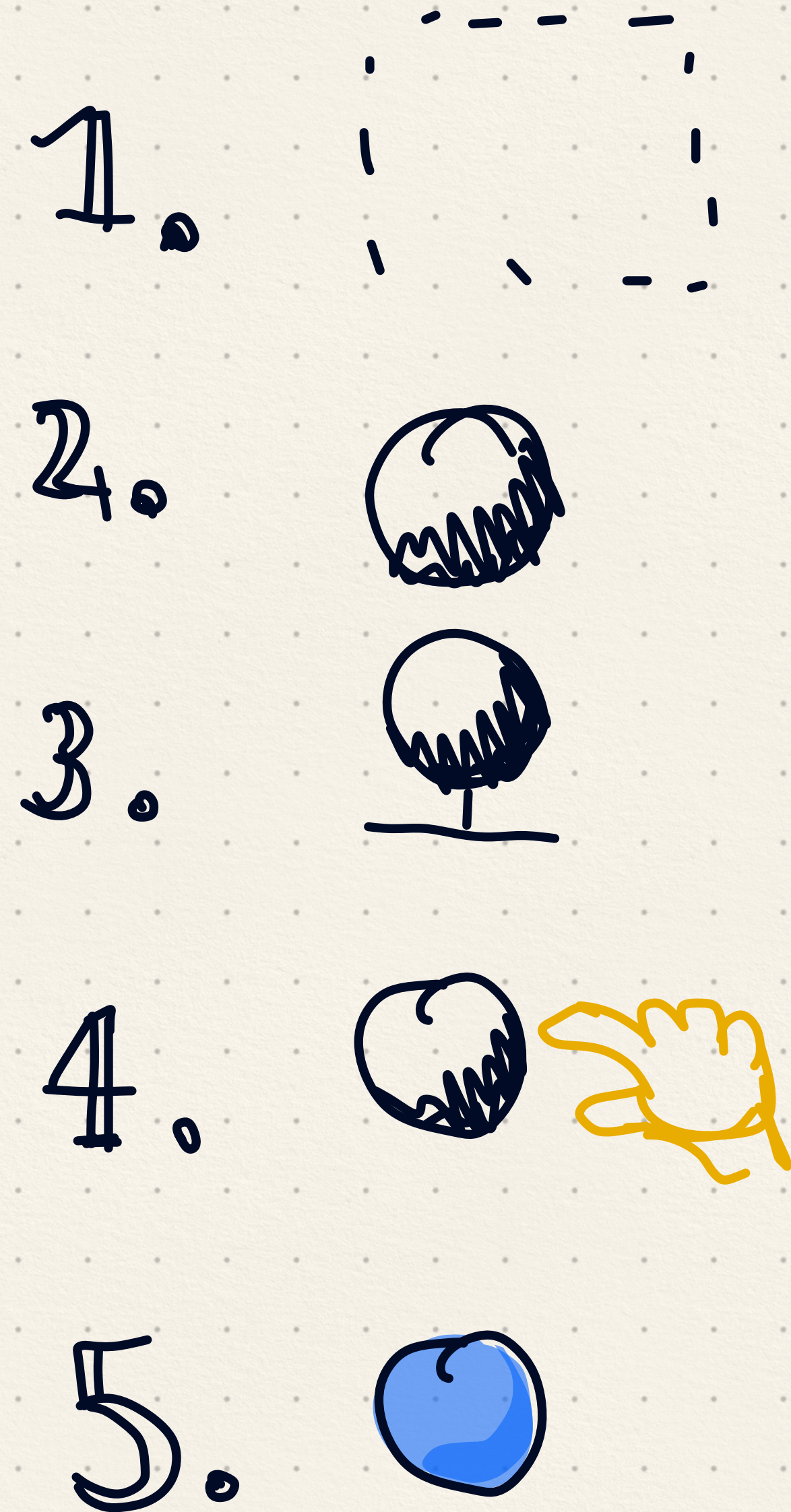
Systems run very often.
**DO NOT DO HEAVY WORK IN
THEM.**

You can create lags by performing
time consuming tasks in them.

We are only doing this, because it
is more efficient.

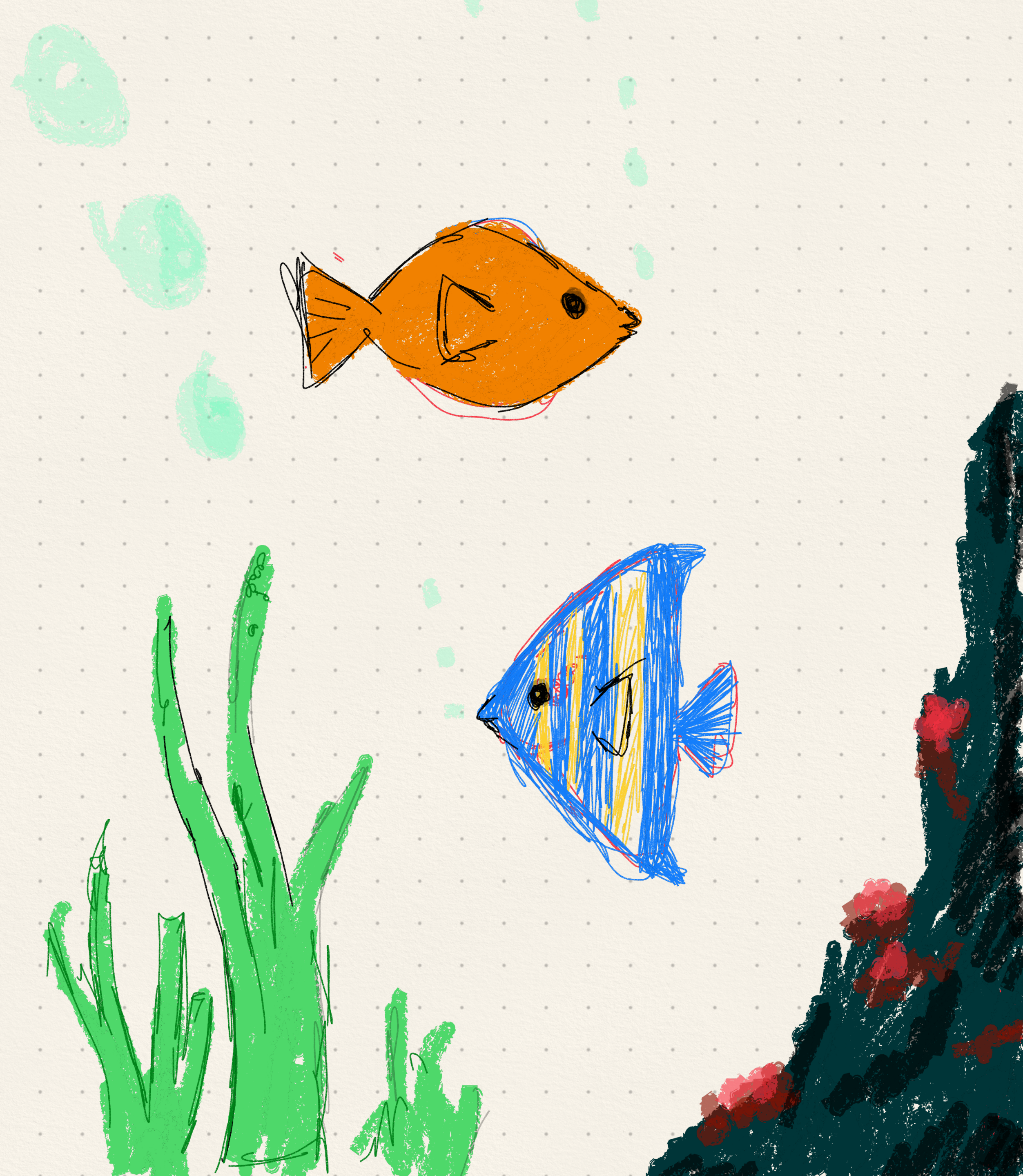
Basic Steps

- Creating Entities
- Adding Components
- Adding to anchor
- Changing Components based on user input
- Let systems do the rest



Good samples to look at

- 'Underwater' Sample project from Apple
 - Flocking and Fear-System for fish
- 'Spatial Drawing' App Sample from Apple
 - Extrusion and metal shaders



Community

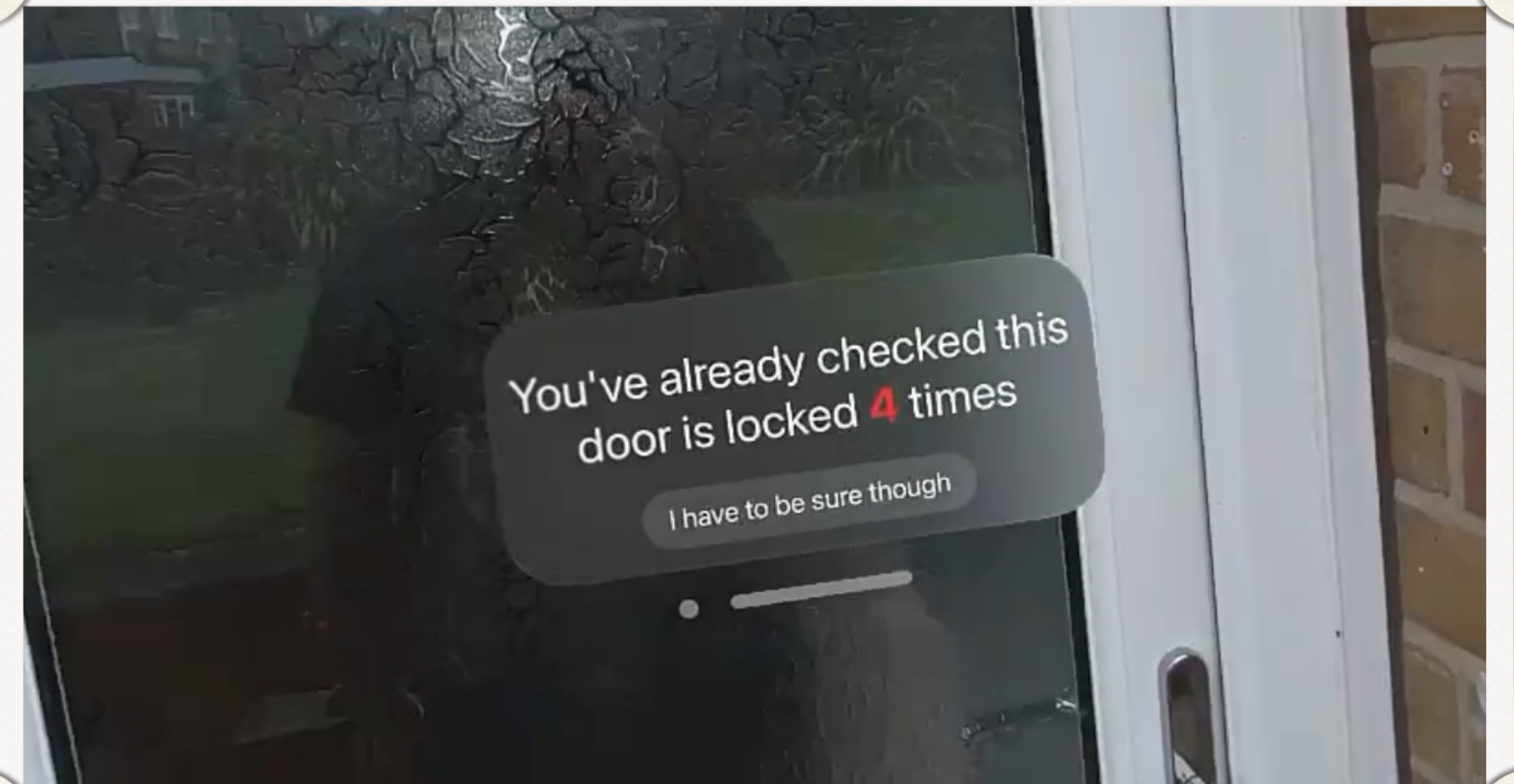
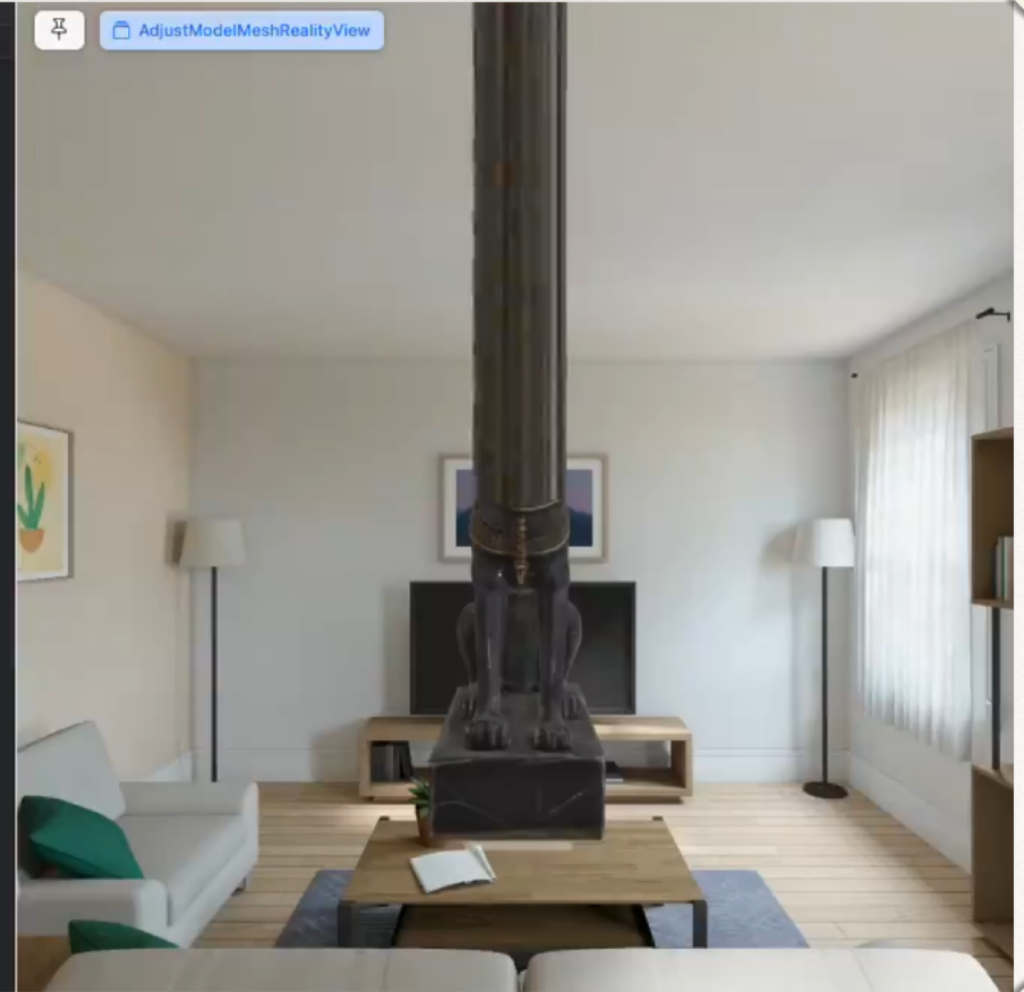
- Matt Pfeiffer's RealityKit Experiments
 - Materials, Mesh creation
- Paolo's Jumpfinity
- Matt Heaney's app ideas on X
 - Growing tasks, did I lock the door?

```
struct LowLevelMeshRevealView: View {
    func createMesh(from modelEntity: ModelEntity) ->
        LowLevelMesh? {
        guard let meshResource = modelEntity.model?.mesh,
            let meshPart =
                meshResource.contents.models.first?.parts
                .first else {
            return nil
        }

        let positions =
            meshPart[MeshBuffers.positions]?.elements ?? []
        let normals =
            meshPart[MeshBuffers.normals]?.elements ?? []
        let textureCoordinates =
            meshPart
            [MeshBuffers.textureCoordinates]?.elements ?? []
        let triangleIndices =
            meshPart.triangleIndices?.elements ?? []

        print("positions.count \(positions.count)")
        print("normals.count \(normals.count)")
        print("textureCoordinates.count
            \(textureCoordinates.count)")
        print("triangleIndices.count
            \(triangleIndices.count)")

        var descriptor = VertexData.descriptor
        descriptor.vertexCapacity = positions.count
        descriptor.indexCapacity = triangleIndices.count
    }
}
```



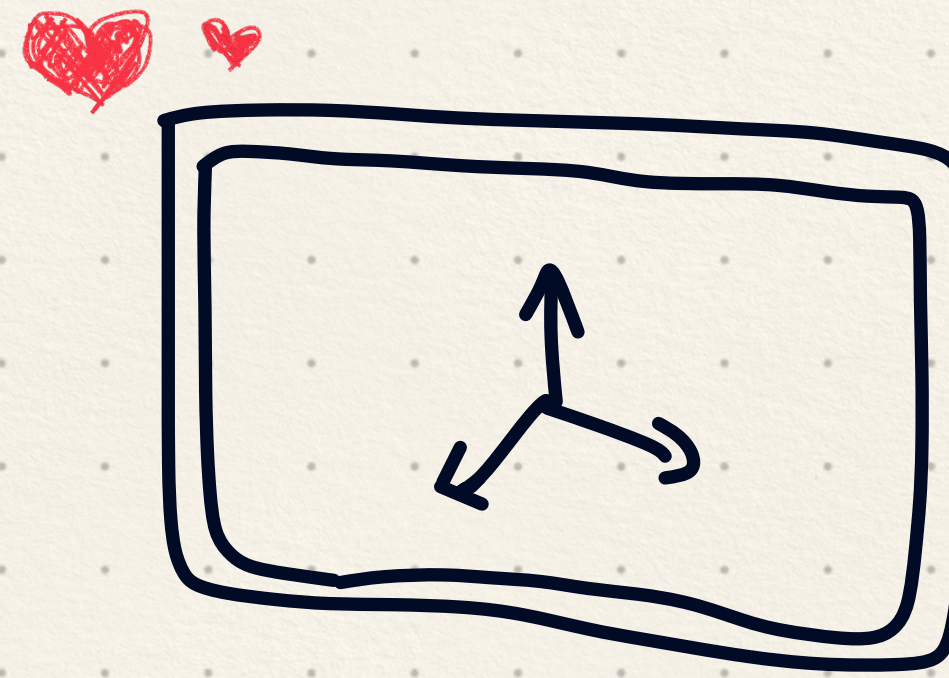
X-Platform

Did I reach my goals?

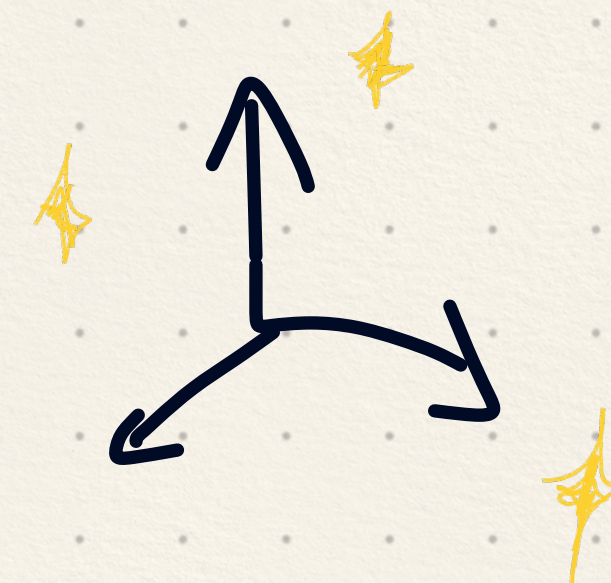
No.

Cross platform

- Platforms are very different
 - Interaction
 - 'Feel' of the UI
 - Anchors



\neq



Concepts

iPad

~~UIKit~~ SwiftUI ✓

Camera Permission ✓

~~ARView~~ RealityView

ARKit Session #

AnchorEntity ✓

RealityKit Models ✓

Apple Vision Pro

SwiftUI ✓

Tracking Permission ✓

RealityView ✓

ARKit Session Provider #

AnchorEntity ✓

RealityKit Models ✓

Concepts

iPad

UIKit

Camera Permission

ARView

ARKit Session

RealityKit Models ✓

Apple Vision Pro

Swift UI

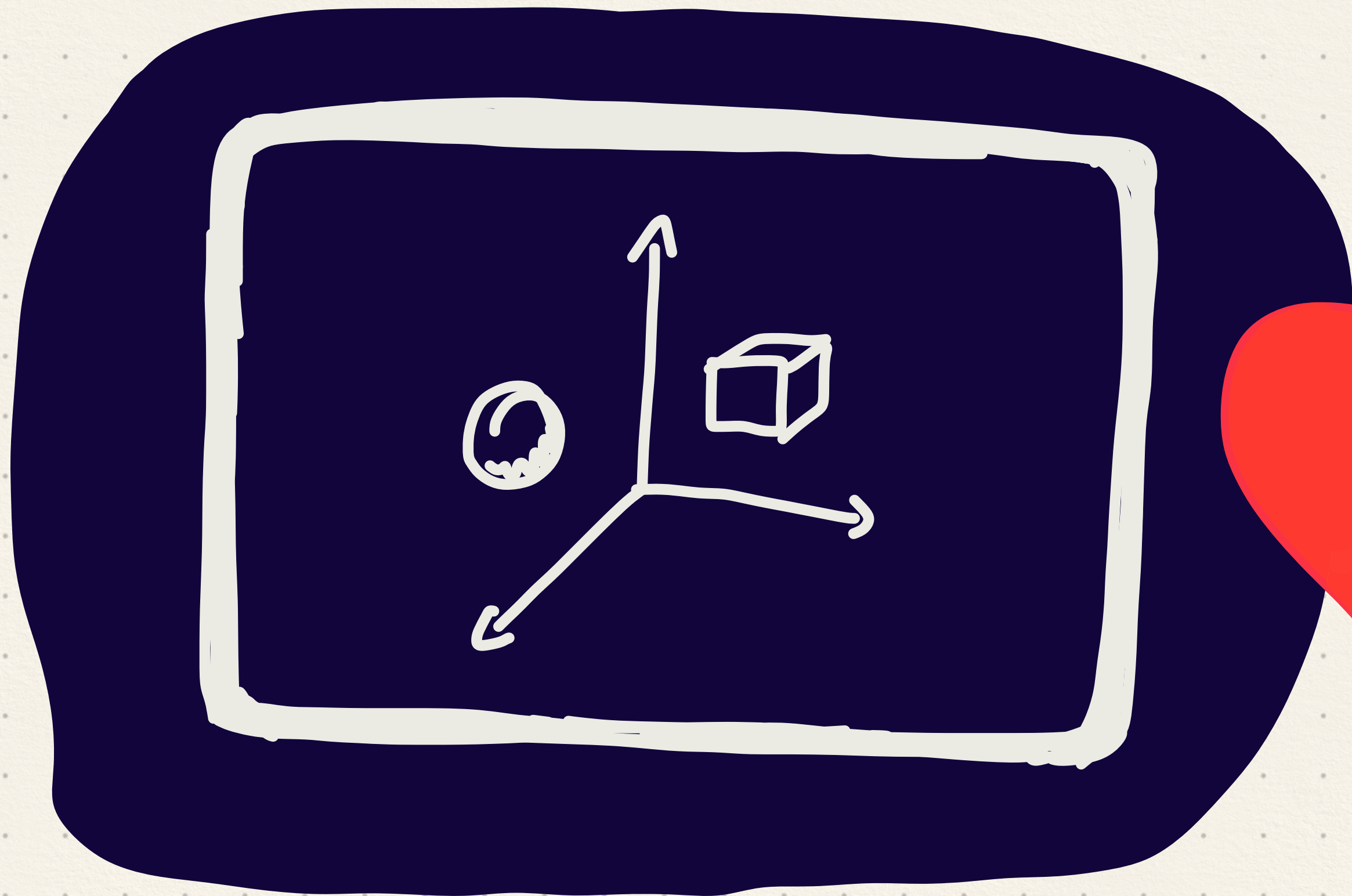
Tracking Permission

RealityView

ARKit Session Provider

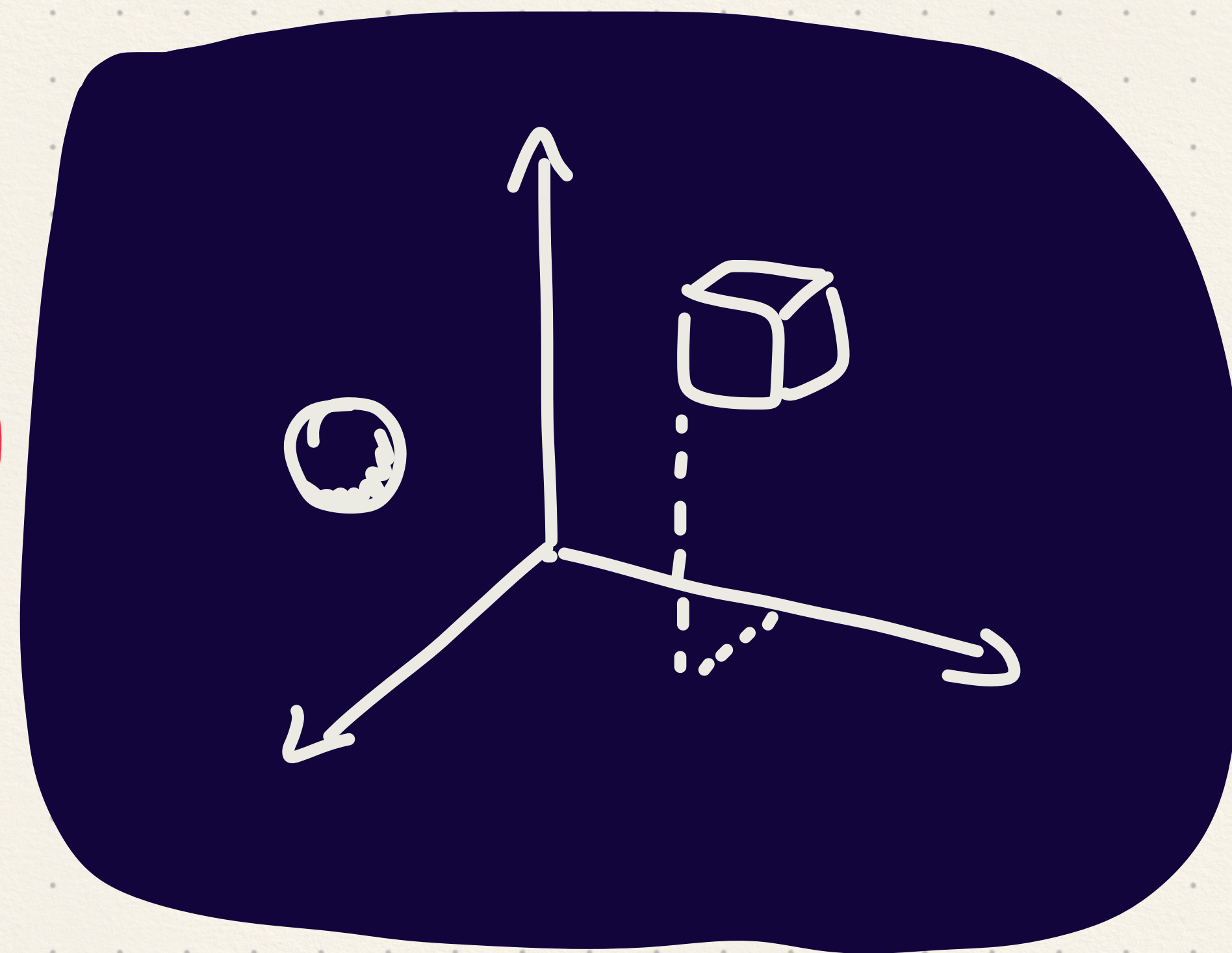
RealityKit Models ✓

iPad



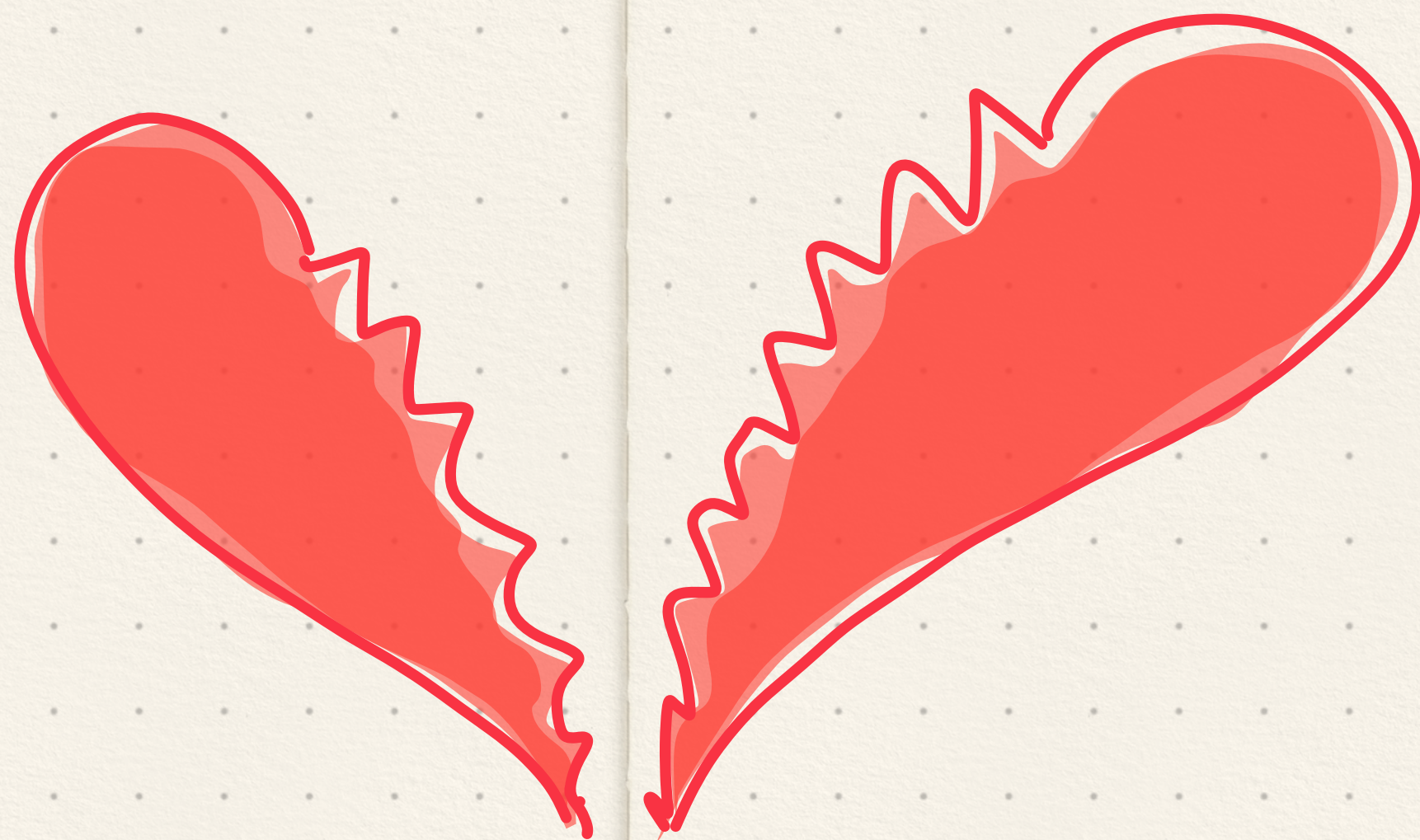
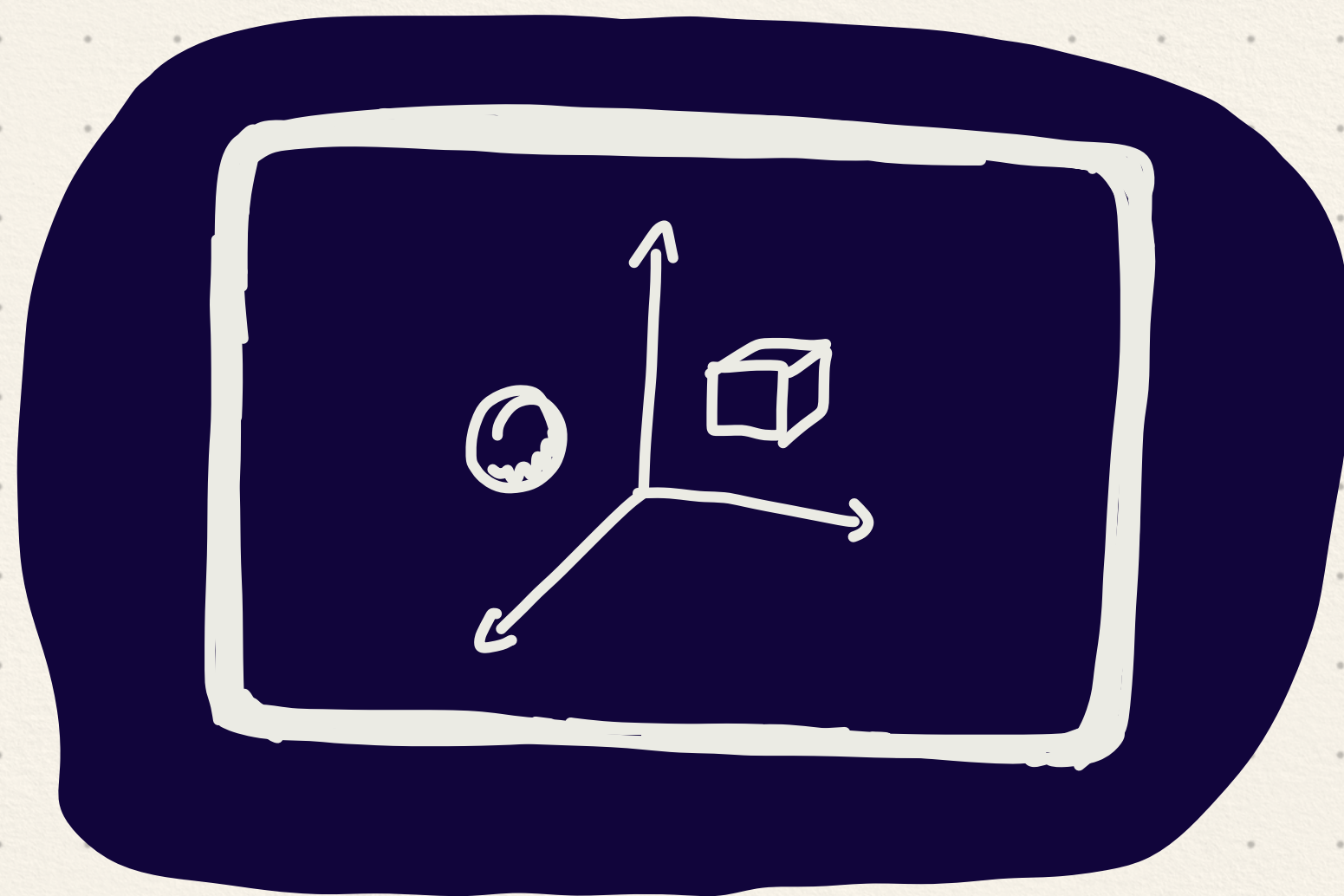
This is AR

Vision Pro

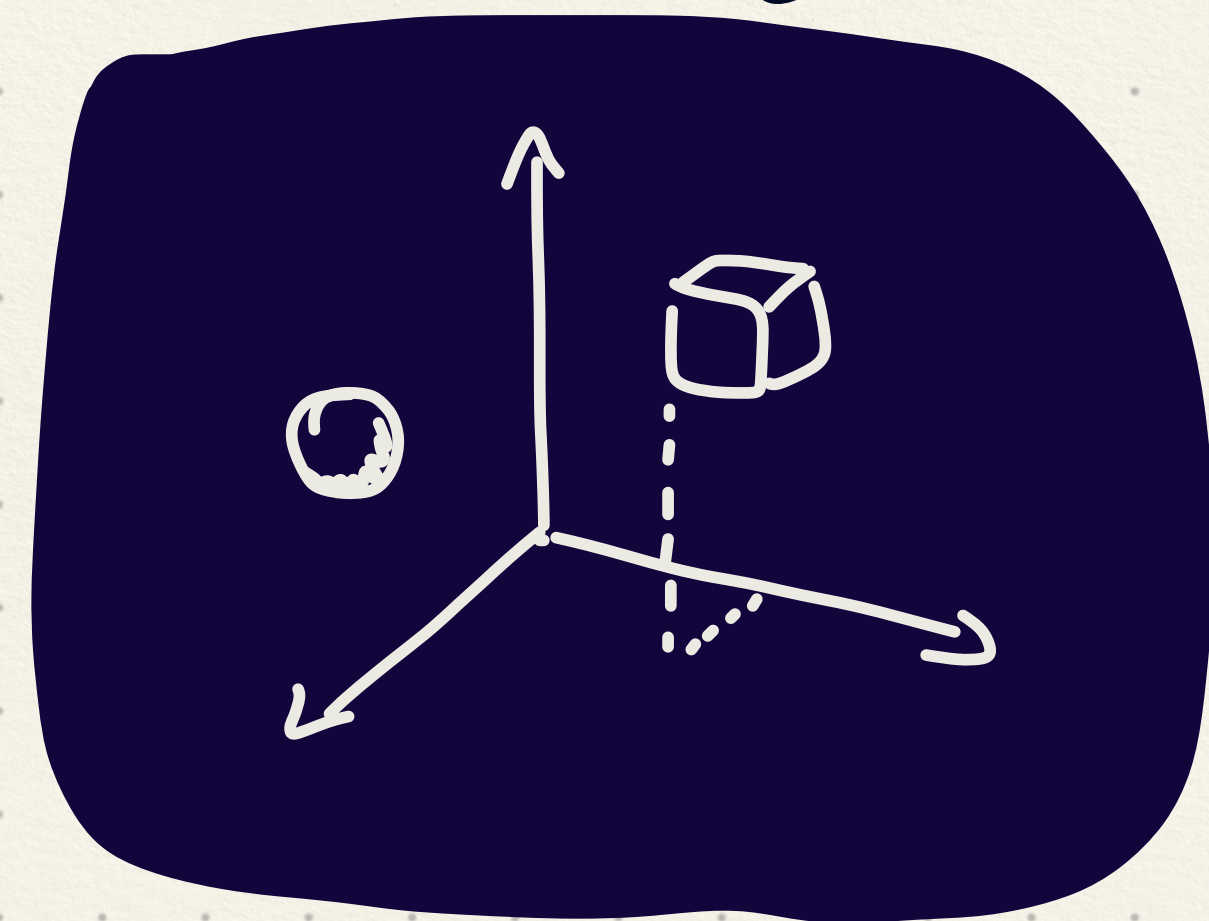


This is also AR

iP2d



Vision Pro



Why AR?

Easy Creativity Starter

- It's easy to start
- Everything **is** cooler in space
- Not about 3D modeling
- It's about creativity



Thank you