



Module 9 Part 1

Mobile application architecture

BITS Pilani

Harvinder S Jabbal
SSZG653 Software Architectures

Architecting Mobile applications

Can you give some examples of mobile apps?

Mobile applications



Examples

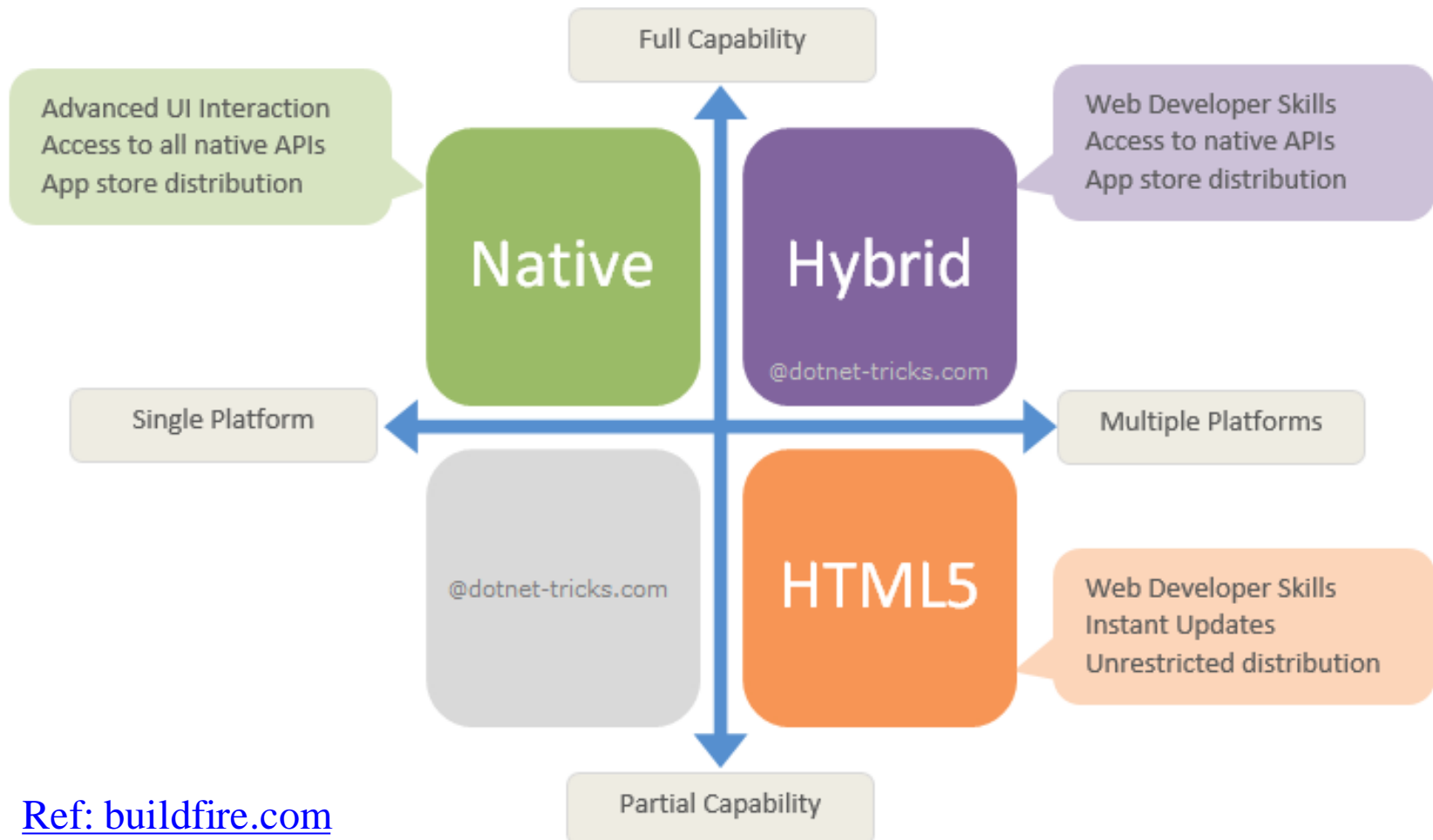
- Uber
- Swiggy
- Courier delivery
- eCom
- Banking
- Spotify
- Where Is My Train
- ...

Mobile Application: Design considerations



- Simple User interface: Easy to type, Large buttons, Minimal features, menu options, actions
- Responsive design: Adapt to different screen sizes & orientations
- Compact code: Less usage of CPU, memory, storage
- Few layers to ensure performance
- Connectivity: Store data locally and synchronize later if connection is poor

Types of Mobile Apps



[Ref: buildfire.com](http://buildfire.com)

Mobile Native app



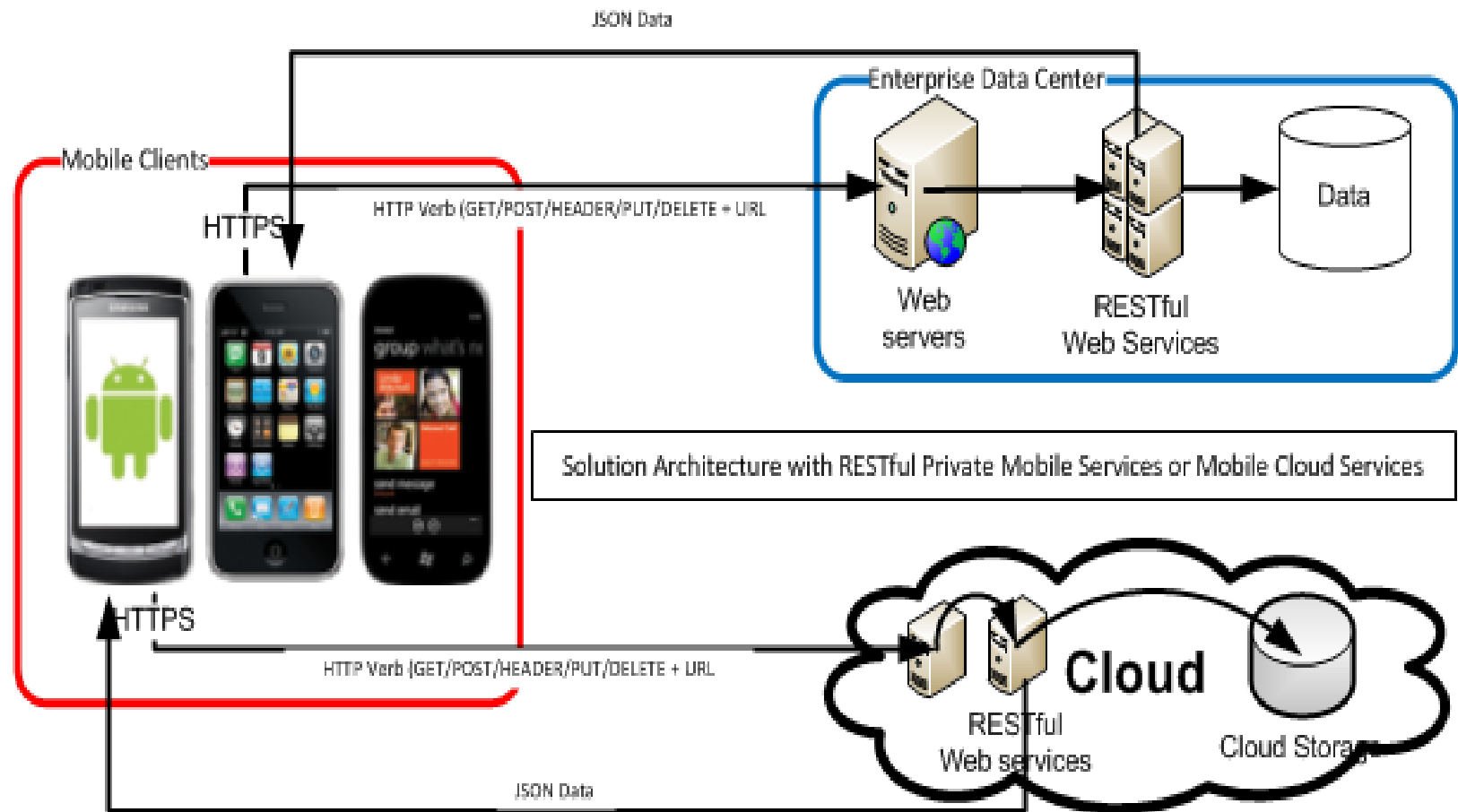
- They are built for specific platforms
 - Examples: Google Maps, Facebook, LinkedIn – one version for iOS and one for Android
- Languages used:
 - Native iOS : Swift or Objective-C
 - Native Android: Java or Kotlin
- Integrated Development Environment (IDE) such as Android studio are used for this

Mobile Web apps



- Progressive web apps use modern web technology to deliver app-like experiences to users, right in their browsers.
- Examples:
 - Flipkart
 - BookMyShow
 - MakeMyTrip
- Uses HTML5, CSS3, JavaScript and runs on a browser

Mobile web application



Hybrid app



- A hybrid app combines elements of both [native apps](#) and [web applications](#).
 - Examples: Twitter, Uber, Instagram
- Hybrid apps are essentially web apps (HTML, CSS, Javascript) that have been put in a native app [shell](#).
- The shell is able to connect to native capabilities of the mobile platform such as camera, accelerometer, GPS, etc.
- Tools such as Xamarin and React Native allows app to run across platforms

Pros & Cons



App type	Pro	Con
Native	<ul style="list-style-type: none">• High performance• Superior user experience• Access to all features of OS	<ul style="list-style-type: none">• Runs only on one platform• Need to know special language• Need to update versions
Web App	<ul style="list-style-type: none">• Easy to deploy new versions• Common code base	<ul style="list-style-type: none">• Little scope to use device hardware• Lower user experience• Need to search for app
Hybrid	<ul style="list-style-type: none">• Does not need browser• Single code base• Access to device hardware	<ul style="list-style-type: none">• Slower

UI design patterns



- Action bars for quick access to frequently used actions
- Login using Facebook, Google, etc. instead of separate user id / password
- Large buttons for ease of use
- Notifications of recent activity
- Discoverable controls: Controls show up only when an item is selected (ex. In WhatsApp, the Forward button shows up when a message is selected)

Mobile optimized web site



All features



Reduced features

DECCAN HERALD
DH E paper | Prajavani | PV E Paper | Sudha | Mayura | The Printers Mysore | DH Classifieds

Home | News | Karnataka | Bengaluru | Business | Budget 2018 | Supplements | Sports | Entertainment

Three soldiers killed as avalanche hits army post in Kashmir | Bad luck follows the 10 rupee coin | Swaraj, Ne



U-19 World Cup final: India limit Australia to 216 after brilliant bowling show



National health scheme to cost govt Rs 12K cr a year

November 2, 2024

For 2018-19, the government has made an initial provision of Rs 2,000 crore



Ahead of Modi's rally, BJP taps into techie support base



Bad luck follows the 10 rupee coin



Karnataka to launch universal health coverage scheme by Feb end



Cow Bill debate see subtle sarcasm to passionate arguments



Ramya turns social media teacher for Youth Cong workers

- Muslims opposing Ram temple must SEZ 605/ US Zia War Software Architectures
- Mehbooba says she can accept going to hell to save Kashmir

DH **DECCAN HERALD** GET APP

TOP STORIES NATION CITY BUDGET 2018

National health scheme to cost govt Rs 12K cr a year



The Union government is likely to pay an annual premium of less than Rs 1,200 per family for the ambitious national health protection scheme, for which approximately Rs 12,000 crore...

Bandh called off after High Court terms it illegal

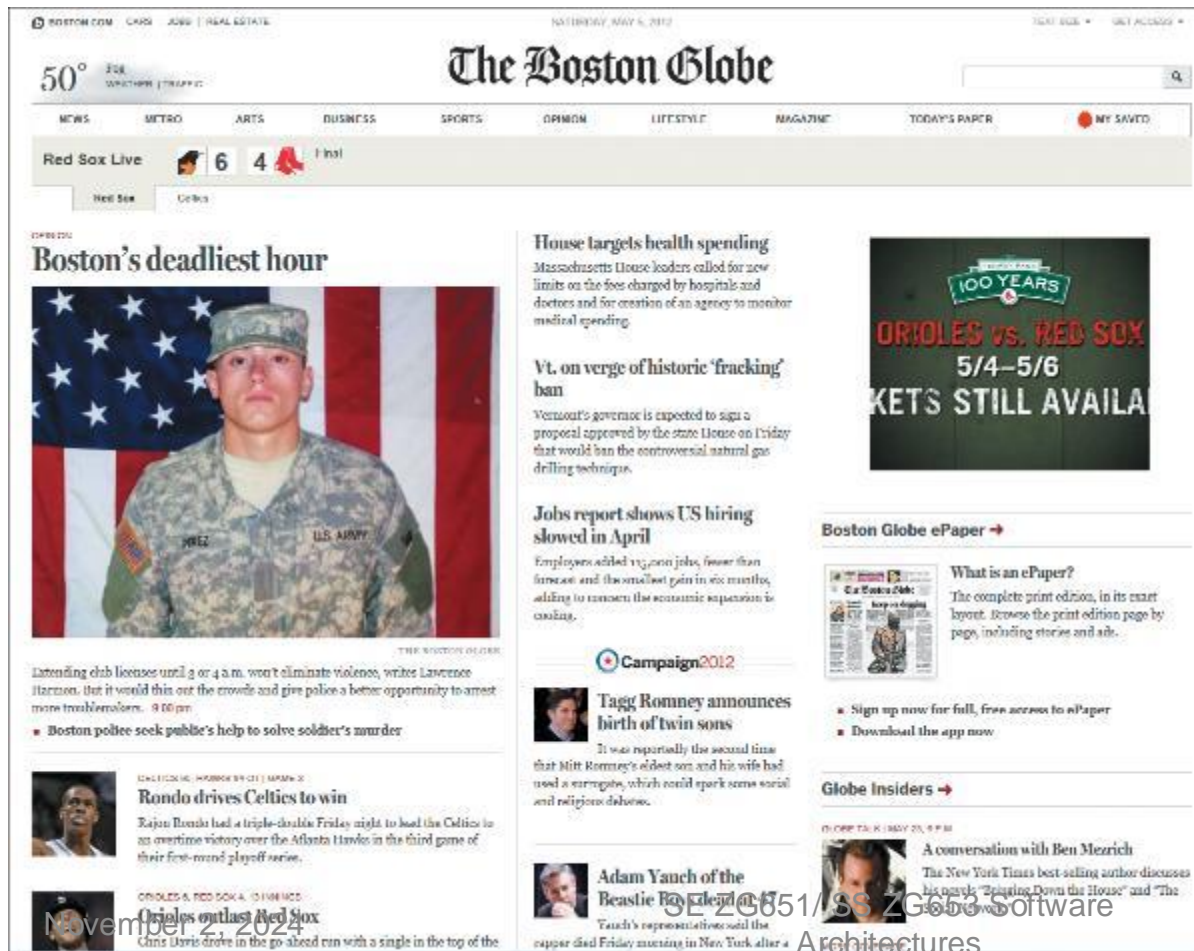
Responsive design

innovate

achieve

lead

Example: Boston Globe News



This is how
the display
looks on
desktop

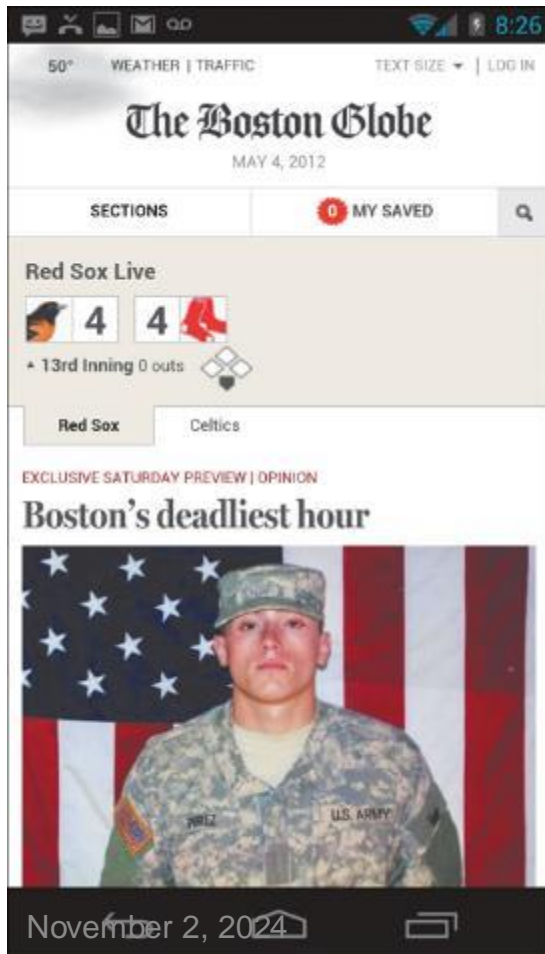
Responsive design

innovate

achieve

lead

Example: Boston Globe News



This is how
the display
looks on
mobile

Adjusted
vertically

Responsive design



Single website for laptop & mobile & tablet

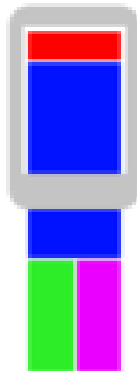
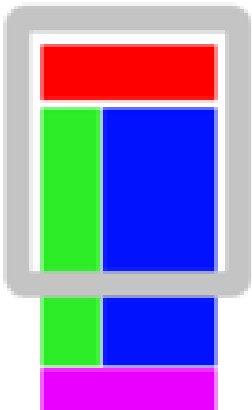
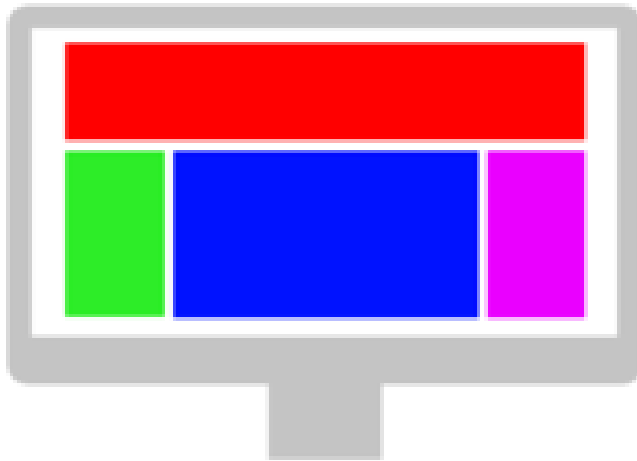
Principle: Adapt rendering depending on screen sizes & orientation



Responsive design



Ref: Wikipedia



Flexi grids

- Divide a screen into multiple columns
- Assign HTML elements to one or more columns
- Choose a different layout depending on screen size

Responsive design

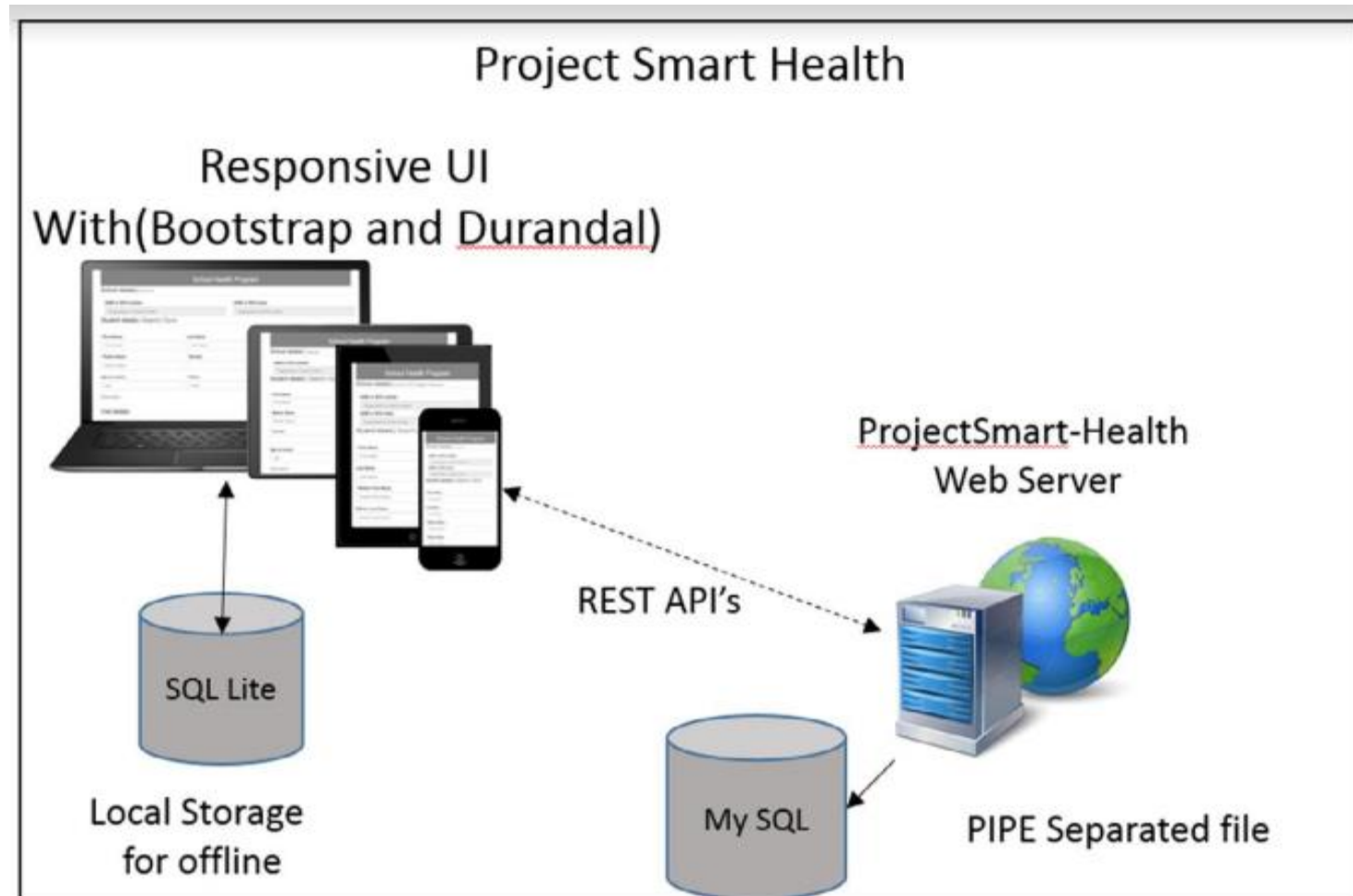


Technique

- Use CSS and HTML to shrink, hide or move content
- Flexible grids (CSS 3)
 - Use media queries to determine screen size
 - Specify grid width as % of screen size rather than fixed pixels
- Flexible images – Specify image size as % of grid size

Store locally, Sync later

In case of intermittent connectivity



Doctors enter patient data in mobile, which gets synced with server later

Mobile Application Architecture



Android application architecture

4 types of components

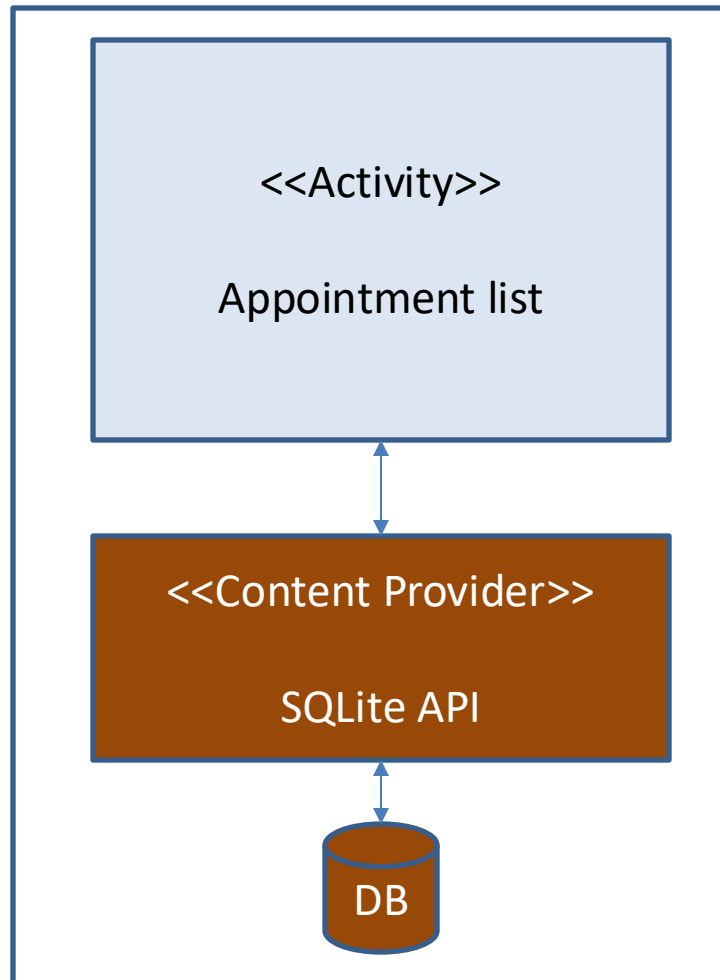
- Activity (UI)
- Service (background process) - ex. playing music, download
- Content provider (Storage) - ex. SQLite, files,
- Broadcast receiver (Acts on events received from OS and other apps) - Ex arrival of SMS

Examples of components

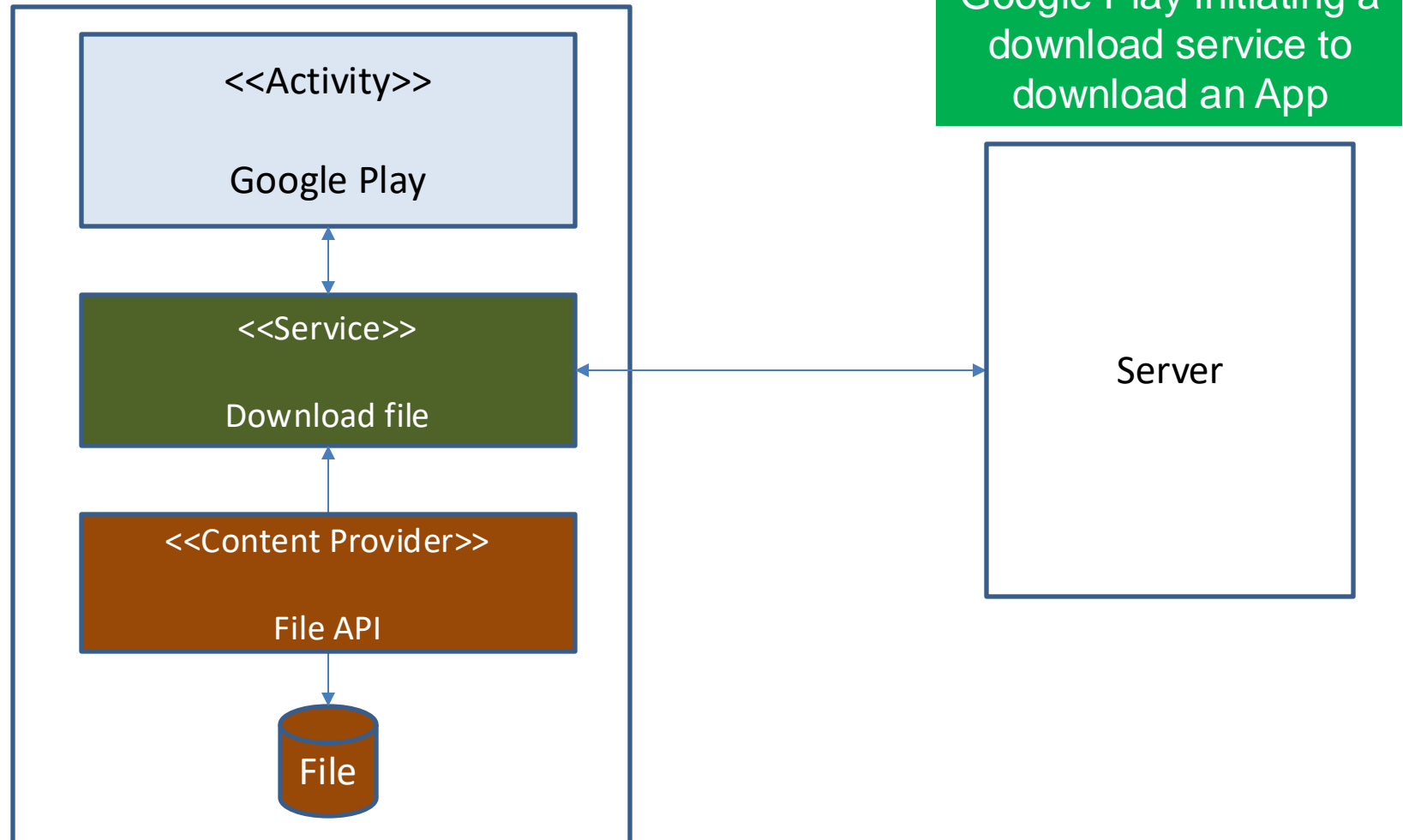
Activity & Content providers



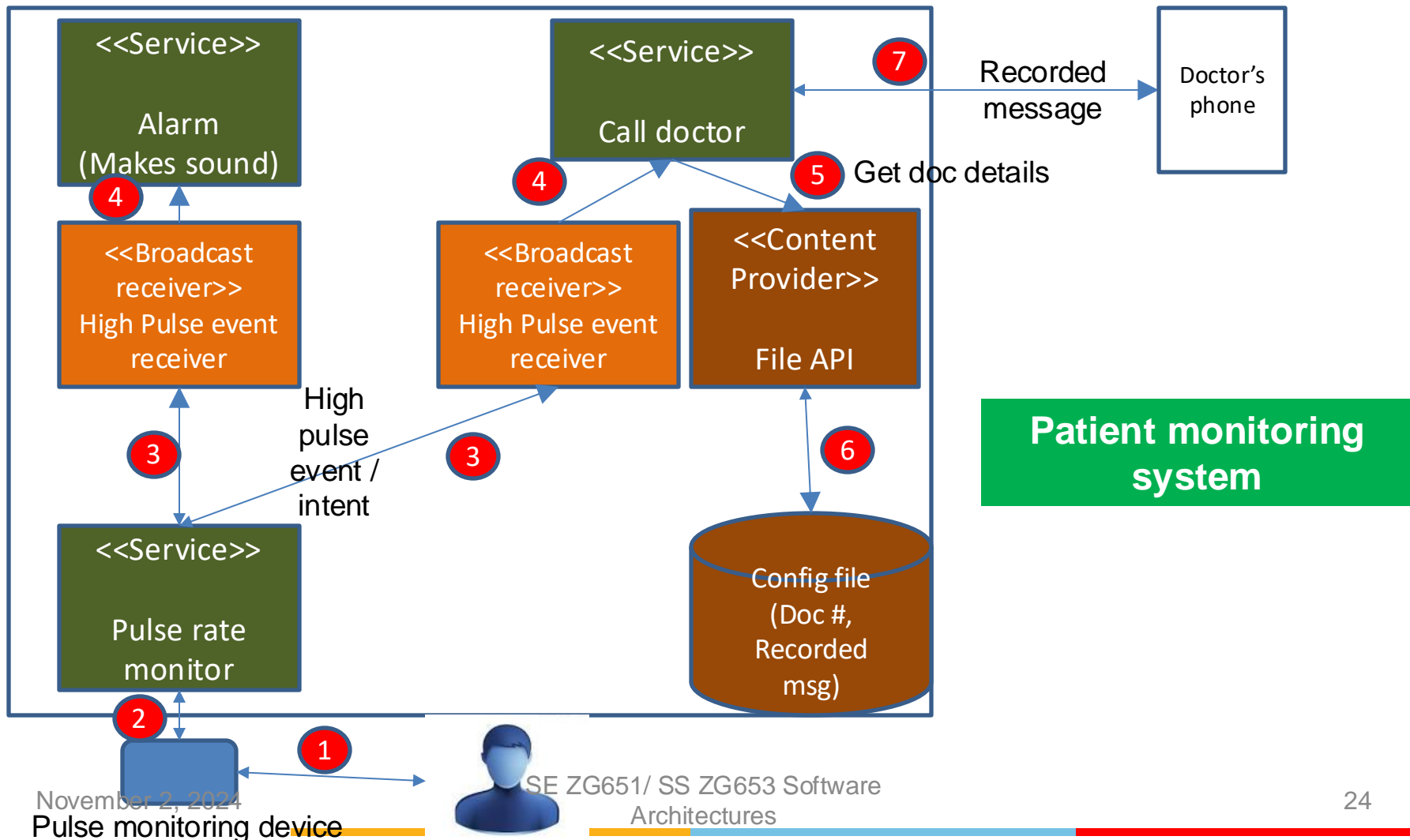
Personal Calendar App



(Background) Service



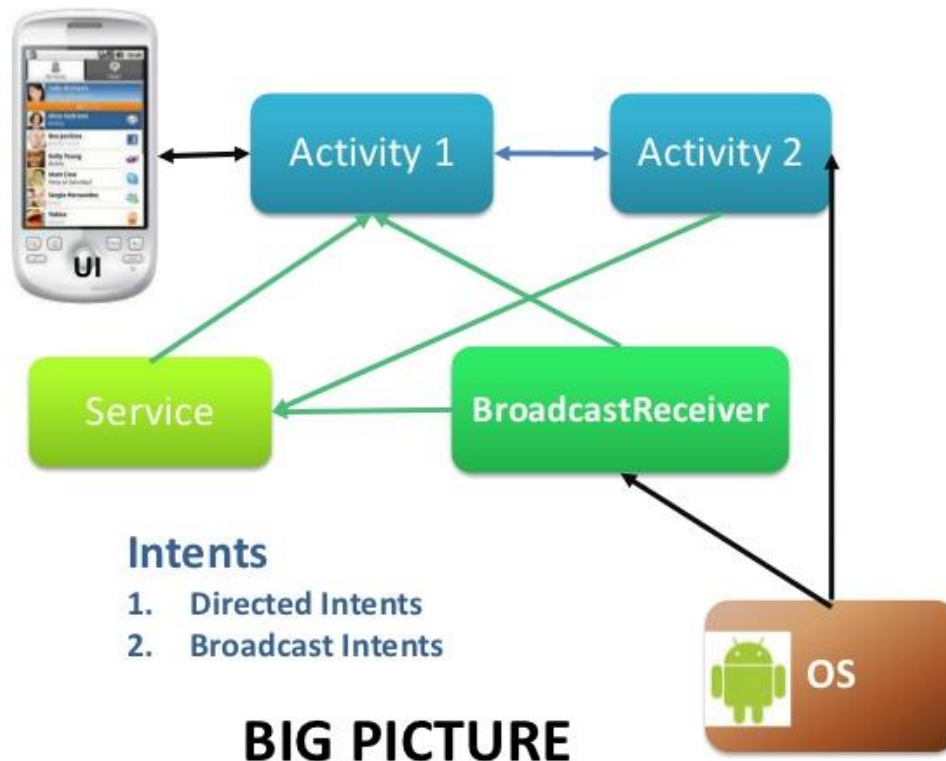
Broadcast receiver (Event handlers)



Android application structure



Android Application Anatomy



Exercise: Mobile app components



Give examples of mobile app components in **Uber app**.

UI

Screen to book a cab

Broadcast receiver

Receive location of cab from backend server and provide to UI for display

Service

Provide cab location to backend server after journey starts

Database

Configuration file containing user data

Exercise



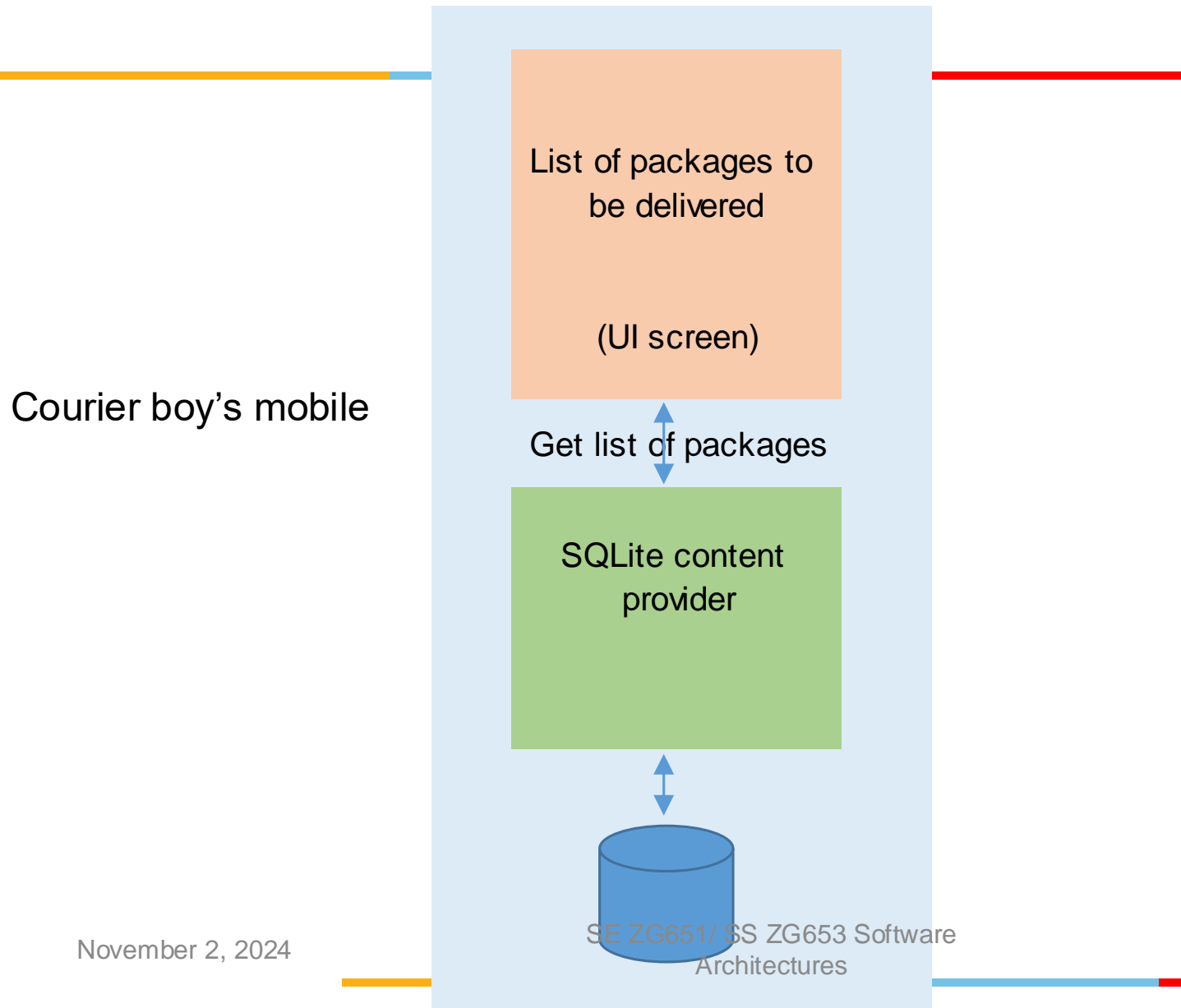
Consider a mobile app carried by the courier delivery boy.

The app should support the following functions:

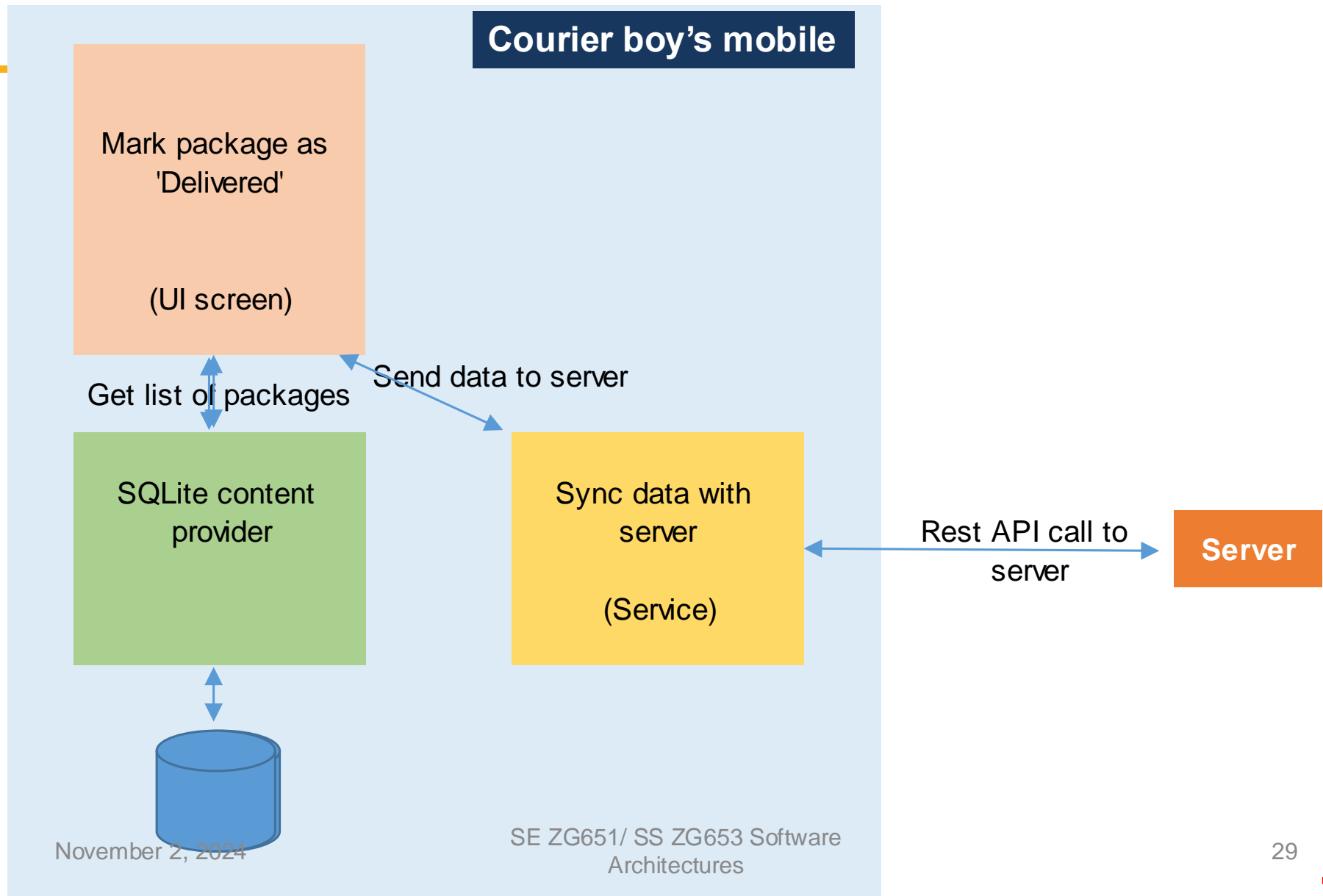
- a) View courier packages to be delivered
- b) Mark a package as delivered.
- c) Upon this event, the app should send information to central server

Identify the components of a mobile app & its inter-connections and draw an appropriate software architecture diagram

a) View courier packages to be delivered



b) Mark a package as delivered. Upon this event, the app sends information to central server



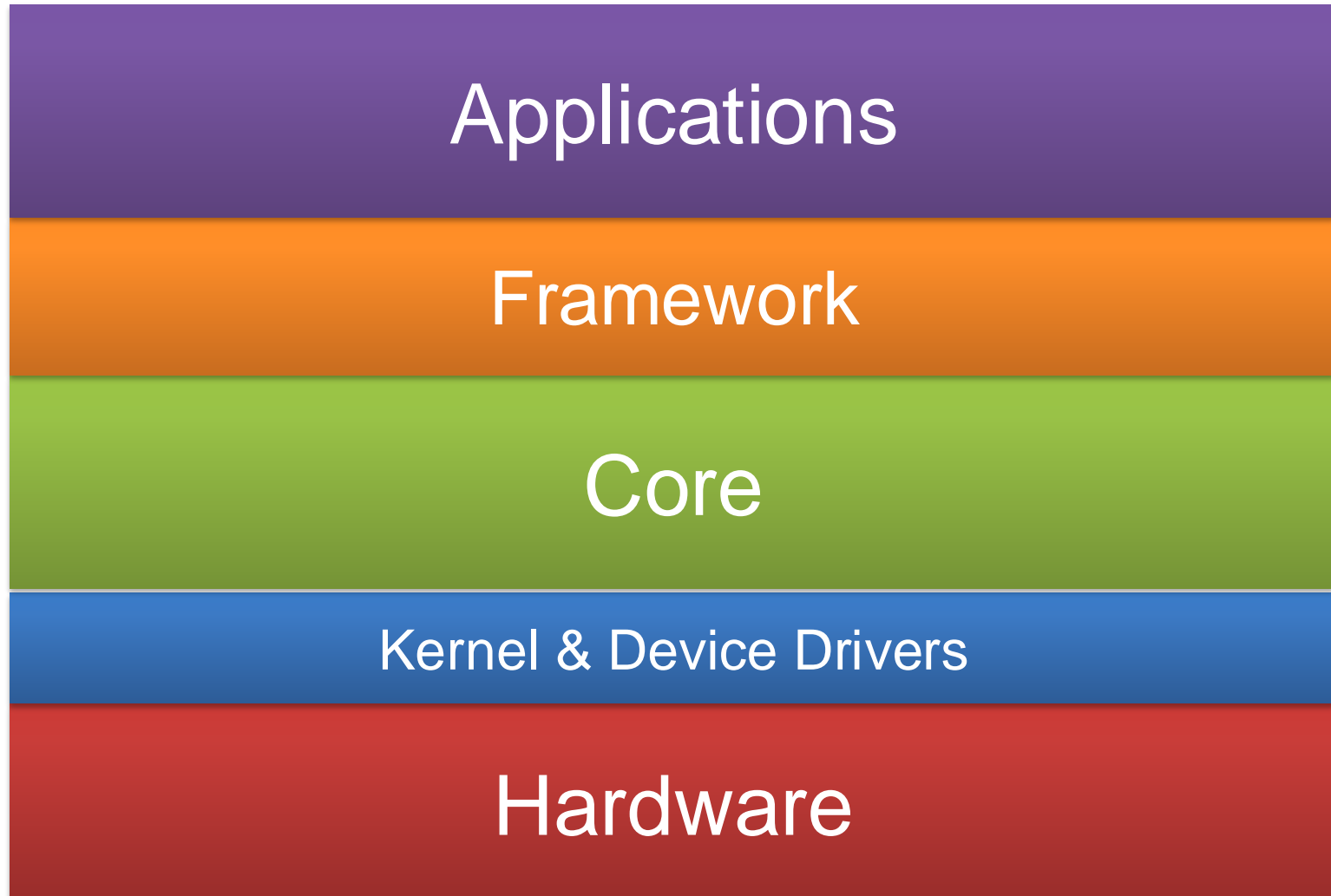
Review questions



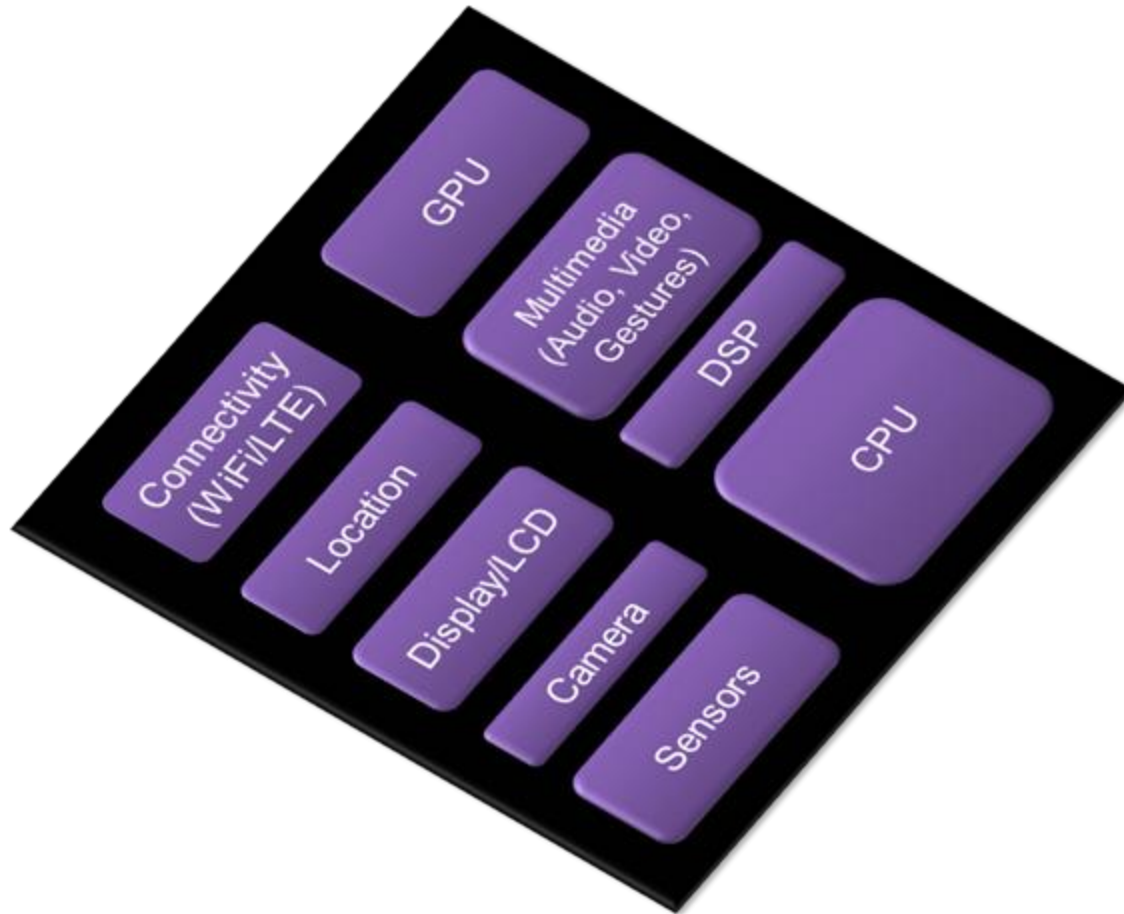
Mobile apps

1. What is a cross platform mobile app?
2. If connectivity is poor, how do we ensure consistency between data in mobile phone and backend server?
3. What is 'Broadcast receiver' in an Android app?

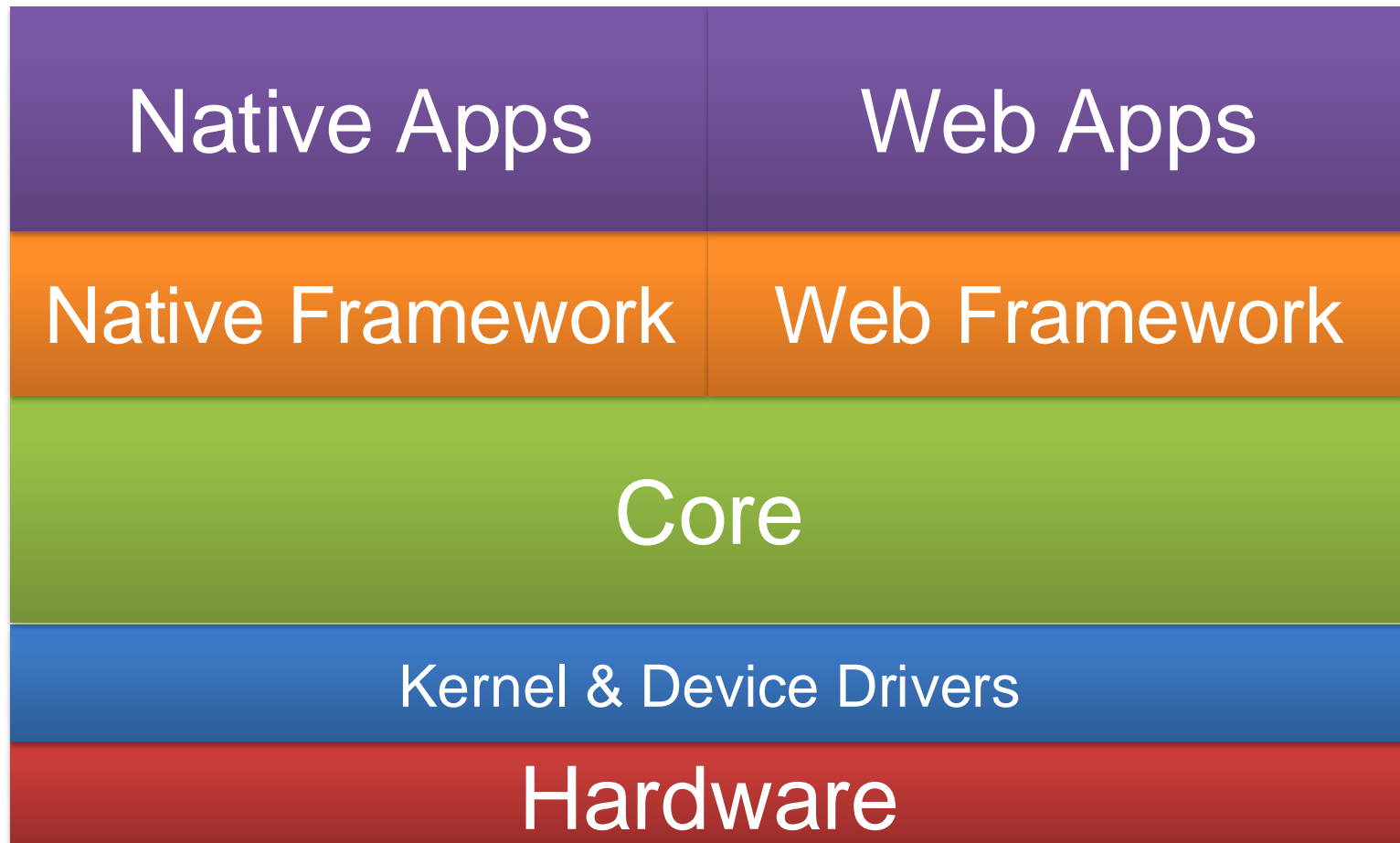
Mobile - Software Platform



Mobile – Hardware Platform



Typical Software Platform



Typical Software Platforms



iOS

- Mobile Operating System by Apple
- Multitasking
- Supports iPhone, iPad and other devices

Windows

- Windows Phone OS by Microsoft
- Multitasking
- Features Metro (Modern UI, Tiled UI)

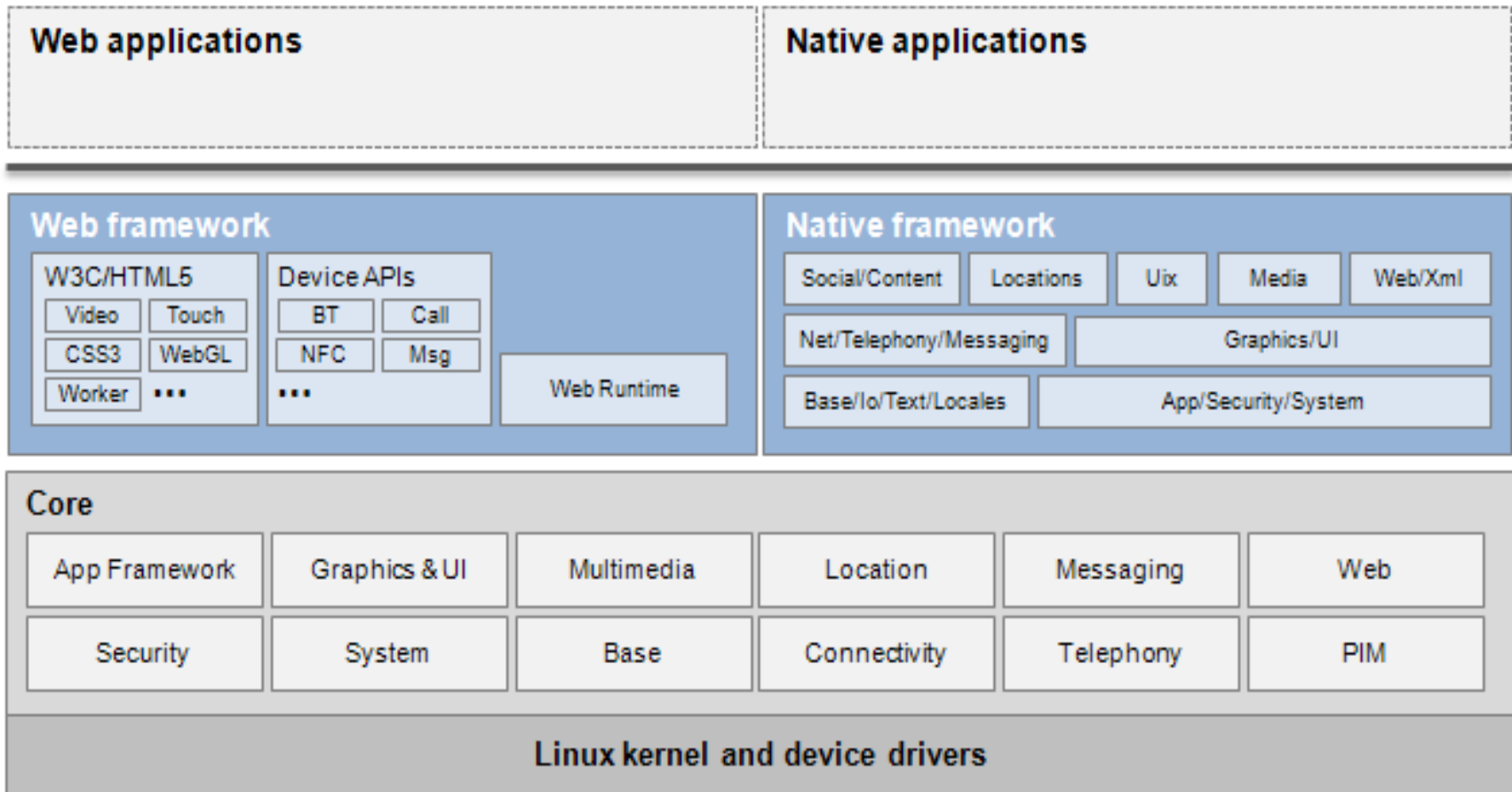
Android

- Google's Mobile OS based on Linux Kernel
- Multitasking
- Software Stack

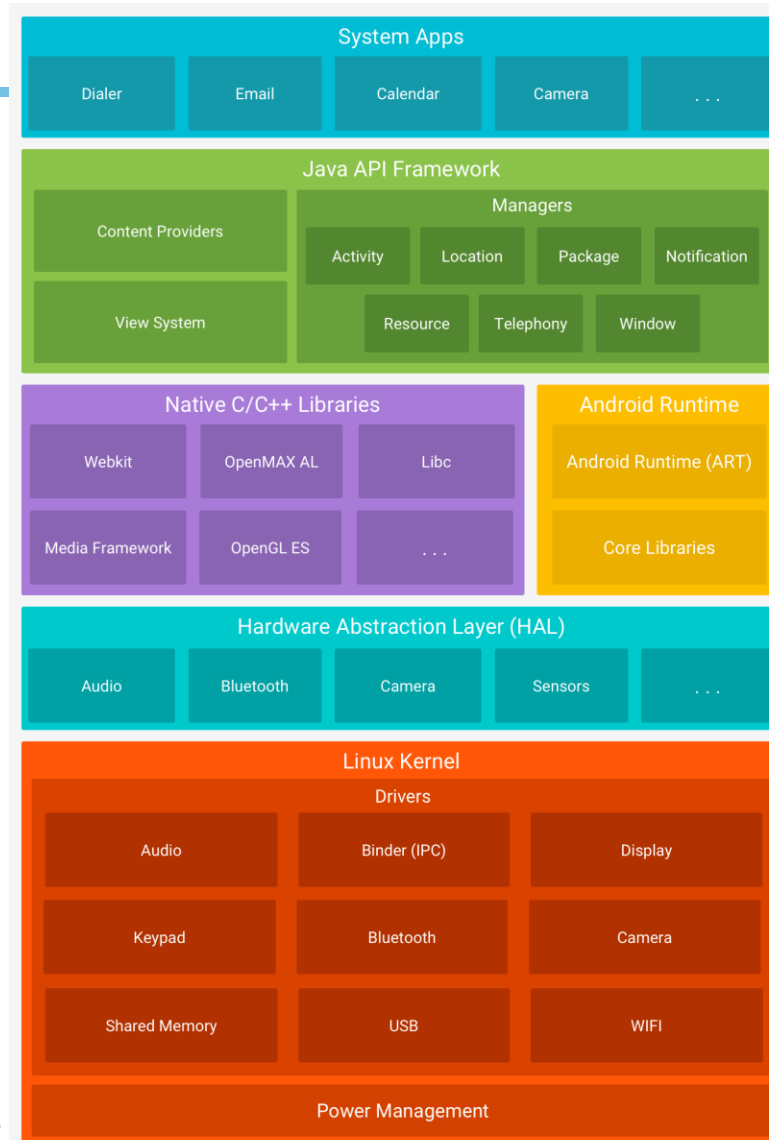
Tizen

- An open source standards based platform
- Based on Linux, part of the Linux Foundation
- OS with Multiple profiles

Tizen Architecture



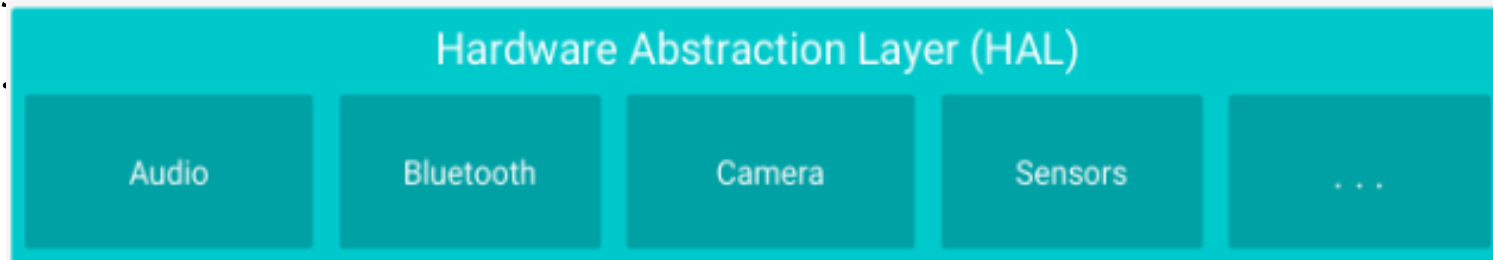
Android Stack



Hardware Abstraction Layer (HAL)

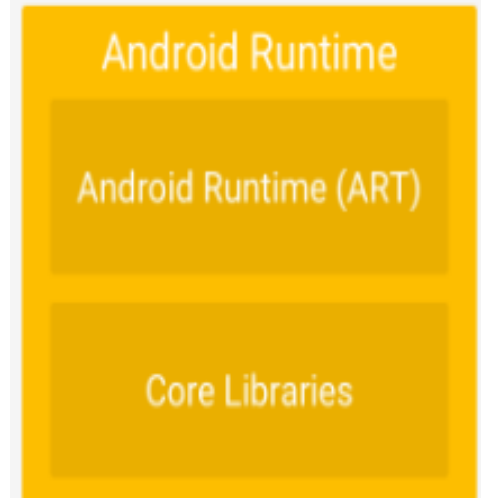


- ❑ HAL provides standard interfaces that expose device hardware capabilities to the higher-level Java API framework.
- ❑ The HAL consists of multiple library modules, each of which implements an interface for specific hardware components, such as the camera or Bluetooth module.
- ❑ When a framework API makes a call to access device hardware, the Android system loads the library module for that hardware component.



Android Runtime (ART).

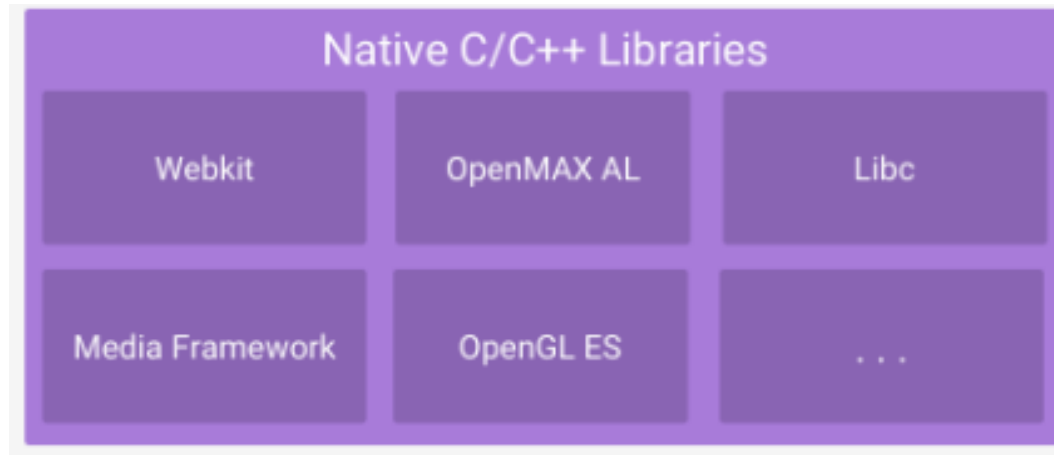
- ❑ Each app runs in its own process and with its own instance of the Android Runtime (ART).
- ❑ ART is written to run multiple virtual machines on low-memory devices by executing DEX files, a bytecode format designed especially for Android that's optimized for minimal memory footprint.



Android Runtime

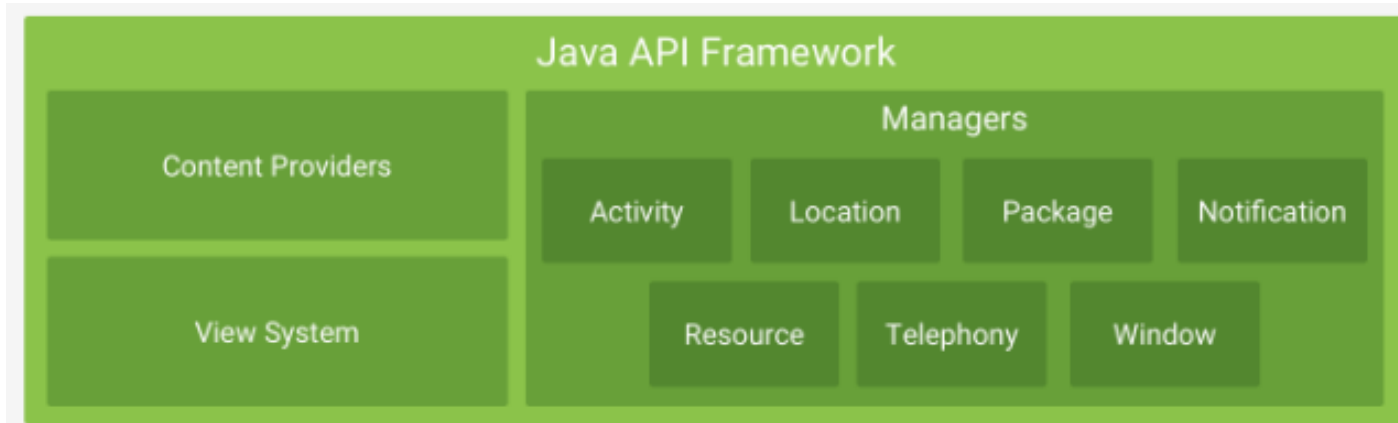
- Include core libraries and the Dalvik VM
- Engine that powers the applications
- Forms the basis of the application framework
- Dalvik VM is a register-based VM to ensure the device can run multiple instances efficiently

Native C/C++ Libraries



- ❑ Many core Android system components and services, such as ART and HAL, are built from native code that requires native libraries written in C and C++.
- ❑ The Android platform provides Java framework APIs to expose the functionality of some of these native libraries to apps

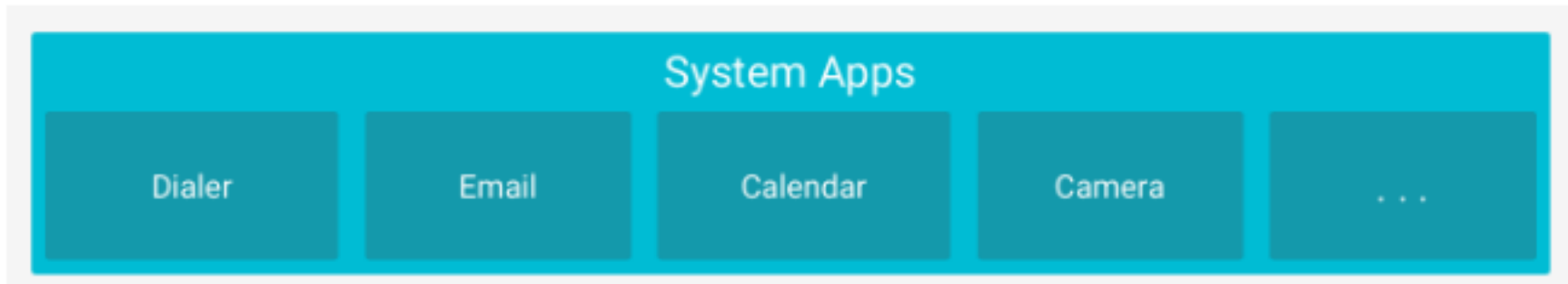
Java API Framework



- ❑ The entire feature set of the Android OS is available to you through APIs written in the Java language.
- ❑ These APIs form the building blocks you need to create Android apps by simplifying the reuse of core, modular system components, and services,

System Apps

- ❑ Android comes with a set of core apps for email, SMS messaging, calendars, internet browsing, contacts, and more.
- ❑ Apps included with the platform have no special status among the apps the user chooses to install.
- ❑ So, a third-party app can become the user's default web browser, SMS messenger, or even the default keyboard (some exceptions apply, such as the system's Settings app).



Linux Kernel



Linux Kernel

Drivers

Audio

Binder (IPC)

Display

Keypad

Bluetooth

Camera

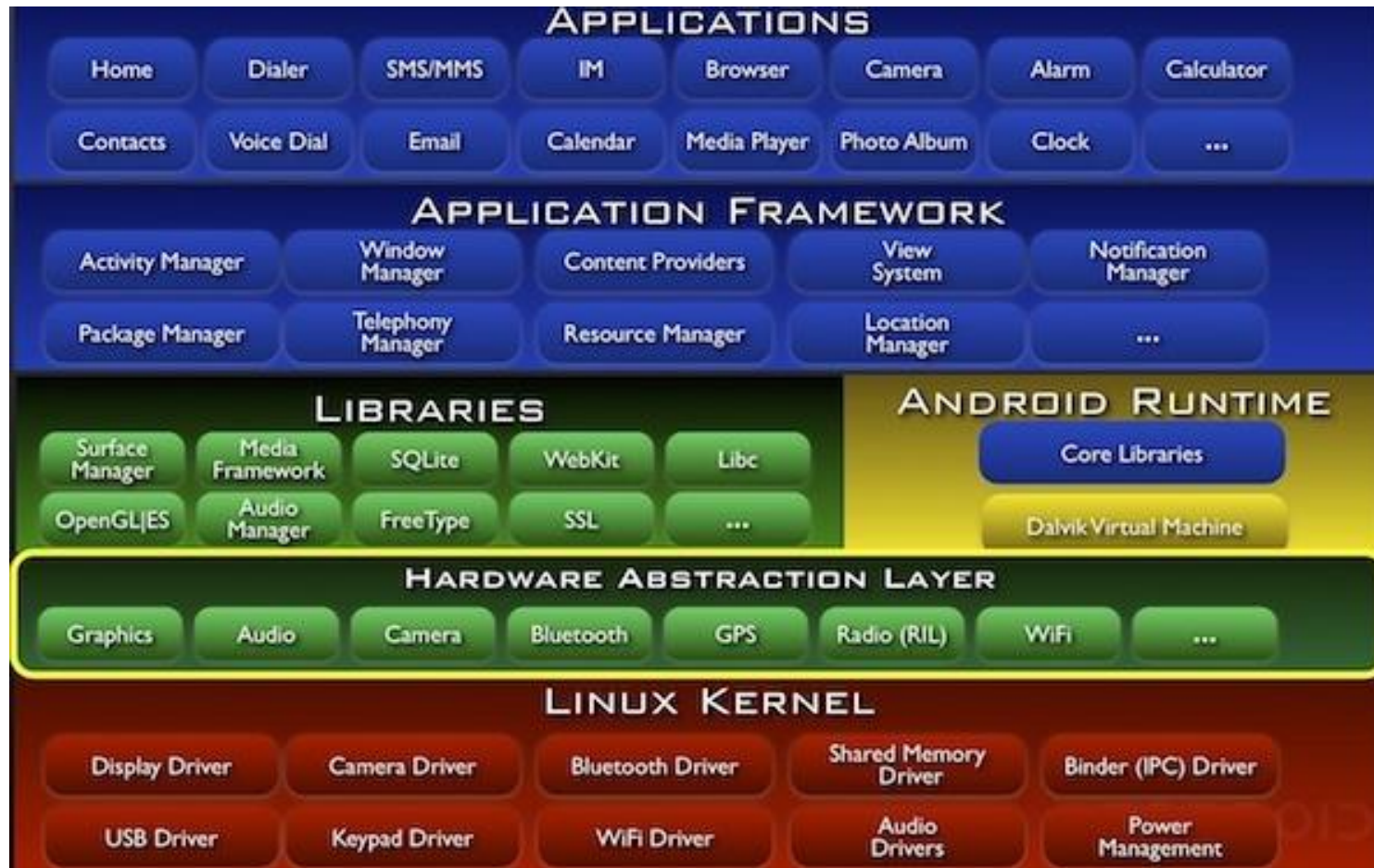
Shared Memory

USB

WIFI

Power Management

Android Architecture

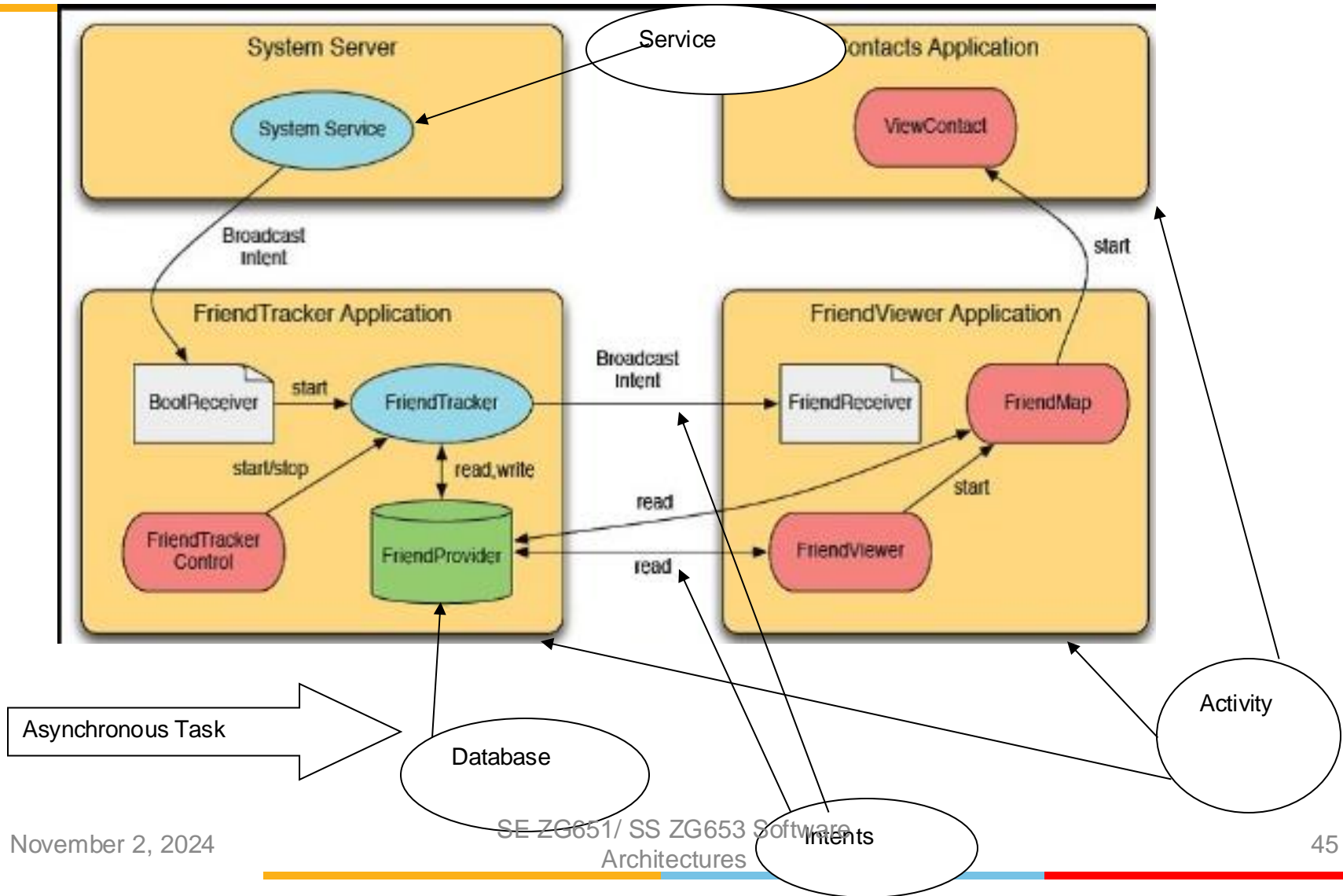


Popular services of cloud vendors



Google App Engine	Microsoft Azure	Amazon Web Services
<ul style="list-style-type: none">• Python & Java development environment• Auto scaling• Database replication	<ul style="list-style-type: none">• .Net environment• Auto scaling• Load balancing• Failure detection & auto replication of services• Database replication	<ul style="list-style-type: none">• Java development• Auto scaling• Load balancing• Failure detection & auto replication of services• Database replication• Data caching (Memcached)• Service discovery (SoA)• Notification of events (SNS)• Message queue (SQS)• CDN (Content Delivery Network) – YouTube
November 2, 2024	SE ZG651/ SS ZG653 Software Architectures	44

Mobile app: Example



Amazon's approach to handling issues in distributed systems



- To scale you have to partition, so you are left with choosing either high consistency or high availability for a particular system. You must find the right overlap of availability and consistency.
- Choose a specific approach based on the needs of the service.
- For the checkout process you always want to honor requests to add items to a shopping cart because it's revenue producing. In this case you choose high availability. Errors are hidden from the customer and sorted out later.
- When a customer submits an order you favor consistency because several services--credit card processing, shipping and handling, reporting--are simultaneously accessing the data.

Issues in Cloud based systems



- Availability
 - Cloud vendors promise high availability ex. 99.95%
 - But still not 100%
 - Need to design for the 0.05%
 - One approach: Store same data in different geographical zones as done by Netflix



Scenario

A start-up company is developing a GST tax returns filing system. This software will be deployed on the Cloud and offered to small and medium businesses as a SaaS.

The clients will have to input their data or upload data using an Excel file. They also need to provide other details such as GST #, etc. The software will process the data and file the returns into the Government's GST system on behalf of the client.

After developing the software, what options exist for deploying it in the cloud?

Different ways to deploy applications on the cloud



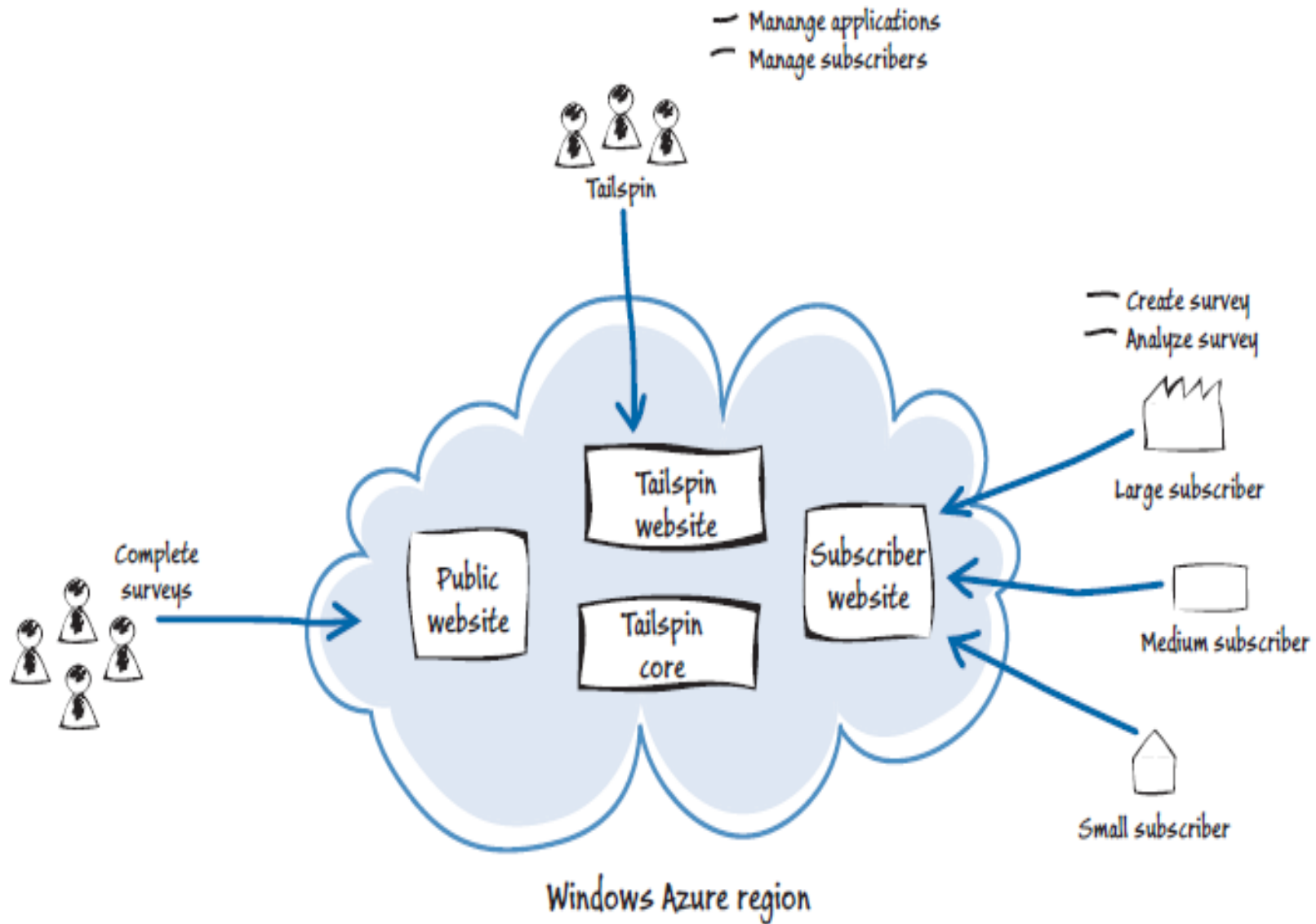
Scenario

A start-up company – Tailspin - is developing a **customer survey & analysis** application. This software will be deployed on the Cloud and offered to clients as a SaaS.

- The clients (subscriber of the application) can create and launch a survey.
- After this the survey participants will access the application and answer the survey questions.
- After the data has been gathered the application will perform analysis and present the results to the client organization

What options exist for Tailspin to deploy the application on the cloud?

Context diagram



November 2, 2024

FIGURE 1

The Surveys application

SE ZG651/ SS ZG653 Software
Architectures

Multi-tenant application Architecture – High level

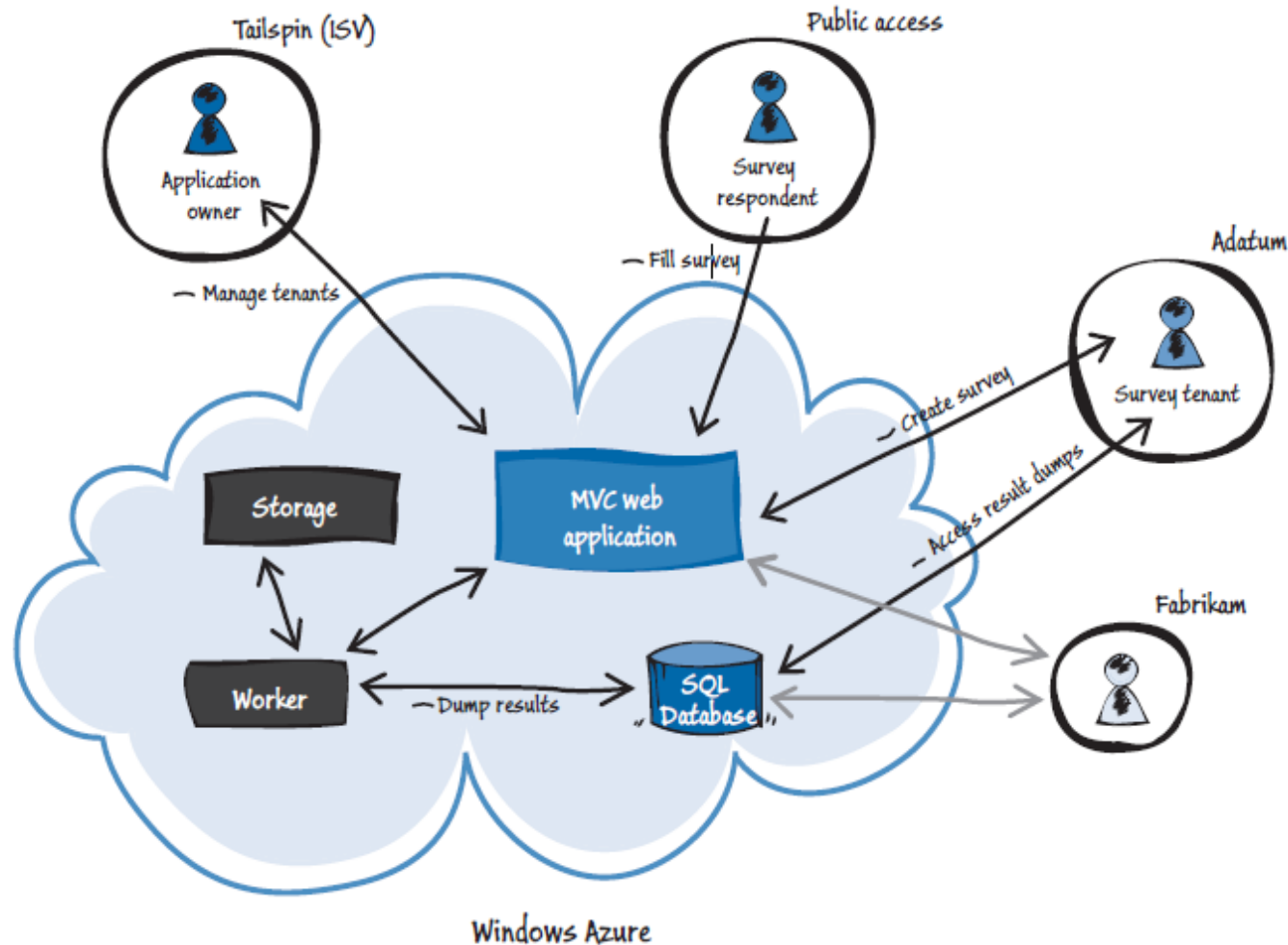


FIGURE 2

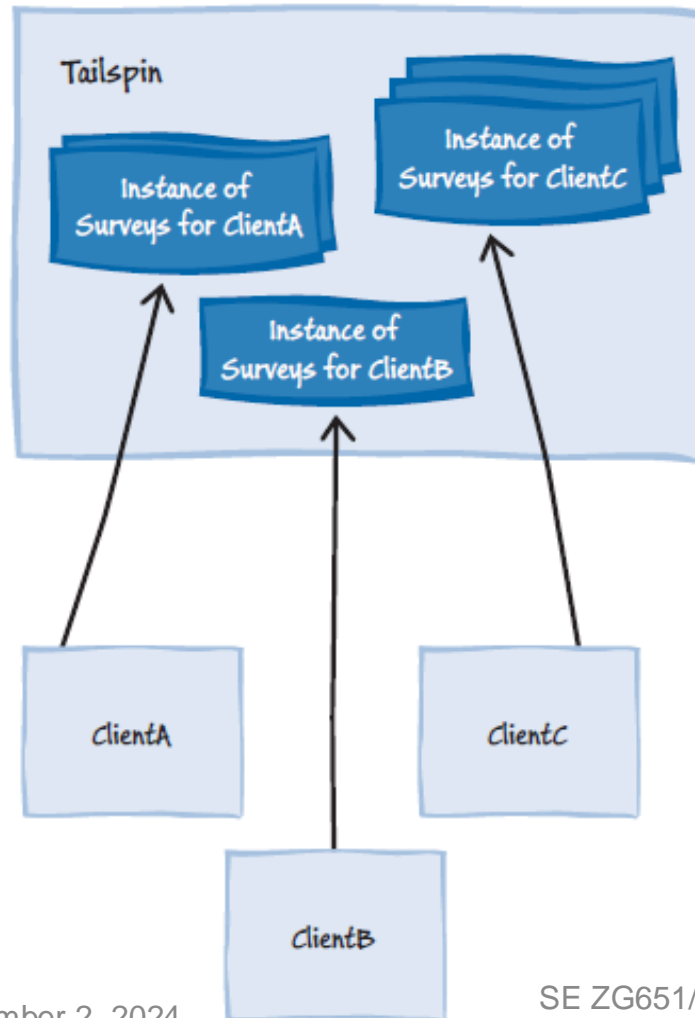
November 2, 2024 The Surveys application architecture

SE ZG651/ SS ZG653 Software Architectures

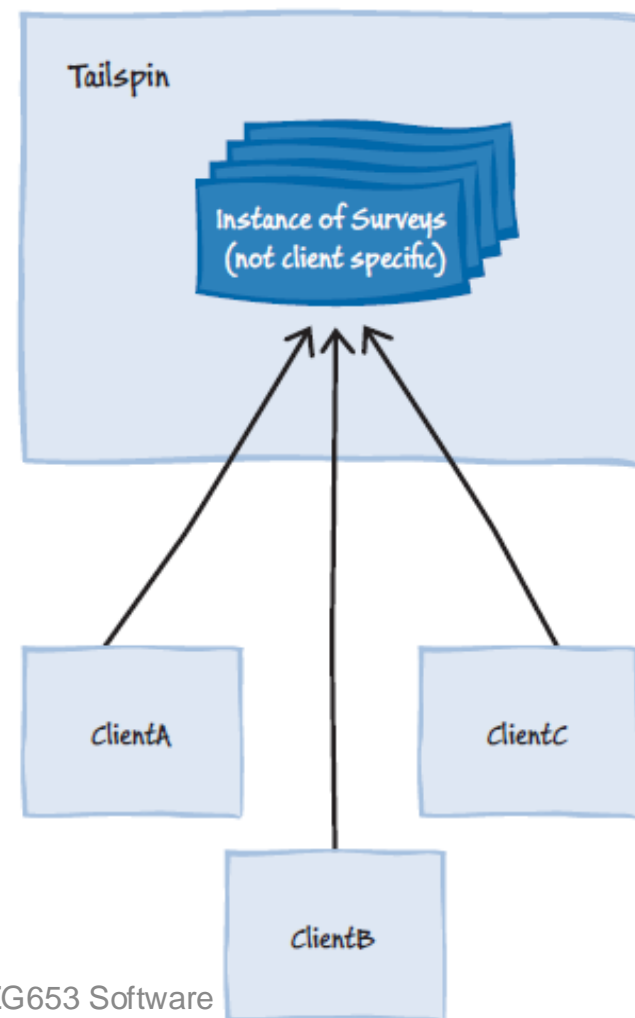
Architecture Options – High level



Multi-instance, single tenant



Single instance, multi-tenant



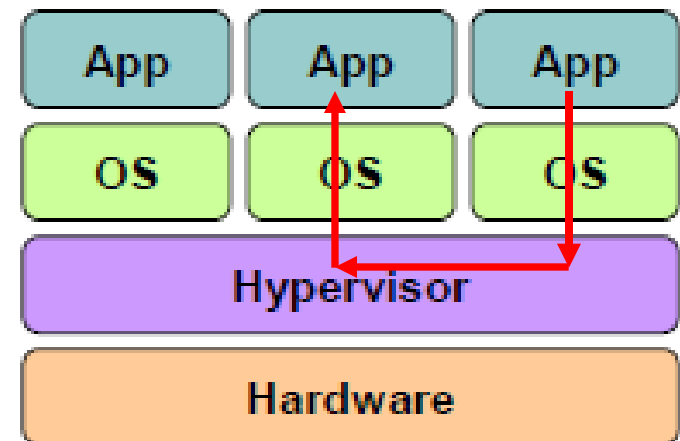
Issues in Cloud based systems



- Security issue due to multi-tenancy
 - Poor design leading to inadvertent sharing of information
 - Virtual machine 'Escape'

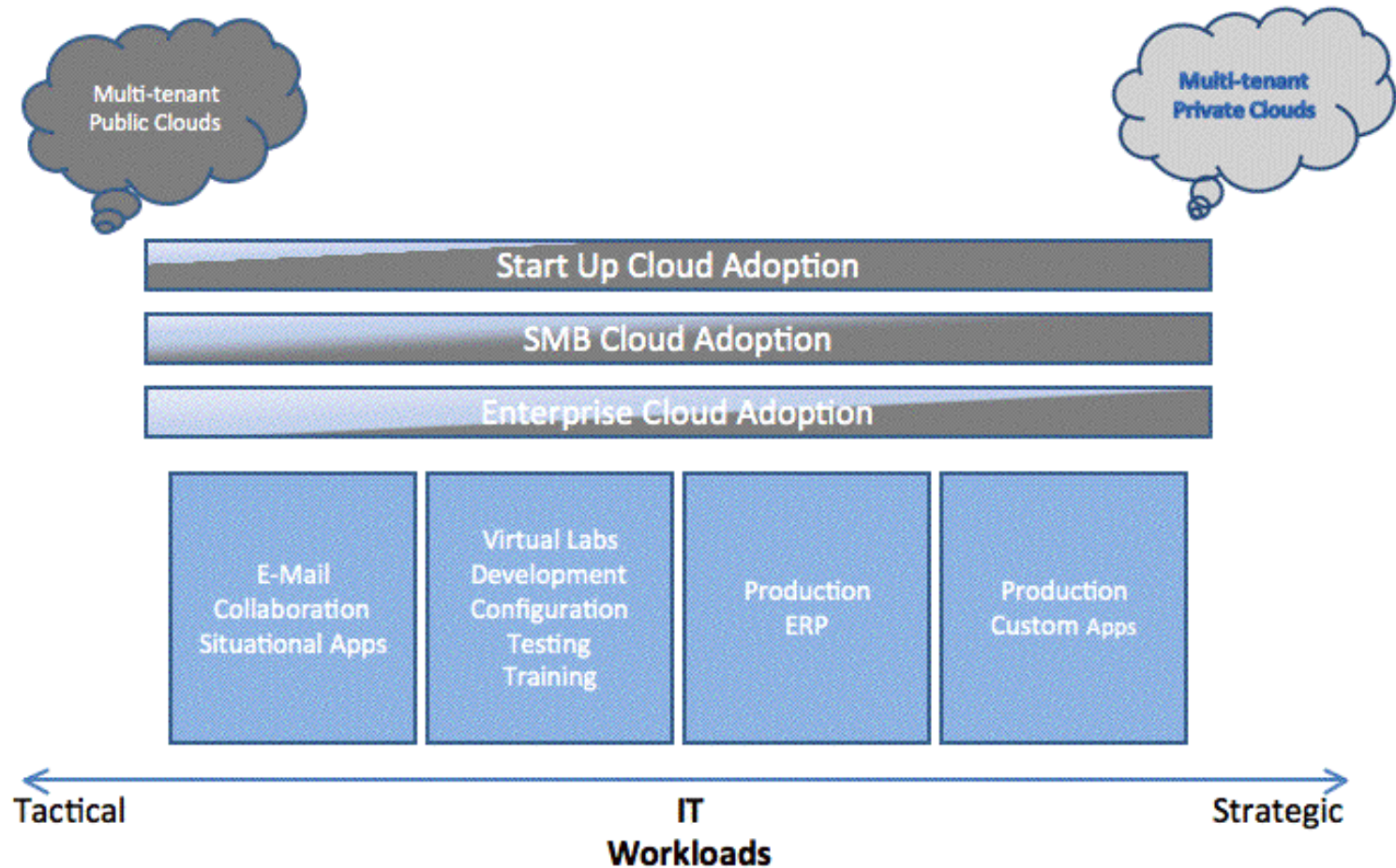
Other fields of the table					OU_ID
					Org1
					Org1
					Org1
					Org2
					Org2

Same table contains data of different organizations



Virtual machine 'Escape'

How to choose your multi-tenancy degree?



IT workloads will reside in a hybrid environment of public and private clouds. Both will be multi-tenant.

How to choose your multi-tenancy degree?



The **characteristics of the workload** in question have to be carefully studied first, including the workload's **utilitarian versus strategic value, volatility, security, etc.**

Higher degrees of multi-tenancy are best suited for cross-industry utilitarian workloads such as **e-mail, expense reporting, travel authorization and sales force management.**

These applications can very easily share the **same schema.**

<https://www.computerworld.com/article/2517005/data-center/multi-tenancy-in-the-cloud--why-it-matters.html>

Degree of multi-tenancy



- Highest degree: IaaS and PaaS are multi-tenant. SaaS is fully multi-tenant also.
- Middle degree: IaaS and PaaS are multi-tenant. Small SaaS clusters are multi-tenant.
- Lowest degree: IaaS and PaaS are multi-tenant. SaaS is single tenant.

'Cloud Native' applications



- Cloud-native computing takes advantage of many modern techniques, including
 - PaaS,
 - multicloud,
 - microservices,
 - agile methodology,
 - containers,
 - CI/CD, and
 - devops



What is cloud
native?

Reference: <https://www.infoworld.com/article/3281046/what-is-cloud-native-the-modern-way-to-develop-software.html>

Exercise



When would you use the following options:

a) App 1

Web tier – Multi-tenant

App tier – Single tenant

Data tier – Multi-tenant

b) App 2

Web tier – Single tenant

App tier – Multi-tenant

Data tier – Single tenant

Exercise



When would you use the following options:

a) App 1

Web tier – Multi-tenant

App tier – Single tenant

Data tier – Multi-tenant

Answer: Web tier processing is light, App processing is heavy, data is not confidential

b) App 2

Web tier – Single tenant

App tier – Multi-tenant

Data tier – Single tenant

Answer: Web tier processing is heavy, App processing is light, data is confidential

Mobile Application Architecture



Types of mobile apps	Characteristics
Native app	Makes use of OS and native devices. Ex. Games
Cross platform app	Same code runs on multiple mobile platforms such as Android and iOS
Mobile web application	Has a mobile component which interacts with a server component. Ex. Uber, PayTM, Banking

Tools and technology for mobile app development



Development Approach	Native	Cross-Mobile Platforms	Mobile Web
Definition and Tools	Build the app using native frameworks: <ul style="list-style-type: none">- iPhone SDK- Android SDK- Windows Phone SDK	Build once, deploy on multiple platforms as native apps: <ul style="list-style-type: none">- RhoMobile- Titanium Appcelerator- PhoneGap- Worklight- Etc.	Build using web technologies: <ul style="list-style-type: none">- HTML5- Sencha- JQuery Mobile- Etc.
Underlying Technology	<ul style="list-style-type: none">- iPhone: Objective C- Android: Java- Windows Phone: .NET	<ul style="list-style-type: none">- RhoMobile: Ruby on Rails- Appcelerator: Javascript, HTML- PhoneGap: Javascript, HTML- Worklight: Javascript, HTML	<ul style="list-style-type: none">- Javascript, HTML

Xamarin – cross platform tool

Thank you

