

# Find Bugs Faster through ViewModel and API Library Testing

Andy Lech

# Code Mash 2026 Sponsors

Unobtanium Exhibitor

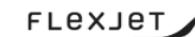


InfernoRed Technology

Adamantium Exhibitor



AWS



Flexjet



TuxCare

Platinum Exhibitor



Umbraco



CodeLogic



Jumpmind



Red Hat AI



Text Control, LLC

Gold Exhibitor



Calliberty



CGI

KidzMash Key Sponsor



NimblePros

Non-Exhibiting Sponsors



Boardgame Room



Waterpark Party



Waterpark Party



Waterpark Party



# 18th Annual Orlando Code Camp | 2026

**April 11th - Seminole State College**

Seminole State College, Sanford, FL

<https://www.orlandocodecamp.com/>

# Audience Polls

- How many have written a mobile app before?
- How many have published an app to an app store?
- How many have used the MVVM pattern in their app?
- How many have separate API libraries in their app?

# Topics

- What makes cross-platform and mobile different
- Real-world mobile design components
- Separating API consumption into testable libraries
- Testing how data flows through your ViewModels

# Part 1

Imagine your house was an app ...

# Energy - User

*[Web Site or Mobile App]*

- Energy Consumption
  - Just Works
- Energy Production
  - Somebody Else's Problem
- Resiliency
  - Generator (Maybe?)
- Focus
  - Living My Best Life

[Image by macrovector on Freepik](#)

Andy Lech - Find Bugs Faster through  
ViewModel and API Library Testing



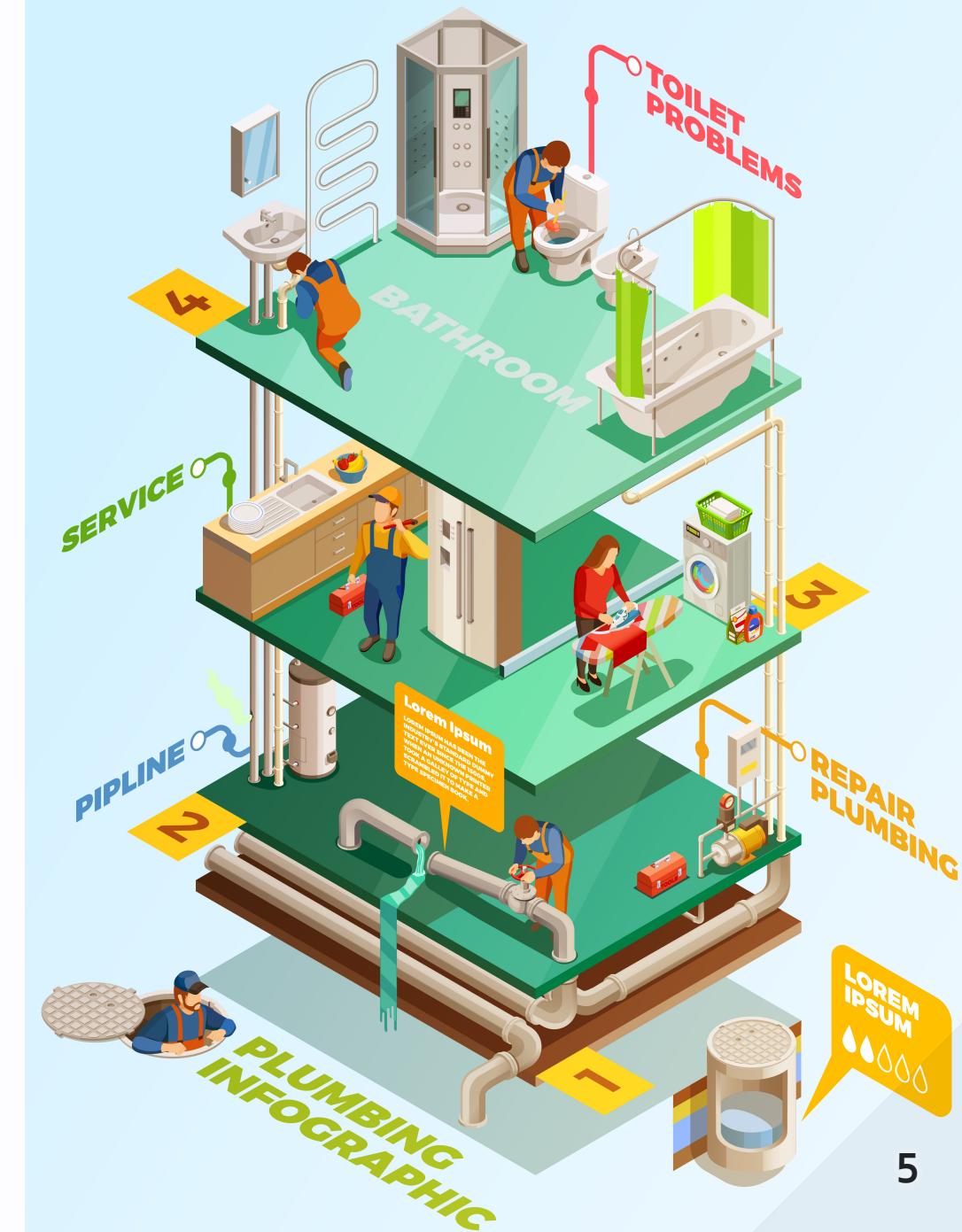
# Energy - Power Grid

[Web Site]

- User Consumption
  - Always
- Grid Production
  - Our Problem
- Resiliency
  - Our Backups Have Backups (Hopefully)
- Focus
  - Keeping the Lights On (Literally)

[Image by macrovector on Freepik](#)  
*[Equivalent energy image not found]*

Andy Lech - Find Bugs Faster through ViewModel and API  
Library Testing



# Energy - Solar/Wind

[Mobile App]

- Energy Production
  - Local
- Grid Consumption
  - Backup
- Grid Production
  - Their Problem
- Resiliency
  - Battery, Power Grid
- Focus
  - Smartly Using and Storing Energy

[Image by macrovector on Freepik](#)

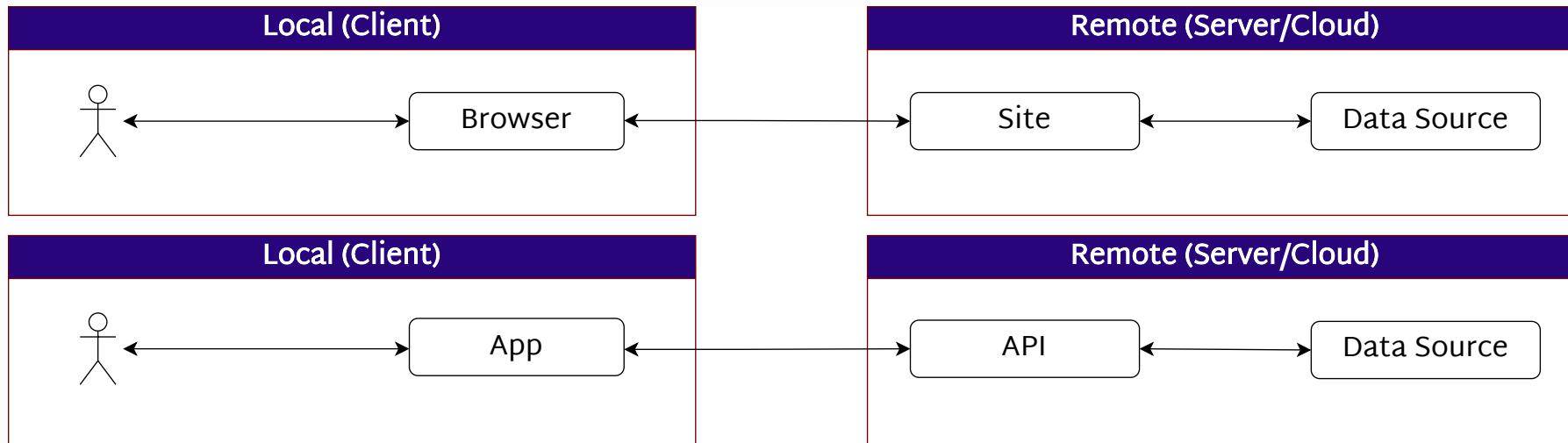
Andy Lech - Find Bugs Faster through ViewModel  
and API Library Testing



## Part 2

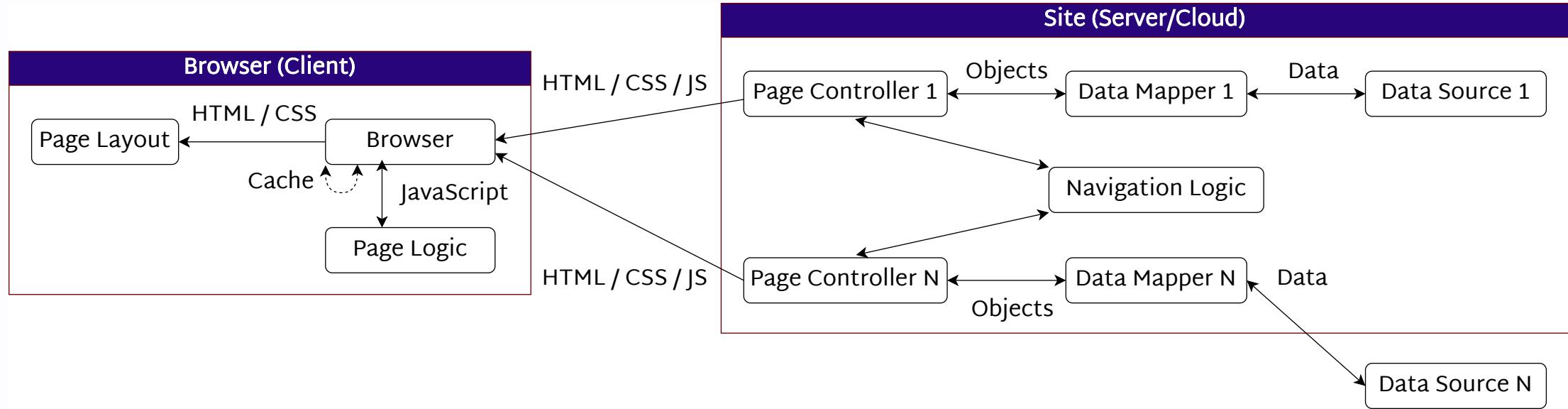
Mobile is just like Web but smaller,  
right!?

# Web vs Mobile - High Level



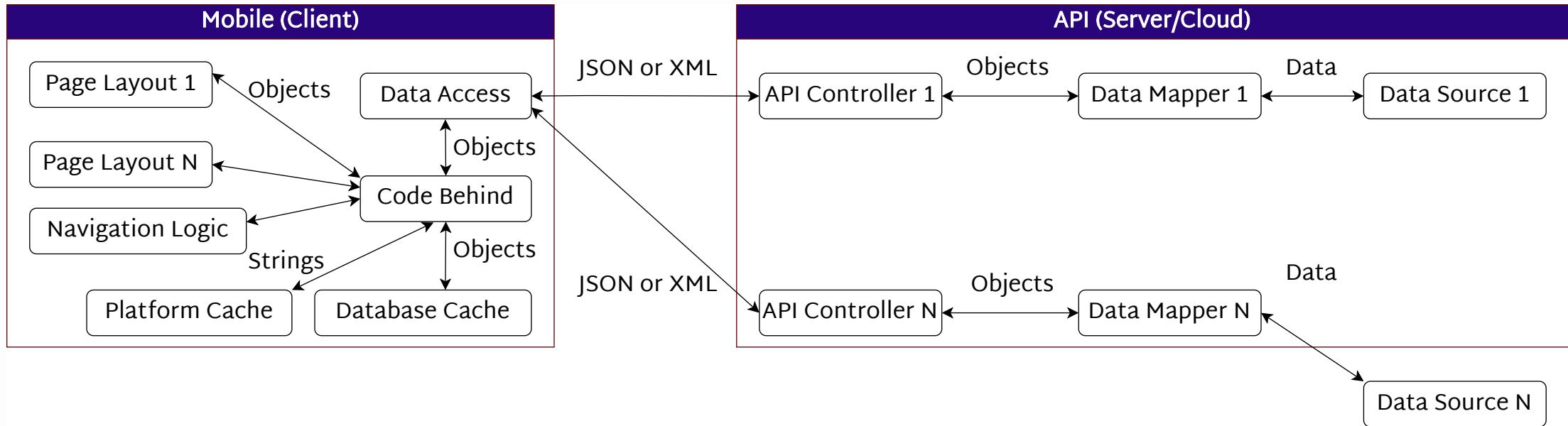
**Same thing, right? Problem solved! Crisis averted!**  
**Thank you and good night!**

# Web Architecture - Components



- Server/cloud stack is the focus here, turning data into pages and layouts
- Complex layouts are generally built on top of templating frameworks
- **Web devs expect fat data pipes in the server/cloud stack to get large data payloads (object graphs) to choose what to filter/display in page layouts**

# Mobile Architecture - Components



- App handles interactivity, page layouts, local caching, and API calls
- API stack delivers only data or status codes in response to app requests
- App pages tend to be focused on single tasks that call the API selectively
- **Mobile/API devs focus more on reliable, just-in-time data delivery**

## Part 3

You got Presentation Logic on my Business Logic

You got Business Logic in my Presentation Logic

# MVVM Pattern - Theory

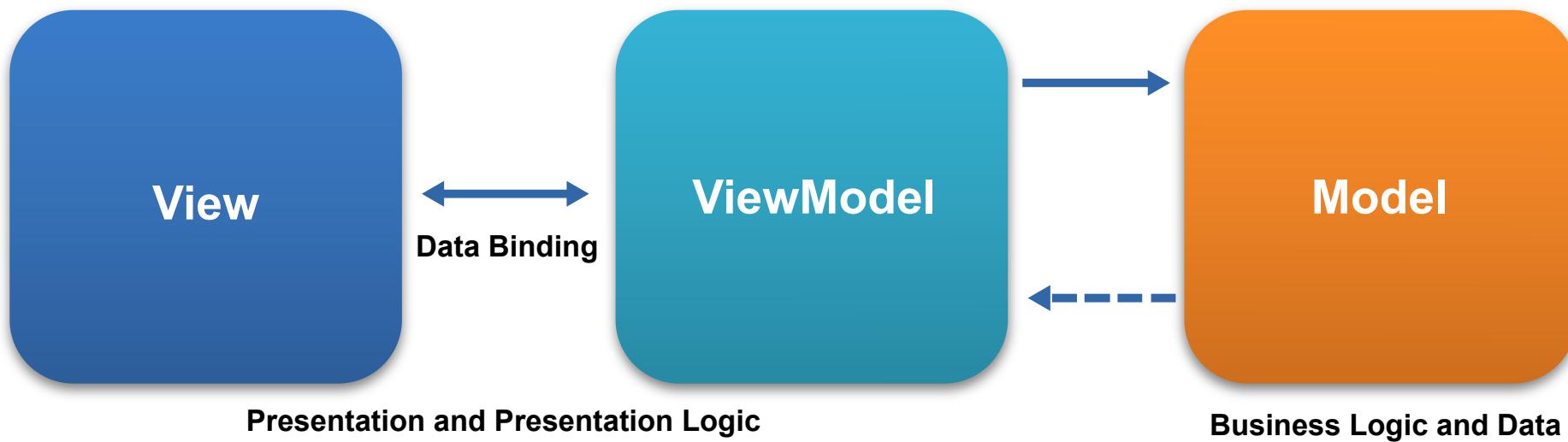
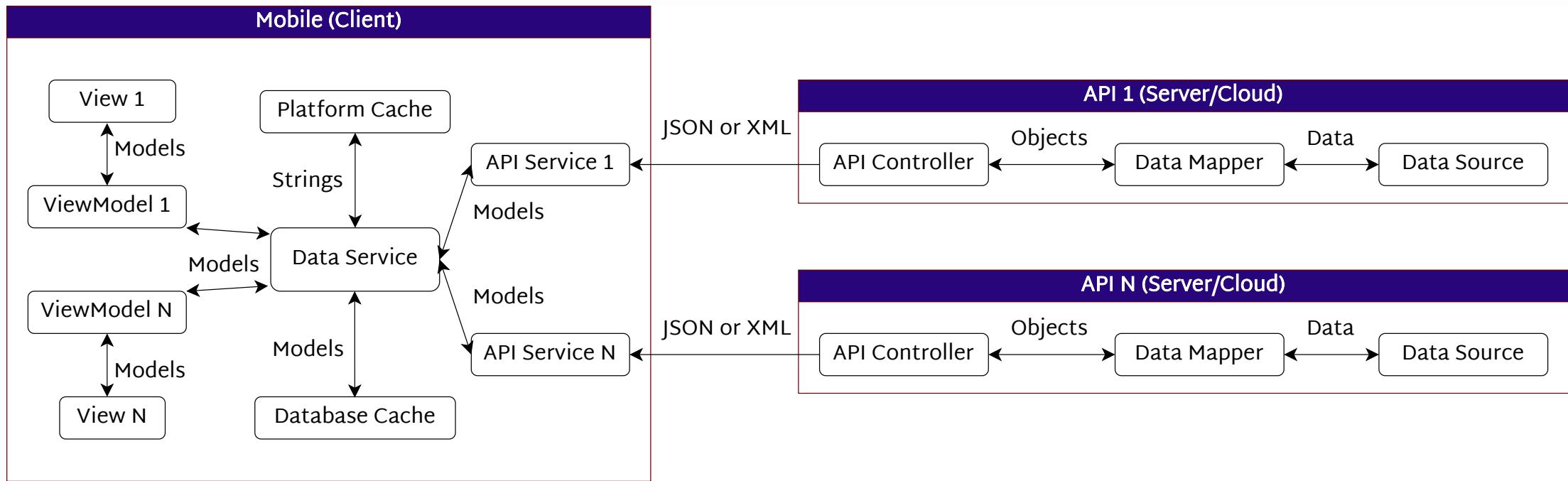


Image from Wikipedia

# Mobile Architecture - Real-world

Component	Architecture	Purpose
Page Layouts	Views	Presentation
Navigation Logic	ViewModels	Navigation Router
Code Behind	ViewModels	Presentation Logic
Code Behind	ViewModels	Data Manipulation Logic
Local Caches	Data Service	Data Source Logic
Data Access	API Services	Data Translation Logic
Data Access	Models	Internal Data Definitions
Data Access	Data Transfer Objects	External Data Definitions

# Mobile Architecture - Network



# Part 4

Why?



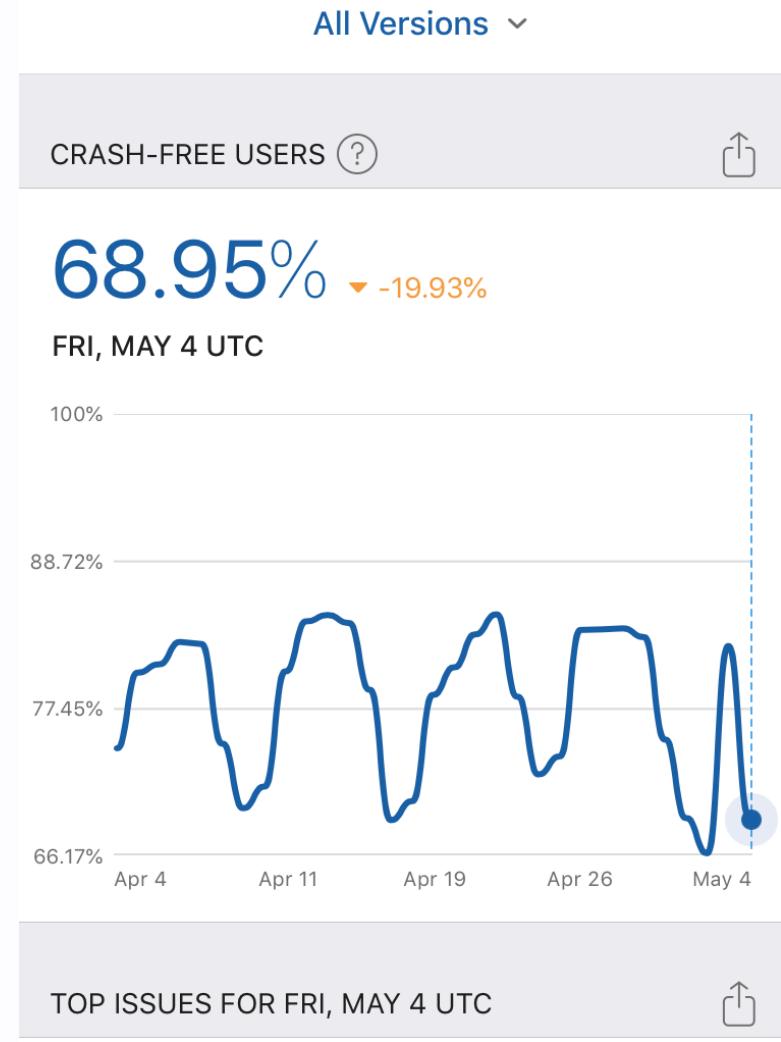
11:14 PM

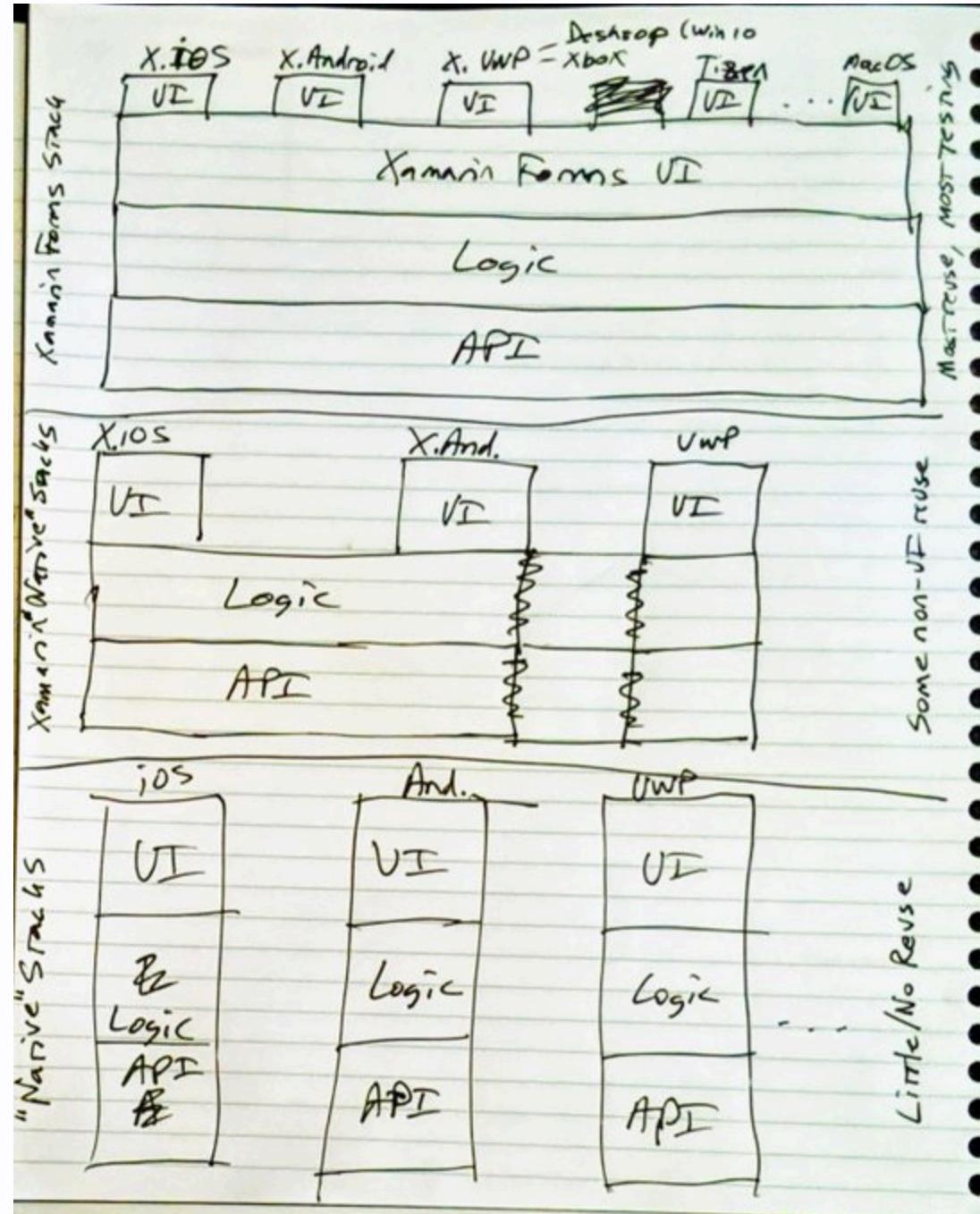


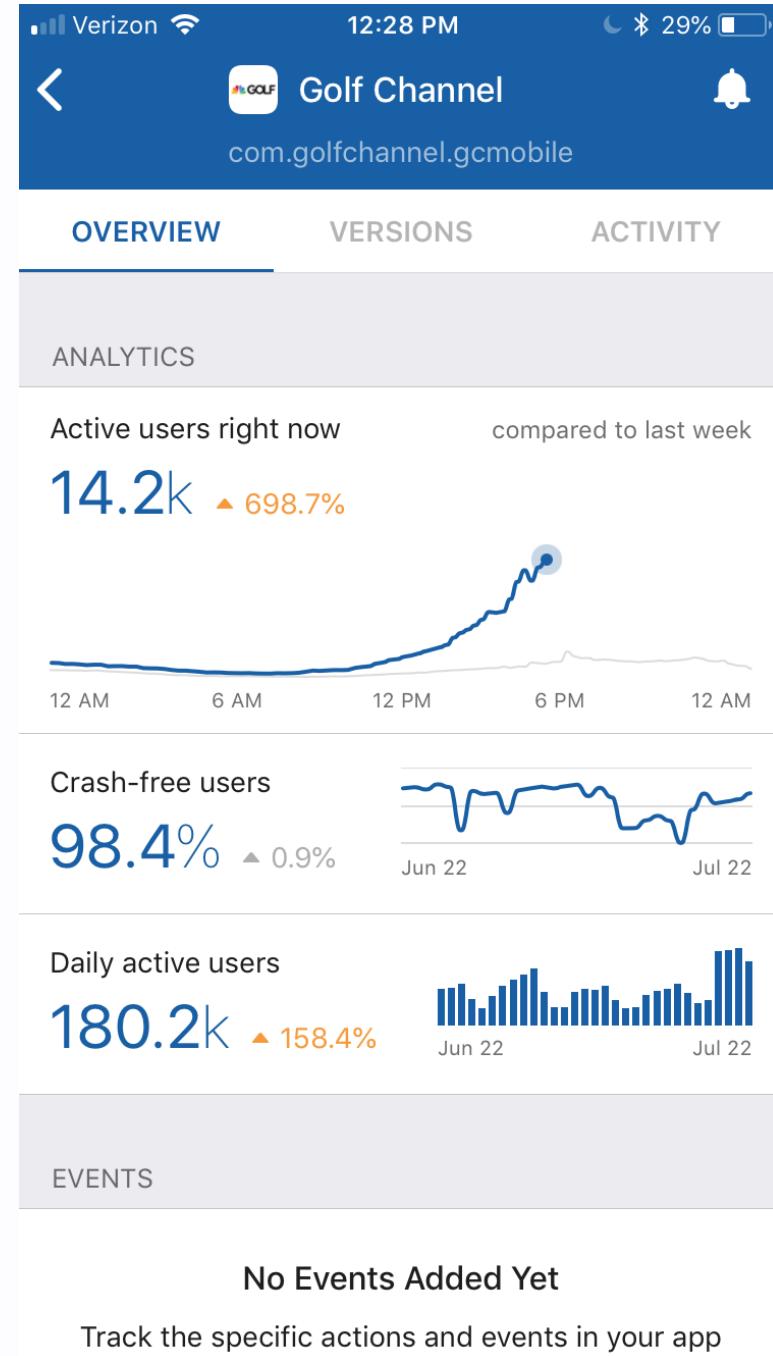
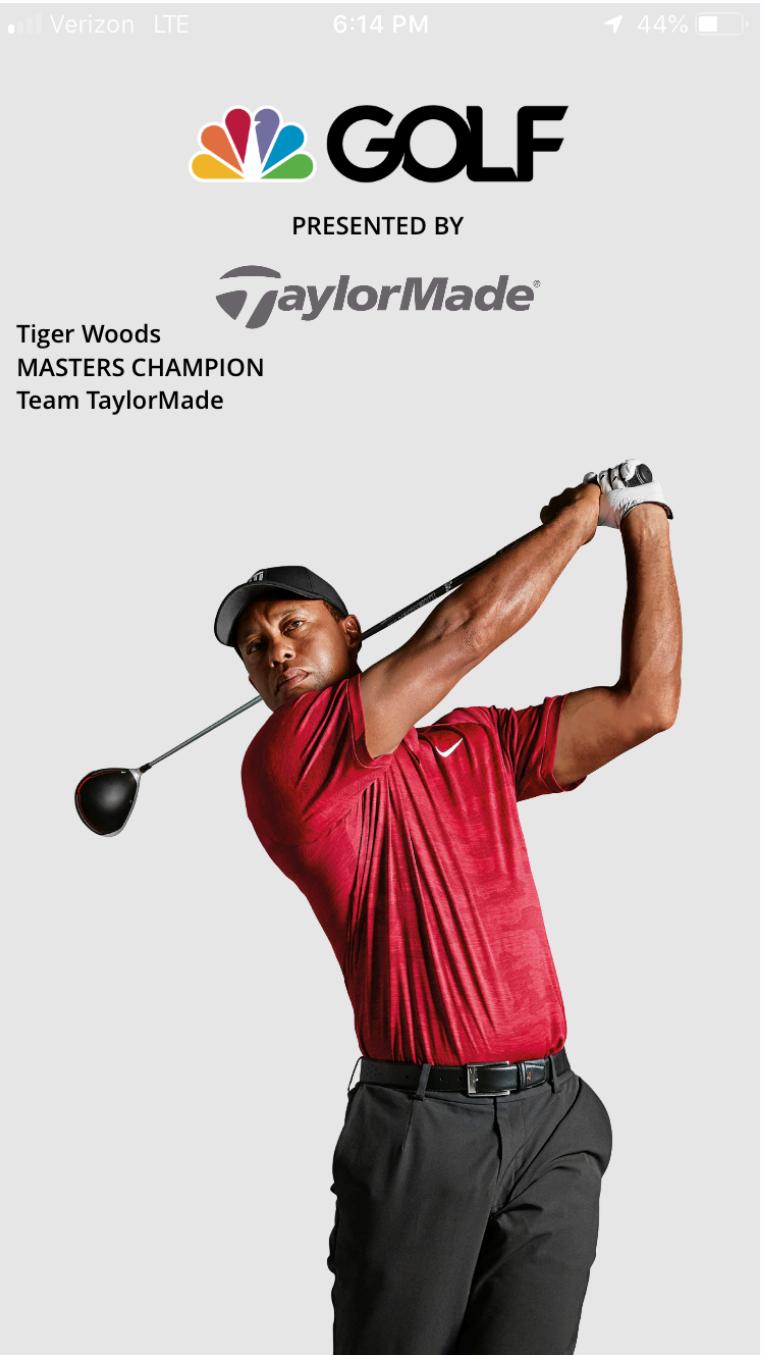
PRESENTED BY



Tiger Woods







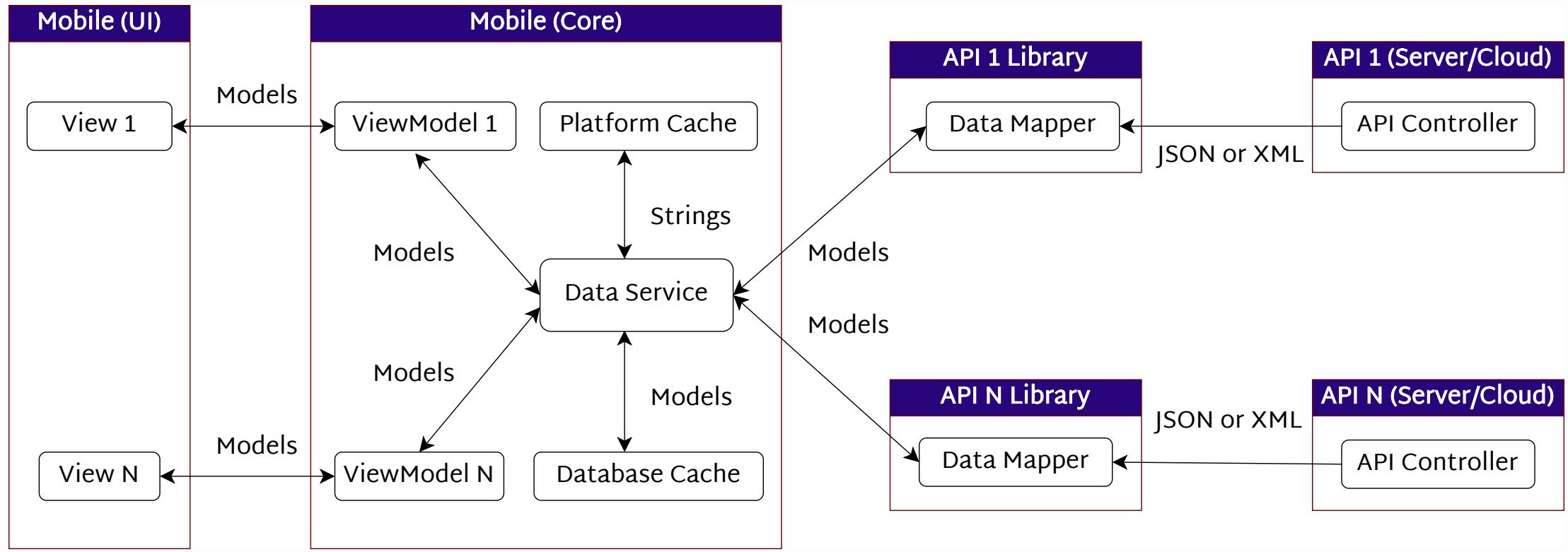
# Part 5

## How?

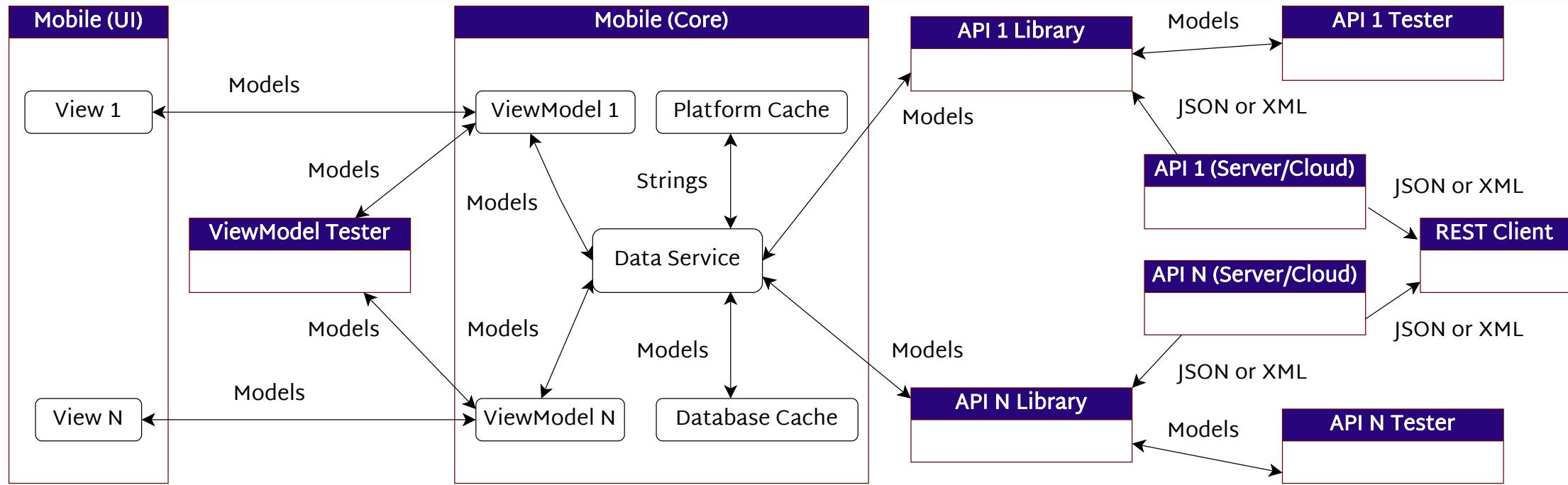
# Mobile Architecture - Components

- **ViewModels**
  - Moving navigation out of the View allows ViewModels to be tested individually with frameworks like ReactiveUI (RxUI)
- **Api Services**
  - Individually testable to return app Models and not just Data Transfer Objects (DTOs)
- **Data Service**
  - Decides whether to call the local cache DB, the platform cache, an API endpoint, or a service based on the requested data and the app lifecycle

# Mobile Architecture - Solution



# Mobile Architecture - Testing



# Part 6

There is another way ...

# Problem Details

## Pros

- Extensible JSON object
- Provides context to status code
- Can report errors or warnings
- Supported in .NET without ASP.NET
- Returned by Refit's ApiException

## Cons

- Buggy return type in ASP.NET
- .NET implementations follow JSON

```
# NOTE: '\' line wrapping per RFC 8792
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "An RFC 7807 problem object",
  "type": "object",
  "properties": {
    "type": {
      "type": "string",
      "format": "uri-reference",
      "description": "A URI reference that identifies the problem type."
    },
    "title": {
      "type": "string",
      "description": "A short, human-readable summary of the problem type."
    },
    "status": {
      "type": "integer",
      "description": "The HTTP status code generated by the origin server \
for this occurrence of the problem.",
      "minimum": 100,
      "maximum": 599
    },
    "detail": {
      "type": "string",
      "description": "A human-readable explanation specific to this occurrence \
of the problem."
    },
    "instance": {
      "type": "string",
      "format": "uri-reference",
      "description": "A URI reference that identifies the specific occurrence \
of the problem. It may or may not yield further information if dereferenced."
    }
  }
}
```

Source: IETF RFC 9457 - Problem Details for HTTP APIs [www.rfc-editor.org/rfc/rfc9457.html](http://www.rfc-editor.org/rfc/rfc9457.html)

# Problem Details - Logic Failure Example

```
POST /purchase HTTP/1.1
Host: store.example.com
Content-Type: application/json
Accept: application/json, application/problem+json

{
  "item": 123456,
  "quantity": 2
}
```

```
HTTP/1.1 403 Forbidden
Content-Type: application/problem+json
Content-Language: en

{
  "type": "https://example.com/probs/out-of-credit",
  "title": "You do not have enough credit.",
  "detail": "Your current balance is 30, but that costs 50.",
  "instance": "/account/12345msgs/abc",
  "balance": 30,
  "accounts": ["/account/12345",
              "/account/67890"]
}
```

# Problem Details - Validation Failure Example

```
POST /details HTTP/1.1
Host: account.example.com
Content-Type: application/json
Accept: application/json, application/problem+json

{
  "age": 42.3,
  "profile": {
    "color": "yellow"
  }
}
```

```
HTTP/1.1 422 Unprocessable Content
Content-Type: application/problem+json
Content-Language: en

{
  "type": "https://example.net/validation-error",
  "title": "Your request is not valid.",
  "errors": [
    {
      "detail": "must be a positive integer",
      "pointer": "#/age"
    },
    {
      "detail": "must be 'green', 'red' or 'blue'",
      "pointer": "#/profile/color"
    }
  ]
}
```

# Part 7

Questions?



Find Mobile Bugs Faster through ViewModel and API Library  
Testing