GOVERNMENT ARTS COLLEGE CBE

MOBILE APPLICATION DEVELOPMENT

UNIT-3

MSc COMPUTER SCIENCE

CONTENT

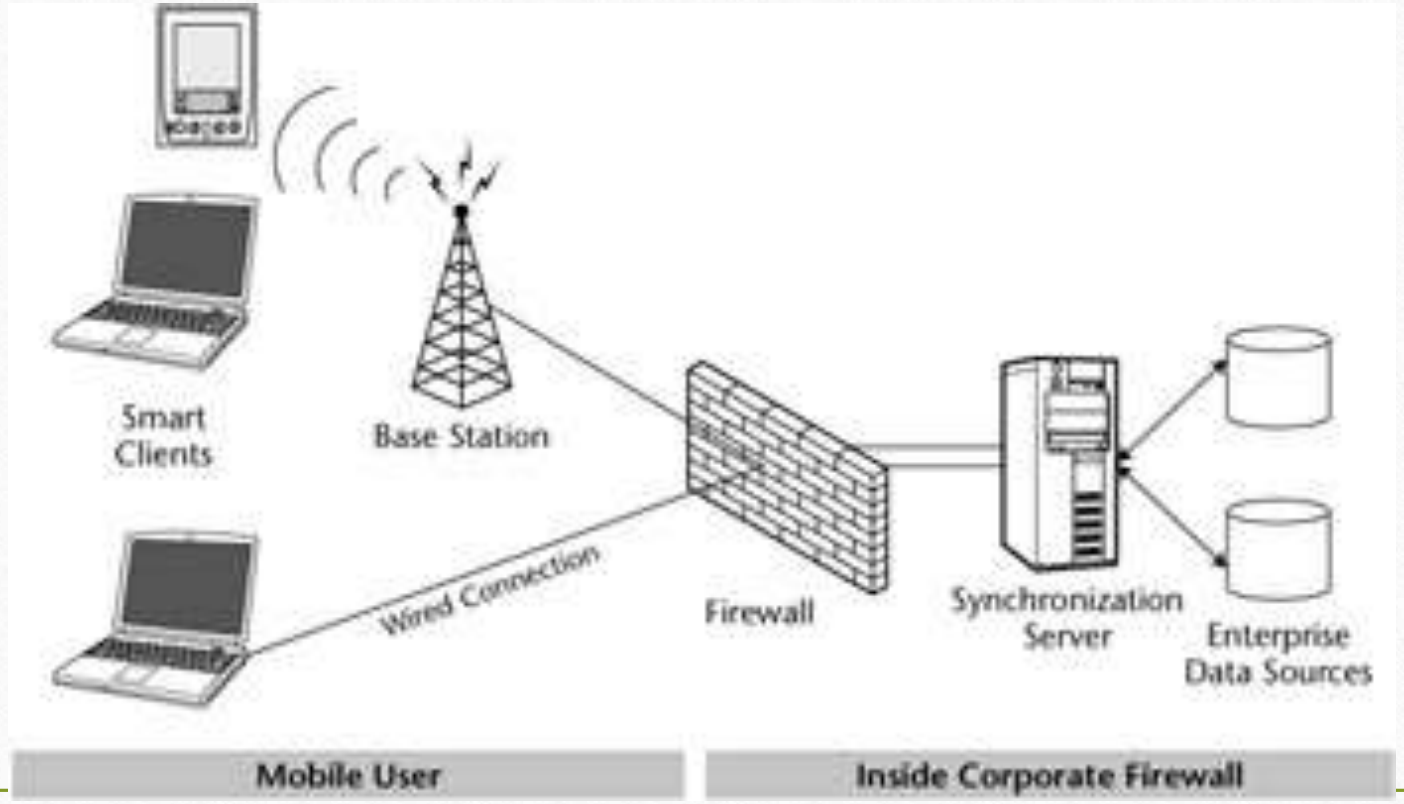
- SMART CLIENT
- SMART ARCHITECTURE
- MESSAGING ARCHITECTURE
- THE MODEL VIEW CONTROLLER MODEL
- DELEGATE PATTERN
- BUILDING SMART CLIENT APPLICATION
- MVVM MOBILE ARCHITECTURE DESIGN

SMART CLIENT

Smart Client

- Smart client applications are a powerful alternative to wireless Internet applications.
- Instead of using a micro browser on the client, custom software is developed.
- This software typically contains a persistent data storage mechanism as well as business logic. This means that smart client applications can be executed at any time, even when a wireless connection is unavailable

Smart Client



SMART ARCHITECTURE

Smart Client Architecture Components

- **Smart Client**
- The smart client application is where client-side business logic is executed. The application itself is either a native executable or Java application that is deployed to the mobile device. To provide offline data access, mobile data store products are incorporated into the application

Synchronization Server

- Data is sent from the client application to the synchronization server.
- This can occur over a wireless or wired connection to the server. From there it is then communicated to the enterprise data sources.
- The synchronization server, with its associated logic, is responsible for ensuring that the minimal amount of data is transferred and that any conflicts are detected and resolved.
- It also provides the communication layer to enterprise systems.

Enterprise Data Source

- The synchronization server will access the enterprise data source using the preferred access mechanism.
- The access to the enterprise source may occur during the synchronization process if it is imperative for the smart client to receive feedback from the synchronization.
- This is a simple process and requires that the client connection to the synchronization server remain active until the enterprise is finished processing the data.

Advantages of Smart Client Applications

- Always-available data
- Rich user interface
- Performance
- Distributed computing
- Security
- Cost

Disadvantages of Smart Client Applications

- Enterprise integration
- Application deployment
- Mobile viruses
- Development complexity
- Multiple development cycles

MESSAGING ARCHITECTURE

Messaging

- Messaging applications can take many forms, ranging from email to alerts and notifications to application-to-application messaging. In some cases messaging is used as an enhancement to an existing mobile application; in other situations, it is itself an application architecture.

User-to-User Messaging

- Messages can be sent from one user to another using a variety of mechanisms, including email, paging, and wireless text messaging such as the Short Message Service (SMS) or Instant Messaging (IM).
- Richer messages that include graphics and formatted text can be sent using the Enhanced Message Service (EMS), while multimedia content can be sent using the Multimedia Message Service (MMS).
- These forms of messaging can also be generated by server-side processes as a means of information dispersal.

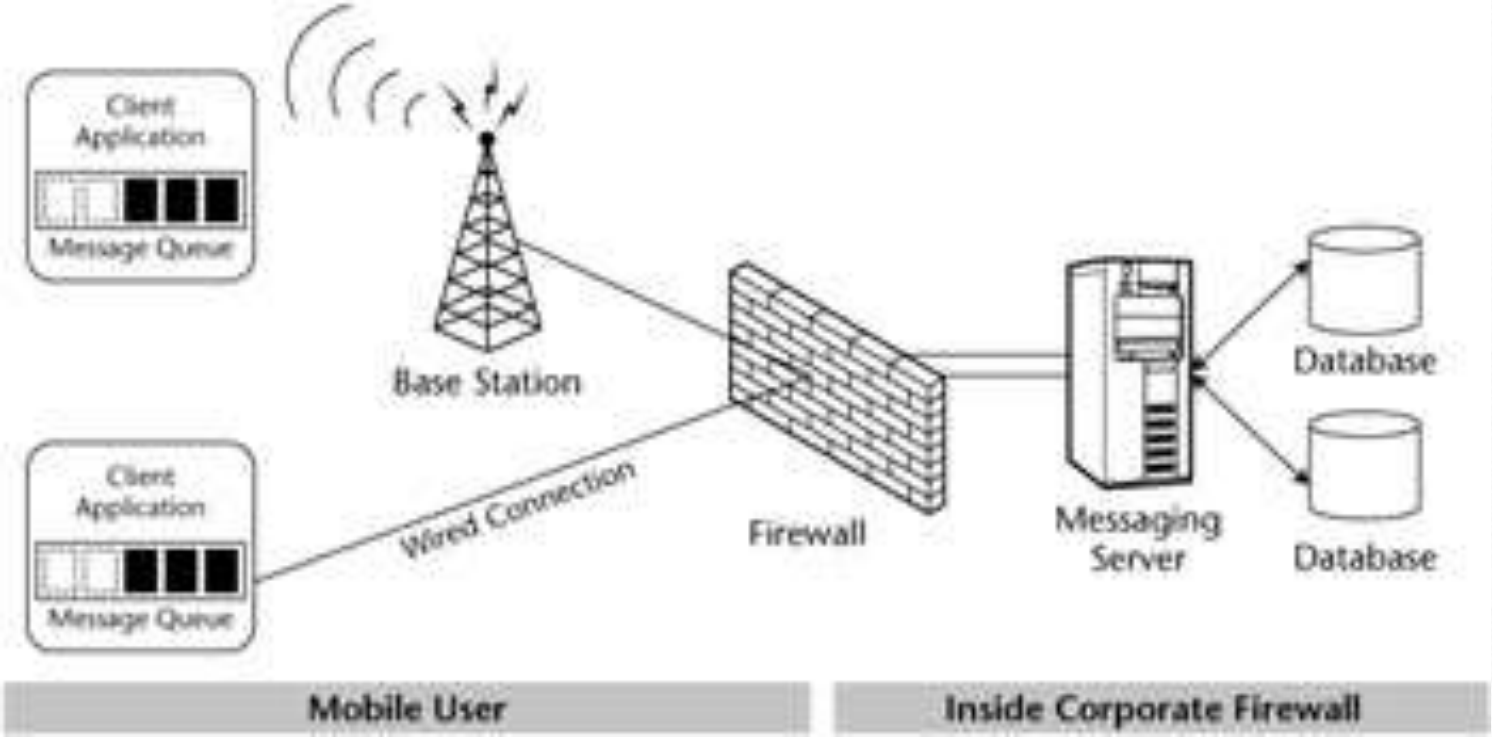
Notifications and Alerts

- Messages that are urgent in nature can be pushed to mobile users on their wireless devices.
- This allows corporations to ensure that information is received in a timely fashion.
- These messages can contain a URL link to a wireless Internet site where the user can obtain additional information. These types of messages are called actionable alerts, since the recipient performs an action based on the message content.

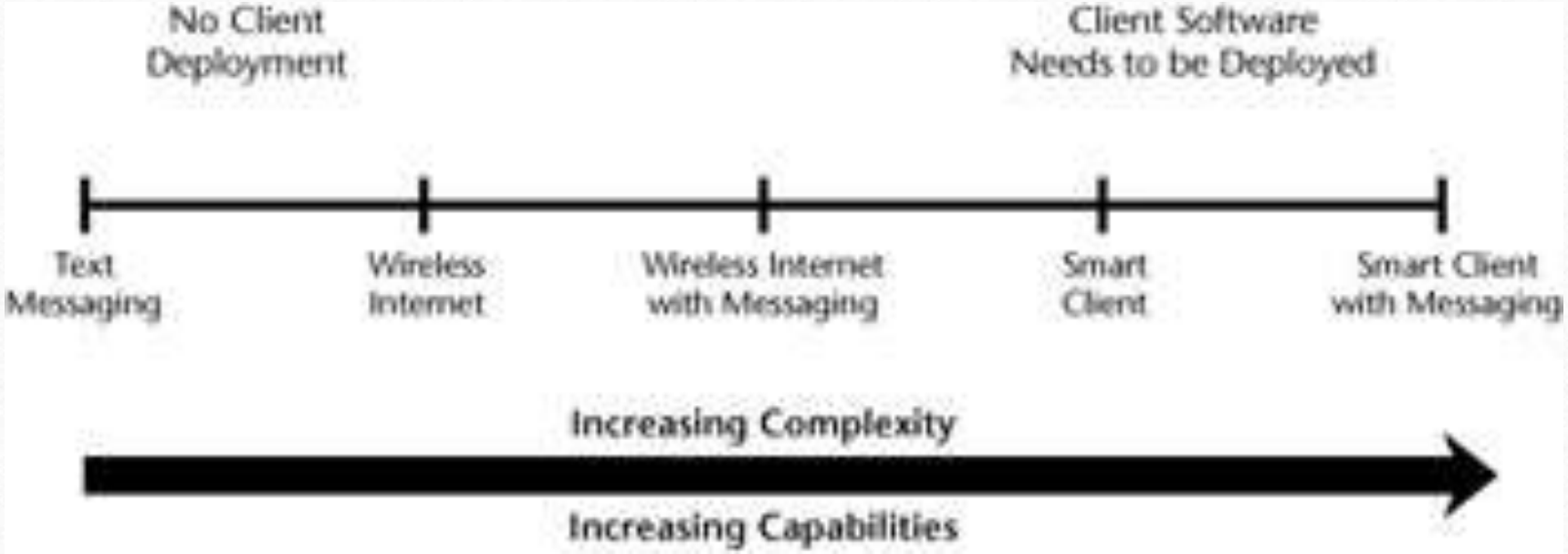
Application-to-Application Messaging

- In many cases, user interaction is not required for the message to be successful.
- Enterprises can communicate data directly from an enterprise server to a client application without user interaction.
- This can be useful to enhance smart client applications with server-initiated synchronization

Application-to-Application Messaging Architecture



Application Architectures



Application-to-Application Messaging Architecture Components

- **Messaging client.** The client application contains message queues as well as clientside logic.
- The message queues can store incoming and outgoing transactions for future access.
- **For example,** if an application attempts to send data to the server when a connection is unavailable, it can be stored in an outgoing queue and be sent automatically when the user establishes a connection to the server. This technique is called store-and-forward messaging.

Messaging Server

- The server component is the part of the system that communicates with the messaging client, as well as the enterprise systems.
- The industry name for messaging servers is Message Oriented Middleware (MOM). Many of these systems are built on the Java Message Service (JMS).
- JMS provides a reliable and scalable base platform with store-and-forward capabilities.

Messaging Server

- This form of messaging is useful for m-business applications because the sender of the message does not have to wait for the recipient to receive the information, allowing him or her to continue working while the message is routed and acted upon.

Enterprise data source

- The messaging server can interact with a variety of backend systems, including databases, business applications, and other messaging systems. This integration can use the preferred access technique for the enterprise system.
- The asynchronous nature of messaging systems is well suited for systems that require complex enterprise integration since the user does not have to wait for a response.

Advantages of Messaging

- Push capabilities
- Store-and-forward
- Personalized data delivery
- Wired and wireless communication

MODEL VIEW CONTROLLER MODEL

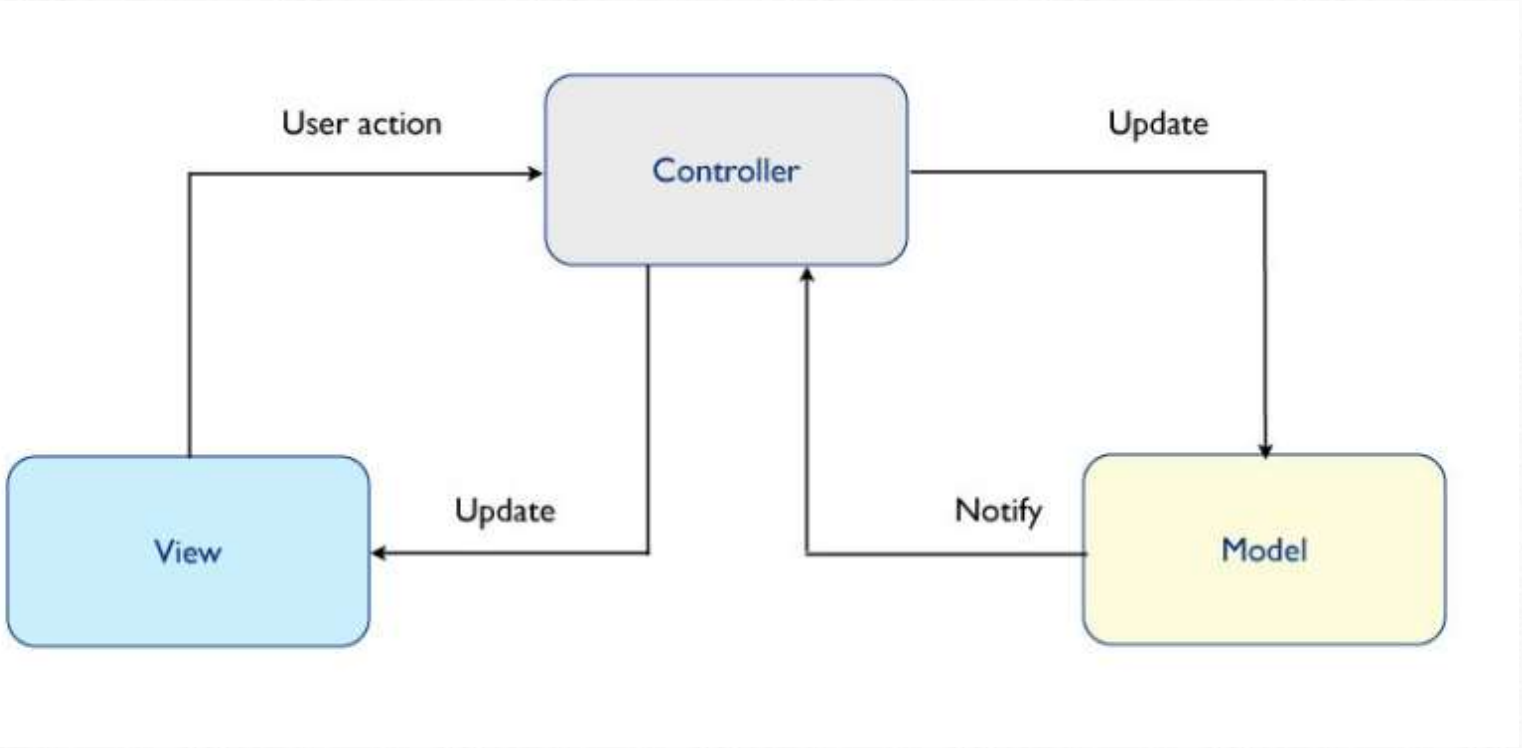
Model View Controller (MVC) Pattern

- A way to separate the code into three functionally independent areas.
- Assigns the objects in an app to one of three roles : model , view , or controller.
- The main purpose for MVC is reusability where user can reuse the same model for different views.

MVC

- Models
- Keep track of your app’s data
- Views
- Display your user interface and make up the content of an app
- Controllers
  - Manage user views by responding to user actions and populating views with content from the data model
  - Serve as a gateway for communication between the model and views

Model View Controller…



Model Objects

- Model objects encapsulate the data specific to an application and define the logic and computation that manipulate and process that data.
- For example, a model object might represent a character in a game or a contact in an address book.
- A model object can have to-one and to-many relationships with other model objects, and so sometimes the model layer of an application effectively is one or more object graphs.

Communication

- User actions in the view layer that create or modify data are communicated through a controller object and result in the creation or updating of a model object.
- When a model object changes (for example, new data is received over a network connection), it notifies a controller object, which updates the appropriate view objects.

Controller Objects

- A controller object acts as an intermediary between one or more of an application’s view objects and one or more of its model objects.
- Controller objects are thus a conduit through which view objects learn about changes in model objects and vice versa.
- Controller objects can also perform setup and coordinating tasks for an application and manage the life cycles of other objects

DELEGATE PATTERN

Delegate Pattern

- A simple and powerful pattern in which one object in an app acts on behalf of/or in coordination with another object.

 - Delegating object

- Keeps a reference to the other object (the delegate)
- The delegating object sends a message to the delegate at appropriate time

Delegating Object

- The Message informs the delegate of an event that the delegating object is about to handle/has just handled
- The Delegate may respond to the message by updating the appearance/state of itself or of other objects in the app
- In some cases it will return a value that affects how an impending event is handled

SMART CLIENT APPLICATION

Android Development

- Released under the open source Apache License
- Built on Linux kernel version 2.6
- A project of the Open Handset Alliance (OHA)
- Founded by Google

Android Development

- On Windows/Linux/Mac platforms
- No Java Virtual Machine on the platform
- Java classes are recompiled in to Dalvik bytecode and run on a Dalvik virtual machine

iOS Development

- Advanced OS
- – With iOS SDK and Xcode IDE creates revolutionary mobile apps

Xcode

- Xcode suite includes Interface Builder and Instruments –
  - Interface Builder helps user create user interfaces for his app
  - – Instruments provides a thorough analysis of user app’s
- Runtime performance
- Memory usage
- Allowing user to efficiently find memory leaks and bottlenecks to help improve the user experience

Blackberry Development

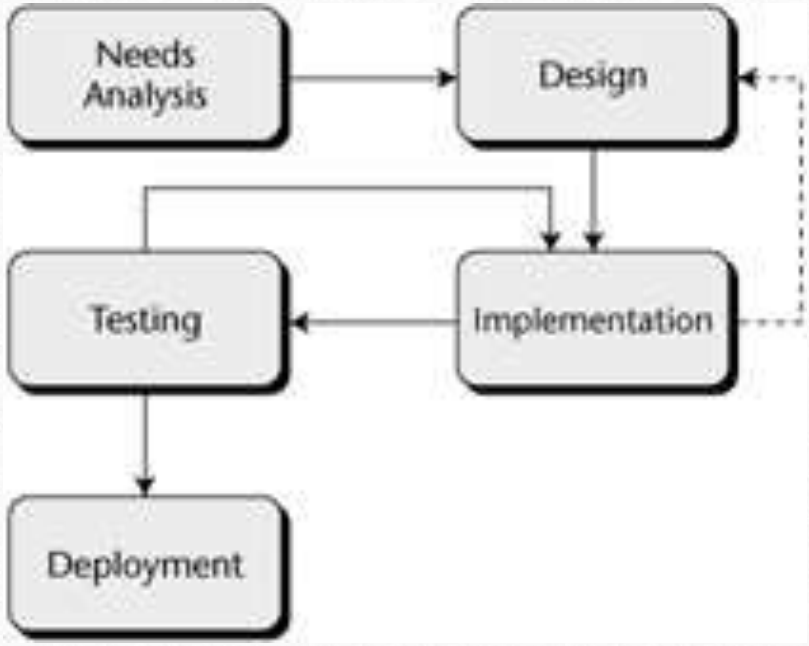
- A product of Research in Motion (RIM)
- Runs a proprietary multitasking OS

BlackBerry Development

- BlackBerry Web Development
- – The newest offering from RIM using the Widget SDK
- – BlackBerry Widgets are small, discrete, standalone web applications that use HTML/CSS/JavaScript

DESIGN-DEVELOPMENT-IMPLEMENTATION-TESTING-DEPLOYMENT PHASE

Smart Client Development



Application Goals

- Across the globe, companies are implementing mobile and wireless applications for a variety of reasons.
- In some cases, the goal is to increase productivity; in others, it is to reduce costs; while in still others, the reason may be simply that it seemed like the next logical thing to do, regardless of the return on the investment.

End User

- The end users of the application play a key role in the application design and rollout.
- Their level of technical ability determines some of the intricacies that may be required.
- If the user is not comfortable setting up wireless network connection parameters, such details will need to be automated within the application.
- Many mobile applications fail to reach their potential because they were not developed with the end user in mind.

Design Phase

- **Client Data Access**

- Enterprise data sources will contain much more data than user can hope to store on the mobile device.
- The amount of data stored on the device will depend on the mobile data store solution being used, device performance characteristics, and physical limitations of the device.
- When designing a smart client application, start by examining the subset of data that is required for the mobile user.
- This subset can be determined by looking at many factors, often depending on the type of application being developed.

Enterprise Integration

- Enterprise integration is a term used to describe any communication to systems not on the device.
- It encapsulates integration with enterprise databases, business applications, XML data, Web content, and legacy data, among other things.
- For the purpose of designing your mobile solution, you will need to determine to which enterprise systems you require access. There are two levels of integration that you may require: basic integration and complex integration.

Enterprise Integration

- Basic levels of enterprise integration include the ability to access enterprise databases using defined communication protocols. These capabilities may include the following:
- Device communication using the standard synchronization software such as HotSync for Palm devices or ActiveSync for Pocket PC devices
- Communication over IP-based networks
- Direct integration with a relational database or flat-file system
- Limited support for transactions

Enterprise Integration

- Support for a variety of mobile clients including laptops, handheld PCs, and PDAs
- Communication over networks that may not be IP-based
- Support for synchronizing multiple users simultaneously to a central back-end data store
- Communicating with systems that do not have well-defined interfaces, often requiring custom adapters
- Synchronization of complex data models

User Interface

- The user interface can account for as much as 80 percent of the total code in a mobile solution.
- When one part of the application accounts for such a large portion of the development effort, it has to be designed correctly to avoid costly changes later in the development cycle.

Screen Size

- When targeting mobile applications, there is a possibility to have a one-half VGA, one-quarter VGA, or even smaller screen to work.
- Using the screen to its maximum benefit is crucial for successful applications.
- There are many different ways to accomplish this goal, depending on the operating system you are using. Windows CE, for example, supports a tab-based interface, allowing easy navigation to multiple forms with the click of a button.

Human Interaction

- Studying human interaction with user application will prove to be invaluable in determining its overall usability.
- There is no way that these ways can be predicted, so testing is critical.
- If the application focuses on data input, then the main input screens should be easy to navigate using the input properties of the device.

Human Interaction

- For example, if the device offers keyboard support, make it possible for the user to quickly tab between entry fields, rather than having to use a scroll-wheel to get there.
- If the only means of input is a stylus with character recognition, then including radio buttons and drop-down lists are effective ways to improve the efficiency of data input.

Wireless Connectivity

- **Wireless networks operate over a variety of protocols.** Some of these are not IP-based, so if application requires IP for data communication, user may have to add an additional IP layer for connectivity.

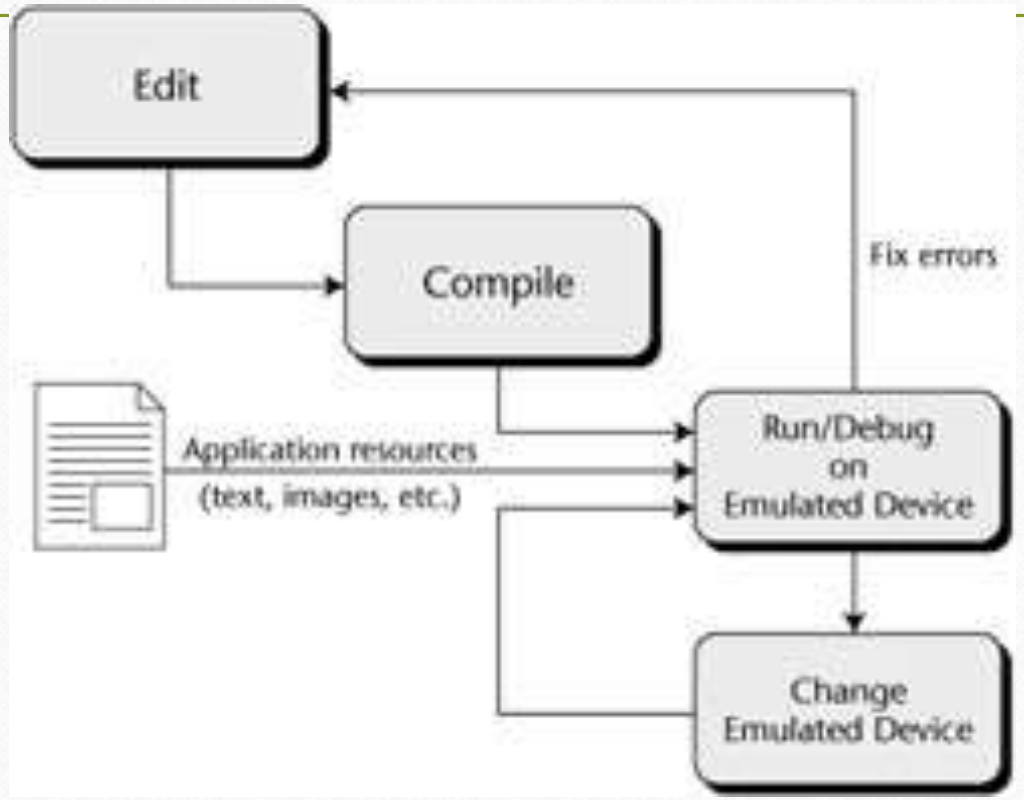
Wireless Connectivity

- **Wireless coverage is not guaranteed.** In fact, many areas do not have adequate coverage for data communication. This situation applies even in countries that have excellent overall wireless coverage.
- **Network penetration issues can arise even in areas that do have coverage.** Network penetration can be problematical not only in obvious places like subways and tunnels but also in many corporate buildings.

Wireless Connectivity

- **Limit the frequency of wireless data transfer.** Because there are potential problems with the reliability of wireless networks, this consideration will make the application more effective.
- Having suitable persistent data storage within the application will make it possible to limit the frequency of network connectivity within the application.
- **Limit the amount of data transferred.** This limitation is urged, once again because of the nature of the wireless networks.

Implementation and Testing Phase



Development Tools

- Choosing which development tool to use is not a decision to take lightly.
- Many development tools are available for smart client applications, each with benefits and drawbacks.
- Some of the more important ones include the target mobile operating system, the preferred programming language, the tool's feature set, and the tool's layout.

Deployment Phase

- **Wide range of devices that need to be supported.** These can range from two-way pagers to PDAs to laptops. Ideally, a way can be found to manage all of these devices in the same manner.
- **Deployment of applications to these devices.** This includes the original application as well as updates as they occur.
- **Management of mobile assets.** This includes keeping an inventory of the devices in the field, as well the software on these devices.

.

Deployment Phase

- **Backup and recovery.** Because the primary functions of many of the mobile applications are to retrieve data and to make sure the data is safe in the event of a system crash.
- **Working with wireless networks.** In most cases, the rules that were developed for LAN-based applications do not apply to wireless deployment and management. The bandwidth of the wireless networks makes the efficiency of the solution a top priority

Deployment Phase

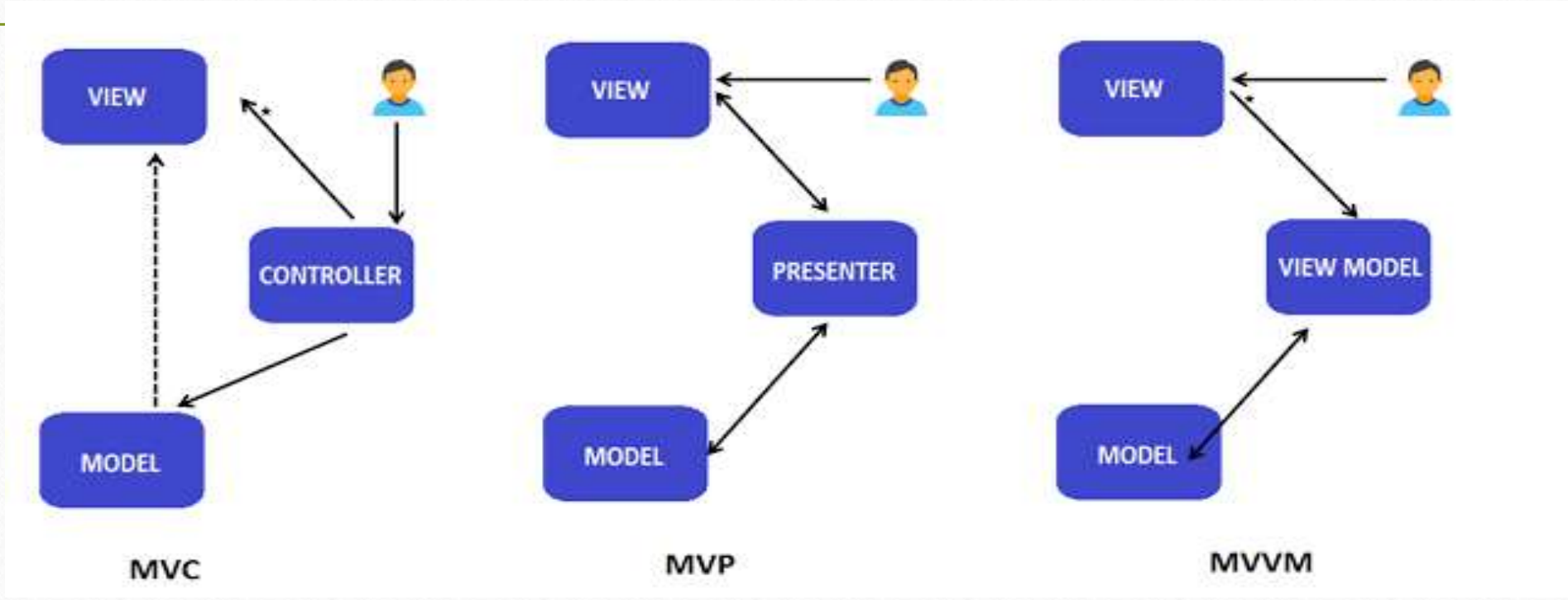
- **End users have little access to technical assistance.** Very often remote workers do not have direct access to technical support, making troubleshooting and repair difficult.
- **Participating in business analysis to determine application effectiveness.** In today's economic environment, businesses need ways to determine if their mobile applications are meeting their goals of improving employee productivity and cost savings.

MVVM MOBILE ARCHITECTURE DESIGN

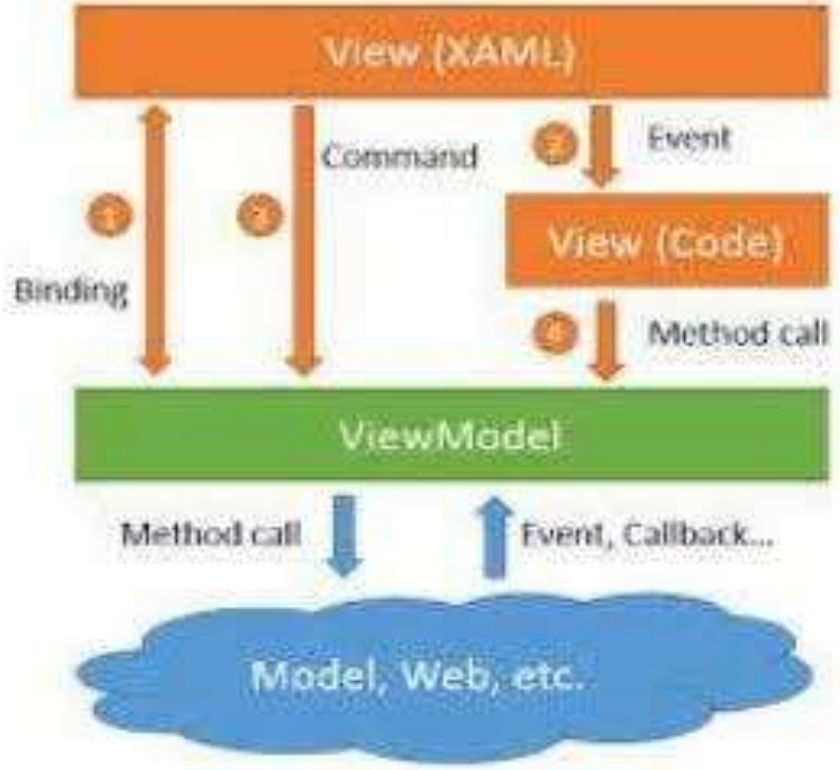
WHY MVVM…

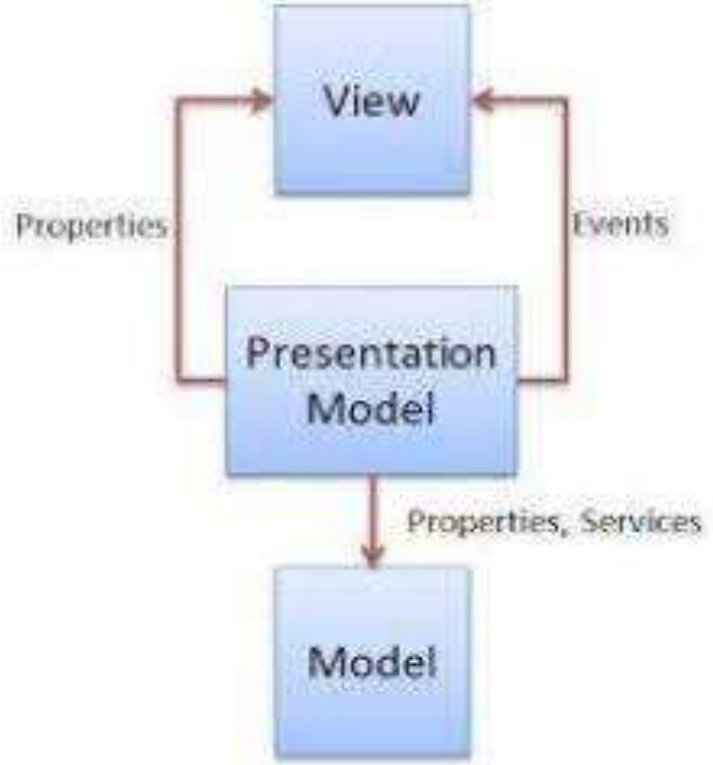
- Model-View-View-Model(MVVM) is a design pattern for building user interfaces.
- Was created in 2005 by John Gossman, the Microsoft WPF and Silverlight architect.
- Similar to MVC and MVP, we can say that MVVM is MVP specialization.

Model-View-View-Model…



MVVM Architecture





MVVM Architecture

- **VIEW:**

  The view is responsible for defining the structure, layout, and appearance of what the user sees on the screen.

- **MODEL:**

  Encapsulates the application’s business logic and data.

MVVM Architecture

- **VIEW MODEL:**

➢ The view model acts as an intermediary between the view and the model and is responsible for handling the view logic.

➢ The view model retrieves data from the model and them makes the data available to the view, and may reformat the data in some way that makes it simpler for the view to handle.

ADVANTAGES…

- Helps user to cleanly separate the business and presentation logic of your application from its user interface.
- Easier to test.
- Improve code reuse opportunities.

DISADVANTAGES…

- Bindings (but wait android did it well).
- In bigger cases, it can be hard to design the ViewModel.

REFERENCES….

- Martyn Mallick, “Mobile and Wireless Design Essentials” ,Wiley, 2003.
- https://www.slideshare.net/mobile/tamirk/smart-client-development
- https://www.slideserver.com/chimalsi/smart-client-architecture-patterns-designs-best-practices
- http://www.slideshare.net/mobile/CarolineKealey/message-architecture
- http://www.slideshare.net/mobile/javierhumaran/model-view-controller
- http://www.google.com/amp/s/www.geeksforgeeks.org/delegate-pattern
- http://www.slideshare.net/mobile/inovaeg/mvvm-presentation