Mobile Computing

Devices
Platforms

Mobile Devices - Categories

- Mobile phones

- Smartphones (no keyboard + touch)
- PDAs (personal digital assistants)
  - pocket assistant without phone

- Wearables (watches, glasses)
- Handhelds (and ultra mobiles)

- Tablets (and phablets)

Mobile devices - capabilities

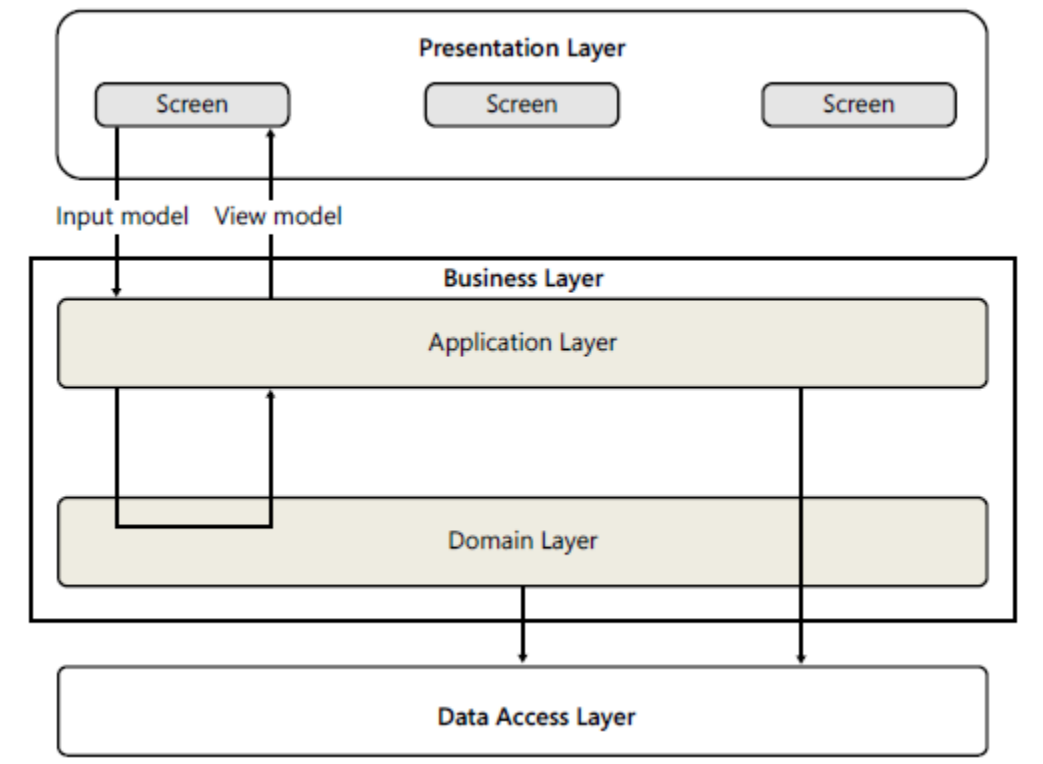
- Software: distri
  specificities of
  - Information ac
  - Storage and loc
  - Specific interfa
    - Mainly based
  - Special functio
    - GPS and loca
    - Compass (ma
    - Acceleration
  - Communicatio
    - 3/4/5G, WiFi,
- Applications
  - For the enterpr
  - For learning / e

Platforms

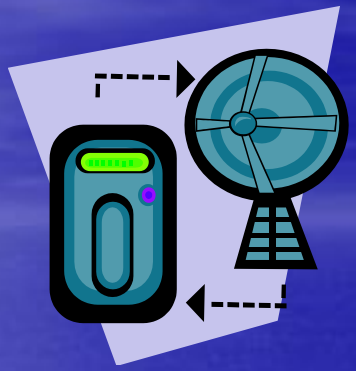
- Operating Systems (many were developed and are available)
  - Android
  - iOS (Apple)
  - BlackBerry
  - Symbian
  - Bada (Samsung)
  - WebOS (Palm)
  - Chrome OS
  - Others …
- Generic frameworks
  - Java ME (some models still support it)
  - PhoneGap (Apache Cordova), Ionic, Titanium Mobile, RhoMobile, …
  - Xamarin (.NET C#) (iOS, Android, Mac, Windows)
  - React Native (JS), Flutter (Dart)
- Application types
  - Web apps – generic mobile browsers (XHTML, HTML5, javascript)
  - Hybrid apps – web technologies encapsulated in a native container
  - Native apps – using the OS and a specific or generic framework

Generic architecture

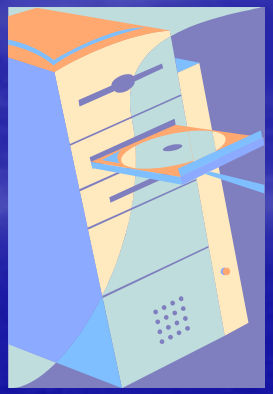
- The typical three tier architecture
  - In a connected environment some tiers can be remote or shared

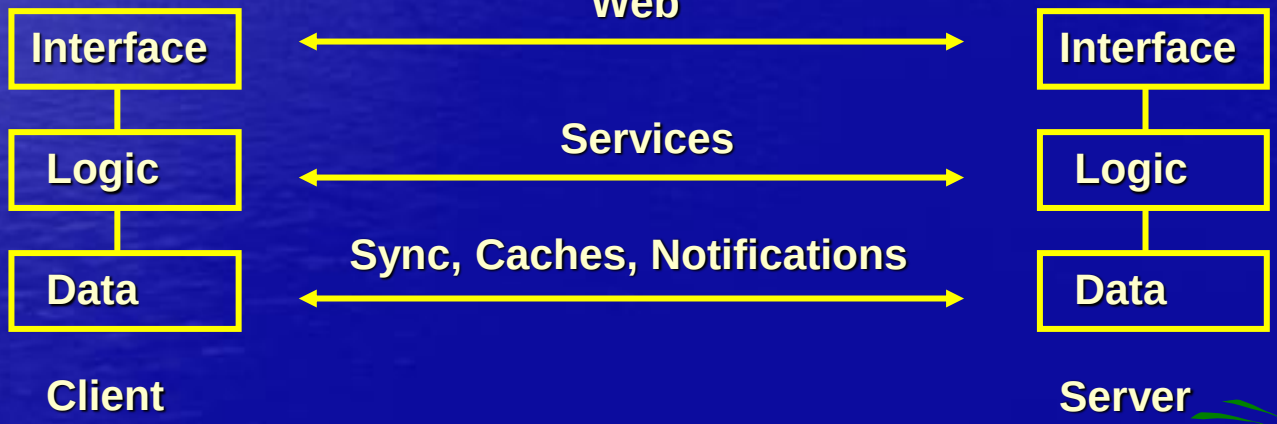


Generic connected application architecture



wireless





Areas of special interest

- Application life cycle
  - Usually different from desktop applications
  - Mobile apps could be suspended by another app
  - It should be possible to resume a background app without loosing state
- Local data storage
  - Several forms: settings, files, databases, …
  - Full relational databases are available on the device
  - Even NoSQL flavors are now available
- Connectivity
  - Despite all advertisement, it's not 100% reliable
  - Data synchronization
  - Occasionally connected functionalities

Specially interesting mobile design patterns (1)

- Interaction patterns
  - Back-and-save
    - Save input screen data when the user leaves the screen
  - Auto save
    - Save the user input periodically
  - Guess-Don’t-Ask
    - Avoid user input, specially writing text
    - If you can’t guess, remember
  - A-la-Carte-Menu
    - At any time the user should know all actions and options available
  - Sink-or-Async
    - Operations longer than 1 s should be asynchronous
  - Logon-and-forget
    - When possible, credentials should be asked only once

Specially interesting mobile design patterns (2)

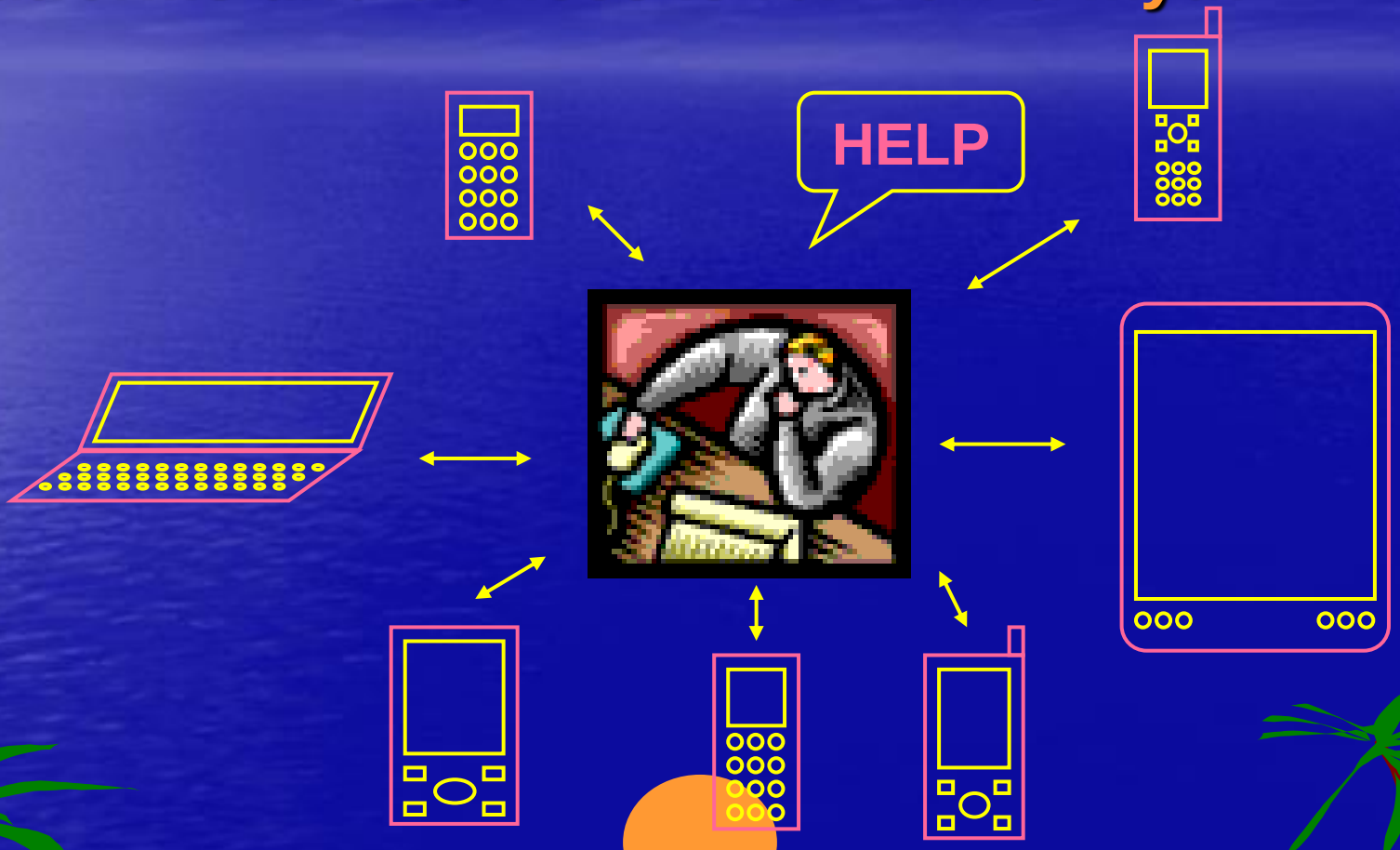
- **Presentation patterns**
  - **Babel-Tower**
    - Avoid hard-coded and fixed layouts
    - Support alternative adaptable layouts
  - **Do-as-Romans-Do**
    - Use the recommended look-and-fill for the platform (native)
  - **List-and-Scroll**
    - Use lists and vertical scrolling only
    - Avoid horizontal scrolling (to read text)
      - Ok for showing extra columns on a table

Specially interesting mobile design patterns (3)

- **Behavioral patterns**
  - **Predictive Fetch**
    - If the app depends on connectivity download data, likely to be used later, whenever connectivity is available
  - **Memento-Mori**
    - Save relevant state and info whenever the app goes into the background
  - **As-Soon-As-Possible**
    - Insist on remote operations and don't fail at first attempt
    - In case of failure, record, and playback when connectivity returns

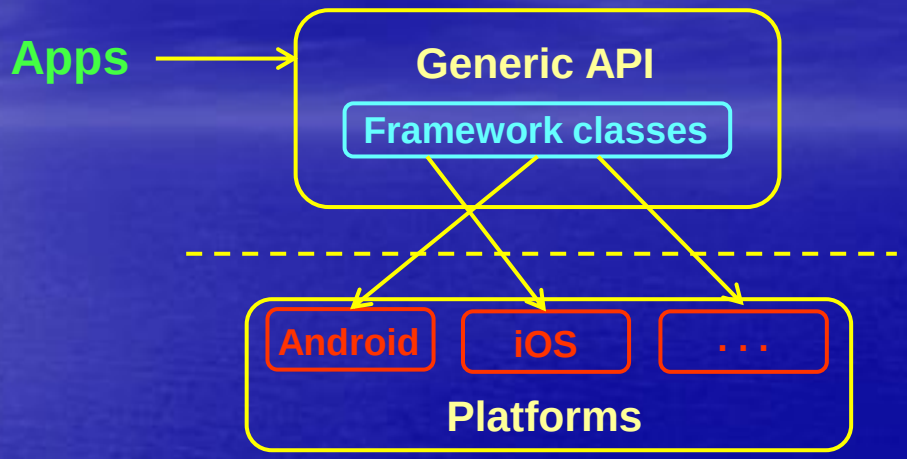
Diversity / Cross platform development

- A single framework for a large number of platforms and devices is for now a myth



Cross platform approaches

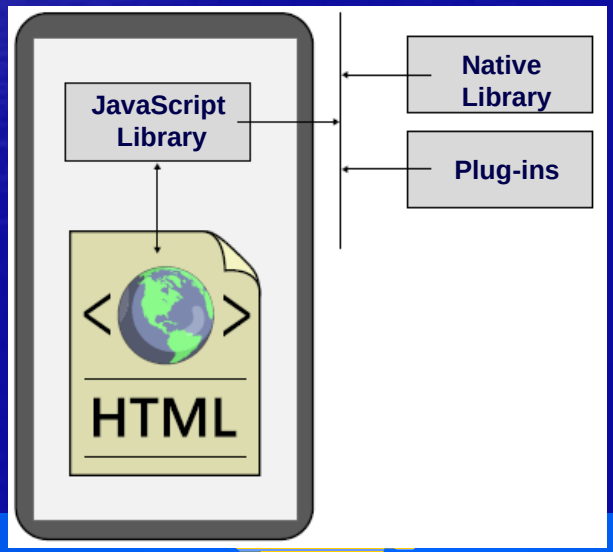
Abstraction layer of translation



Ex: Titanium Mobile
(in run time)

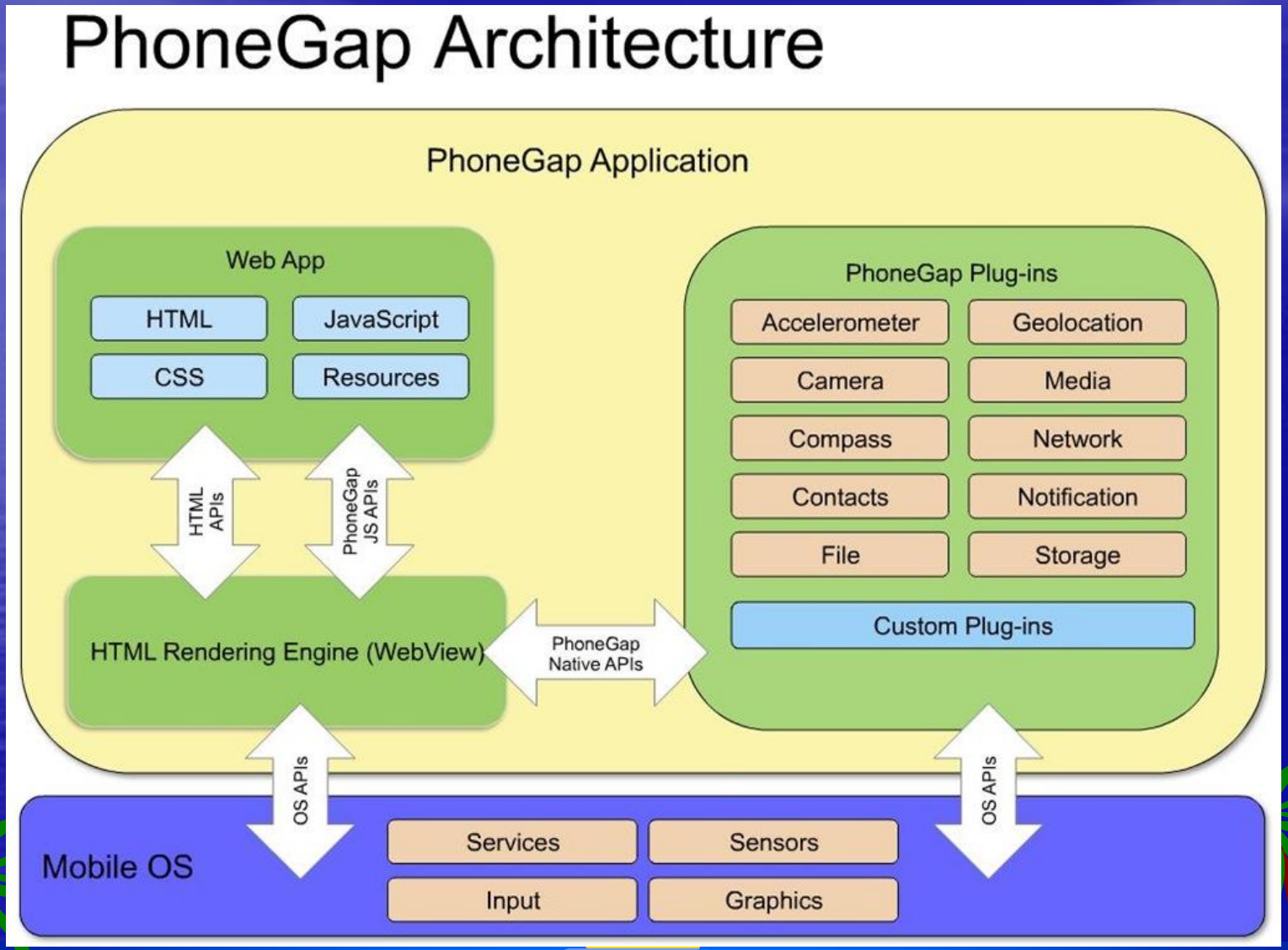
Xamarin
(in build time)

Web shell

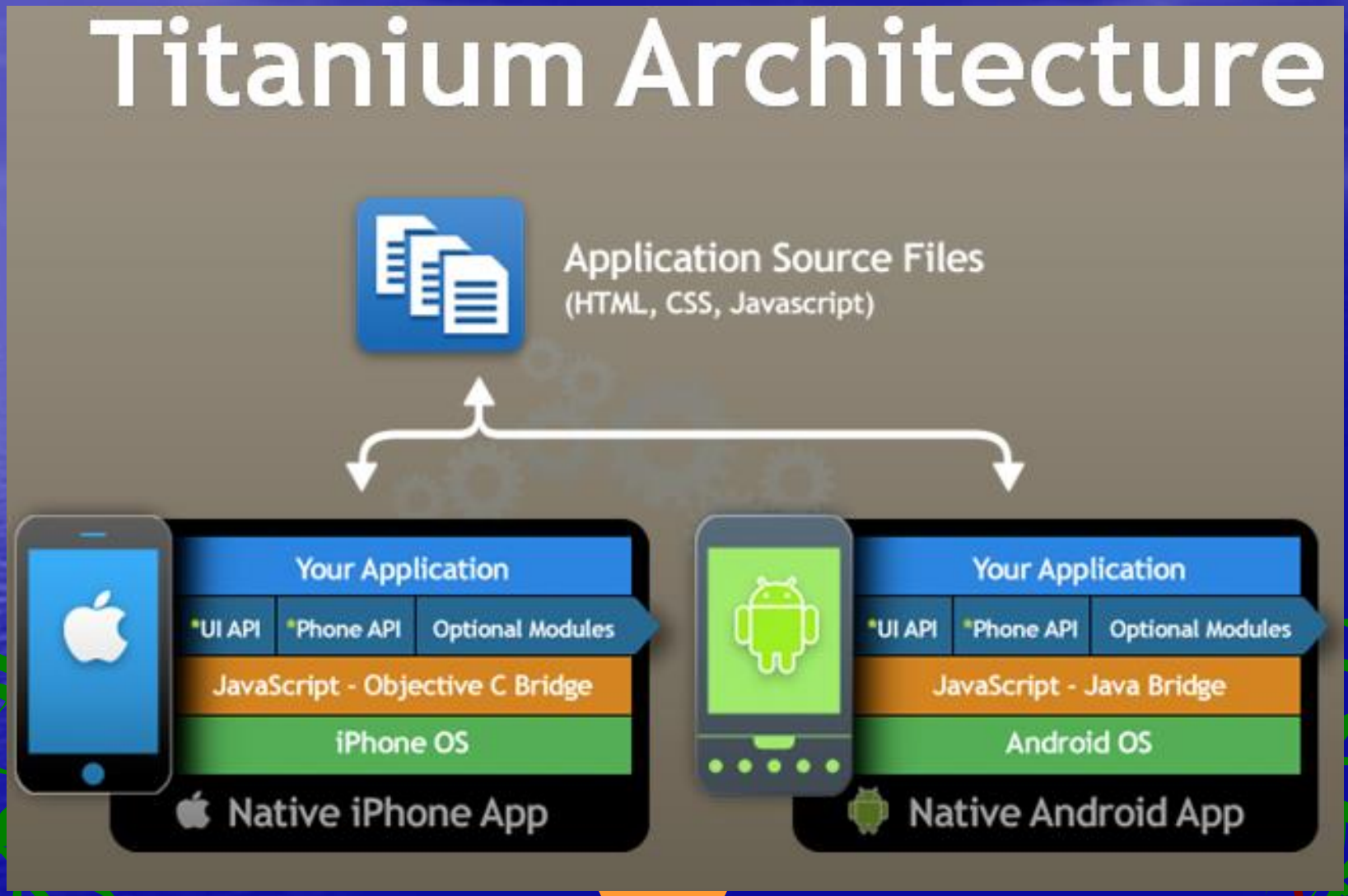


Ex: PhoneGap

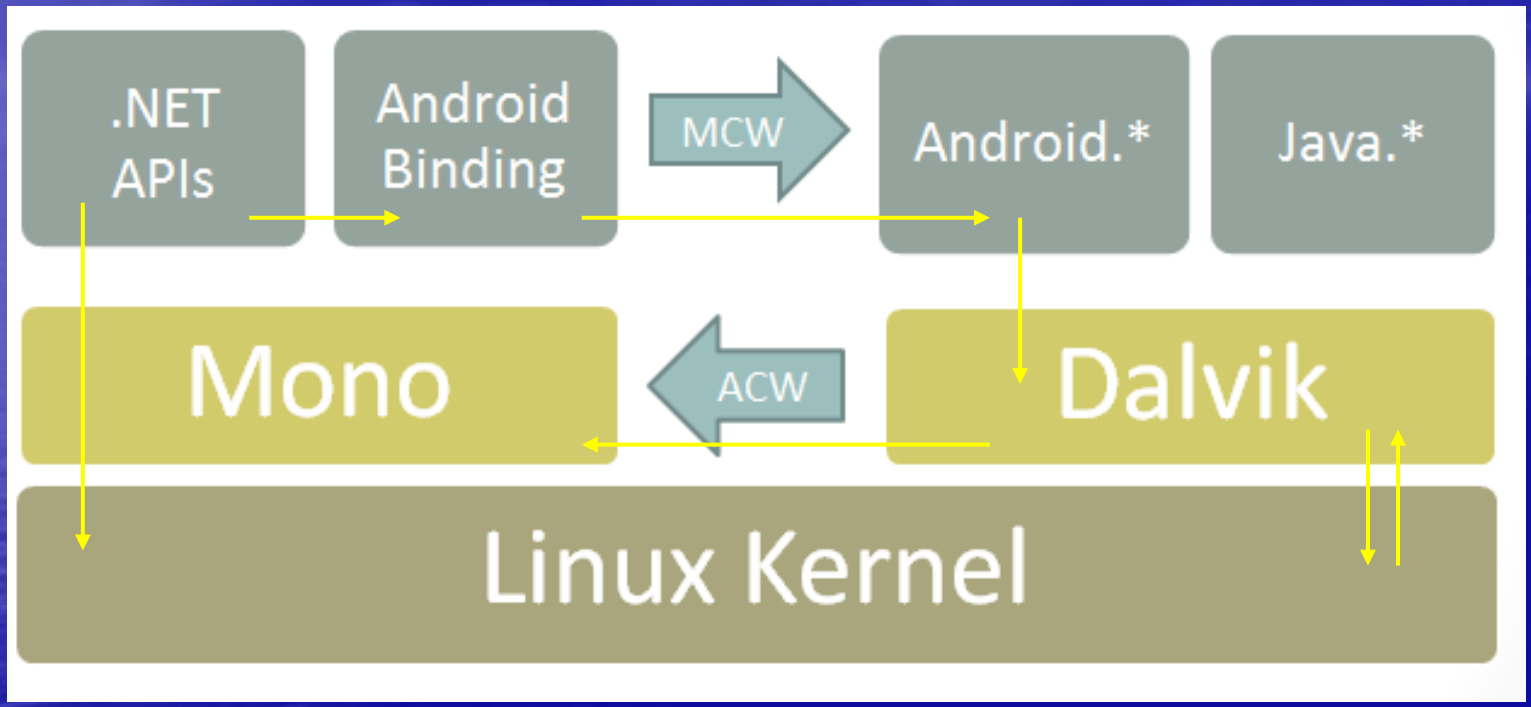
PhoneGap


PhoneGap Architecture
PhoneGap Application
Web App
HTML
JavaScript
CSS
Resources
HTML APIs
PhoneGap JS APIs
HTML Rendering Engine (WebView)
PhoneGap Native APIs
PhoneGap Plug-ins
Accelerometer
Geolocation
Camera
Media
Compass
Network
Contacts
Notification
File
Storage
Custom Plug-ins
OS APIs
OS APIs
Mobile OS
Services
Sensors
Input
Graphics

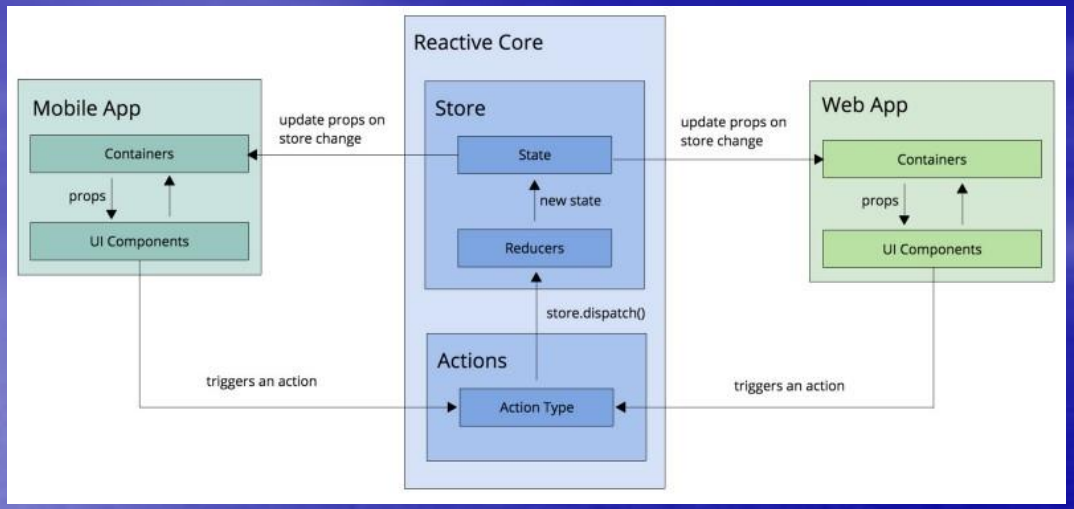
Titanium


Titanium Architecture
Application Source Files
(HTML, CSS, Javascript)
Your Application
*UI API
*Phone API
Optional Modules
JavaScript - Objective C Bridge
iPhone OS
Native iPhone App
Your Application
*UI API
*Phone API
Optional Modules
JavaScript - Java Bridge
Android OS
Native Android App

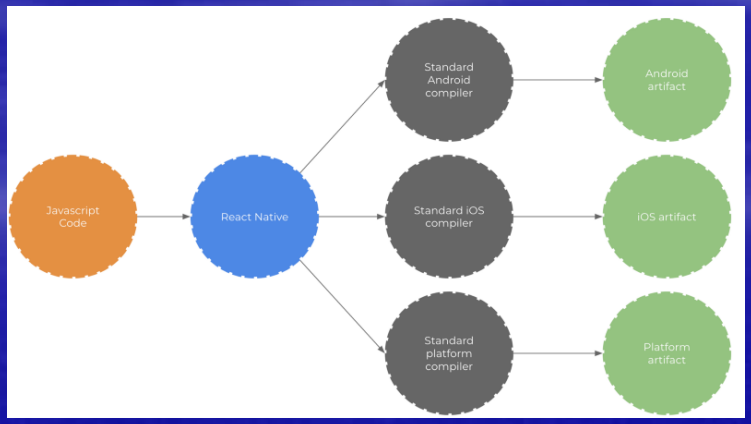
Xamarin


.NET APIs
Android Binding
MCW
Android.*
Java.*
Mono
ACW
Dalvik
Linux Kernel

React Native

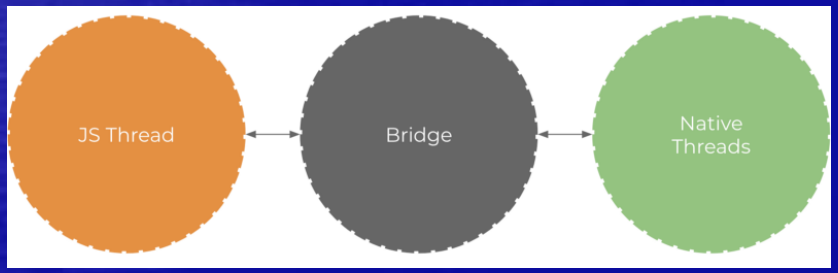


Architecture



Building

Run-time




```
class HelloWorldApp extends Component {
  render() { return ( <View style={{padding:40}}>
                        <Text>Hello world!</Text>
                      </View>
                     );
           }
}
```

JSX source example