# The Secret to Mobile

- API Design
- App Architecture
- Data Handling

# Topics

- Infrastructure differences between Web and Mobile
- Architecturing your Mobile app for better results
- Making better data decisions for your Mobile app
- Replacing exceptions with extensible error models

# Part 1

**Mobile is just like the web but smaller**

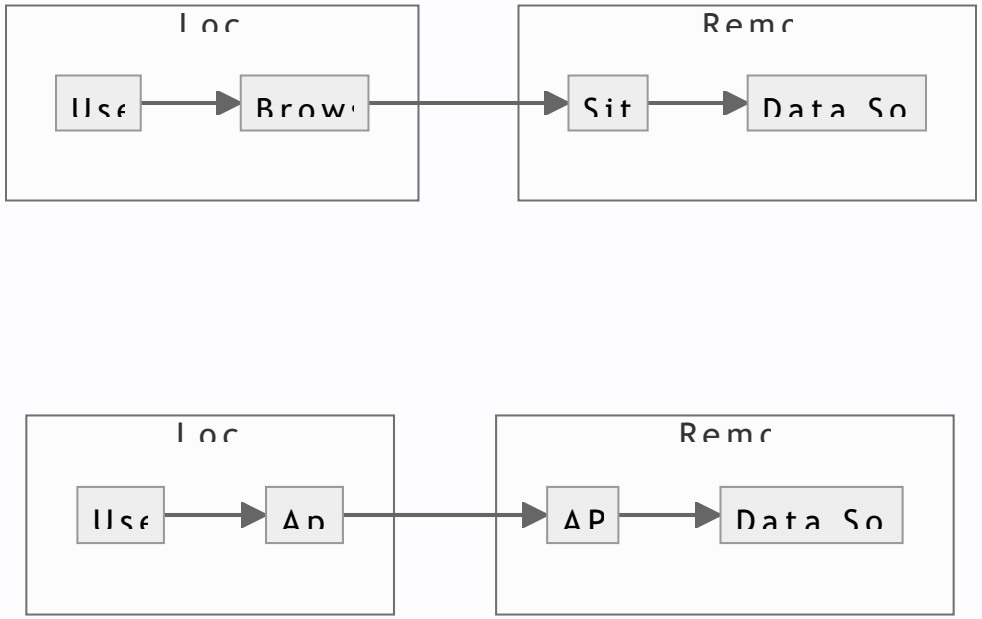
**Right!?**

**or**

**Web devs make bad Mobile devs**

**... but they can learn**

# Web vs Mobile - High Level View




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

# Web vs Mobile - Comparable (.NET)

- Languages: C#, LINQ, VB.NET, F#, etc.
- Frameworks: .NET, Entity Framework, etc.
- Data: MSSQL, Azure, etc.
- IDEs: Visual Studio, Visual Studio Code, Rider, etc.
- Tools: ReSharper, etc.

# Web Sites (Traditional / SPAs)

## *Browser*

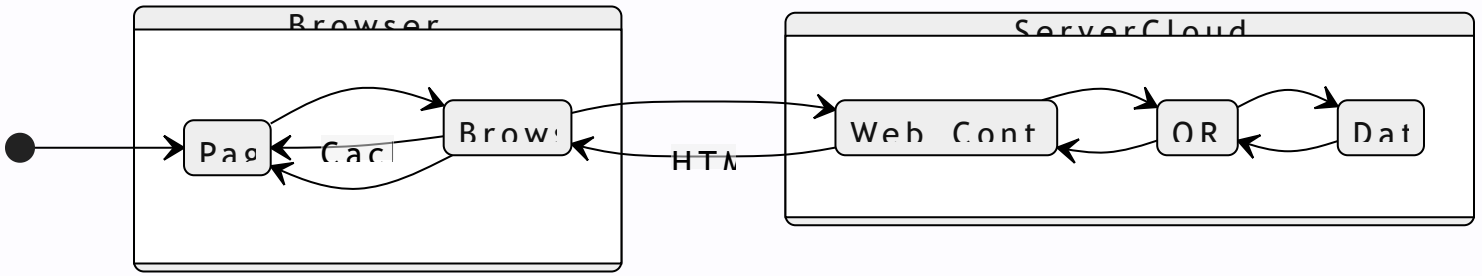
- Synchronous Browse, Submit (*)
- Caching: Browser
- Dev Focus: Interactivity, Updates

## *Server/Cloud*

- Site Response: Page (HTML/JS)
- State Management: Server, Page
- ORM: Data Source to Server/Cloud
- Caching: CDN, Server/Cloud
- Resiliency: Server/Cloud
- Dev Focus: Layouts, State, Services
- Data Goal: Fat Data Pipes

- (*) Single-page application (SPAs) request HTML like synchronous sites and update their content via API calls but still rely on the browser infrastructure

# Web Sites (Traditional / SPAs)




- The server/cloud stack is the focus, turning data into pages and layouts
- Complex page layouts are generally built on top of templating frameworks
- SPAs may modify a page layout, but the initial layout comes from the server
- Result: web devs expect fat data pipes in the server/cloud stack so they can get large data payloads (object graphs) all at once and choose what to filter

# Web vs Mobile - Different Focuses

## *Web Site*

- Synchronous Browse, Submit
- Site Response: Page (HTML/JS)
- State Management: Server, Page
- ORM: Data Source to Server/Cloud
- Client Caching: Browser
- Focus: Heavy Server/Cloud Services
- Resiliency Needed: Server/Cloud
- Data Goal: Fat Data Pipes

## *Mobile App*

- Asynchronous REST (GET, POST, etc.)
- API Response: JSON/XML (DTO)
- State Management: App, App Cache
- ORM: Data Source to API, API to App
- Client Caching: App (Custom or OS)
- Focus: Light API Consumption
- Resiliency Needed: API Calls
- Data Goal: Smart Data Pipes

- Single-page application (SPAs) request HTML like synchronous web sites and update their content via API calls but still rely on the browser infrastructure.

# Web vs Mobile - Different Focuses

## *Web Site*

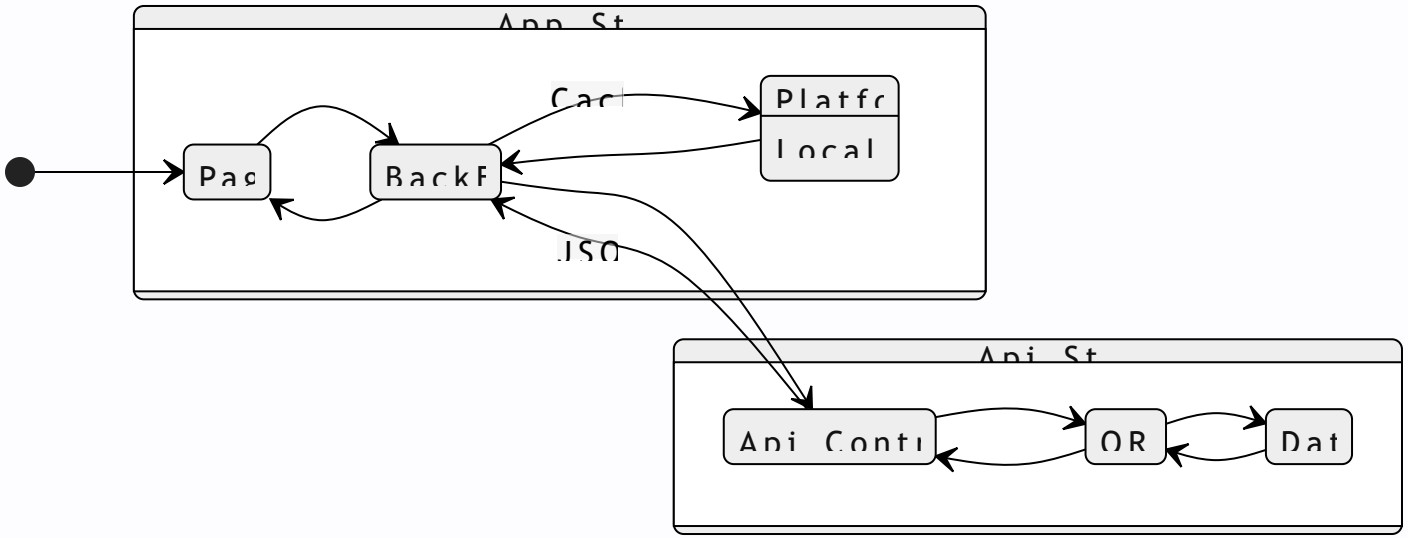
- Synchronous Browse, Submit
- Site Response: Page (HTML/JS)
- State Management: Server, Page
- ORM: Data Source to Server/Cloud
- Client Caching: Browser
- Focus: Heavy Server/Cloud Services
- Resiliency Needed: Server/Cloud
- Data Goal: Fat Data Pipes

## *Mobile App*

- Asynchronous REST (GET, POST, etc.)
- API Response: JSON/XML (DTO)
- State Management: App, App Cache
- ORM: Data Source to API, API to App
- Client Caching: App (Custom or OS)
- Focus: Light API Consumption
- Resiliency Needed: API Calls
- Data Goal: Smart Data Pipes

- Single-page application (SPAs) request HTML like synchronous web sites and update their content via API calls but still rely on the browser infrastructure.

# Mobile Apps




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

# Part 2

## Imagine your house was an app ...

# Part 3

## It's the data, stupid

## Picture of James Carville

## Picture of War Room whiteboard

- **User perspective of house: cross-section and/or floor plan**

    TODO Need cross-section diagram of house

    TODO Need floor plan diagram of house

- **Infrastructure perspective of house:**

TODO Need cross-section diagram of house with electrical, water, etc.

- Turn house on side and pull

# Part 4

## ???

# Part 4

## Explain yourself, dammit

## Picture of Lewis Black

# Part 5

# Closing remarks