iOS
DeCal

lecture 5

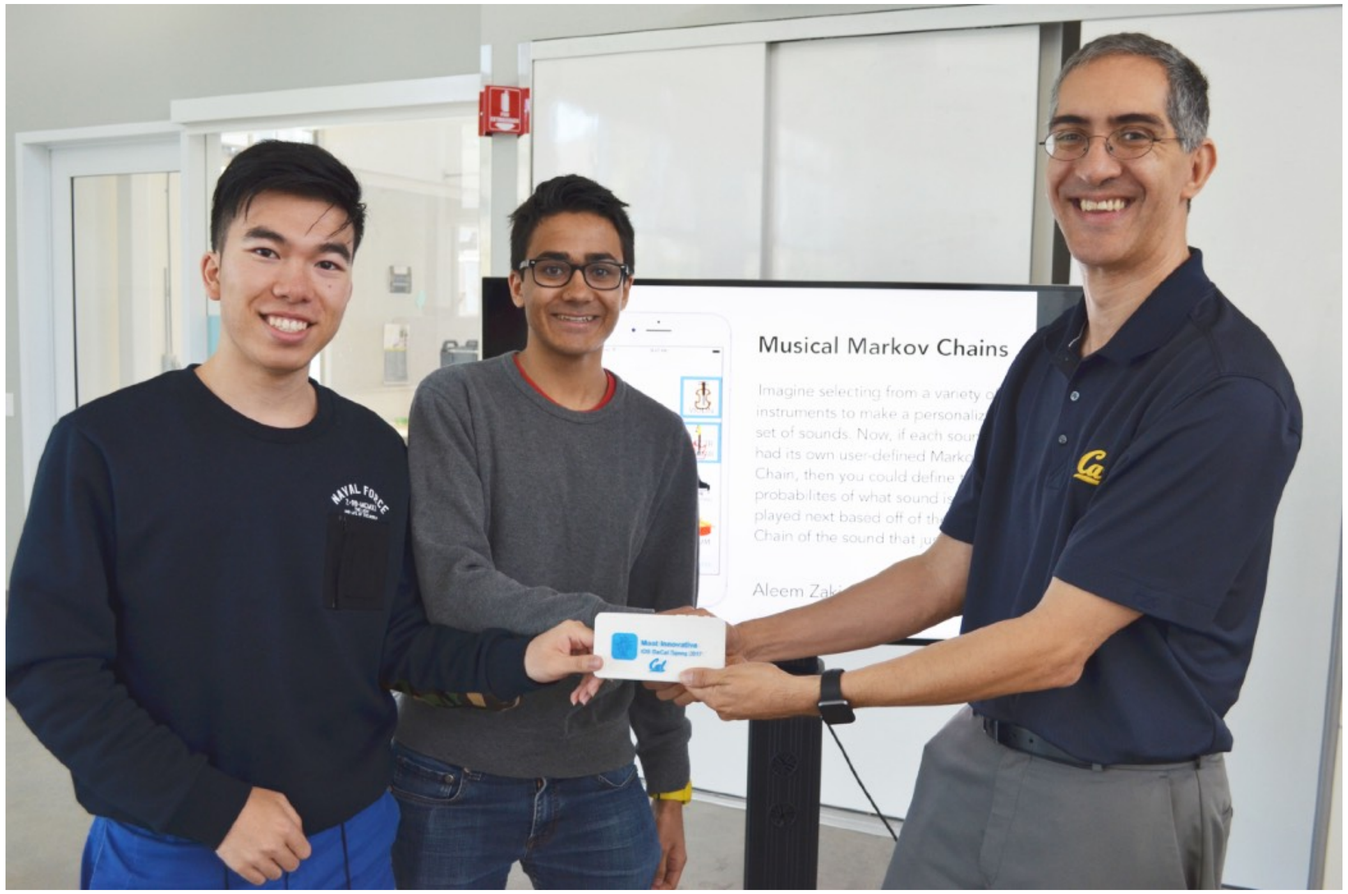
Networking
CocoaPods and AFNetworking

cs198-001 : fall 2017

announcements

- no new lab next week - continuation of Pokedex lab (attendance required) + hw3 help
  - Pokedex due **Wednesday at 11:59 pm**
  - submit on Gradescope ONLY if you do not finish during lab
- hw3 pt 1 (snapchat clone) due next Monday
  - long assignment - start now!

final presentations



- either Monday or Tuesday of dead week
- staff selected top app submissions will be asked to present
- spec released next week, but start thinking of ideas now

today's lecture

- Networking
- CocoaPods
- Alamofire

closures

Can capture and store references to any constants and variables from the context in which they are defined

- **Global closure functions**
  - Named, do not capture values
- **Nested closure functions**
  - Named, capture values from enclosing function

closures - format

```
{ (parameters) -> return type in
    statements
}
```

global closures

```
let intPow = {(val1: Int, val2: Int) -> Int in
    return Int(pow(Double(val1),
                 Double(val2)))
}

let result = intPow(2, 10)
print(result)
```

nested closures

```
func makeIncrementer(forIncrement amount:
Int) -> () -> Int {
    var runningTotal = 0
    func incrementer() -> Int {
        runningTotal += amount
        return runningTotal
    }
    return incrementer
}
var incrementer =
makeIncrementer(forIncrement: 5)
```

nested closures

```
func makeIncrementer(forIncrement amount:
Int) -> () -> Int {
    var runningTotal = 0
    func incrementer() -> Int {
        runningTotal += amount
        return runningTotal
    }
    return incrementer
}
var incrementer =
makeIncrementer(forIncrement: 5)
```

**used frequently in
network calls!**

networking

networking an iOS

Networking is acquiring/passing data to/from some URL that exists on the world wide web or local

General structure as it relates to iOS

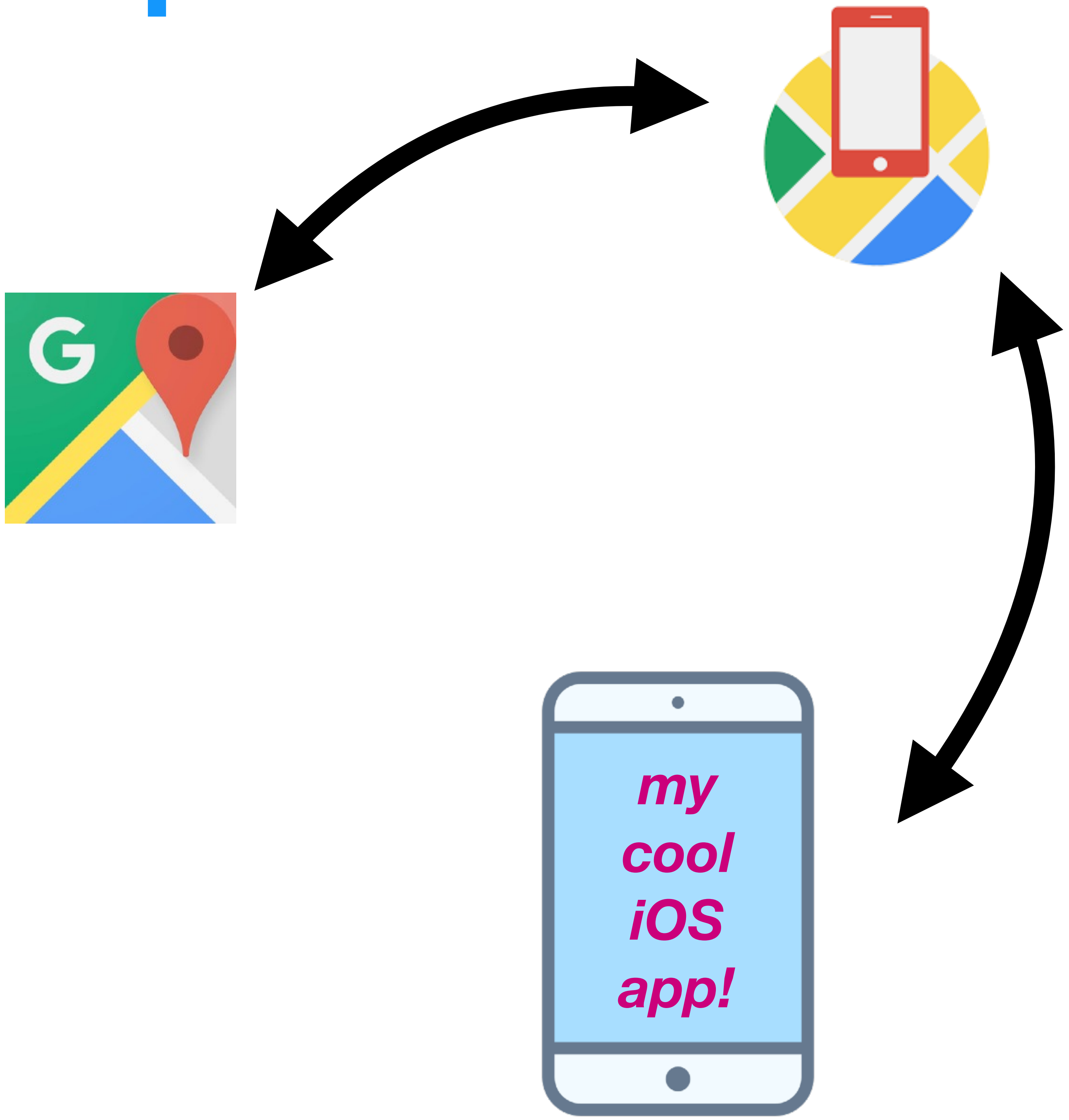
- Recipient Address
- Parameters
- Response

what is an api?

application programming interface

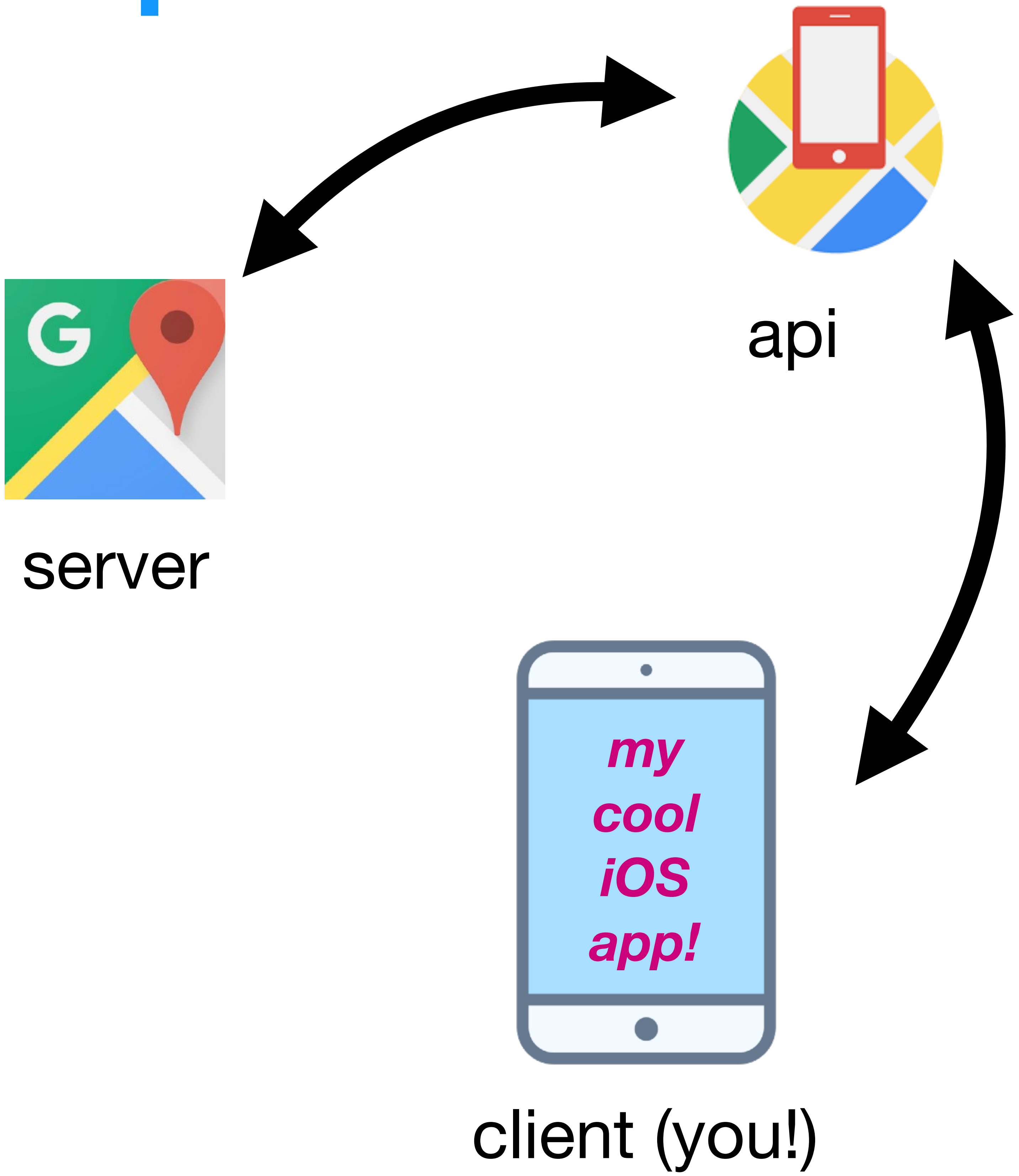
abstraction layer between two software components

used throughout computer science (OS's, hardware, databases)



my cool iOS app!

what is an api?



server

api

my cool iOS app!

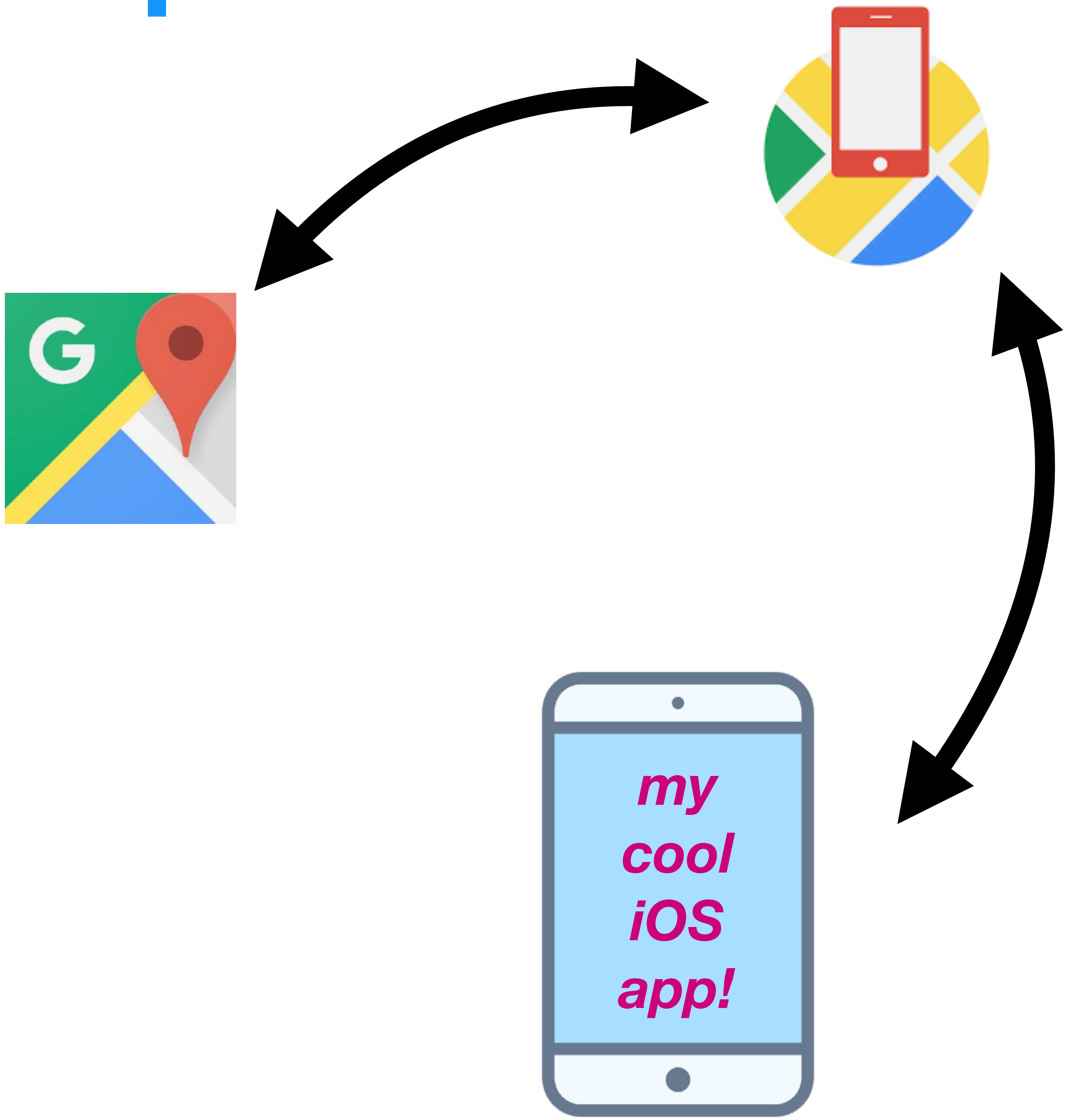
client (you!)

what is an api?

example: you want to create an map related app!

one option: create a map application from scratch

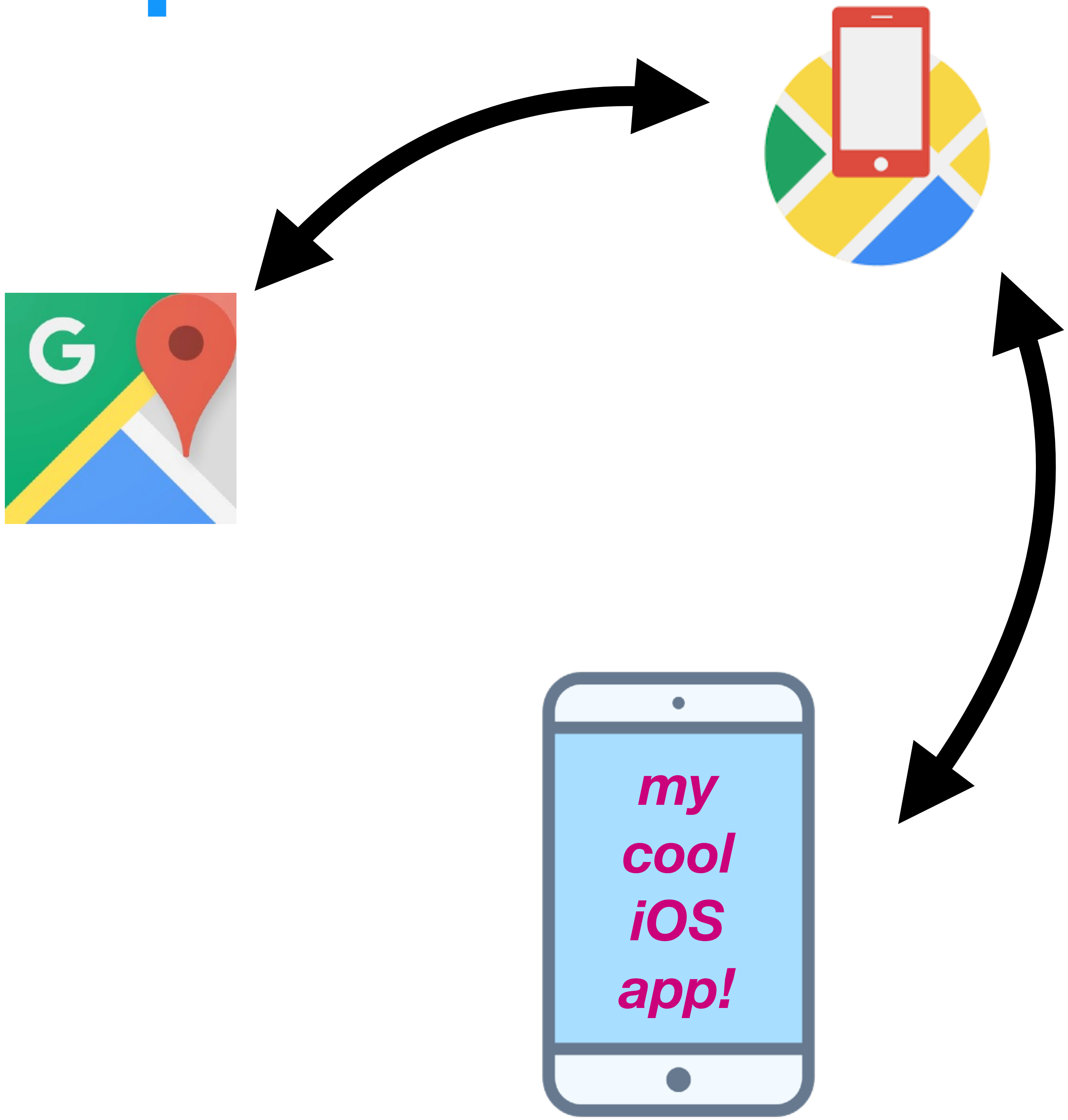
another option: integrate google maps data into your app, using an… api!



what is an api?

another example: you’ve gathered extensive data and have created a database/app using it

share this data with other developers through your own api



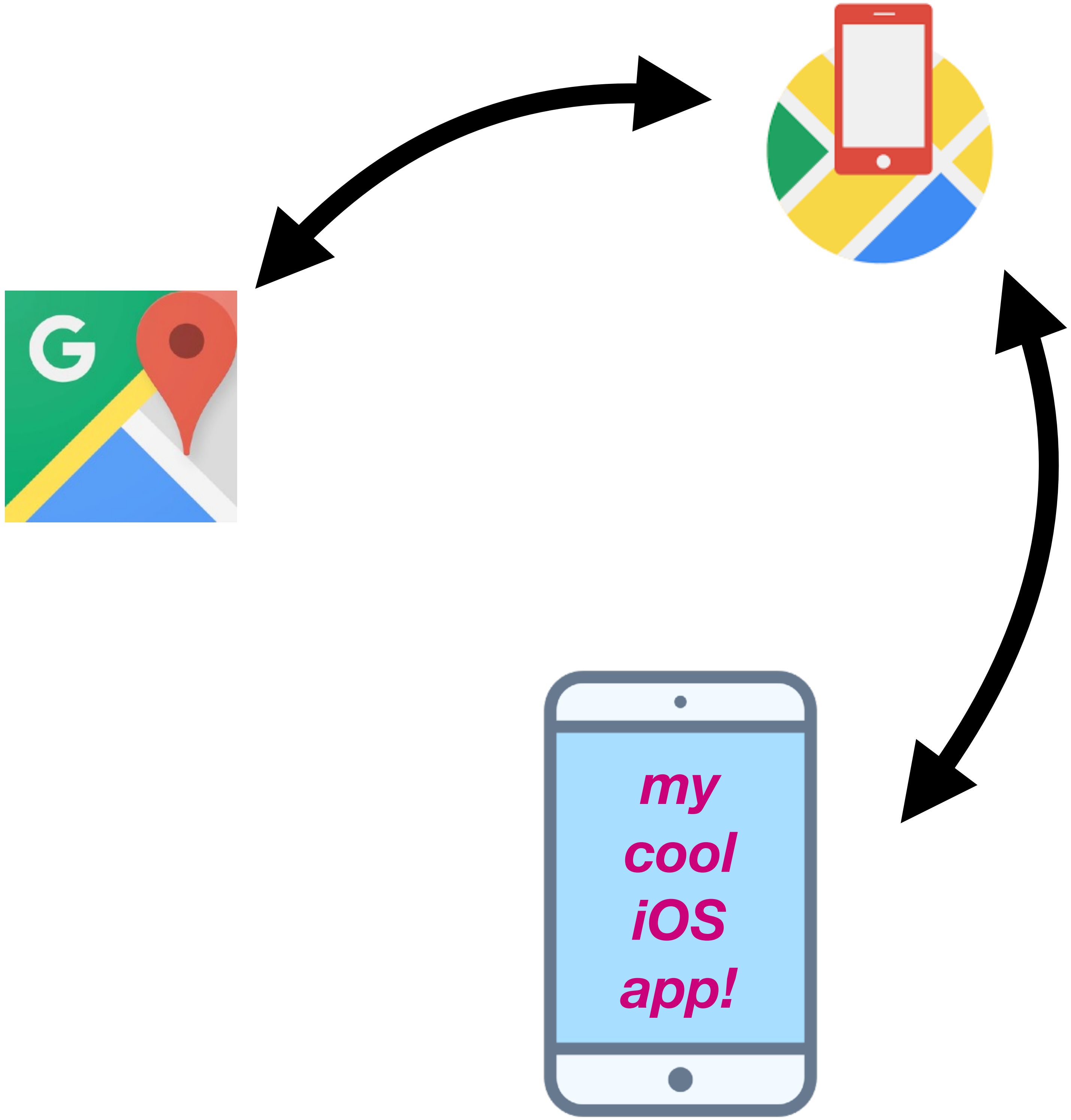
my cool iOS app!

why are api's used?

save time

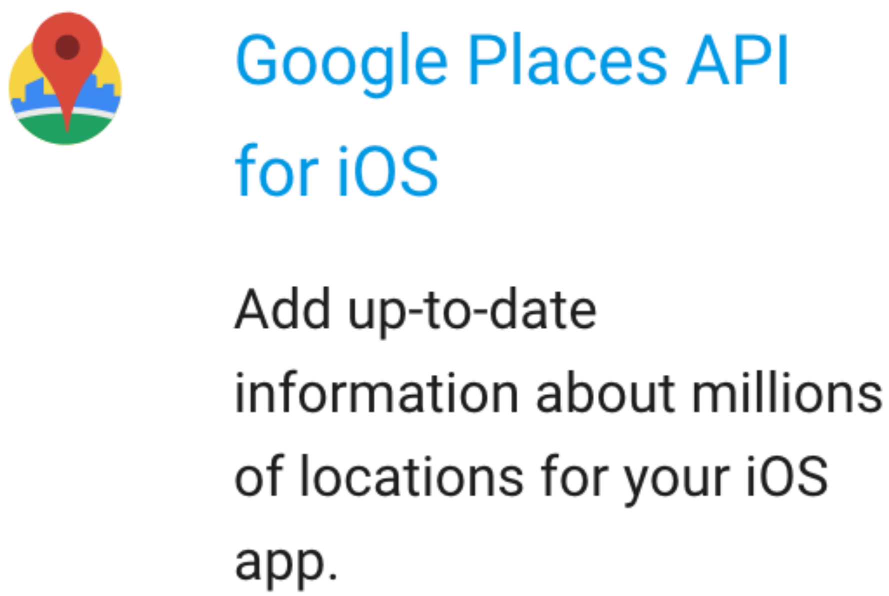
hide implementation details while providing functionality

modularity



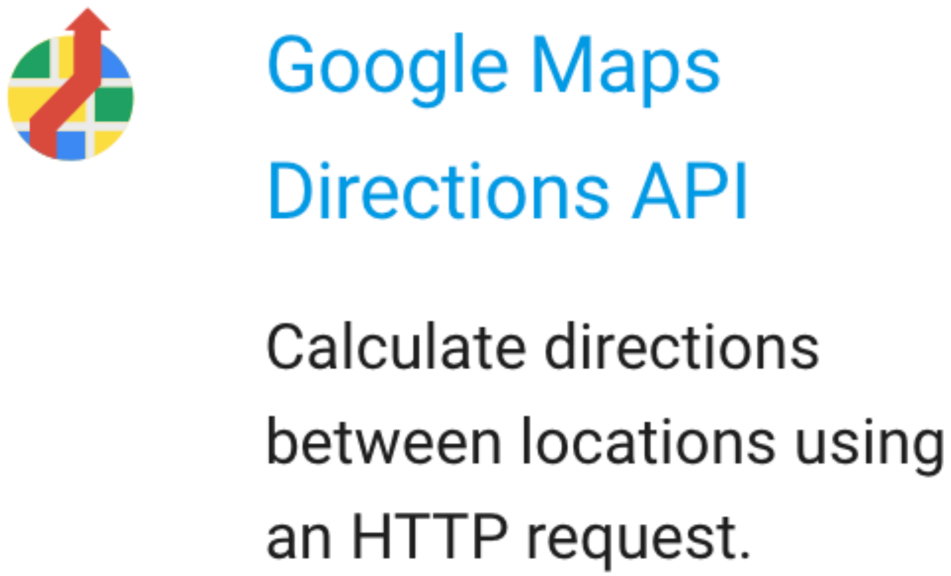
how to use API’s in iOS applications

direct RESTful api calls (via URL)
iOS sdk’s (often still in objective C, but many now written in Swift)



Google Places API for iOS

Add up-to-date information about millions of locations for your iOS app.



Google Maps Directions API

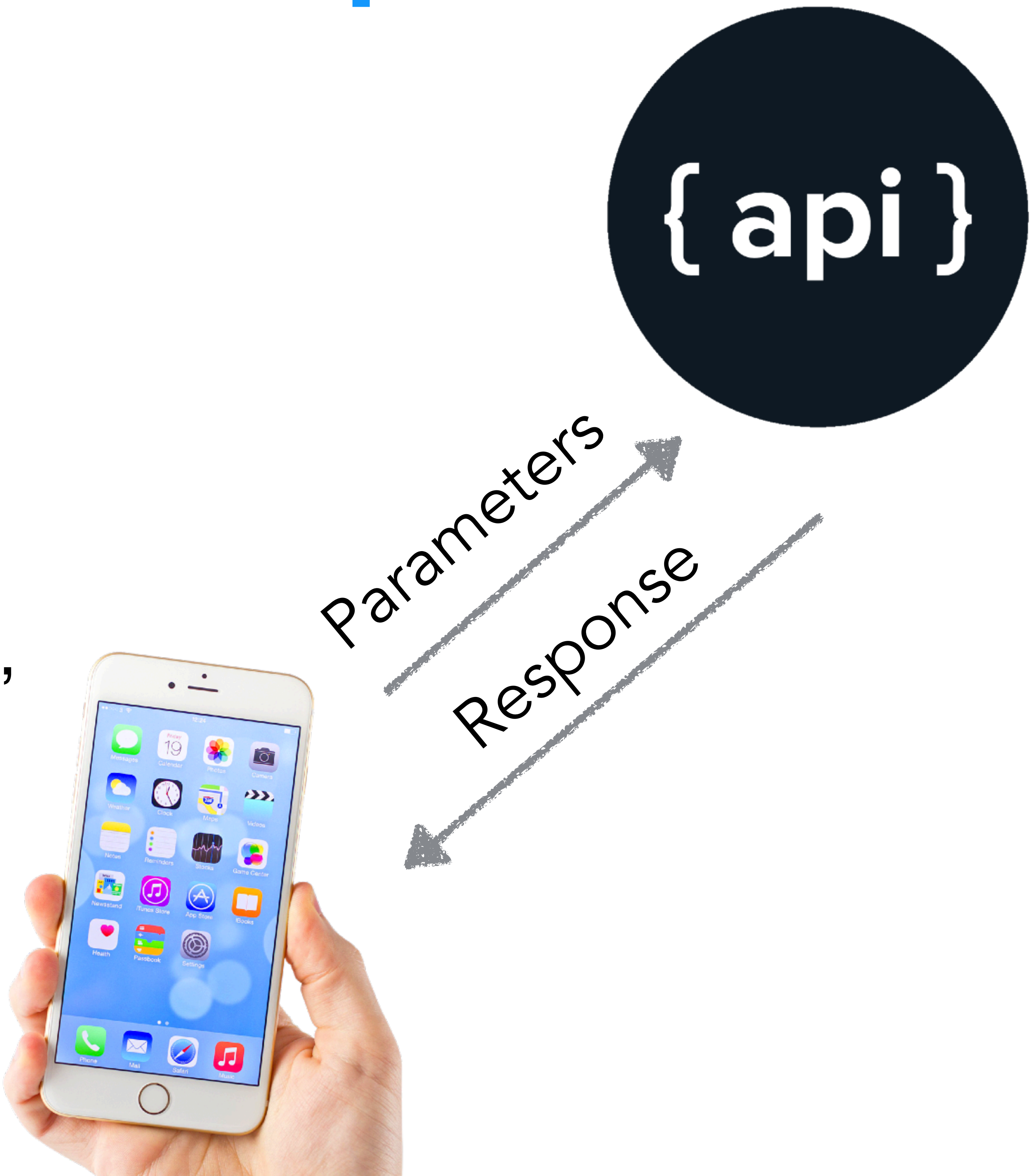
Calculate directions between locations using an HTTP request.

RESTful apis

RESTful apis

representational
state
transfer

requests will return responses (XML, JSON, HTML)



{ api }

Parameters

Response

RESTful apis + HTTP methods

http://**maps.google.com**/maps/api/geocode/json?address=berkeley

server

RESTful apis + HTTP methods

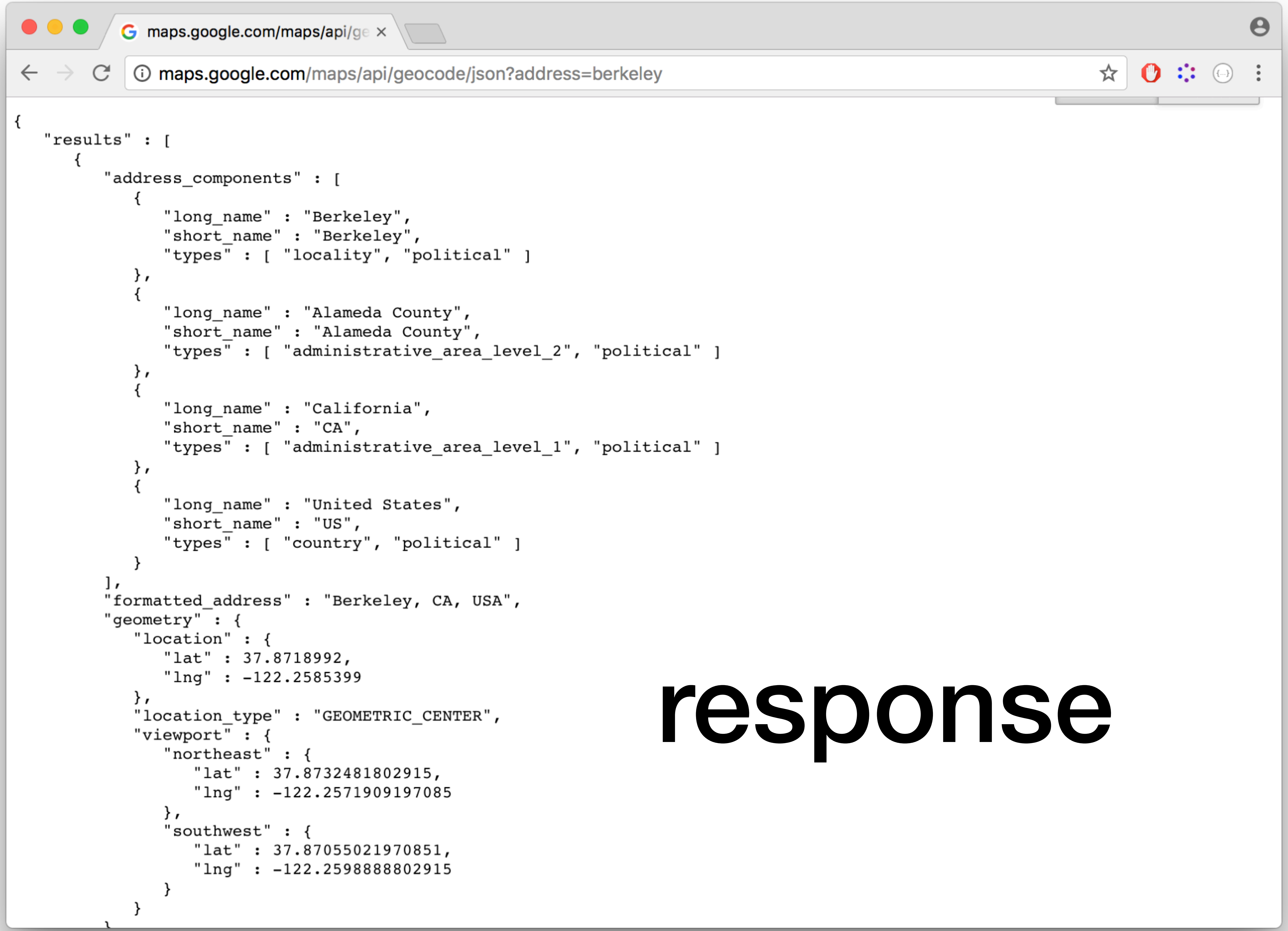
http://maps.google.com/**maps/api/geocode/json**?address=berkeley

resources

RESTful apis + HTTP methods

http://maps.google.com/maps/api/geocode/json?**address=berkeley**

parameters


maps.google.com/maps/api/ge ×
maps.google.com/maps/api/geocode/json?address=berkeley

```
{
   "results" : [
      {
         "address_components" : [
            {
               "long_name" : "Berkeley",
               "short_name" : "Berkeley",
               "types" : [ "locality", "political" ]
            },
            {
               "long_name" : "Alameda County",
               "short_name" : "Alameda County",
               "types" : [ "administrative_area_level_2", "political" ]
            },
            {
               "long_name" : "California",
               "short_name" : "CA",
               "types" : [ "administrative_area_level_1", "political" ]
            },
            {
               "long_name" : "United States",
               "short_name" : "US",
               "types" : [ "country", "political" ]
            }
         ],
         "formatted_address" : "Berkeley, CA, USA",
         "geometry" : {
            "location" : {
               "lat" : 37.8718992,
               "lng" : -122.2585399
            },
            "location_type" : "GEOMETRIC_CENTER",
            "viewport" : {
               "northeast" : {
                  "lat" : 37.8732481802915,
                  "lng" : -122.2571909197085
               },
               "southwest" : {
                  "lat" : 37.87055021970851,
                  "lng" : -122.2598888802915
               }
            }
```

response

JSON

Javascript
Object
Notation

text only format

- easy to send to / from server
- language independent

```
{
  "id": "883836255014203",
  "likes": {
    "data": [
      {
        "name": “Soda Hall",
        "id": "213133108715544",
        "created_time": "2017-08-22:"
      },
      {
        "name": “iOS decal",
        "id": "104681762916482",
        "created_time": "2017-08-21"
      },
      {
        "name": “Oski",
        "id": "548276038608761",
        "created_time": “2017-07-17"
  ...
```

but how do we do this in iOS?

making requests - URLRequest, URLSession (built in API’s), or Alamofire (3rd party)

parsing - JSONSerialization or SwiftyJSON (3rd party)

URL Session

URLSession

Apple’s API for downloading content

Support various URL schemes

- HTTP, HTTPS, FTP, Data, File

Pass in a URL

- URL object, allocated from String

Some Relevant Classes

URL

Object that contains URL

URLRequest

Contains URL, request method, etc.

URLResponse

Contains info for server's response

URLSession Workflow

1) Create URL from a String

2) Create URLSession

3) Create a URLSessionDataTask

Get data from the task and save it

URLSession

- **URLSessionDataTask**
    - dataTaskWithURL - Default HTTP GET
    - dataTaskWithRequest - Can specify HTTP

URLSession

```
func loadImage() {
  let url = URL(string:”https://instagram.com/img.jpg")

  let session = URLSession.shared

  let task = session.dataTask(with: url!,
                                completionHandler: {
       (data, response, error) -> Void in
       if error == nil {
            let img = UIImage.init(data: data!)
            self.imageView.image = img
       }
     })
        task.resume()
}
```

JSON parsing in iOS

JSONSerialization

jsonObject(with:options:)

JSON parsing in iOS

JSONSerialization

jsonObject(with:options:)

```
let data: Data // received from a network request
let json = try? JSONSerialization.jsonObject(with: data,
options: [])
```

accessing values from json

```
// Example JSON with object root:
/*
  {
    "someKey": 42.0,
    "anotherKey": {
      "someNestedKey": true
    }
  }
*/
if let dictionary = json as? [String: Any] {
  if let number = dictionary["someKey"] as? Double {
    // access individual value in dictionary
  }
```

[example link]

parsing via initializers

```
extension Dog {
  init?(json: [String: Any]) {
    guard let name = json["name"] as? String,
      let friends = json["friends"] as? [String],
      let bestFriendName = friends[0],
      let secondBestFriendName = friends[1],
      let meals = json[“meals"] as? [String: Any],
    else {
      return nil
}
```

creating Model objects from json example

CocoaPods

What are CocoaPods

A dependency manager for Cocoa Projects

CocoaPods are essentially Swift classes that other people write for you that you can use in your project:

What are CocoaPods

A dependency manager for Cocoa Projects

CocoaPods are essentially Swift classes that other people write for you that you can use in your project:

- Make life more **efficient**

What are CocoaPods

A dependency manager for Cocoa Projects

CocoaPods are essentially Swift classes that other people write for you that you can use in your project:

- Make life more **efficient**
- Make life **easier**

What are CocoaPods

A dependency manager for Cocoa Projects

CocoaPods are essentially Swift classes that other people write for you that you can use in your project:

- Make life more **efficient**
- Make life **easier**
- Make life **funner**

Timepiece

Adding A Year To the Current Date:

Without Timepiece:

```
let calendar = NSCalendar.currentCalendar()
let newDate = calendar.dateByAddingUnit(.Year, value:
1, toDate: NSDate(), options:
NSCalendarOptions.MatchNextTime)
```

Timepiece

Adding A Year To the Current Date:

Without Timepiece:

```
let calendar = NSCalendar.currentCalendar()
let newDate = calendar.dateByAddingUnit(.Year, value:
1, toDate: NSDate(), options:
NSCalendarOptions.MatchNextTime)
```

With Timepiece

```
let newDate = now + 1.year
```

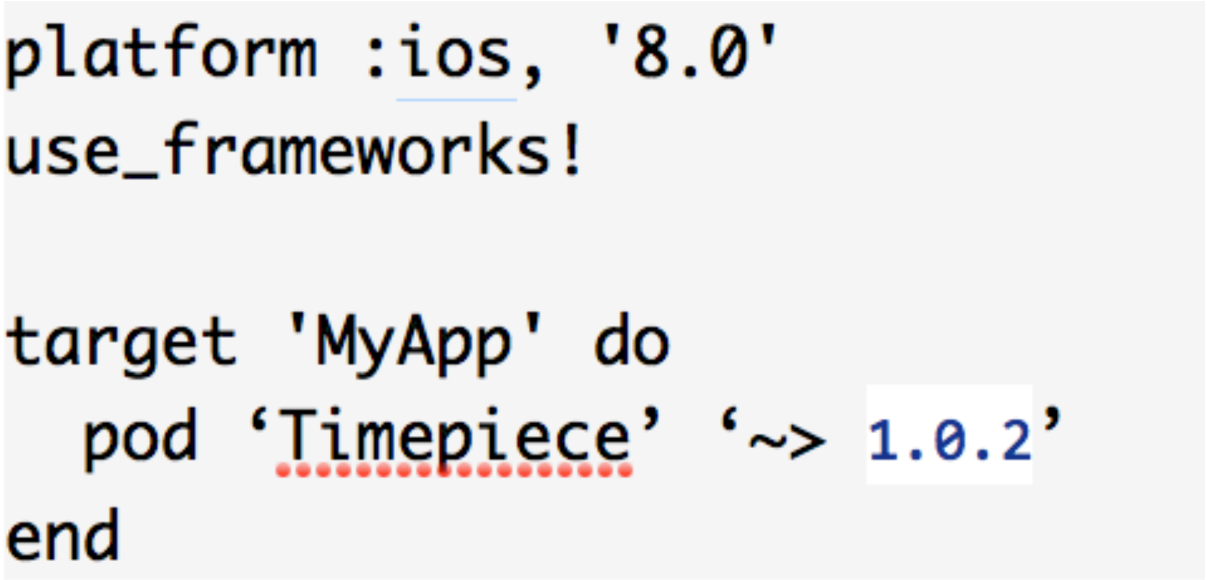
How to use CocoaPods

Install CocoaPods

```
sudo gem install cocoapods
```

Make a Podfile

1. Just named Podfile with no extension
2. Format (Trick: Use “pod init”):



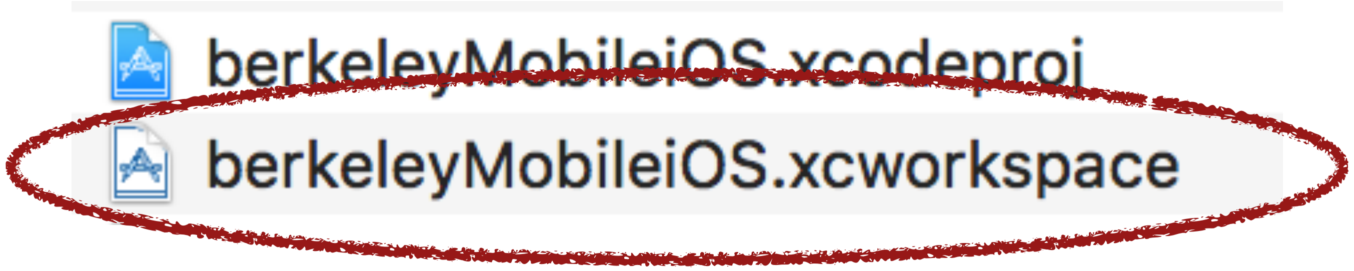
```
platform :ios, '8.0'
use_frameworks!

target 'MyApp' do
  pod ‘Timepiece’ ‘~> 1.0.2’
end
```

Update Dependencies

```
pod install
```

Open .xcworkspace



berkeleyMobileiOS.xcodeproj

berkeleyMobileiOS.xcworkspace

Check Ins

Alamofire

Alamofire : Networking in Swift

HTTP networking library written in Swift

Simplifies common networking tasks

- Request/Response methods
- JSON serialization
- Authentication

GitHub link

Alamofire : Requests

.request: HTTP requests
.upload: Upload large files
.download: Download large files or resume a download already in progress.

Alamofire : Request types

```
Alamofire.request("https://httpbin.org/
get") // default is GET

Alamofire.request("https://httpbin.org/
post", method: .post)
```

Alamofire : Response Handlers

```
// Response Data Handler - Serialized
into Data
func responseData(queue: DispatchQueue?,
completionHandler: @escaping
(DataResponse<Data>) -> Void) -> Self


// Response JSON Handler - Serialized
into Any
func responseJSON(queue: DispatchQueue?,
completionHandler: @escaping
(DataResponse<Any>) -> Void) -> Self
```

Alamofire : Response Validation

```
Alamofire.request("https://httpbin.org/get”)
        .validate().responseJSON { response in
           switch response.result {
              case .success:
                 print("Validation Successful")
              case .failure(let error):
                 print(error)
            }
       }
```

Alamofire : Parameters

```
Alamofire.request("https://httpbin.org/post",
        method: .post,
        parameters: parameters)

Alamofire.request("https://httpbin.org/post",
        method: .post,
        parameters: parameters,
        encoding: URLEncoding.default)
```

Alamofire : Authentication

```
let user = "user"
let password = "password"

Alamofire.request(“https://
       httpbin.org/basic-auth/\(user)/
       \(password)")
     .authenticate(user: user,
                  password: password)
     .responseJSON { response in
     debugPrint(response)
}
```