

IEMS 5722

Mobile Network Programming and Distributed Server Architecture

Lecture 1

Course Introduction

Lecturer: Albert C. M. Au Yeung

14th January, 2016

Overview of This Lecture

Course Administration

- Course details
- Course schedule
- Assessment Schemes
- Policies and Rules

Course Content Overview

- Computer networking
- Mobile network programming
- Distributed server architecture
- Introduction to Android programming

Course Instructors

Albert Au Yeung – Lecturer

- Email: cmauyeung@ie.cuhk.edu.hk
- For lecture content, materials, details of assignments, project arrangements, reference materials, etc.

Marco Leung – Teaching Assistant

- Email: mtleung@cuhk.edu.hk
- Contact Marco if you need specific help when working on your assignments and project

Note: Both of us do NOT have regular office hours on campus, email us first whenever you need help.

Course Details

Venue & Time

- YIA LT9
- 7:00pm – 9:30pm
- 12 Lectures + 1 Project Presentation Session (28th April, 2016)

Jan 2015						
S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Feb 2015						
S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29					

Mar 2015						
S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Apr 2015						
S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Course Schedule

Our 12 Lectures will following (roughly) the schedule below:

Lecture	Date	Content	Assignments
1	14 / 01	Course Introduction	Assignment 0
2	21 / 01	Android Programming	Assignment 1
3	28 / 01	Data Communications & Client-Server Architecture	
4	04 / 02	Multi-threading & HTTP Networking in Android	
5	11 / 02	Web and Application Servers	Assignment 2
6	18 / 02	Database and Caching	
7	25 / 02	Instant Messaging & Google Cloud Messaging	Assignment 3
8	03 / 03	Peer-to-Peer Networking in Android	
9	10 / 03	Asynchronous Tasks & Message Queues	
10	17 / 03	Cloud Computing & APIs	Assignment 4
11	24 / 03	Socket.io	
12	31 / 03	Advanced Android Programming	Project Plan
13	28 / 04	Project Demonstration	

Assessment Scheme

- **10% - Attendance**

(From Lecture 2 to Lecture 12)

- **10% - Short Essay**

(Write a 2-page essay on an Android app that you find interesting)

- **40% - Programming Assignment**

(Android and server applications development)

- **40% - Course Project**

(On top of the app developed in the assignments, develop new features and functions that requires data communication over the network)

Assessment Scheme

40% - Programming Assignment

- Assignments involve **Android** and **Python** programming
- Using the **Android Studio** environment is recommended
- You are required to submit the source code for marking
- We will cover some Android programming and network programming
- You will have to **re-use** your assignment codes in your project
- A total of **5** assignments (2%, 8%, 8%, 10%, 12%)

Assessment Scheme

40% - Course Project

- You are going to build on top of the app developed in your assignment
- You will give a **presentation** of your project (in PPT and/or as a demo) in the last lecture
- A **brief report** should be handed in after the presentation (in PPT format)

Assignments & Project

In the **assignments**, you will:

- Learn how to develop an Android app
- Learn how to develop a server application in Python that supports the app
- Learn how to build an instant messaging app using various components and services

In the **project**, you will:

- Build on top the app in the assignment, and add interesting features to it (e.g. group chat, send and receive images, files, audio clips; mini games; NFC for adding new friends, etc.)

Notices

Take this Course?

Take this course if you:

- Have basic understanding of **Java programming**
- Would like to learn how to become a **full stack developer** of mobile applications
- Interested in mobile app development on **Android**, and in Web applications using **Python**
- Would like to challenge yourself with interesting programming problems

Some Rules

What you should do in this course?

- Attend the **lectures**, and raise questions whenever you have any
- Seek help as early as possible (e.g. if you have difficulties in picking up Java/Android programming, or if you cannot set up the development kit)
- Feel free to make **suggestions** to the course and/or lectures
- Do your own assignments, and do NOT make your work publicly available before the deadline
- All late submissions of assignments will **receive 30% penalty**

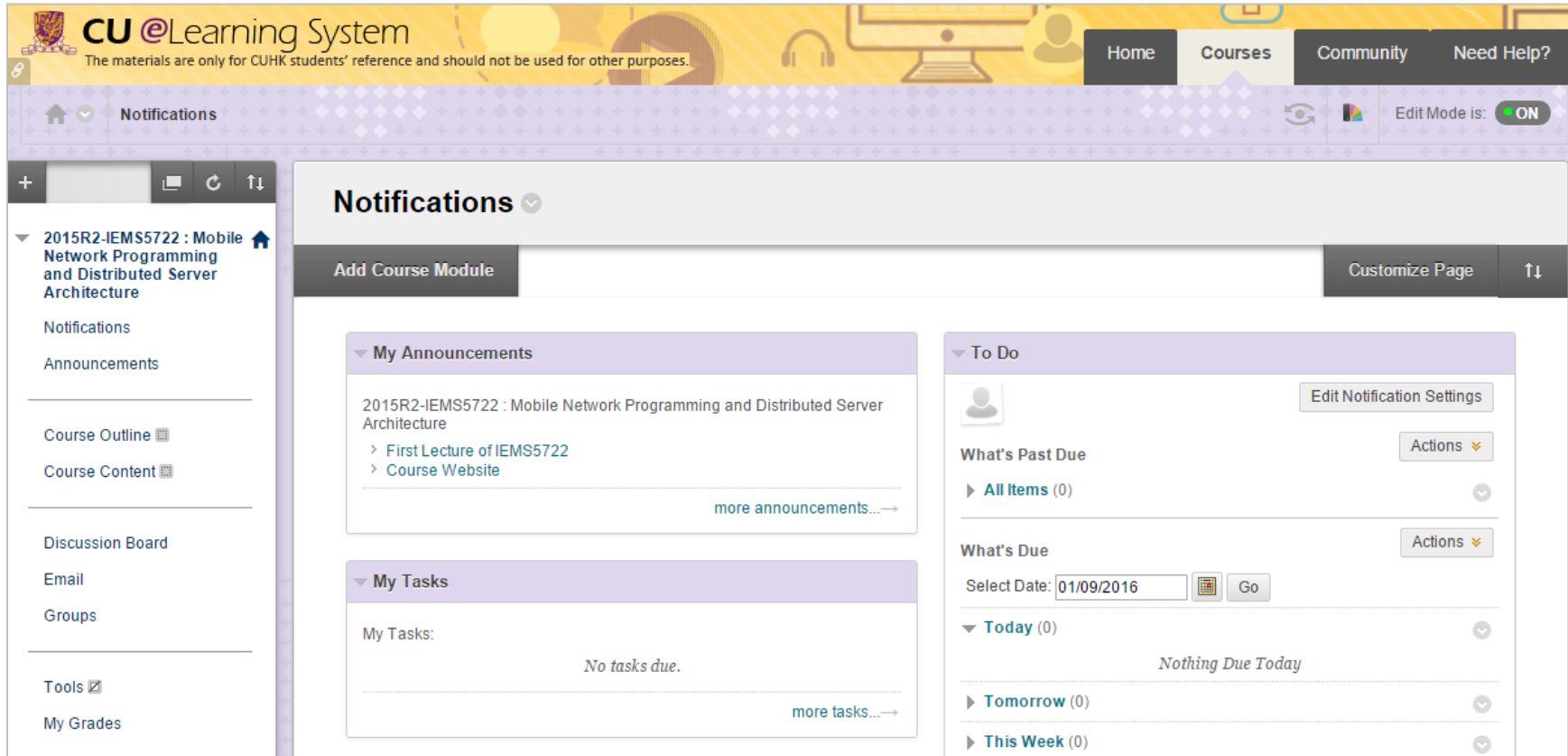
Some Rules

Honesty in Academic Work

- Zero tolerance on cheating and plagiarism
- <http://www.cuhk.edu.hk/policy/academichonesty/>
- Cite references whenever you use materials from any other sources
- It will be considered **plagiarism** no matter you **copy other's work** or **allow other to copy your work**

Online Materials

- Assignments will be released and collected on the **CUHK E-Learning System**: <https://elearn.cuhk.edu.hk/>
- You will **submit your assignments** here



The screenshot displays the CUHK E-Learning System interface. At the top, the header includes the CUHK logo, the text "CU@Learning System", a disclaimer "The materials are only for CUHK students' reference and should not be used for other purposes.", and navigation links for Home, Courses, Community, and Need Help?. Below the header, a "Notifications" section is active, showing a sidebar with course details for "2015R2-IEMS5722 : Mobile Network Programming and Distributed Server Architecture" and a main content area with "My Announcements" and "My Tasks".

CU@Learning System
The materials are only for CUHK students' reference and should not be used for other purposes.

Home Courses Community Need Help?

Notifications

2015R2-IEMS5722 : Mobile Network Programming and Distributed Server Architecture

Notifications

Announcements

Course Outline

Course Content

Discussion Board

Email

Groups

Tools

My Grades

Notifications

Add Course Module

Customize Page

My Announcements

2015R2-IEMS5722 : Mobile Network Programming and Distributed Server Architecture

- > First Lecture of IEMS5722
- > Course Website

more announcements...

My Tasks

My Tasks:

No tasks due.

more tasks...

To Do

Edit Notification Settings

What's Past Due

Actions

All Items (0)

What's Due

Select Date: 01/09/2016 Go

Actions

Today (0)

Nothing Due Today

Tomorrow (0)

This Week (0)

Online Materials

- **Course Website:**

<https://course.ie.cuhk.edu.hk/~iems5722/> or

<http://iems5722.github.io/>

- Lecture slides, assignments, project details, references will be available here

IEMS 5722 - Spring Term 2015-2016

CourseScheduleAssignmentsProjectReferences

IEMS5722 Mobile Network Programming and Distributed Server Architecture

Course Details

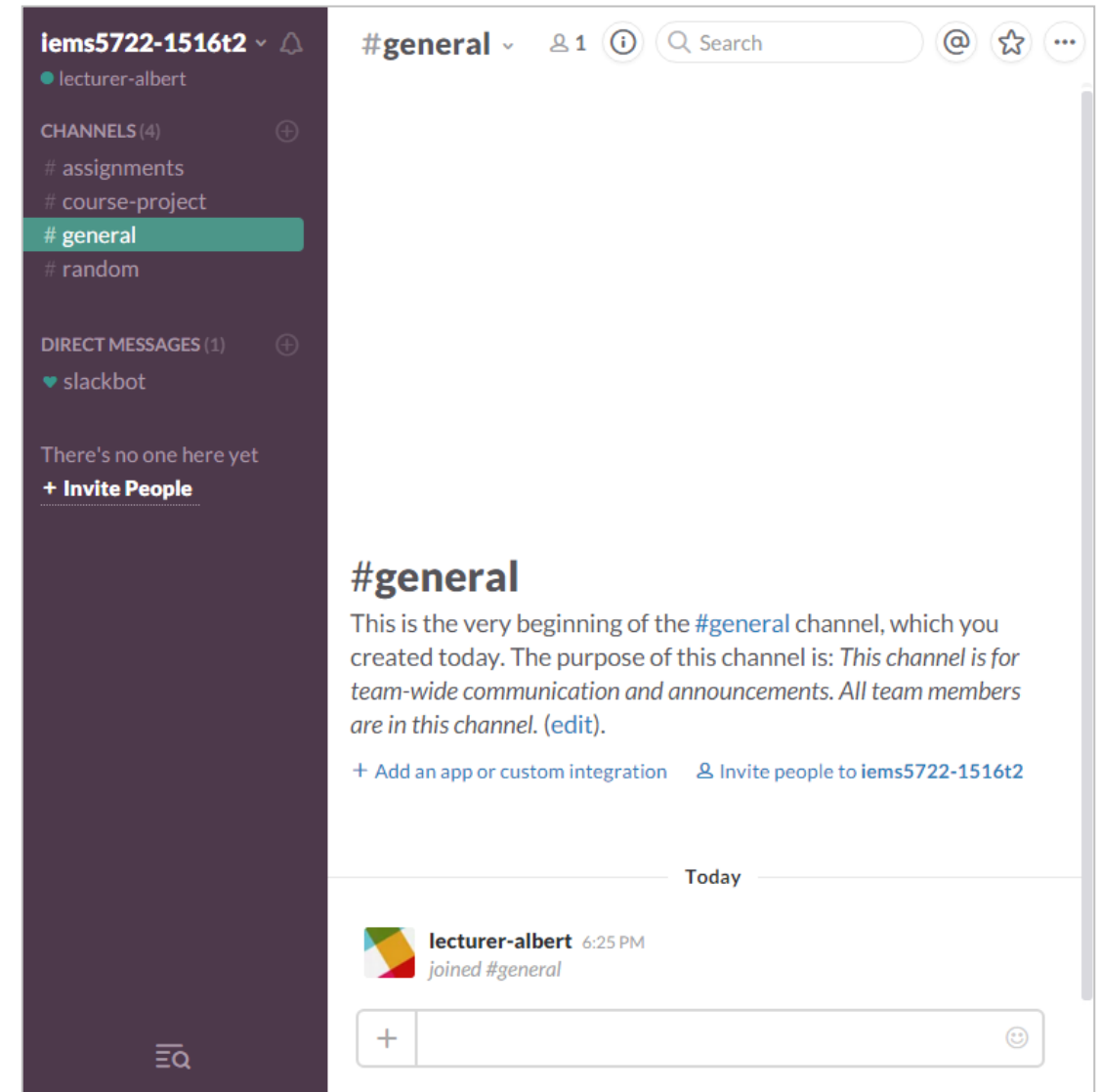
This course will cover topics and emerging technologies on mobile network programming mainly on the Android platform. Topics include data communication and networking, such as HTTP, socket programming, Bluetooth and NFC, in the context of mobile application development. Integration of third-party APIs and SDKs will be introduced. The course will also cover the major concepts and design principles of distributed server architectures that are used to support network-based mobile applications. Students should have acquired good knowledge in Java programming before enrolling in this course.

Lectures

- Every Thursday 19:00 - 21:30
- YIA (Yasumoto International Academic Park) LT9
- **Instructor:** Dr. Albert Au Yeung [cmauyeung@ie]
- **Teaching Assistant:** Mr. Marco Leung

Online Materials

- For more convenient communication among us and discussions among yourselves, we will use **Slack** in this course:
<https://iems5722-1516t2.slack.com/>
- Sign up for an account on slack and join the above team
- NOTE: **DO NOT** post any solution of assignments on Slack or any other public channels



Course Content Overview

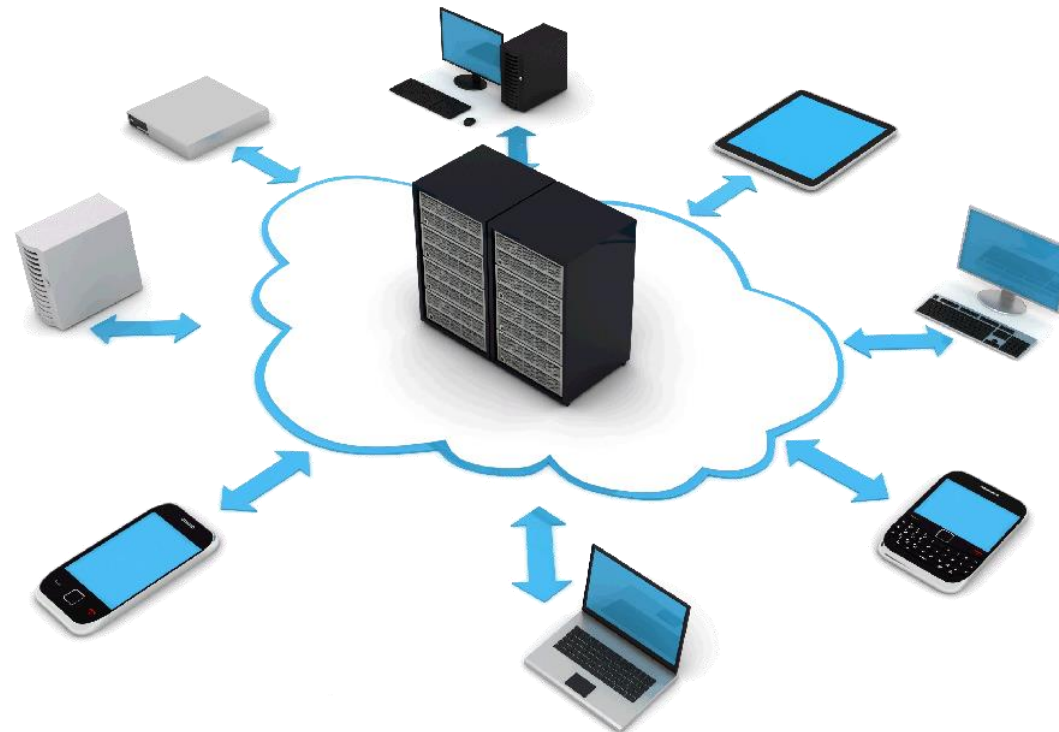
What is this Course About?

Mobile Network Programming and Distributed Server Architecture

Computer Networking

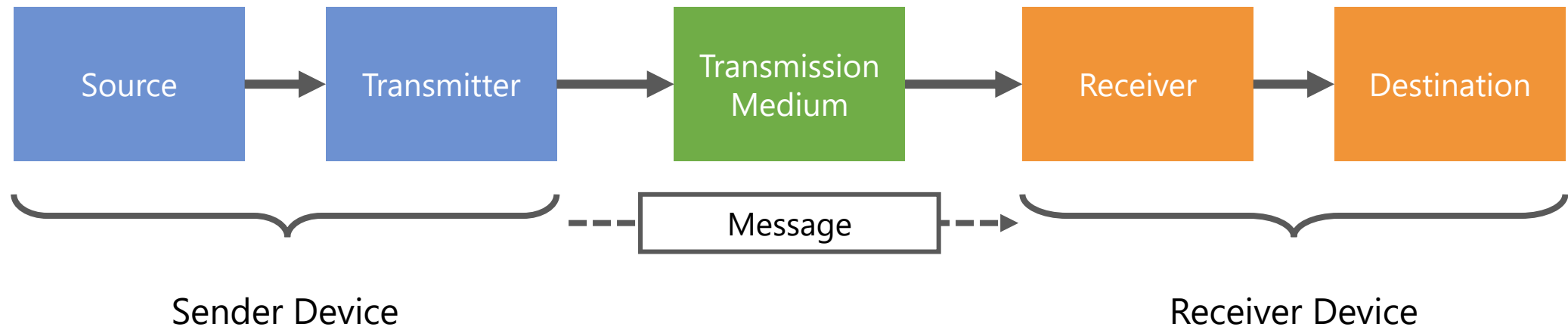
Computer Network

- A network that allows computers to perform **data communication** with one another



Data Communication

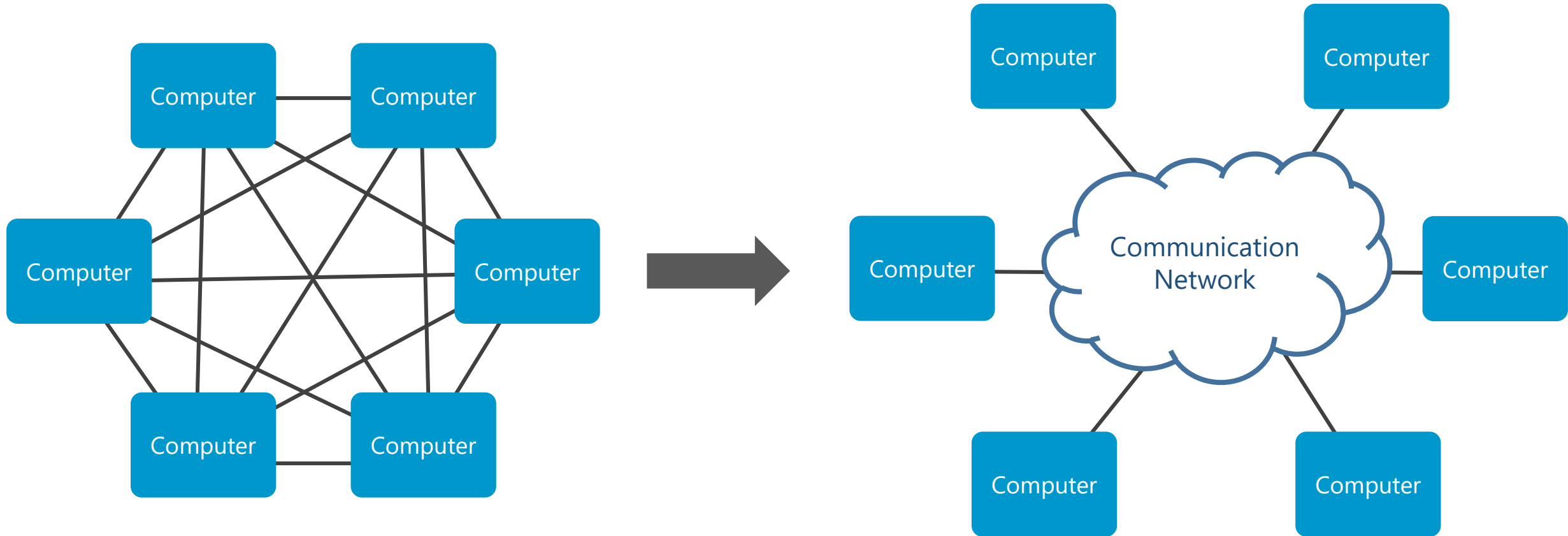
- Exchange of data between two devices using some form of transmission medium
- A simplified communication model:



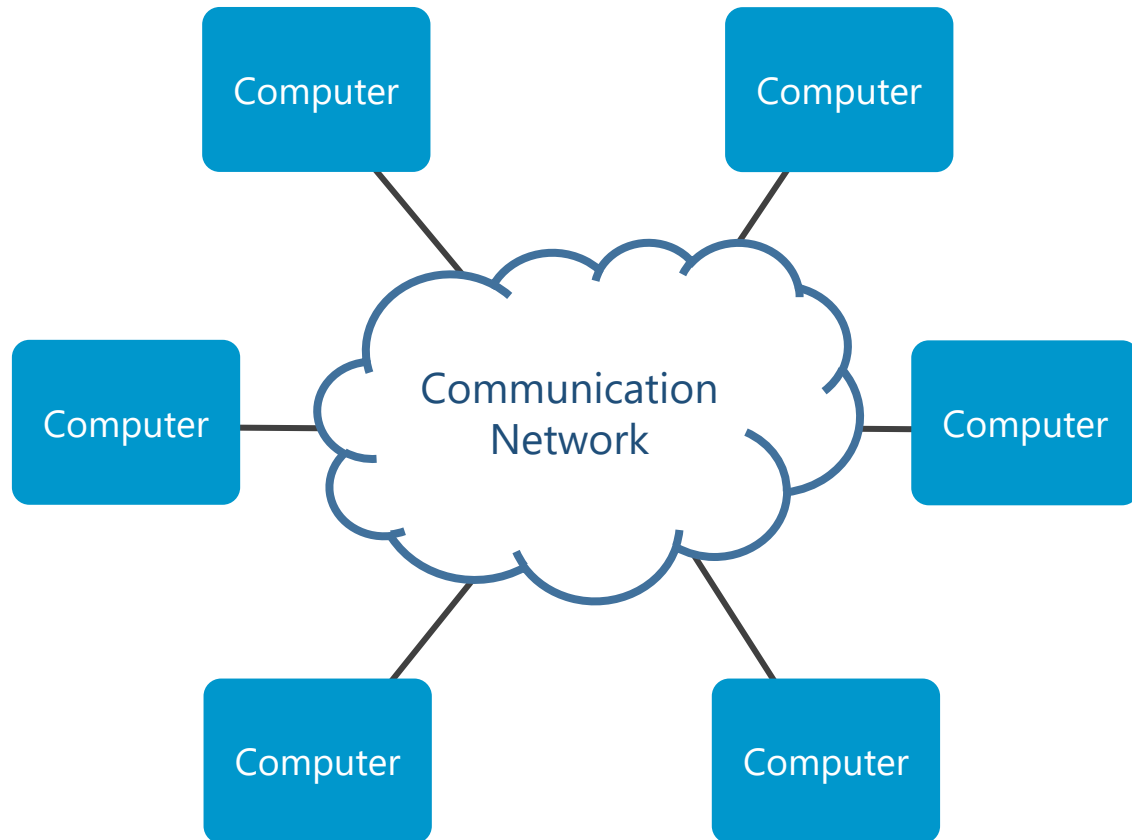
- **Protocols:** rules that govern how data is transmitted in this system

Computer Networks

- When we have many computers that want to talk to one another, point-to-point links become not practical, especially when the distance is too far



Computer Networks



Challenges in Networking:

- **How can data be transmitted** from one node to another through the network?
- How can we **address** the computers?
- How can we identify which **applications** on the computers the data should be delivered to?
- How to handle **error** or **missing** data?
- What if a **large amount of data** is transmitted at the same time?

Computer Networks

Challenges in Networking:

- **How can data be transmitted** from one node to another through the network?
- How can we **address** the computers?
- How can we identify which **applications** on the computers the data should be delivered to?
- How to handle **error** or **missing** data?
- What if a **large amount of data** is transmitted at the same time?

Data is transmitted through **routing/switching** in the network

E.g. **IP address** is one way to uniquely identify a computer in a network

Applications can be configured to listen to a specific **port** on the computer

Protocols such as **TCP** or **DCP** handle these problems

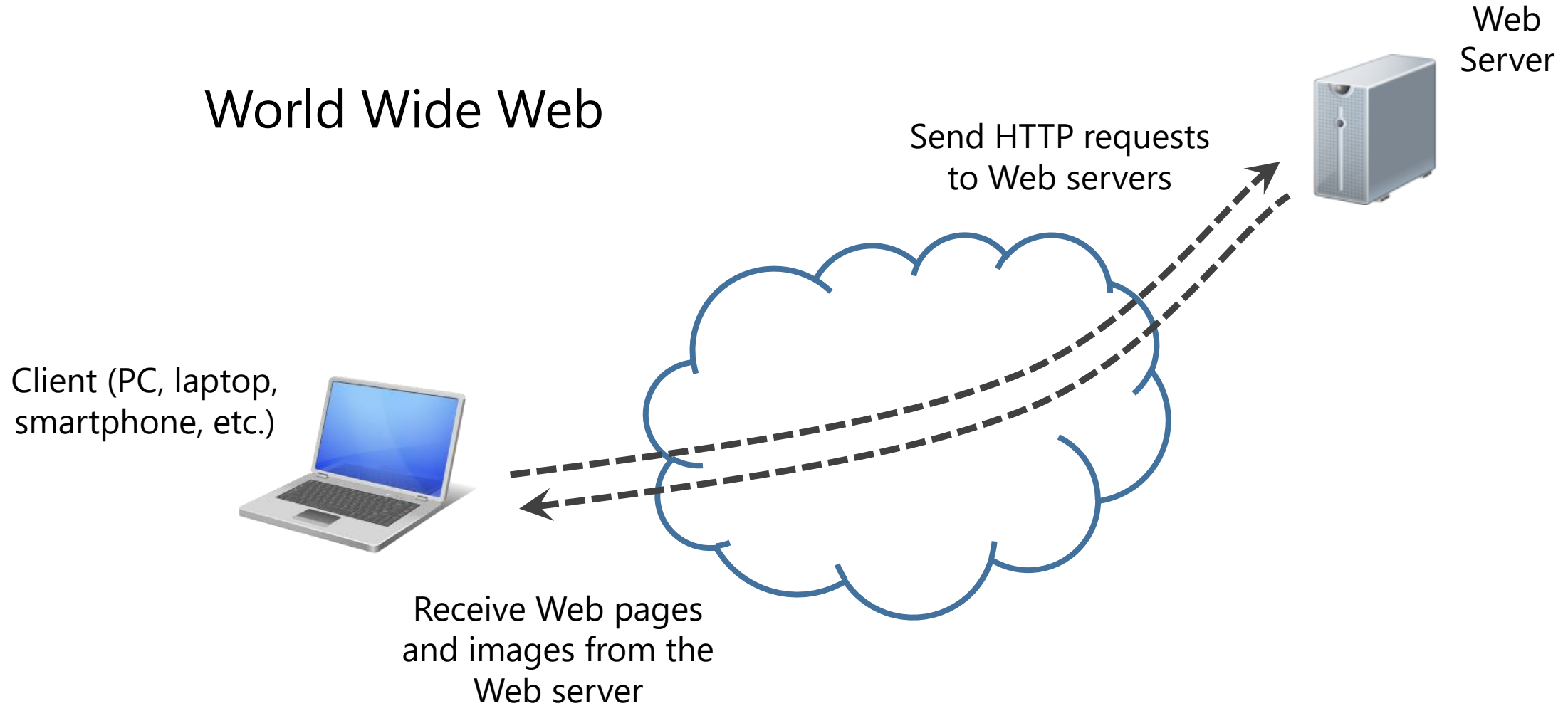
Applications in Computer Networks

Common **Applications** on the Internet

- The World Wide Web (Web servers and browsers)
- File transfer (FTP servers and clients)
- Instant messaging & video conferencing (e.g. Skype, Whatsapp, Wechat)
- Peer-to-peer file sharing
- Cloud storage
- ...

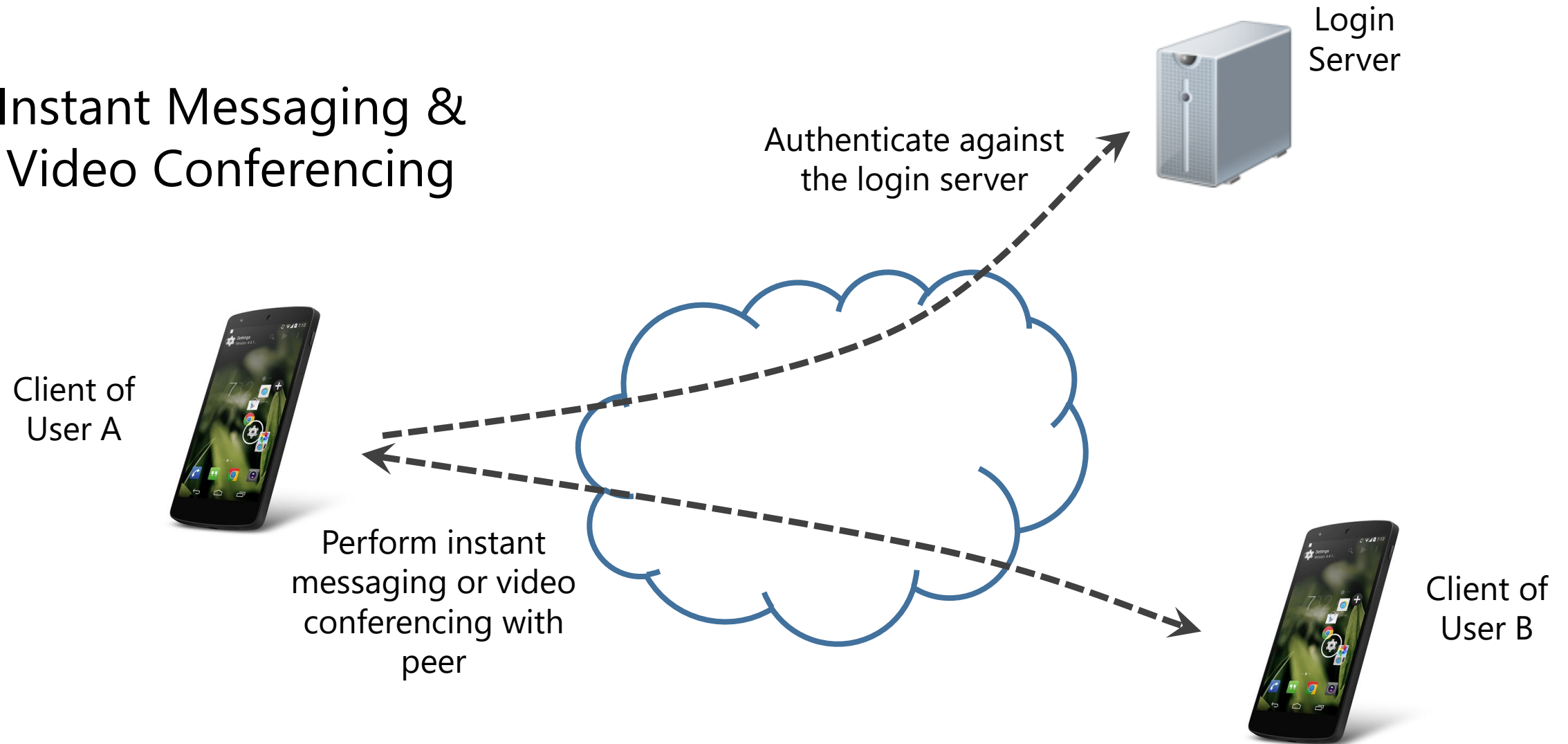
Applications in Computer Networks

World Wide Web

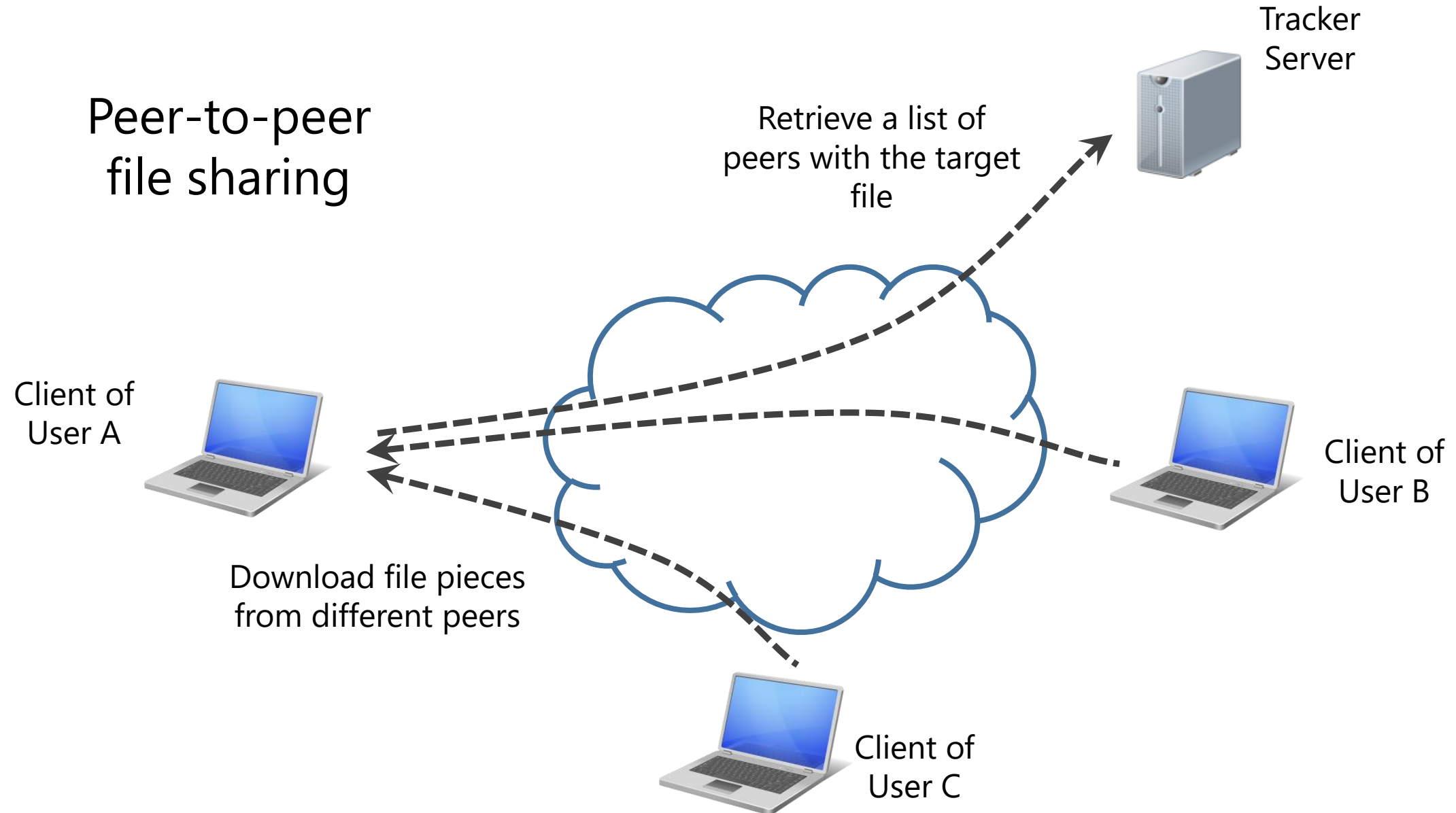


Applications in Computer Networks

Instant Messaging & Video Conferencing

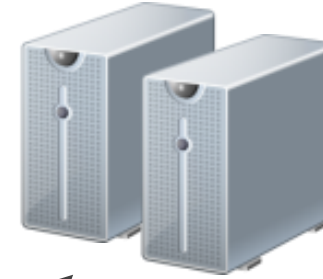


Applications in Computer Networks



Focus of this Course

Format for data exchange between servers and clients



Distributed Server Architecture
(How servers support networking apps, what are the design principles)

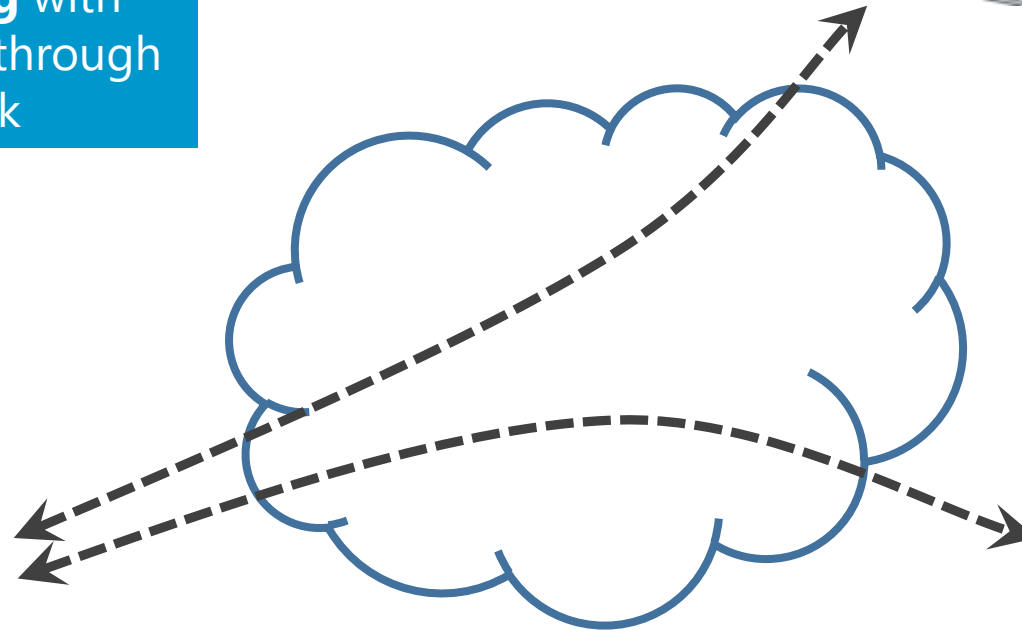
Different ways for **communicating** with other computers through the network

Applications

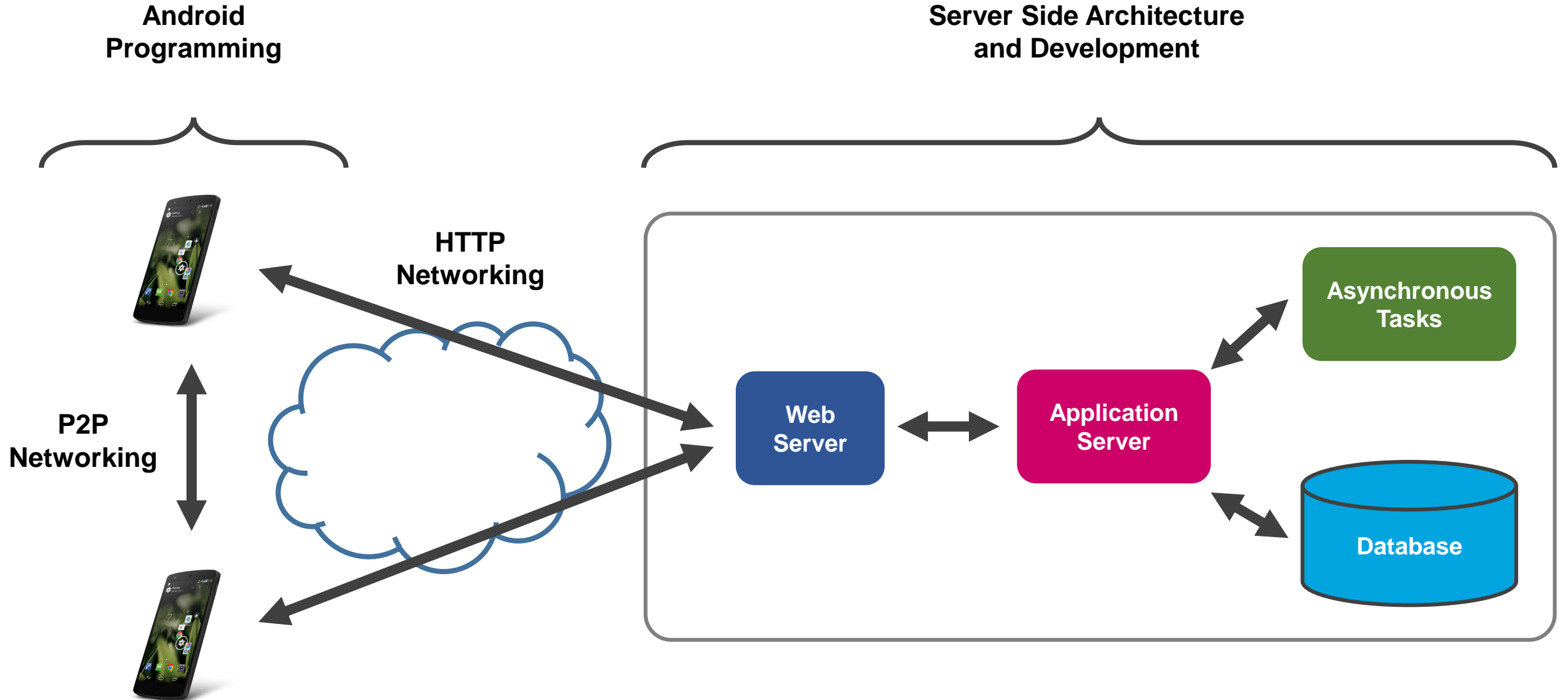
Network
Protocols



Android Network Programming



What you will Learn in this Course

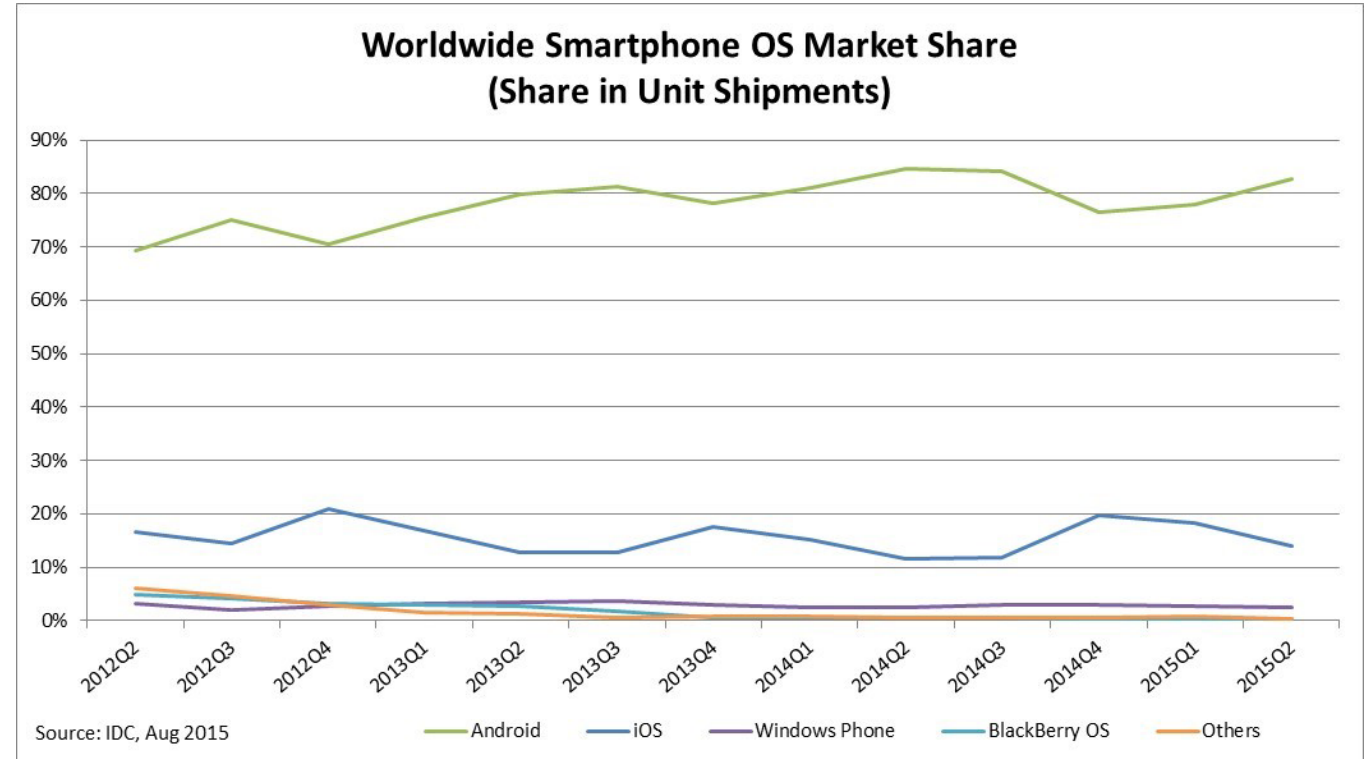


Android Mobile Application Development

Android

Why **Android**?

- The **most popular** OS for mobile devices
- **Java** as the programming language
- Many **third-party** libraries and components to use
- Relatively easier to develop and test



Source: IDC Report on Smartphone OS
Market Share, Q2 2015

Android

What is **Android**?

- Initially developed by Android Inc. (which had intended to develop OS for cameras in the beginning)
- Bought by Google in 2005
- First released in 2007, first smartphone (HTC) released in 2008
- Based on the Linux kernel, targeting touch devices
- Latest version: **6.0** (Marshmallow)
<https://www.android.com/versions/marshmallow-6-0/>



Mobile Network Programming

Many mobile apps require access to the **Internet** to

- Receive news, updates, data, etc. (e.g. News apps)
- Send and retrieve data for backup (e.g. Dropbox, Google Drive)
- Perform networking or messaging functions (e.g. Facebook, Whatsapp, Wechat)
- Transmit images or files
- ...



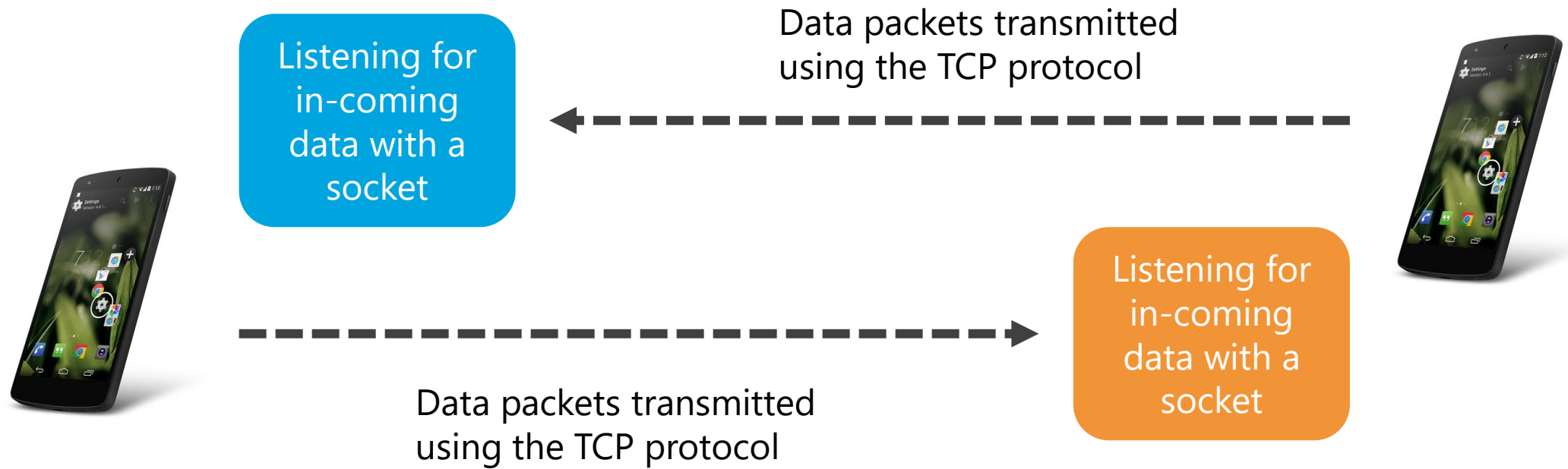
Mobile Network Programming

Topics we will cover in this course

- Data communication using the TCP/IP protocol (with peers and with servers)
- HTTP based data exchange
- Push technology & cloud messaging
- Emerging tools for socket programming
- Data exchange formats (e.g. JSON & XML)

Mobile Network Programming

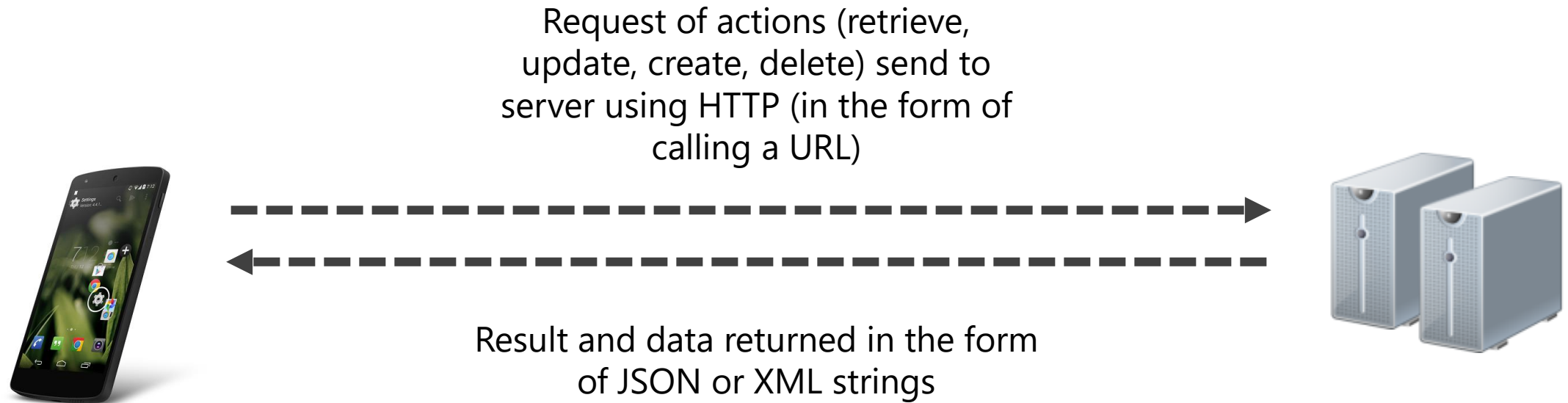
Data communication using the TCP/IP protocol



(Involves some socket programming and multi-threading in Java)

Mobile Network Programming

HTTP-based Data Exchange

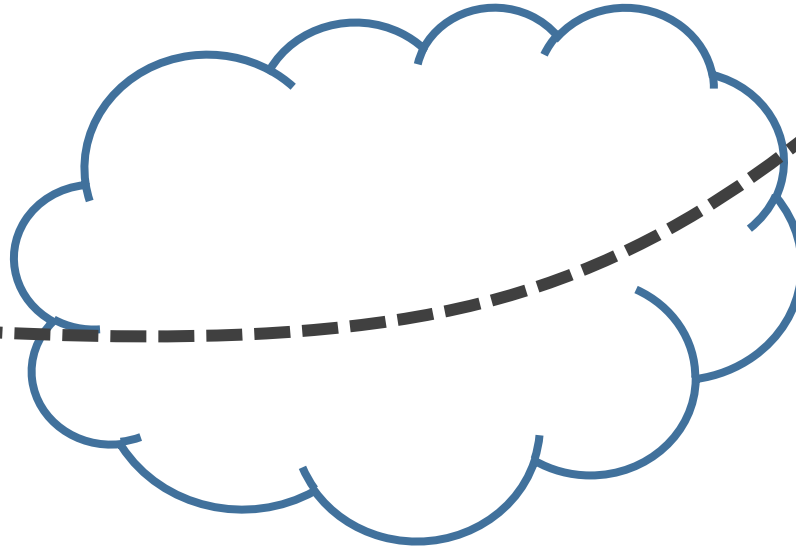


(Involves using an HTTP client and multi-threading on the client side)

Mobile Network Programming

Push Technology & Cloud Messaging

A service running in the client receives the message, and carry out corresponding actions (e.g. notify the user, launch an app)



Push a message to the client when there is an update, a new message, etc.

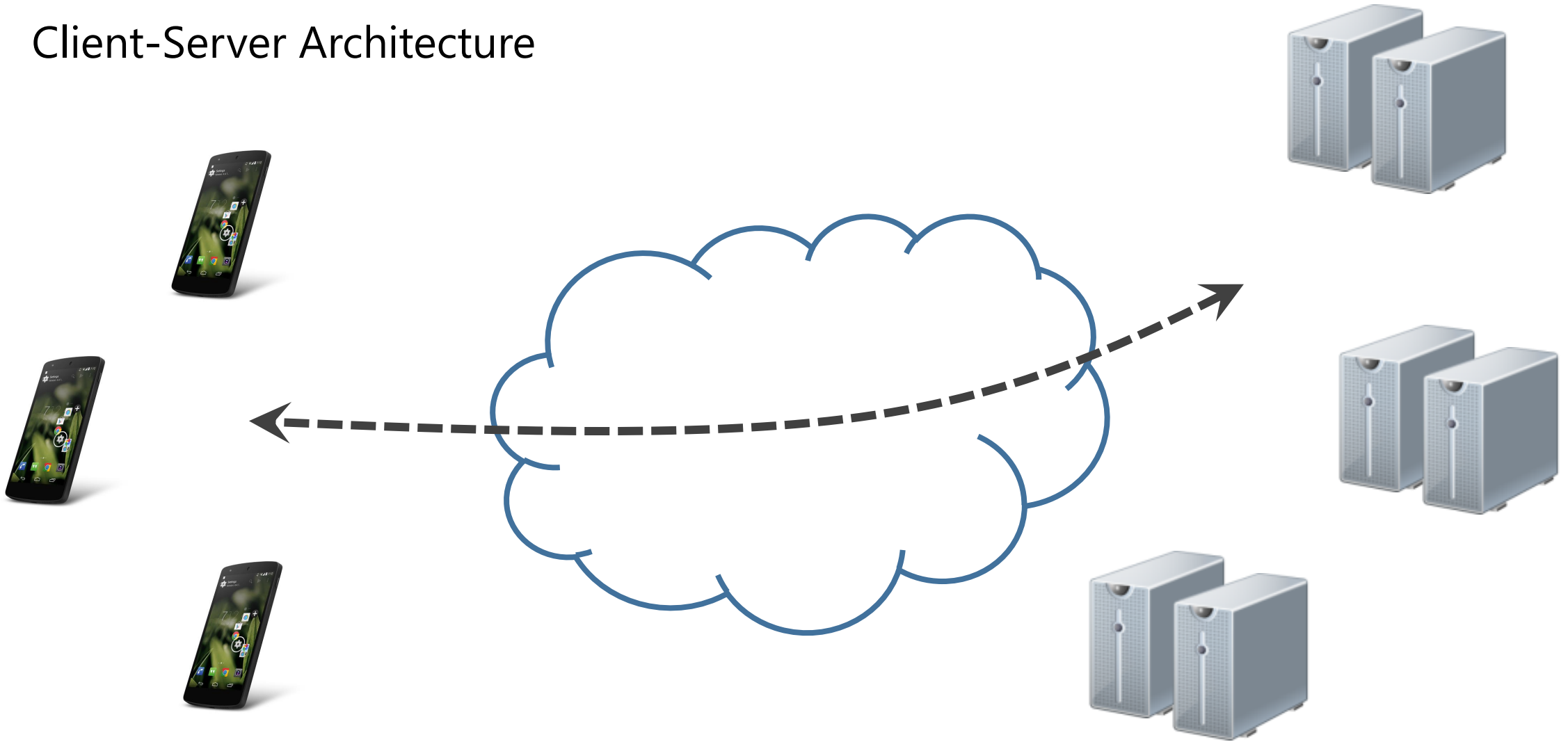
Application Servers



(We will talk about common push technologies, and also the Google Cloud Messaging service)

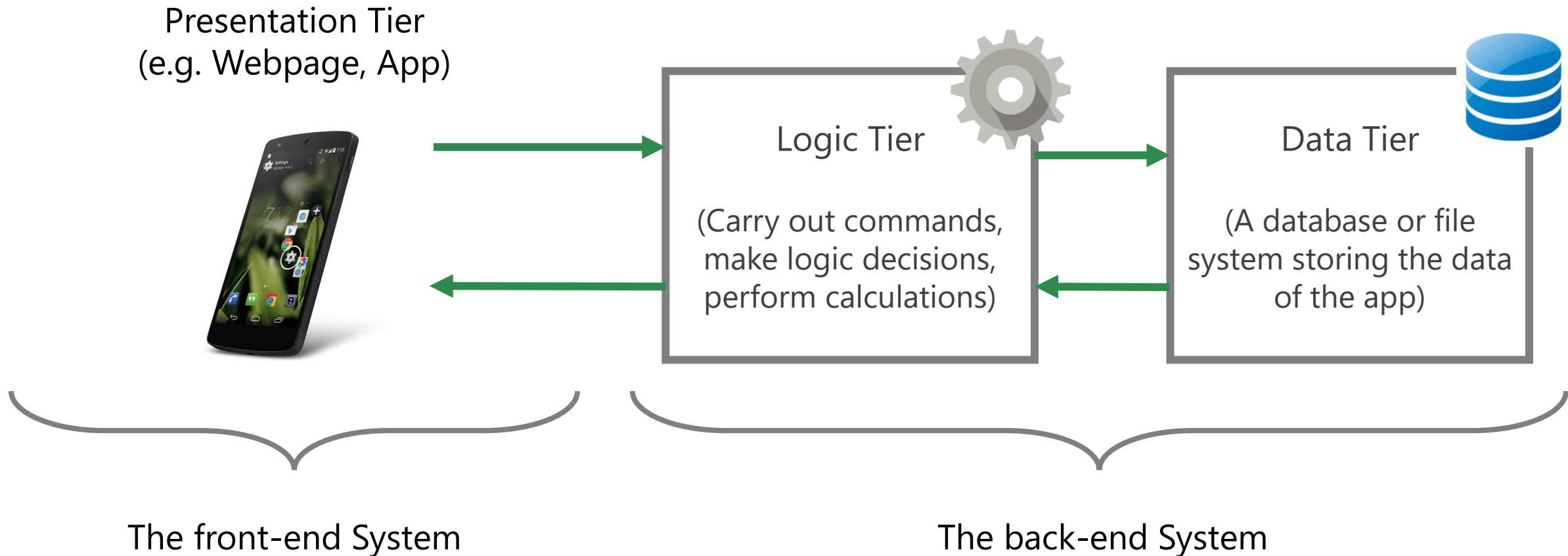
Distributed Server Architecture

Client-Server Architecture



Distributed Server Architecture

Common **Three Tier** Architecture



Break

Please fill in this questionnaire



<https://goo.gl/0mKKJY>

Introduction to Android Programming

The Android Platform

A **software stack** for **mobile devices** such as smartphones and tablets

- Include an OS kernel, system libraries, application frameworks and some important apps (e.g. telephony, SMS, camera)

For developing Android apps, we use the **Android SDK**

- With libraries and development tools
- Guideline and trainings here: <http://developer.android.com/training>

Android Runtime

Android programs are run on the **Dalvik Virtual Machine** (replaced by **Android Runtime (ART)** in Android 5.0)

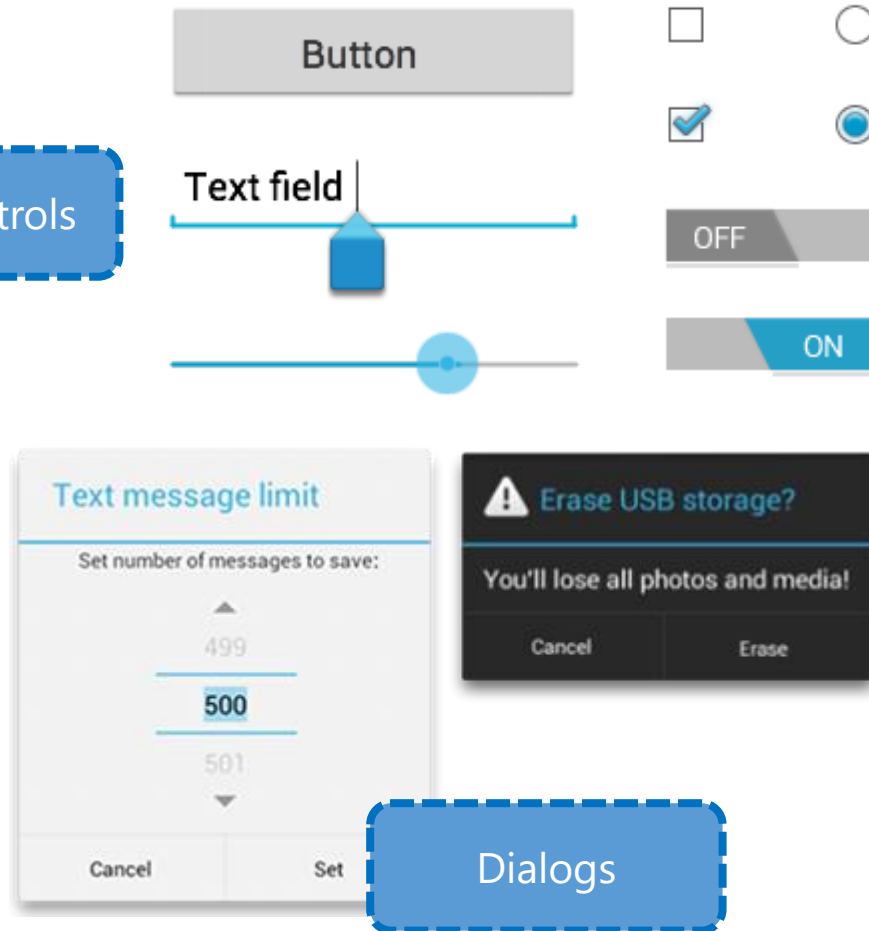
Workflow of App Deployment

1. Write the app in Java
2. Compile the Java source code into Java bytecode files
3. Convert the Java bytecode files to a DEX bytecode file
4. Dalvik executes DEX bytecode file

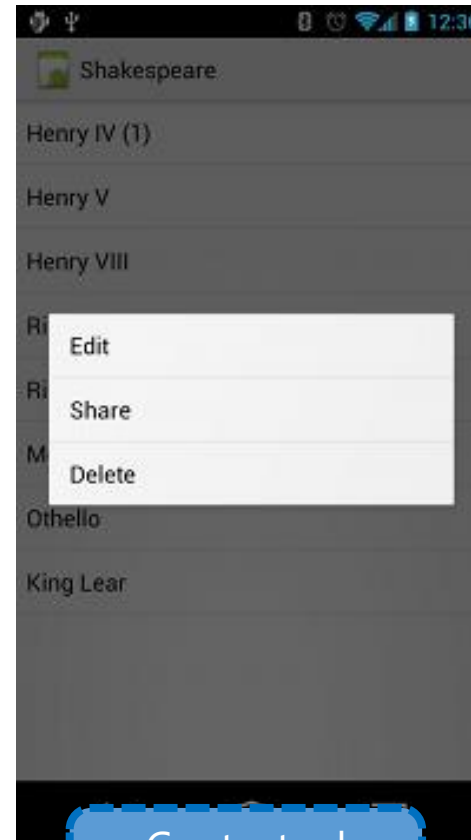
Android Apps

Android has many pre-defined UI components, based on the **View** class

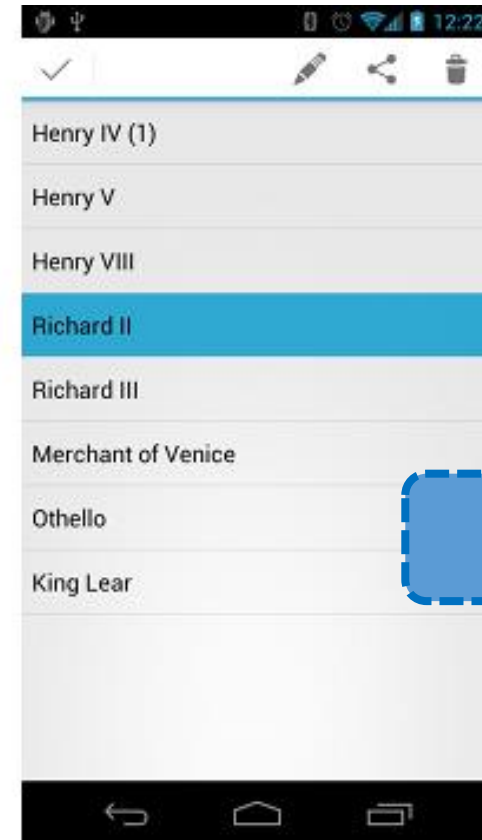
Input Controls



Dialogs



Contextual Menu



Lists

Android App Development

Each “**page**” in an Android is called an **Activity**

Logic of the Activity

In the **Java source code**, we bind variables to UI elements, and then write codes to describe what actions to perform or how data are fetched from different sources, or how information is presented on the UI



Layout of the UI

The layout of the UI is specified in an XML file, which describes:

- The **positions** and **layout** method of the UI elements
- The **style** of each element (e.g. width, height, font colour and size, etc.)
- The **unique name (ID)** of each UI element

Android App Development

An example XML file describing the layout of an activity

```
<LinearLayout xmlns:"http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >
    <EditText android:id="@+id/edit_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="@string/edit_message" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_send" />
</LinearLayout>
```

Android App Development

An example Java source code file of an activity

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_display_message);  
  
        EditText input = (EditText)findViewById(R.id.edit_message);  
        ...  
    }  
}
```


Android App Development

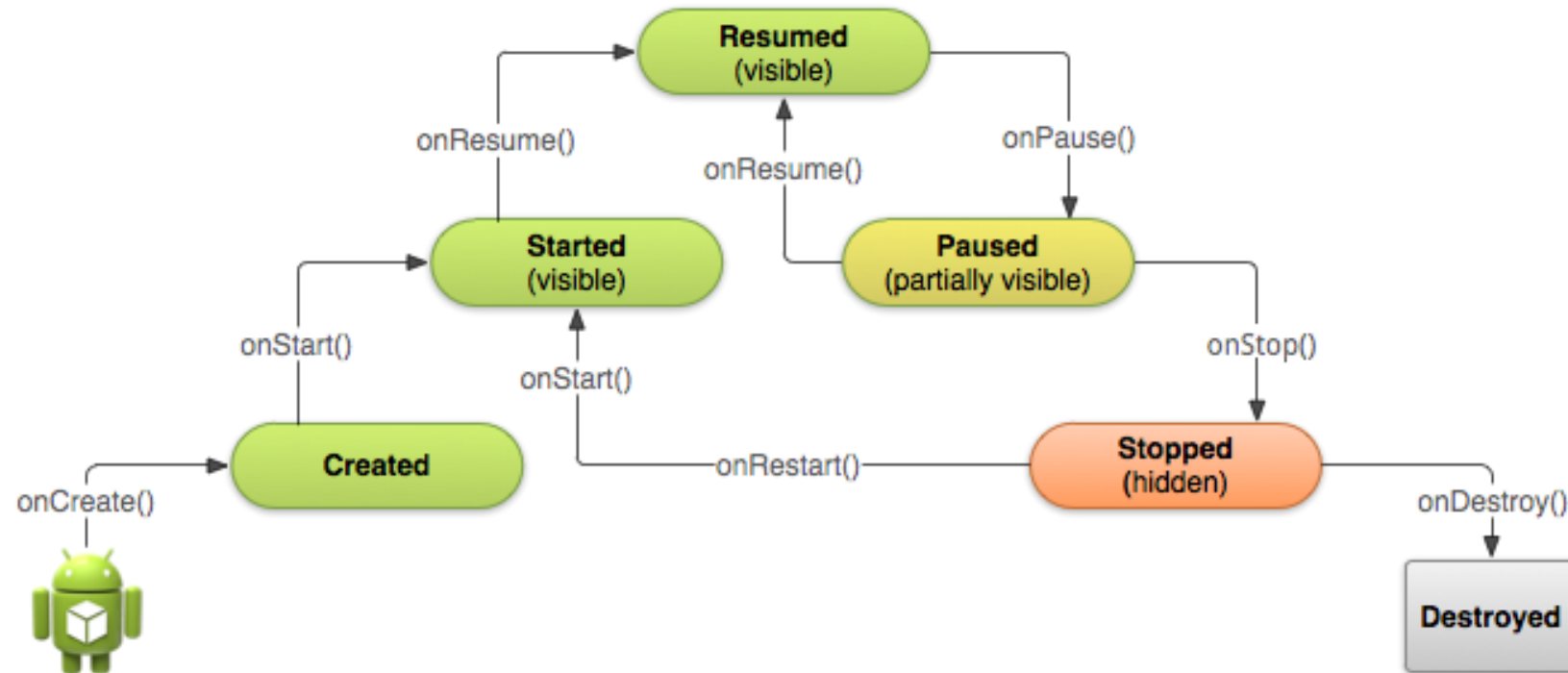
Android apps are mainly **event driven**

- Actions are performed when there is an event (e.g. the app is **launched**, a button is **clicked**, a check box is **checked**)
- You will implement specific functions or **event handlers** and attach these to UI elements
- An example:

```
button.setOnClickListener(new Button.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Toast.makeText(getApplicationContext(), "Hello!", Toast.LENGTH_SHORT).show();  
    }  
});
```

Android App Life Cycle

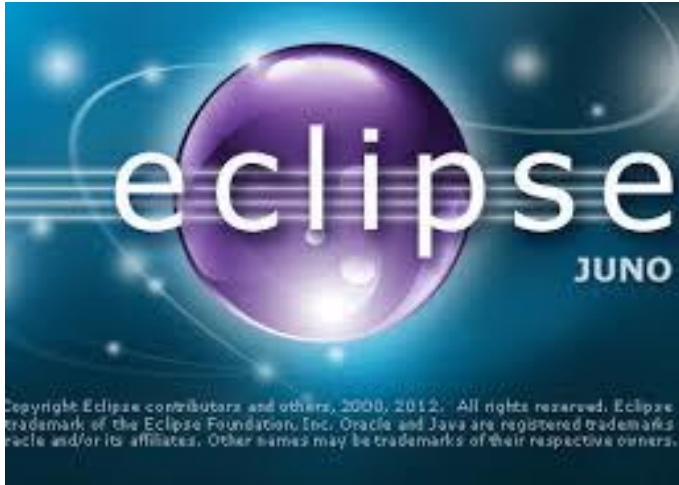
- Android is intended to be run on mobile devices with limited power and computing resources
- Android takes care of the life cycle of an app (launch, suspend, resume, destroy)



Android Development Environment

Android Development Environment

Both **Eclipse** and **Android Studio** can be used to develop Android apps



An IDE for software development in Java (and other languages), support Android development through the Android Development Tools (ADT) plugin

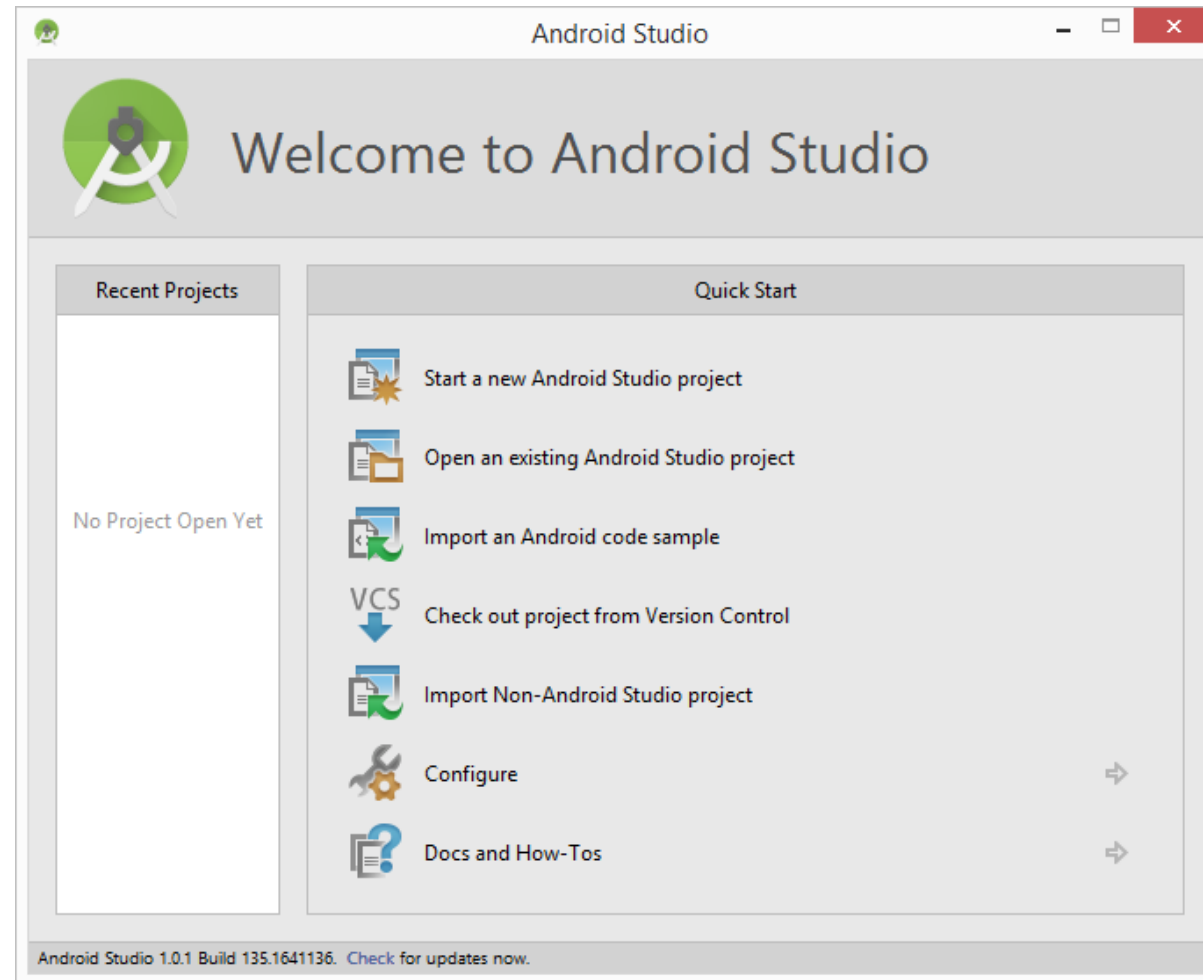


The official Android development IDE release recently, based on the IntelliJ IDE software

In this course, you are recommended to use **Android Studio** to develop the app in your assignments and projects

Android Studio

- Download Android Studio and get familiar with using it
<http://developer.android.com/sdk/index.html>



Assignment 0

Android Development Basics

What you will do in this assignment:

1. Learn the basic concepts and procedures in developing an Android application
2. Set up the Android development environment using the **Android Studio**
3. Create a **simple Android app**
4. Test your app on the **emulator** or on a **physical device**
5. Make some **changes** to the app by editing the source files

Android Development Basics

Notes on **Assignment 0**

1. You should go through all the steps to set up the development environment (in your own computer if possible)
2. Seek help or look for solutions as early as possible
3. Your work should be submitted to the CUHK eLearning system before the deadline (Friday 22nd Jan, 23:59)

Learning Resources

Java Tutorial

<https://www.udemy.com/java-tutorial/>

<http://eclipsetutorial.sourceforge.net/totalbeginnerlessons.html>

Google Android Official Guide

<https://developer.android.com/training/>

Sample Apps

<http://developer.android.com/samples/index.html>

Android Development Tutorial

<http://www.vogella.com/tutorials/Android/article.html>

And use **Google** and **Stack Overflow** to find answers to your questions

Next Lecture: Android Programming

End of Lecture 1