I. Introduction
II. Category of Applications
III. Key Technical Criteria for Evaluating Mobile Architecture
IV. What is Smart App?
V. Designing of Reference Mobile Clients Architecture
VI. Exemplified Scenarios (1~5)
VII. Conclusion
Choosing appropriate client architectures is very important to develop mobile applications with high quality and productivity.

The design of mobile applications must take into account several unique aspects like the device form factor, sporadic connectivity, variable bandwidth, multi-platform support and user experience.

The three client architectures – Web App, Native App and Hybrid App – differ significantly.

In this speech, I will discuss the mobile client architectures and its environments.
One of the first steps for development of any mobile application is selecting the right client architecture. There are three popular approaches today:

- **Web Apps**
  - the application runs on a mobile browser. The browser only hosts the application’s presentation layer that is designed using HTML5.

- **Native Apps**
  - the mobile application is custom built for the target device operating system with a compiled programming language like Objective-C and using the native SDK

- **Hybrid Apps**
  - This approach emerged to address the inability of the Web App approach to access device sensors (like cameras and Bluetooth) while preserving its highly desirable cross-platform support.
## Layout of Client 3 tiers

<table>
<thead>
<tr>
<th></th>
<th>Device Access</th>
<th>Speed</th>
<th>Development Cost</th>
<th>App Store</th>
<th>Approval Process</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Native</strong></td>
<td>Full</td>
<td>Very Fast</td>
<td>Expensive</td>
<td>Available</td>
<td>Mandatory</td>
</tr>
<tr>
<td><strong>Hybrid</strong></td>
<td>Full</td>
<td>Native Speed as Necessary</td>
<td>Reasonable</td>
<td>Available</td>
<td>Low Overhead</td>
</tr>
<tr>
<td><strong>Web</strong></td>
<td>Partial</td>
<td>Fast</td>
<td>Reasonable</td>
<td>Not Available</td>
<td>None</td>
</tr>
</tbody>
</table>
Mobile Web Approach

- Relying on the web browser

- Mobile web apps are designed to run on a mobile web browser. HTML5 is the most popular and promising technology for ‘Write Once Run Anywhere.’

- Almost all mobile web browsers running on high-end mobile devices support HTML5 to a large extent,

- This is thin client and 1 layer Architecture
Mobile Web Approach

Middleware

Components

Synchronization Authentication Notification Etc...

Web Server

HTML 5 Web Pages

Web Services

Enterprise Network

Databases CMS AS

Data WH BI Reporting

AJAX
Native App Approach

- Mobile native apps are built using the native device operating system APIs and SDKs.

- These are coded using a platform specific language like Objective C for iOS, Java for Android, and C# for Windows phone.

- This is Rich client and 2 layer Architecture.
Native App Approach

Diagram showing components such as Middleware, Components, Web Server, Enterprise Network, Databases, Web Services, Connectivity Layer, Native Container App.
This approach achieves the middle ground between native mobile applications and mobile web applications.

While mobile web apps attempt to provide platform independence, the price one pays for this is that they do not function when the device is offline and they cannot access device hardware like the camera, Bluetooth, accelerometer, or compass.

The Hybrid App approach evolved to deliver platform independence while providing access to the device hardware and offline operation.

This is Rich client and 3 layer Architecture
Hybrid App Approach

Middleware

Components
- Synchronization
- Authentication
- Notification
- Etc..

Web Server
- HTML 5
- Web Pages
- Web Services

Hy5 Native Bride/Wrapping
- Javascript Interfaces
  with the Bridgeware

Browse Control running HTML5 GUI

Enterprise Network
- Databases
- CMS
- AS
- Data WH
- BI Reporting

Native Portion to access device capabilities
Key Technical Criteria for Evaluating Mobile Architecture

- **Access to Hardware Sensors**
  - One of the main disadvantages of the Web App approach is the inability to access device capabilities.
  - Hybrid App can access all device sensors. Most popular hybrid application development frameworks provide access to almost all the important device capabilities.
  - Native apps are ideally suited to use all the device sensors and various peripherals. It would provide a seamless and native user experience that is responsive.

- **Performance**
  - Mobile web apps and hybrid apps are slower since their code is interpreted by the JavaScript engine running within the browser.
  - When it comes to computational needs, the Native App approach outperforms the other two approaches by a wide margin.
Key Technical Criteria for Evaluating Mobile Architecture

- **Native Look and Feel**
  - There are several web frameworks that provide libraries that can be used by mobile web apps and hybrid apps to re-create and imitate native mobile interfaces and behavior.
  - Of course with the Native App approach, one automatically gets the native look and feel.

- **Search, Distribution and Upgrades**
  - Mobile web apps can be hosted on a web server like any website; they do not require any download or installation.
  - In contrast, native apps and hybrid apps are typically hosted in an app store and must be downloaded and installed.
Key Technical Criteria for Evaluating Mobile Architecture

- Offline Capability
  - Web apps require connectivity to be operational.
  - Native and Hybrid App approaches, one can access the device database and implement a synchronization engine that would allow seamless operation when the device has sporadic connectivity.
  - Hybrid App approach, the images, or videos are typically inserted inside the app and thus don’t need to be downloaded from any server.

- Development & Testing
  - For a typical native application, roughly 20% of the effort is user experience design, 20% is requirements and design, 40% is development and 20% is testing and miscellaneous.
What is Smart App?

- Combine the benefits of Web App, Native App and Hybrid App.

- All Smart App share some or all of the following characteristics:
  - Make use of local resources
  - Make use of network resources
  - Support occasionally connected users
  - Provide intelligent installation and update
  - Provide client device flexibility
Smart App Architecture Requirements

Client Type
- Thin client
- Rich client

Connection Type
- Connected
- Never Connected

Synchronization
- Continuous
- Store-and-forward

Device Hardware
- Platform independent
- Provide access to device hardware

- Combination of thin and rich client
- 3 layer architecture
- Support Caching
- Not reliable to network connection
- Support local storage

- Sensors
- GPS, etc.
Overall Mobile Clients Software Architecture
The section presents the Reference Mobile Architecture scenario. There are four scenarios to exemplify the thin, rich, and smart clients.

**Scenario 1**

- **Requirement**: Don’t require any network connection.
- **Solution**:
  - Stand-Alone Layer
  - Local Storage
  - Presentation Layer
  - Business Layer
  - Data Access Layer

**Diagram**

- Mobile Device: Presentation Layer → Business Layer → Data Access Layer → DB

**Legend**

- Requirement
- Suggested Architectural patterns
- Solution
- Reference Architectural
Exemplified Scenarios (2)

Scenario 2

Requirement

Reliable network connection
Retrieves information located on App. server

Suggested Architectural patterns

Client-Server
No Local service
No local data

Solution

Least 2 tiers
Web Browser
Presentation Layer
Business Layer
Data Access Layer

Thin Client

Reference Architectural

Mobile Device
Application Server
Database Server

Web Browser
Presentation Layer
Service Layer
Business Layer
Data Access Layer
DB
Exemplified Scenarios (3)

Scenario 3

- **Requirement**: Reliable network connection
- **Suggested Architectural Patterns**: Client-Server
- **Solution**: Least 2 tiers
- **Thin Client**
  - Reliable network connection
  - Retrieves information located on App. server
  - No Local service
  - No local data

**Reference Architectural**

- **Mobile Device**: Web Browser
- **Application Server**: Presentation Layer, Service Layer, Business Layer
- **Database Server**: DB
**Scenario 4**

- **Requirement**: Reliable network connection, Several network resource, Accessing local service

- **Suggested Architectural patterns**: SOA Layer, local service, No local data

- **Solution**: Predefined interface, Presentation Layer, Business Layer, Data Access Layer

- **Reference Architectural**

*Exemplified Scenarios (4)*
Scenario 5

- Requirement
  - Non-reliable network connection
  - Interact with an application server
  - Functional regardless of network connectivity

- Suggested Architectural patterns
  - Client-Server
  - Message bus
  - Caching
  - Synchronization
  - Local storage (Rich Client)

- Solution
  - Least 2 tiers
  - Local storage mechanism
  - Presentation Layer
  - Business Layer
  - Data Access Layer

- Reference Architectural
Presents a generic scheme for its solutions
Describes one or several solution for each particular recurring design challenge
Conclusion

- **Purpose**
  - Discussed the differences between Native App, Web App, Hybrid App and Smart App
  - Discussed the client and server architecture of mobile devices
  - Discussed the mobile connection and synchronization types

- **Introduced the Smart App**
  - Capabilities
  - Requirements

- **Generic Reference Architecture Template**
References

[16] Mobile Application Pocket Guide v1.1
[17] Microsoft patterns & practices 112
Thank you for listening!
Q & A.