POWERED BY UNIKEY™

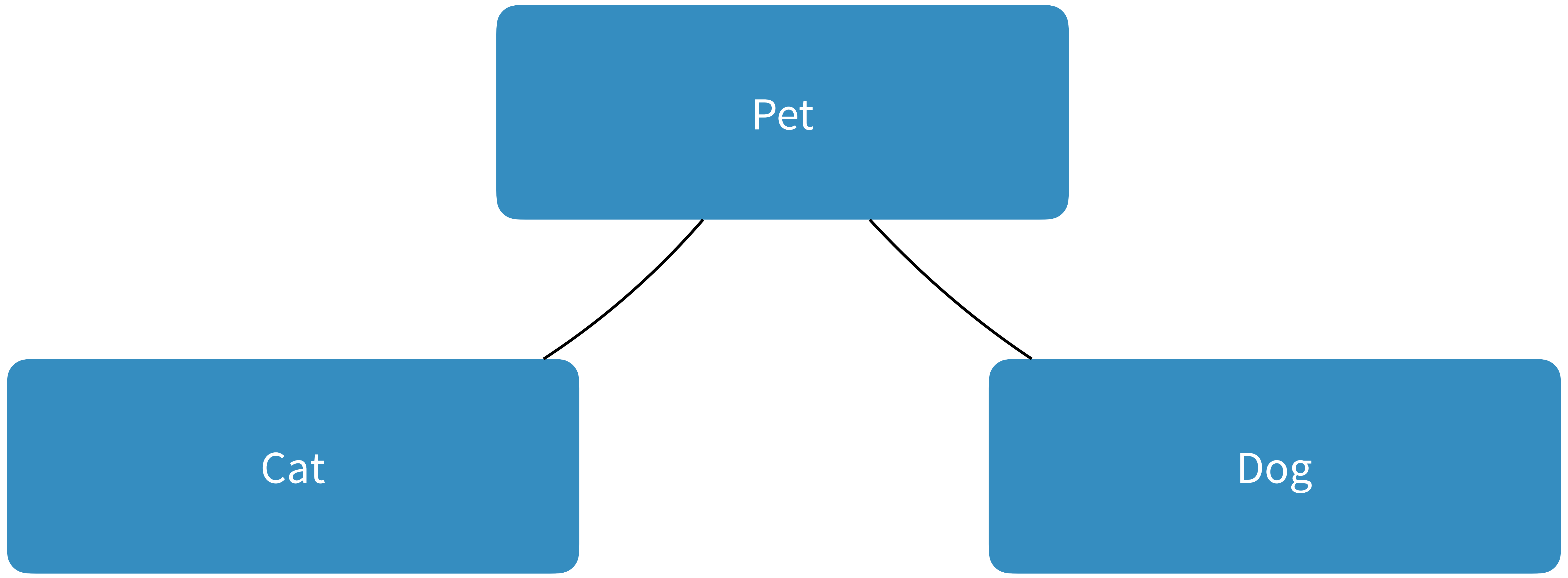
# Protocol Oriented Programming

Tom Marks
Mobile Developer

## Agenda

- Overview
- Demo
  - Creating a new protocol
  - Complying to an existing protocol
- Q & A

# Class Inheritance



## Class Inheritance

```
func find(cats: [Cat], withId anId: NSUUID) -> [Cat] {
    var foundCats = Array<Cat>()

    for aCat in cats {
        if aCat.id == anId {
            foundCats.append(aCat)
        }
    }

    return foundCats
}


func find(dogs: [Dog], withId anId: NSUUID) -> [Dog] {
    var foundDogs = Array<Dog>()

    for aDog in dogs {
        if aDog.id == anId {
            foundDogs.append(aDog)
        }
    }

    return foundDogs
}
```

## Class Inheritance

```
func find(pets: [Pet], withId anId: NSUUID) -> [Pet] {
    var foundPets = Array<Pet>()

    for aPet in pets {
        if aPet.id == anId {
            foundPets.append(aPet)
        }
    }

    return foundPets
}
```

# Unrelated Classes

| Car | Pet |
|---|---|
| vinNumber | id |
| name | firstName |
| colour | lastName |
| serviceDate | isFluffy |
| isElectric | adoptionDate |

## Unrelated Classes

```
func find(cars: [Car], withId anId: NSUUID) -> [Car] {
    var foundCars = Array<Car>()

    for aCar in cars {
        if Car.vinNumber == anId {
            foundCars.append(aCar)
        }
    }

    return foundCars
}


func find(pets: [Pet], withId anId: NSUUID) -> [Pet] {
    var foundPets = Array<Pet>()

    for aPet in pets {
        if aPet.id == anId {
            foundPets.append(aPet)
        }
    }

    return foundPets
}
```

## Unrelated Classes

| Car | Pet |
|---|---|
| vinNumber | id |
| name | firstName |
| colour | lastName |
| serviceDate | isFluffy |
| isElectric | adoptionDate |

## Unrelated Classes

| Car | Pet |
|---|---|
| vinNumber | id |
| name | firstName |
| colour | lastName |
| serviceDate | isFluffy |
| isElectric | adoptionDate |

## A Protocol

```
protocol UniqueId {
    var id: NSUUID
}
```

## Unrelated Classes

| Car: UniqueId | Pet: UniqueId |
|---|---|
| vinNumber | id |
| name | firstName |
| colour | lastName |
| serviceDate | isFluffy |
| isElectric | adoptionDate |

# Unrelated Classes

| Car: UniqueId | Pet: UniqueId |
|---|---|
| id | id |
| name | firstName |
| colour | lastName |
| serviceDate | isFluffy |
| isElectric | adoptionDate |

# Protocol Oriented Programming (P.O.P)

✓

```swift
func find<T: UniqueId>(objects: [T], withId anId: NSUUID) -> [T] {
    var foundObjects = Array<T>()

    for anObject in objects {
        if anObject.id == anId {
            foundObjects.append(anObject)
        }
    }

    return foundObjects
}
```

POWERED BY UNIKEY™

*Demo*

POWERED BY UNIKEY™

# Thank you