

Introduction:

Course Organization & Topics

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA



Learning Objectives in this Part of the Module

- Understand the structure & contents of the course

VANDERBILT UNIVERSITY

Pattern-Oriented Software Architectures: Programming Mobile Services for Android Handheld Systems

Part of the "Mobile Cloud Computing with Android" Specialization »

In this course—the second in a trans-institution sequence of MOOCs on Mobile Cloud Computing with Android—we will learn how to apply patterns, pattern languages, and frameworks to alleviate the complexity of developing concurrent and networked services on mobile devices running Android that connect to popular cloud computing platforms.

[Preview Lectures](#)



About the Course

The confluence of multi-core and distributed-core processors, inexpensive mass storage, ubiquitous wireless connectivity, and commodity software platforms is driving the need for software engineers and programmers who understand how to develop concurrent and networked software for mobile devices that connect to cloud computing platforms. Despite many improvements in processors, storage, and networks, however, developing quality software on-time and on-budget remains hard. Moreover, developing high quality reusable concurrent and networked software apps

Sessions

May 12th 2014 ▼

[Go to class](#)

[Earn a Verified Certificate](#)

Overview of the Course Organization

- The course has three main sections
 - **Section 0: Course Introduction**
 - Part 1: Overview of Mobile Cloud Computing with Android
 - Part 2: Course Structure & Topics
 - Part 3: Course Prerequisites & Learning Strategies
 - Part 4: Overview of Patterns & Frameworks
 - **Section 1: Android Concurrency**
 - Module 1: Concurrency Motivations & Challenges
 - Module 2: Java Concurrency Mechanisms
 - Module 3: Android Concurrency Frameworks
 - **Section 2: Android Services, Local IPC, & Security**
 - Module 1: Android Services & Local IPC
 - Module 2: Android App Security & Risks
 - Module 3: Building More Secure Android Apps
 - **Section 3: Concurrency & Communication Patterns Applied in Android**
 - Module 1: Coordinating Concurrent Access to Shared Data
 - Module 2: Ensuring Only One Looper Per Thread
 - Module 3: Passing Commands to Threads & Services
 - Module 4: Passing Message Requests to Threads
 - Module 5: Decoupling Synchronous & Asynchronous Processing
 - Module 6: Supporting Object-Oriented Remote Method Calls

Overview of the Course Organization

- The course has three main sections
 - **Section 0: Course Introduction**
 - Part 1: Overview of Mobile Cloud Computing with Android
 - Part 2: Course Structure & Topics
 - Part 3: Course Prerequisites & Learning Strategies
 - Part 4: Overview of Patterns & Frameworks
 - **Section 1: Android Concurrency**
 - Module 1: Concurrency Motivations & Challenges
 - Module 2: Java Concurrency Mechanisms
 - Module 3: Android Concurrency Frameworks
 - **Section 2: Android Services, Local IPC, & Security**
 - Module 1: Android Services & Local IPC
 - Module 2: Android App Security & Risks
 - Module 3: Building More Secure Android Apps
 - **Section 3: Concurrency & Communication Patterns Applied in Android**
 - Module 1: Coordinating Concurrent Access to Shared Data
 - Module 2: Ensuring Only One Looper Per Thread
 - Module 3: Passing Commands to Threads & Services
 - Module 4: Passing Message Requests to Threads
 - Module 5: Decoupling Synchronous & Asynchronous Processing
 - Module 6: Supporting Object-Oriented Remote Method Calls

Overview of the Course Organization

- The course has three main sections
 - **Section 0: Course Introduction**
 - Part 1: Overview of Mobile Cloud Computing with Android
 - Part 2: Course Structure & Topics
 - Part 3: Course Prerequisites & Learning Strategies
 - Part 4: Overview of Patterns & Frameworks
 - **Section 1: Android Concurrency**
 - Module 1: Concurrency Motivations & Challenges
 - Module 2: Java Concurrency Mechanisms
 - Module 3: Android Concurrency Frameworks
 - **Section 2: Android Services, Local IPC, & Security**
 - Module 1: Android Services & Local IPC
 - Module 2: Android App Security & Risks
 - Module 3: Building More Secure Android Apps
 - **Section 3: Concurrency & Communication Patterns Applied in Android**
 - Module 1: Coordinating Concurrent Access to Shared Data
 - Module 2: Ensuring Only One Looper Per Thread
 - Module 3: Passing Commands to Threads & Services
 - Module 4: Passing Message Requests to Threads
 - Module 5: Decoupling Synchronous & Asynchronous Processing
 - Module 6: Supporting Object-Oriented Remote Method Calls

Overview of the Course Organization

- The course has three main sections
 - **Section 0: Course Introduction**
 - Part 1: Overview of Mobile Cloud Computing with Android
 - Part 2: Course Structure & Topics
 - Part 3: Course Prerequisites & Learning Strategies
 - Part 4: Overview of Patterns & Frameworks
 - **Section 1: Android Concurrency**
 - Module 1: Concurrency Motivations & Challenges
 - Module 2: Java Concurrency Mechanisms
 - Module 3: Android Concurrency Frameworks
 - **Section 2: Android Services, Local IPC, & Security**
 - Module 1: Android Services & Local IPC
 - Module 2: Android App Security & Risks
 - Module 3: Building More Secure Android Apps
 - **Section 3: Concurrency & Communication Patterns Applied in Android**
 - Module 1: Coordinating Concurrent Access to Shared Data
 - Module 2: Ensuring Only One Looper Per Thread
 - Module 3: Passing Commands to Threads & Services
 - Module 4: Passing Message Requests to Threads
 - Module 5: Decoupling Synchronous & Asynchronous Processing
 - Module 6: Supporting Object-Oriented Remote Method Calls

Overview of the Course Organization

- The course has three main sections
 - Each Section is composed of Modules



Overview of the Course Organization

- The course has three main sections
 - Each Section is composed of Modules
 - Each Module is composed of Parts



Overview of the Course Organization

- The course has three main sections
 - Each Section is composed of Modules
 - Each Module is composed of Parts
 - Each Part is a single video



Overview of the Course Organization

- The course has three main sections
 - Each Section is composed of Modules
 - Each Module is composed of Parts
 - Each Part is a single video
 - Each Part is composed of segments



Overview of the Course Organization

- The course has three main sections
 - Each Section is composed of Modules
 - Each Module is composed of Parts
 - Each Part is a single video
 - Quizzes pop up periodically

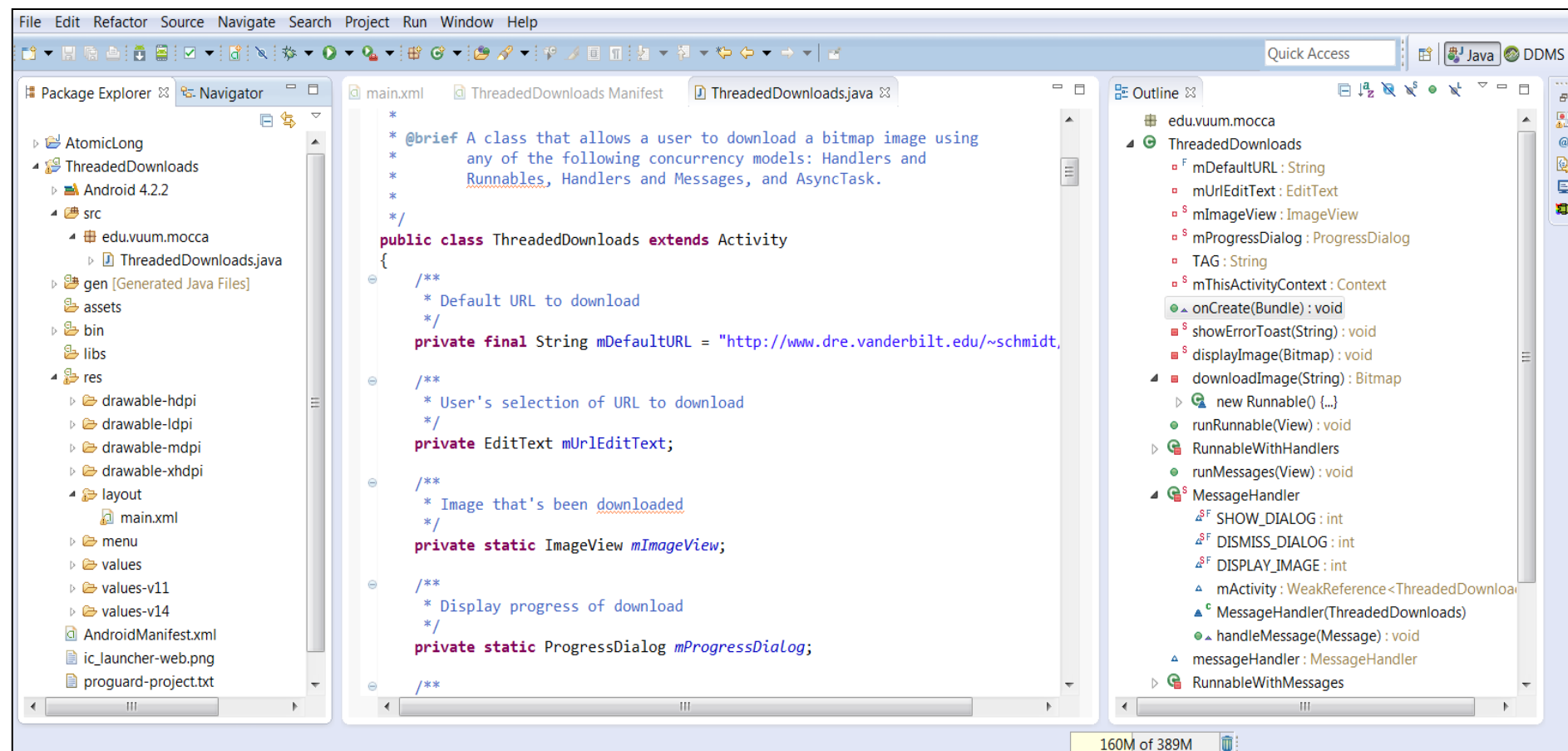


If you download the videos & watch them locally the quizzes won't appear

Overview of the Assignments & Assessments


Overview of Assignments & Assessments

- Programming assignments should be written in Java using Eclipse



Overview of Assignments & Assessments

- Programming assignments should be written in Java using Eclipse
- All source code for assignments & examples available at GitHub

VANDERBILT  **UNIVERSITY**

Pattern-Oriented Software Architectures: Programming Mobile Services..
by Dr. Douglas C. Schmidt, Dr. C. Jules White

Join Signature Track
1 month and 20 hours left



Q

COURSE
Discussion Forums
Announcements
Video Lectures
Source Code By Week

Source Code By Week

[Help](#)

We've created a GitHub repository for the 2014 POSA MOOC assignments and examples, which you can access at

<https://github.com/douglasraigschmidt/POSA-14>

When the 2014 POSA MOOC officially starts (in mid-May) we'll post links to the source code examples by week here. You can also get various parts of the Android 4.0 (ICE Cream Sandwich) software at the following links

bionic
Page : https://github.com/android/platform_bionic/tree/android-4.0.4_r2.1
D/L : https://github.com/android/platform_bionic/archive/android-4.0.4_r2.1.zip

dalvik
Page : https://github.com/android/platform_dalvik/tree/android-4.0.4_r2.1

See class.coursera.org/posa-002/wiki/Source_Code_By_Week

Overview of Assignments & Assessments

- Programming assignments should be written in Java using Eclipse
- All source code for assignments & examples available at GitHub
- The assignments will provide you with a range of experience with Android programming



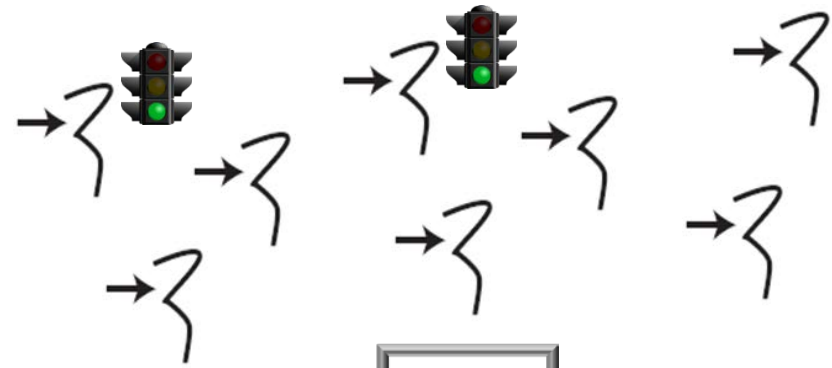
The screenshot shows the GitHub repository page for `douglasraigschmidt / POSA-14`. The repository description states: "This repository contains assignments and examples for the 2014 offering of the POSA MOOC (see www.coursera.org/course/posa for more information) — Edit". The repository statistics show 23 commits, 1 branch, 0 releases, and 2 contributors. The current branch is `master`. A commit message "Fixed a few bugs." is highlighted. Below this, a table lists the repository's structure:

File	Description	Time
<code>assignments</code>	Fixed a few bugs.	3 days ago
<code>ex</code>	Some light editing and reformatting.	4 days ago
<code>.gitattributes</code>	Some Improvements for the ThreadedDownloads example	5 days ago
<code>.gitignore</code>	Some Improvements for the ThreadedDownloads example	5 days ago
<code>README.md</code>	Create README.md	21 days ago

Below the table, the `README.md` file is shown, containing the title **POSA-14** and the same repository description as above.

Overview of Assignments & Assessments

- Programming assignments should be written in Java using Eclipse
- All source code for assignments & examples available at GitHub
- The assignments will provide you with a range of experience with Android programming, e.g.
 - Implement simple AtomicLong & Semaphore classes

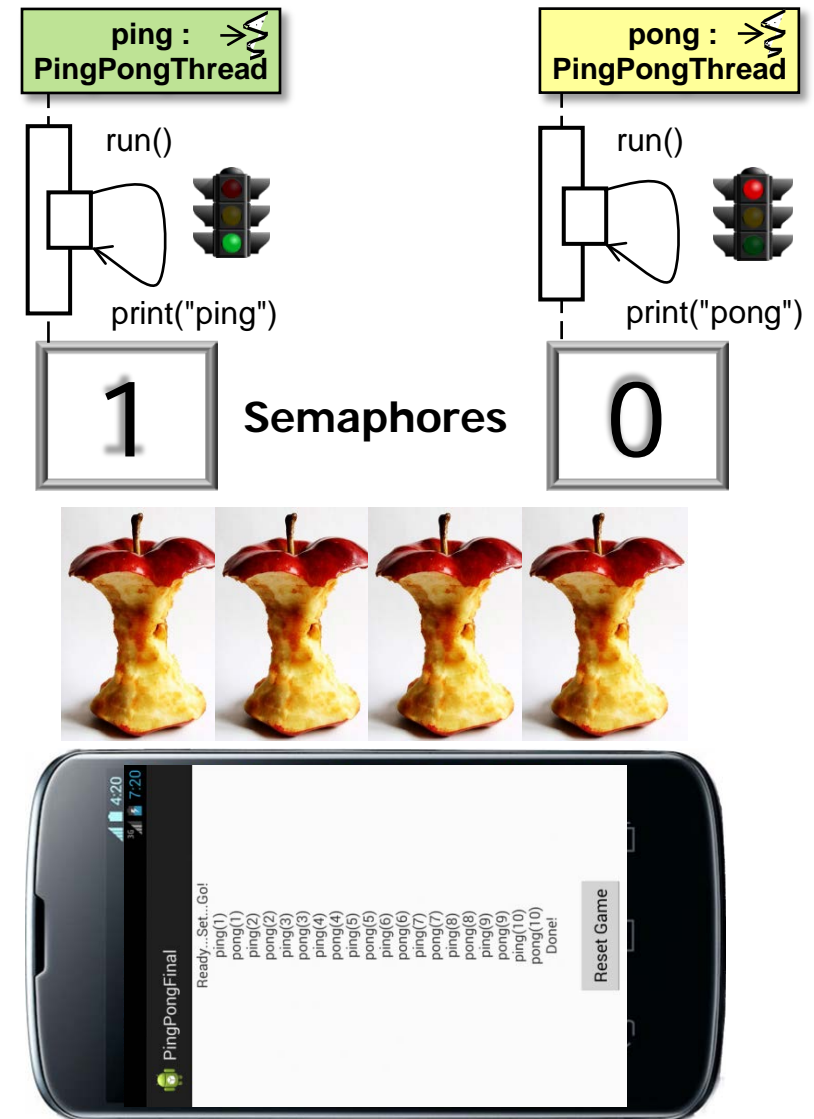


Semaphore



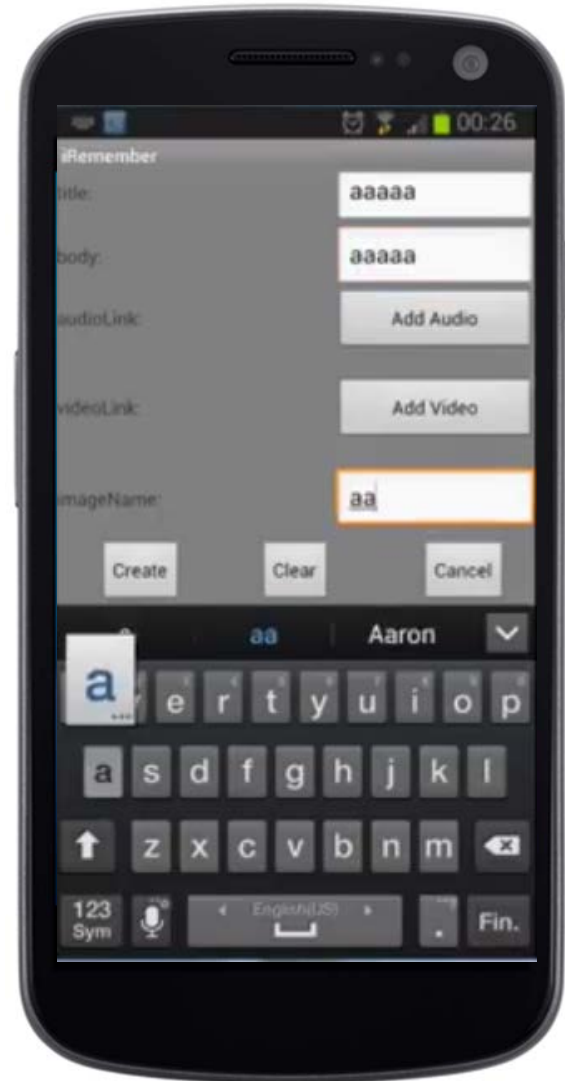
Overview of Assignments & Assessments

- Programming assignments should be written in Java using Eclipse
- All source code for assignments & examples available at GitHub
- The assignments will provide you with a range of experience with Android programming, e.g.
 - Implement simple AtomicLong & Semaphore classes
 - Implement portions of a multi-threaded “pong-pong” application



Overview of Assignments & Assessments

- Programming assignments should be written in Java using Eclipse
- All source code for assignments & examples available at GitHub
- The assignments will provide you with a range of experience with Android programming, e.g.
 - Implement simple AtomicLong & Semaphore classes
 - Implement portions of a multi-threaded “pong-pong” application
 - Extend the iRemember application to support concurrency & security



Overview of Assignments & Assessments

- Programming assignments should be written in Java using Eclipse
- All source code for assignments & examples available at GitHub
- The assignments will provide you with a range of experience with Android programming
- Assessments will be done by auto-grading & peer-grading



Overview of Assignments & Assessments

- Programming assignments should be written in Java using Eclipse
- All source code for assignments & examples available at GitHub
- The assignments will provide you with a range of experience with Android programming
- Assessments will be done by auto-grading & peer-grading
- Statements of Accomplishment require completing quizzes and/or assignments

Statements of Accomplishments

[Help](#)

The POSA MOOC is the second course in the Mobile Cloud Computing with Android (MoCCA) Specialization. It consists of lecture videos with integrated quiz questions designed to ensure students understand the material covered in the videos. In recognition that not all students have the same learning objectives or available time, the POSA MOOC is offered at the following two levels of engagement:

- **Normal Track** – Estimated Time Commitment: 3 – 4 Hours Per Week. Students at this level will be assessed by weekly auto-graded standalone quizzes. This track is designed for those who wish to engage the material by taking the auto-graded quizzes, as well as participating in the online discussion forums, but who may not have the time/interest to complete the auto-/peer-graded programming assignments.
- **Distinction Track** – Estimated Time Commitment: 8 – 12 Hours Per Week. In addition to completing the auto-graded weekly quizzes from the Standard Track, students in this track will also complete auto-/peer-graded programming assignments, including (but not limited to) the next set of enhancements to the iRemember app begun in Dr. Porter's MOOC on [Programming Mobile Applications for Android Handheld Systems](#). The programming assignments will involve writing concurrent Android Services and Applications using its pattern-oriented frameworks written in Java. This track is designed for those students wishing to achieve mastery of the POSA MOOC material and to understand its application in realistic project context. It's also mandatory for students in the Signature Track to successfully achieve a Verified Certificate with Distinction to take the Capstone project at the end of the MoCCA Specialization.

Students need not explicitly choose which of these two Tracks to take since Statement of Accomplishment will automatically be generated corresponding to the work students successfully perform in the POSA MOOC. The final grade for the course will be calculated as follows:

1. **Weekly quizzes** (Weighting: 100% for "Normal Track" and 30% for "Distinction Track")

Each quiz will contain a number of equally-weighted questions. You will normally have 14 days to submit the quiz. There will be roughly eight quizzes.

2. **Programming assignments** (Weighting: 70% for "Distinction Track" and not applicable for "Normal Track")

You will have N programming assignments (where $N = 5$ or 6) by the end of the course, including the iRemember app. You will have roughly 14 days to submit your solution. Each assignment will account for $1/N$ th of the total programming assignment points. The total weekly assignment score will be 70% of the final course score for students taking the "Distinction Track".

See class.coursera.org/posa-002/wiki/SoA for grading criteria

Overview of Assignments & Assessments

- Programming assignments should be written in Java using Eclipse
- All source code for assignments & examples available at GitHub
- The assignments will provide you with a range of experience with Android programming
- Assessments will be done by auto-grading & peer-grading
- Statements of Accomplishment require completing quizzes and/or assignments
- You can also take this MOOC solely to expand your knowledge, without submitting quizzes or assignments



Overview of the Course Topics in Section 1

Overview of Topics Covered in Section 1

- **Section 0: Course Introduction**
 - Part 1: Overview of Mobile Cloud Computing with Android
 - Part 2: Course Structure & Topics
 - Part 3: Course Prerequisites & Learning Strategies
 - Part 4: Overview of Patterns & Frameworks
- **Section 1: Android Concurrency**
 - Module 1: Concurrency Motivations & Challenges
 - Module 2: Java Concurrency Mechanisms
 - Module 3: Android Concurrency Frameworks
- **Section 2: Android Services, Local IPC, & Security**
 - Module 1: Android Services & Local IPC
 - Module 2: Android App Security & Risks
 - Module 3: Building More Secure Android Apps
- **Section 3: Concurrency & Communication Patterns Applied in Android**
 - Module 1: Coordinating Concurrent Access to Shared Data
 - Module 2: Ensuring Only One Looper Per Thread
 - Module 3: Passing Commands to Threads & Services
 - Module 4: Passing Message Requests to Threads
 - Module 5: Decoupling Synchronous & Asynchronous Processing
 - Module 6: Supporting Object-Oriented Remote Method Calls

Overview of Topics Covered in Section 1

Section 1: Android Concurrency

- *Module 1: Concurrency Motivations & Challenges*
 - Part 1: Concurrency Motivations
 - Part 2: Concurrency Challenges
- *Module 2: Java Concurrency Mechanisms*
 - Part 1: Overview of Java Threads (Part 1)
 - Part 2: Overview of Java Threads (Part 2)
 - Part 3: Motivating Java Synchronization & Scheduling Mechanisms
 - Part 4: Java Synchronization & Scheduling Classes
 - Part 5: Java ReentrantLock
 - Part 6: Java ReentrantReadWriteLock
 - Part 7: Java Semaphore
 - Part 8: Java ConditionObject
 - Part 9: Java CountdownLatch
 - Part 10: Java Synchronization & Scheduling Example
 - Part 11: Java Built-in Monitor Objects
- *Module 3: Android Concurrency Frameworks*
 - Part 1: Overview of Android Concurrency Frameworks & Idioms
 - Part 2: Android Looper
 - Part 3: Overview of Android Handler
 - Part 4: Posting & Processing Runnables to Android Handler
 - Part 5: Sending & Handling Messages to Android Handler
 - Part 6: the AsyncTask Framework
 - Part 7: Programming with Android Concurrency Frameworks (Part 1)
 - Part 8: Programming with Android Concurrency Frameworks (Part 2)

Overview of Topics Covered in Section 1

- **Section 1 – Android Concurrency**



Overview of Topics Covered in Section 1

- **Section 1 – Android Concurrency**
 - Concurrency Motivations & Challenges



Overview of Topics Covered in Section 1

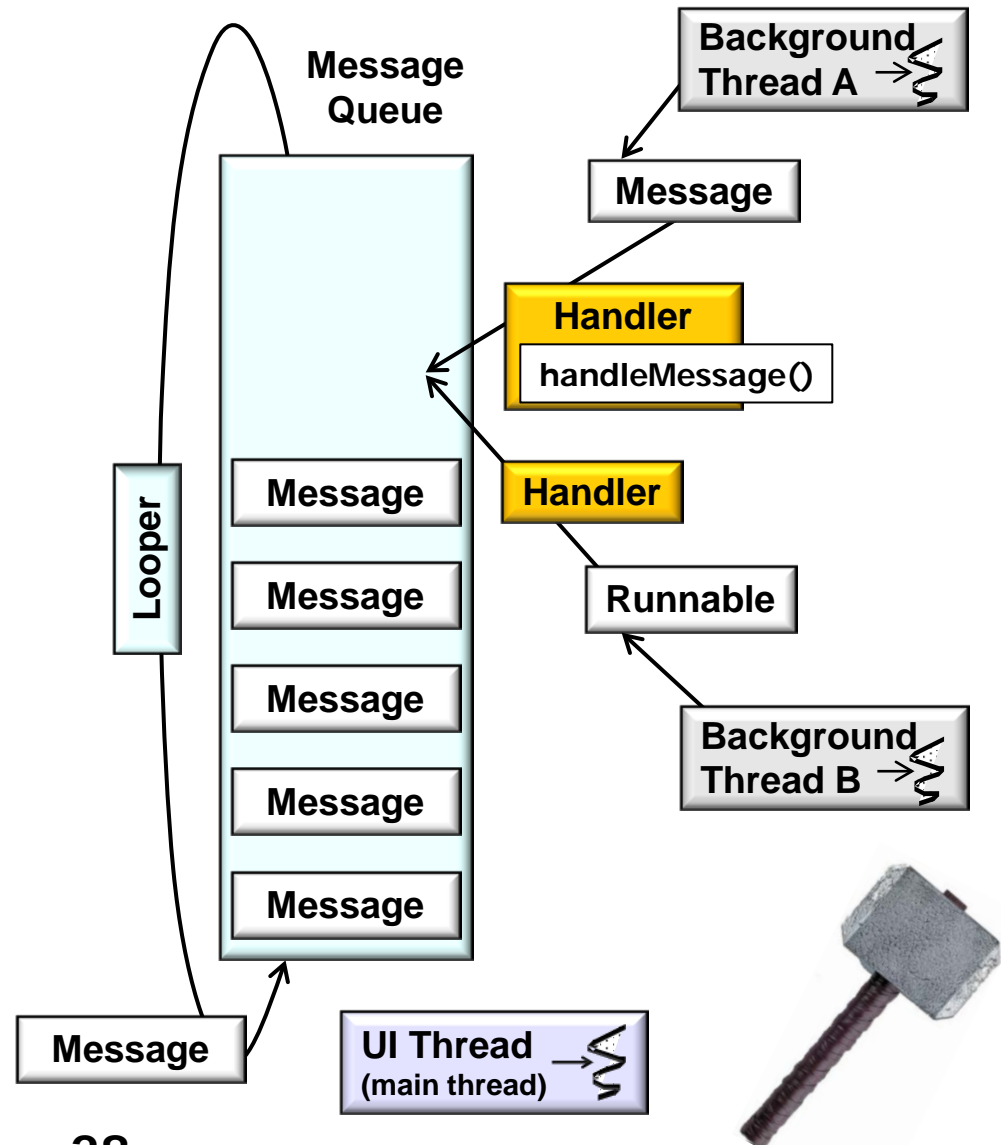
- **Section 1 – Android Concurrency**
 - Concurrency Motivations & Challenges
 - Java Concurrency Mechanisms



Overview of Topics Covered in Section 1

- **Section 1 – Android Concurrency**

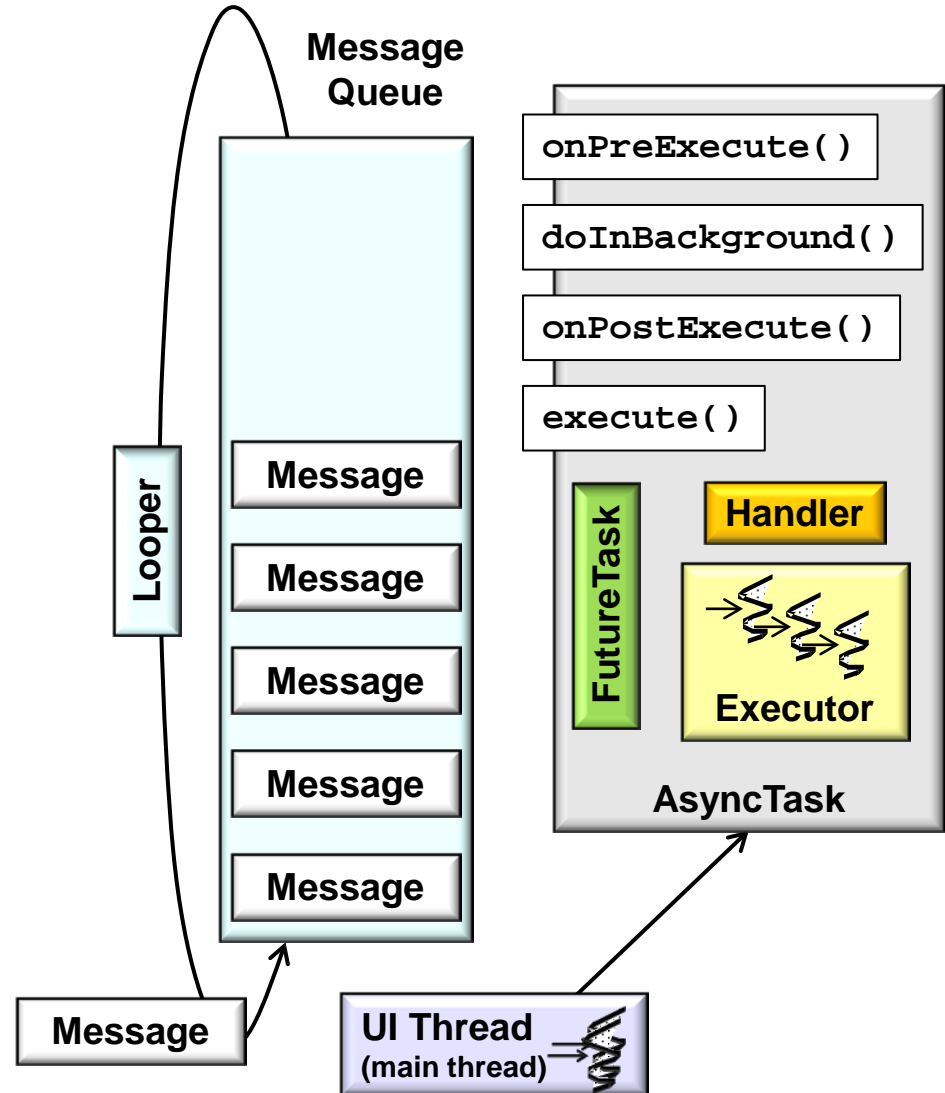
- Concurrency Motivations & Challenges
- Java Concurrency Mechanisms
- Android Concurrency Frameworks



Overview of Topics Covered in Section 1

- **Section 1 – Android Concurrency**

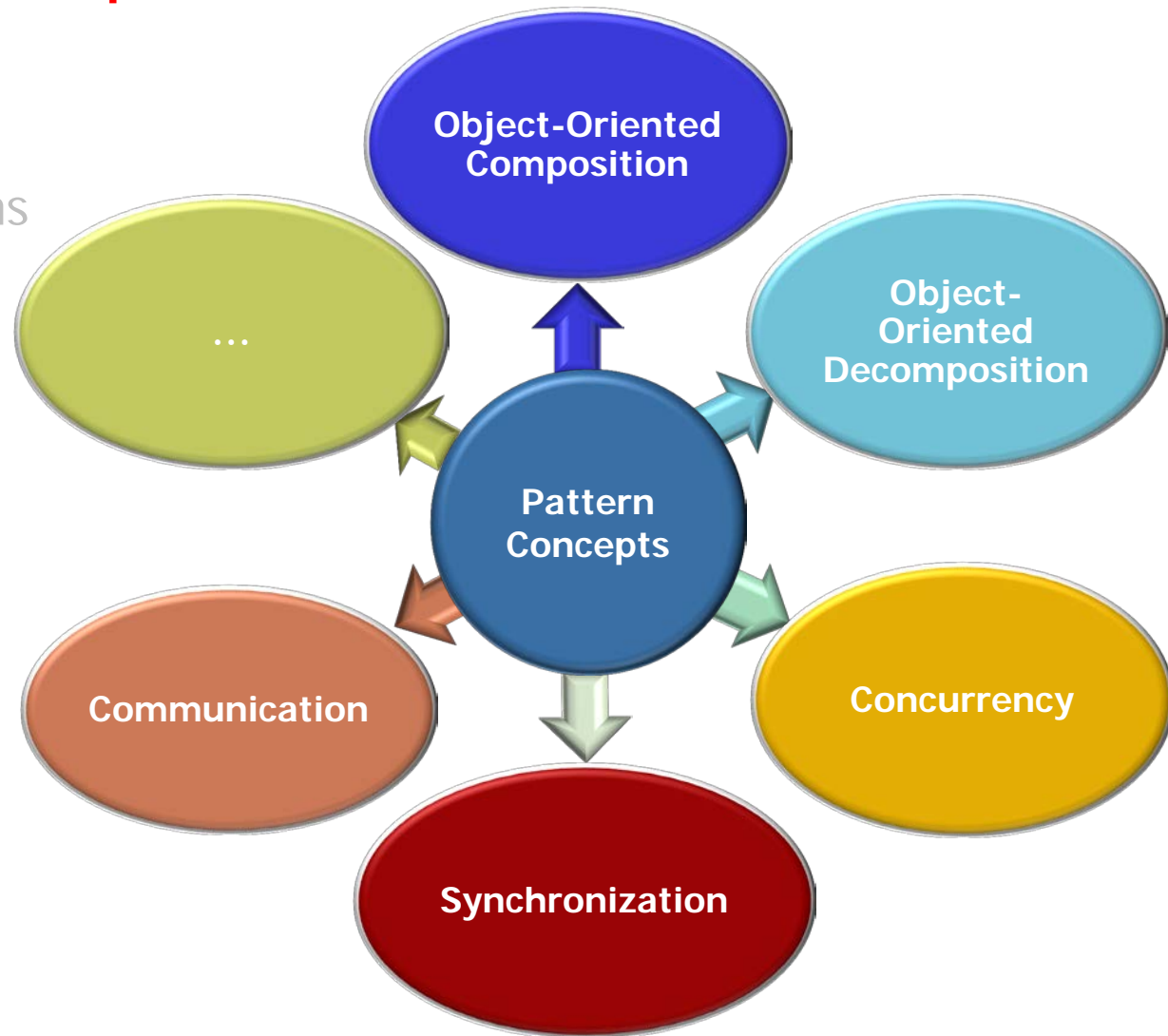
- Concurrency Motivations & Challenges
- Java Concurrency Mechanisms
- Android Concurrency Frameworks



Overview of Topics Covered in Section 1

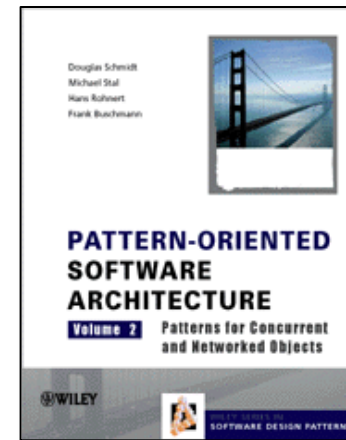
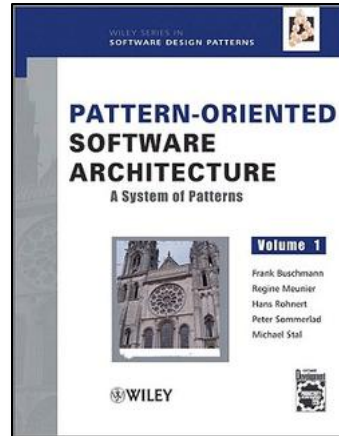
- **Section 1 – Android Concurrency**

- Concurrency Motivations & Challenges
- Java Concurrency Mechanisms
- Android Concurrency Frameworks



Overview of Topics Covered in Section 1

- **Section 1 – Android Concurrency**
 - Concurrency Motivations & Challenges
 - Java Concurrency Mechanisms
 - Android Concurrency Frameworks



Android's concurrency frameworks apply many POSA & GoF patterns

Overview of the Course Topics in Section 2

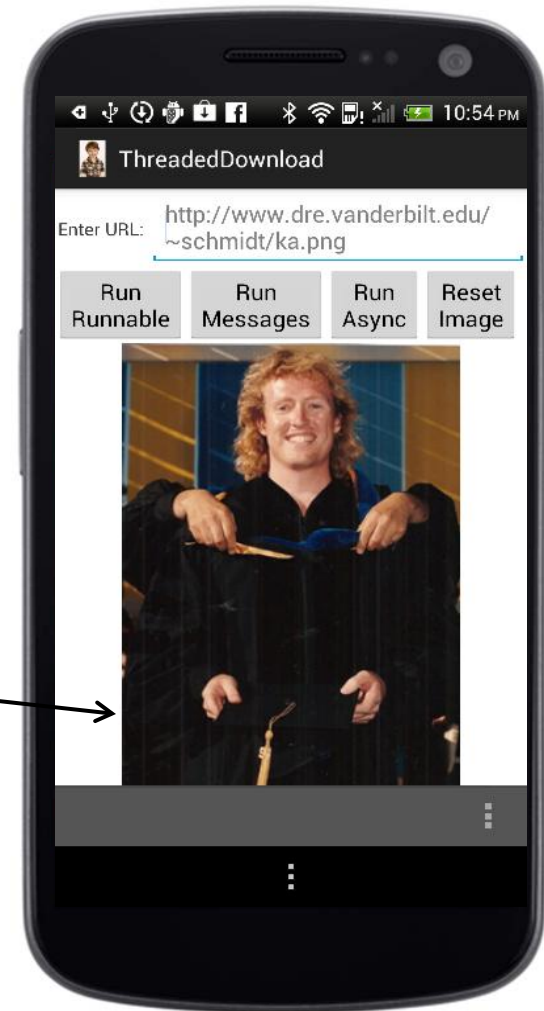
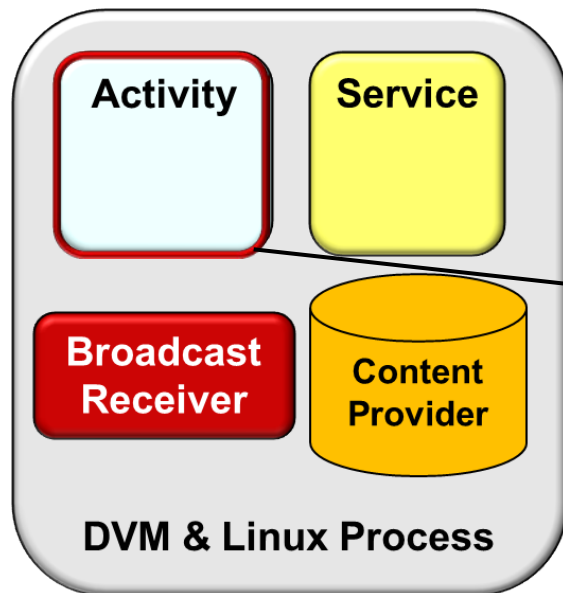
Overview of Topics Covered in Section 2

• **Section 2: Android Services, Local IPC, & Security**

- *Module 1: Android Services & Local IPC*
 - Part 1: Overview of Bound & Unbound Services
 - Part 2: Activity & Service Communication
 - Part 3: Overview of the Android Binder
- *Module 2: Android App Security & Risks*
 - Part 1: Overview of the Android App Security Model
 - Part 2: App User Accounts Permissions
 - Part 3: How Android Protects Your App Limitations of that Protection
 - Part 4: An Overview of App Privilege Escalation Attacks
 - Part 5: Understanding the Risks of Storing Sensitive Data in Apps
 - Part 6: App Code Obfuscation Security Limitations
 - Part 7: User Manipulation via Malware
- *Module 3: Building More Secure Android Apps*
 - Part 1: Properly Storing Private Data
 - Part 2: The Importance of Input Validation
 - Part 3: Securely Handling Intents & IPC
 - Part 4: Avoiding Privilege Escalation Attacks
 - Part 5: Understanding Broadcast Receiver Priority Attacks
 - Part 6: Securing Embedded WebViews
 - Part 7: Understanding the Risks of Hiding Information from the User
 - Part 8: Protecting Sensitive UI Elements from Manipulation
 - Part 9: Avoiding the Leakage of Sensitive Information through Logging

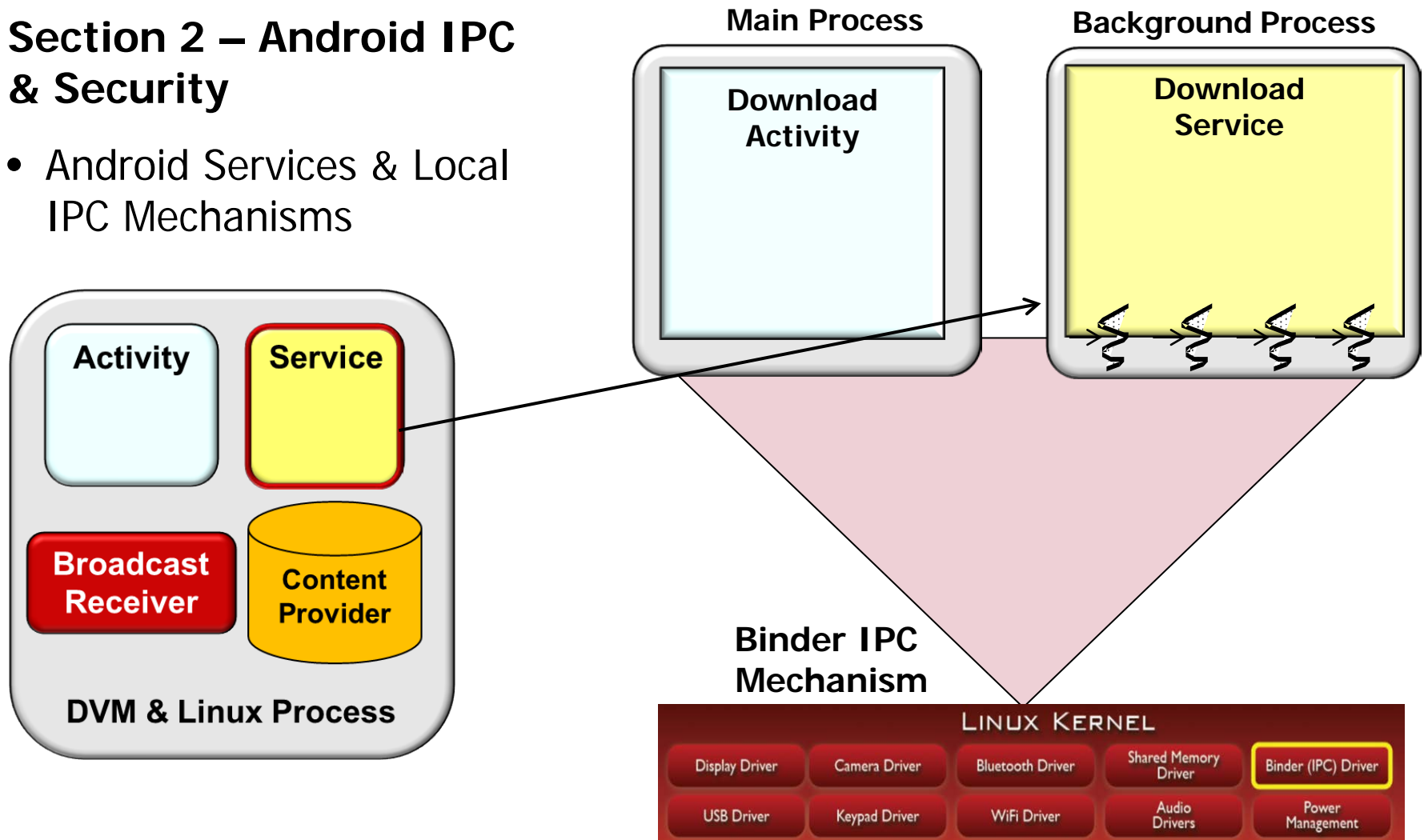
Overview of Topics Covered in Section 2

- **Section 2 – Android IPC & Security**
- Android Services & Local IPC Mechanisms



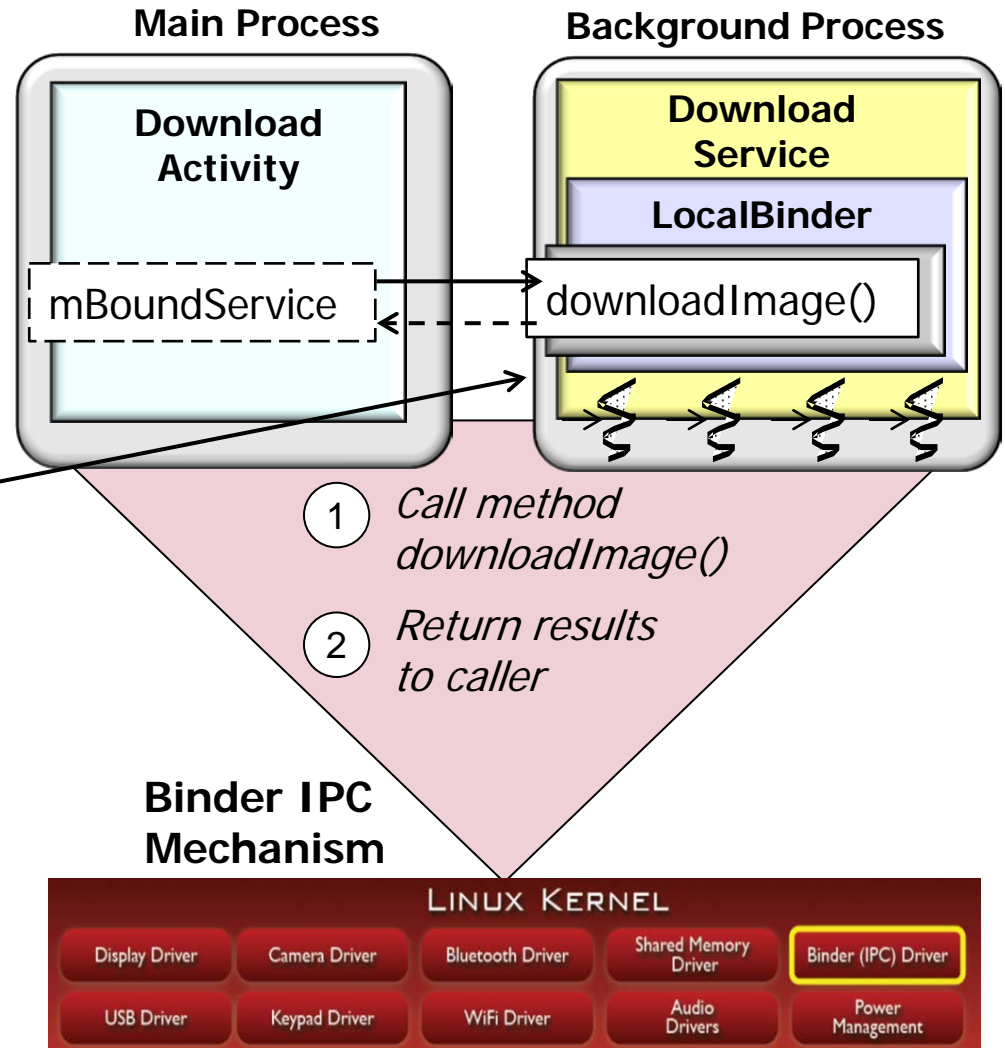
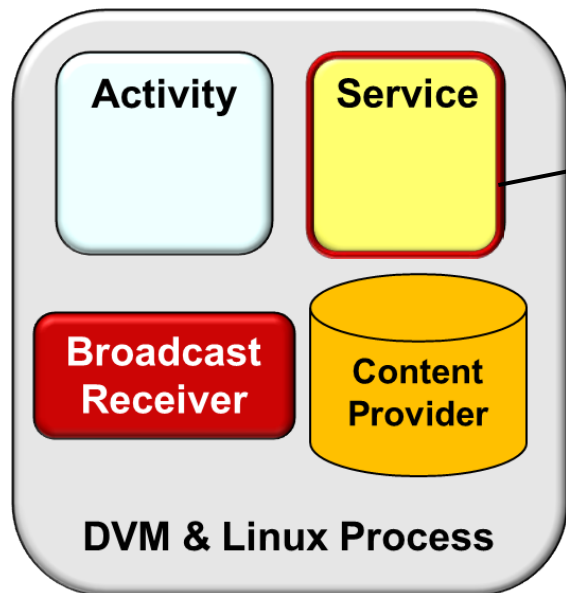
Overview of Topics Covered in Section 2

- **Section 2 – Android IPC & Security**
 - Android Services & Local IPC Mechanisms



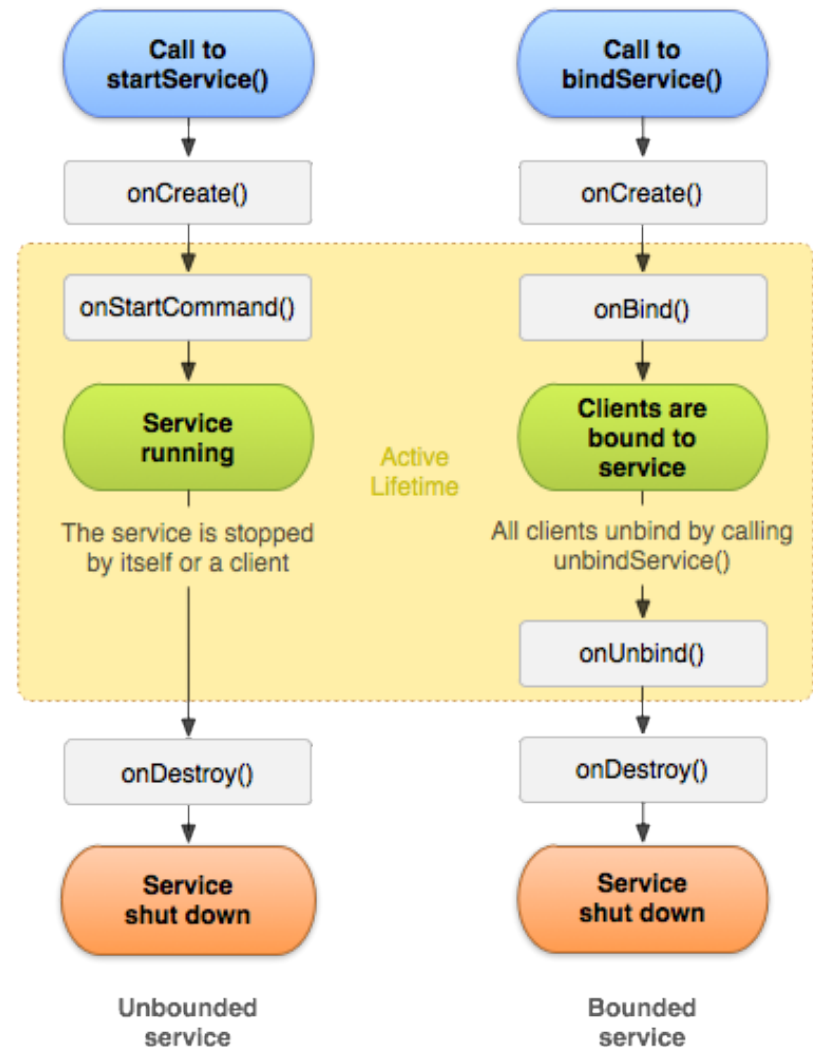
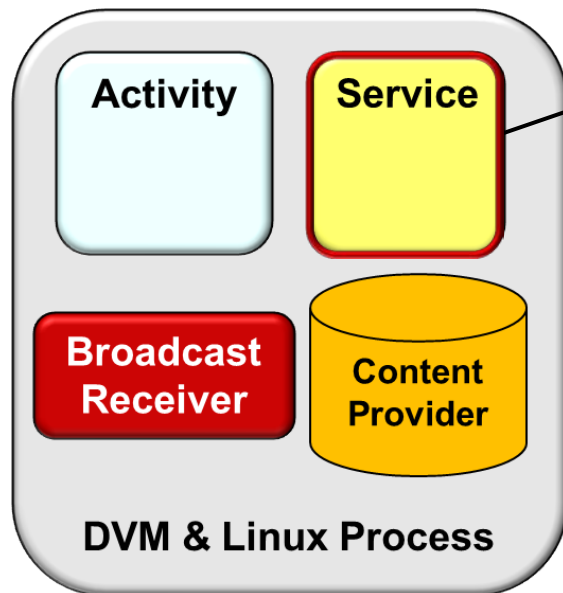
Overview of Topics Covered in Section 2

- **Section 2 – Android IPC & Security**
- Android Services & Local IPC Mechanisms



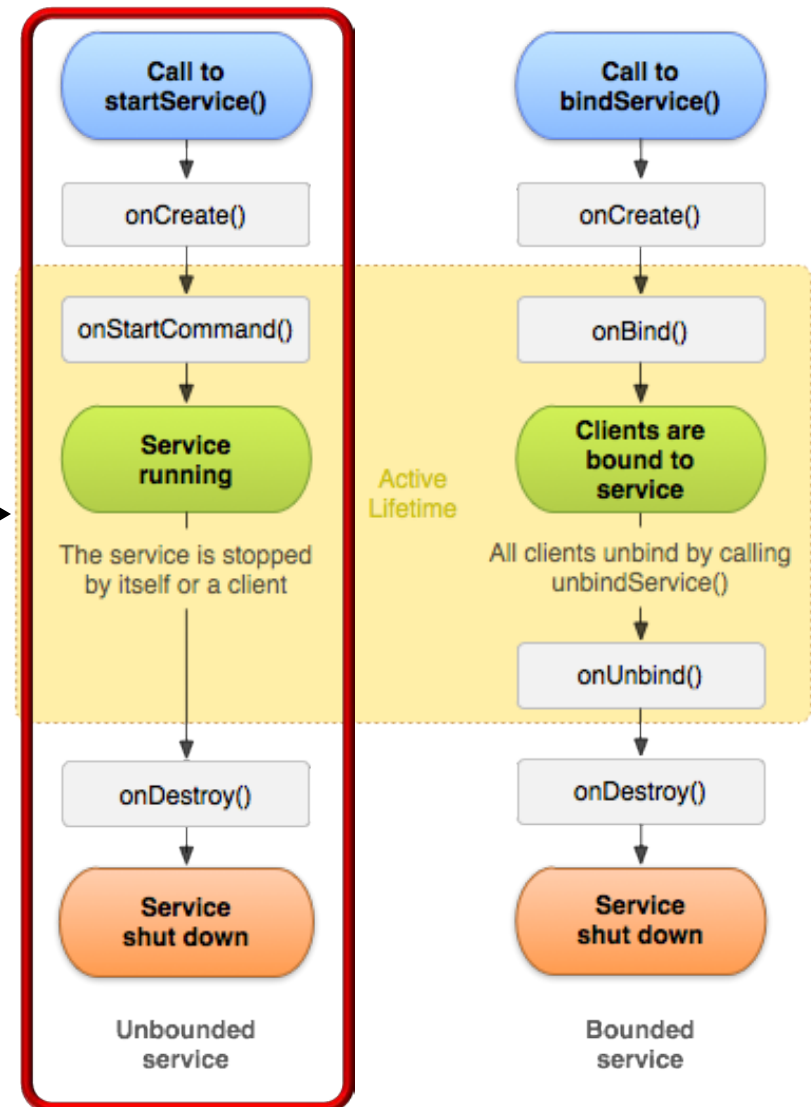
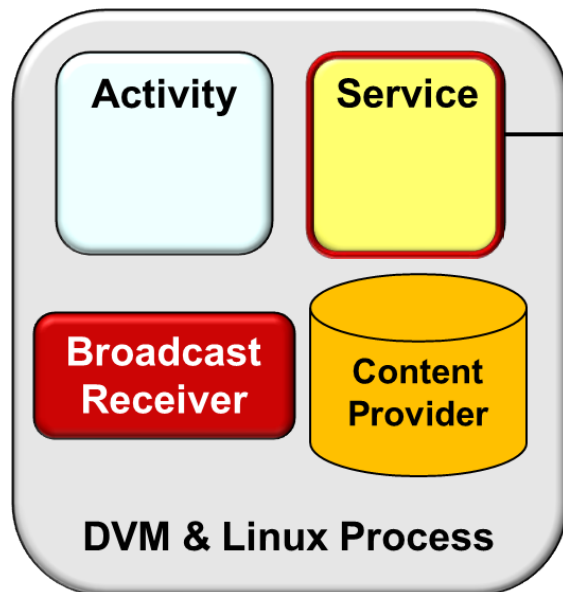
Overview of Topics Covered in Section 2

- **Section 2 – Android IPC & Security**
 - Android Services & Local IPC Mechanisms



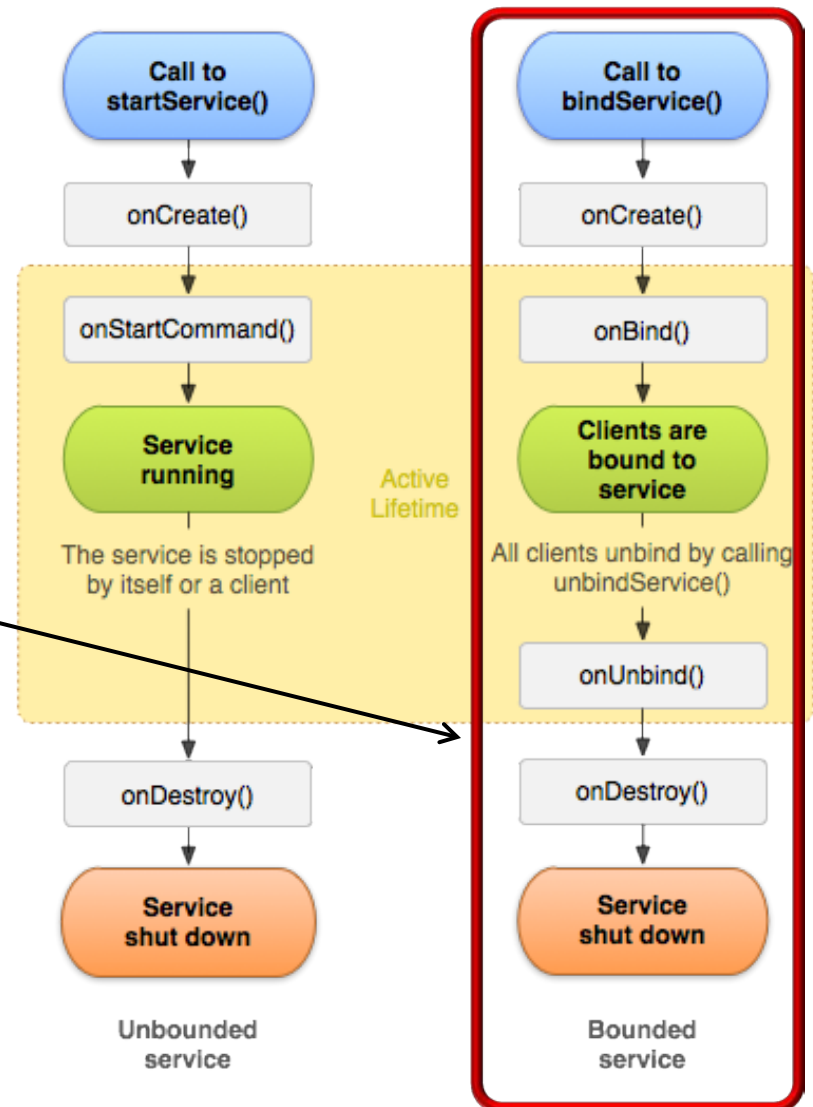
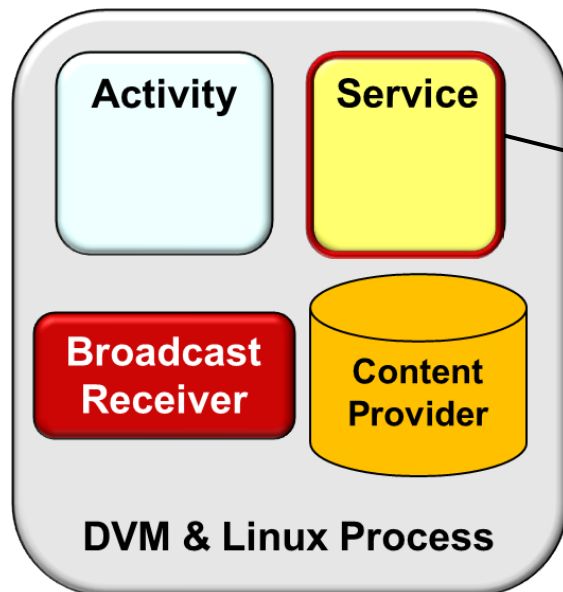
Overview of Topics Covered in Section 2

- **Section 2 – Android IPC & Security**
 - Android Services & Local IPC Mechanisms



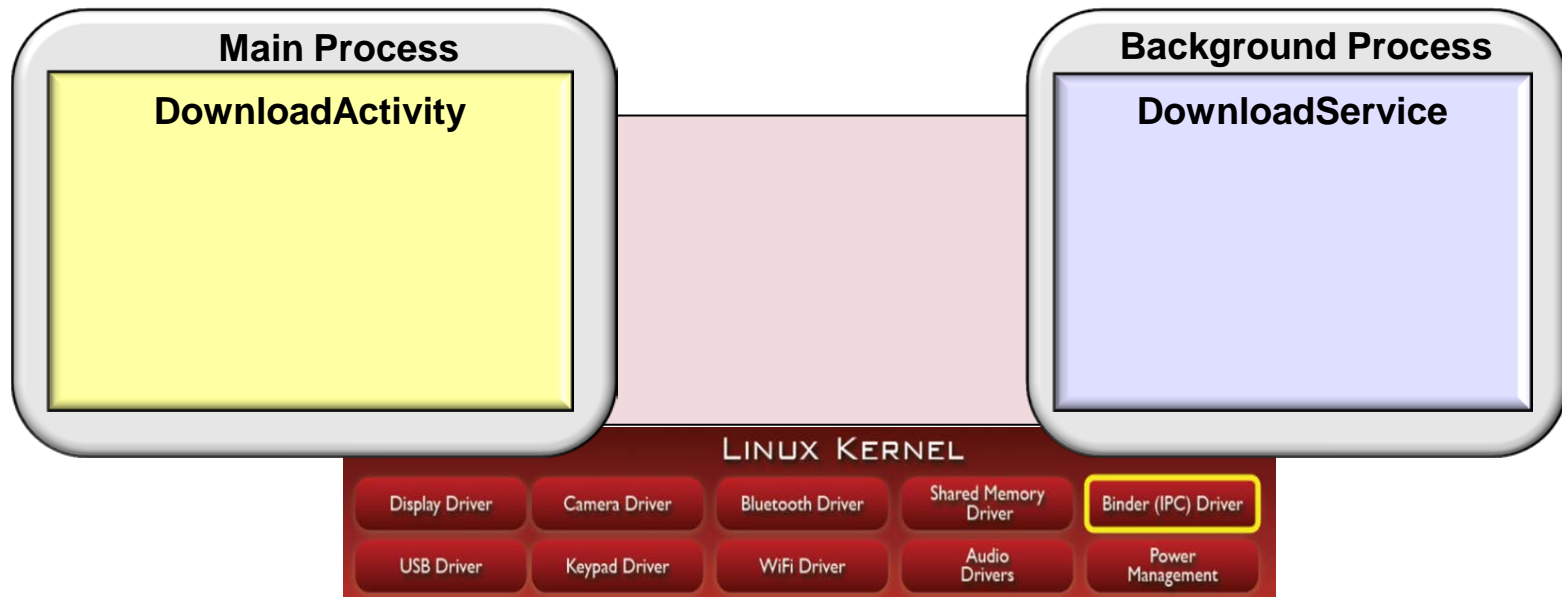
Overview of Topics Covered in Section 2

- **Section 2 – Android IPC & Security**
 - Android Services & Local IPC Mechanisms



Overview of Topics Covered in Section 2

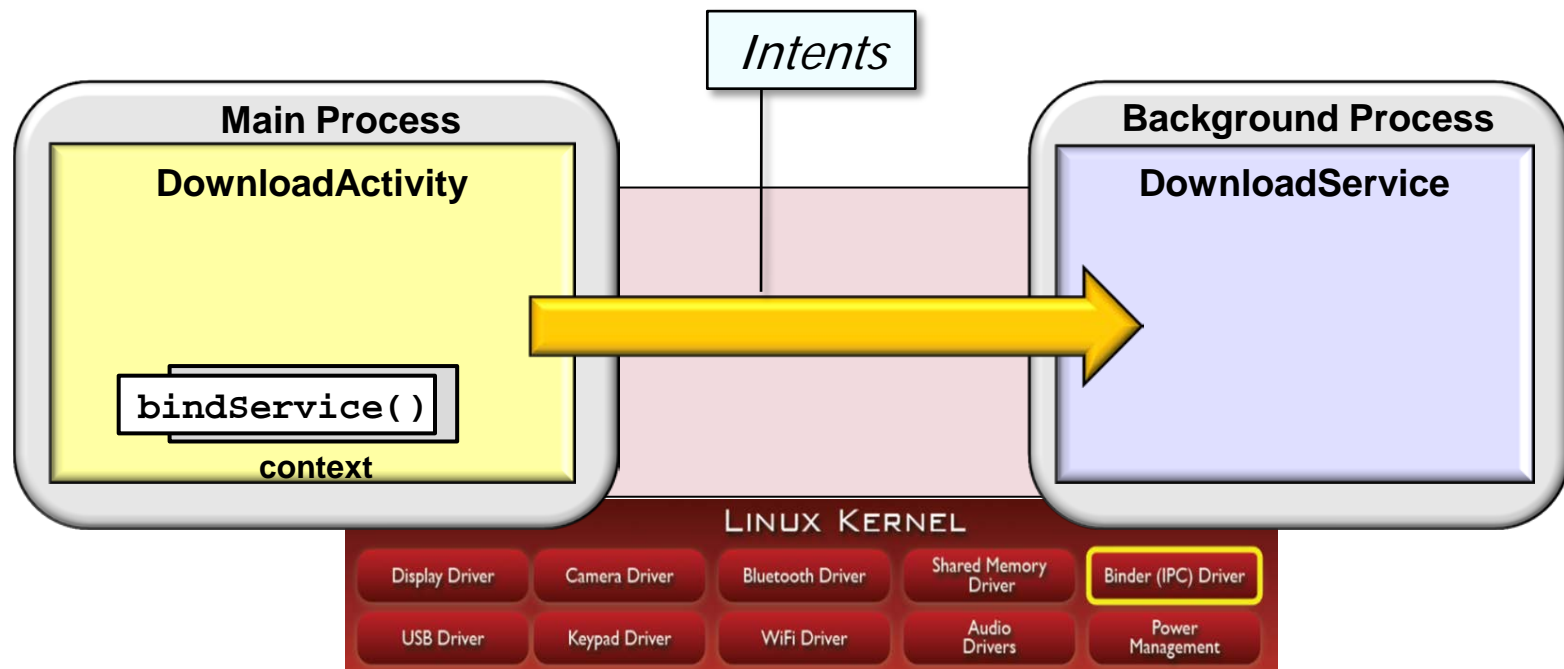
- **Section 2 – Android IPC & Security**
 - Android Services & Local IPC Mechanisms



Android Linux provides mechanisms optimized for inter-process communication

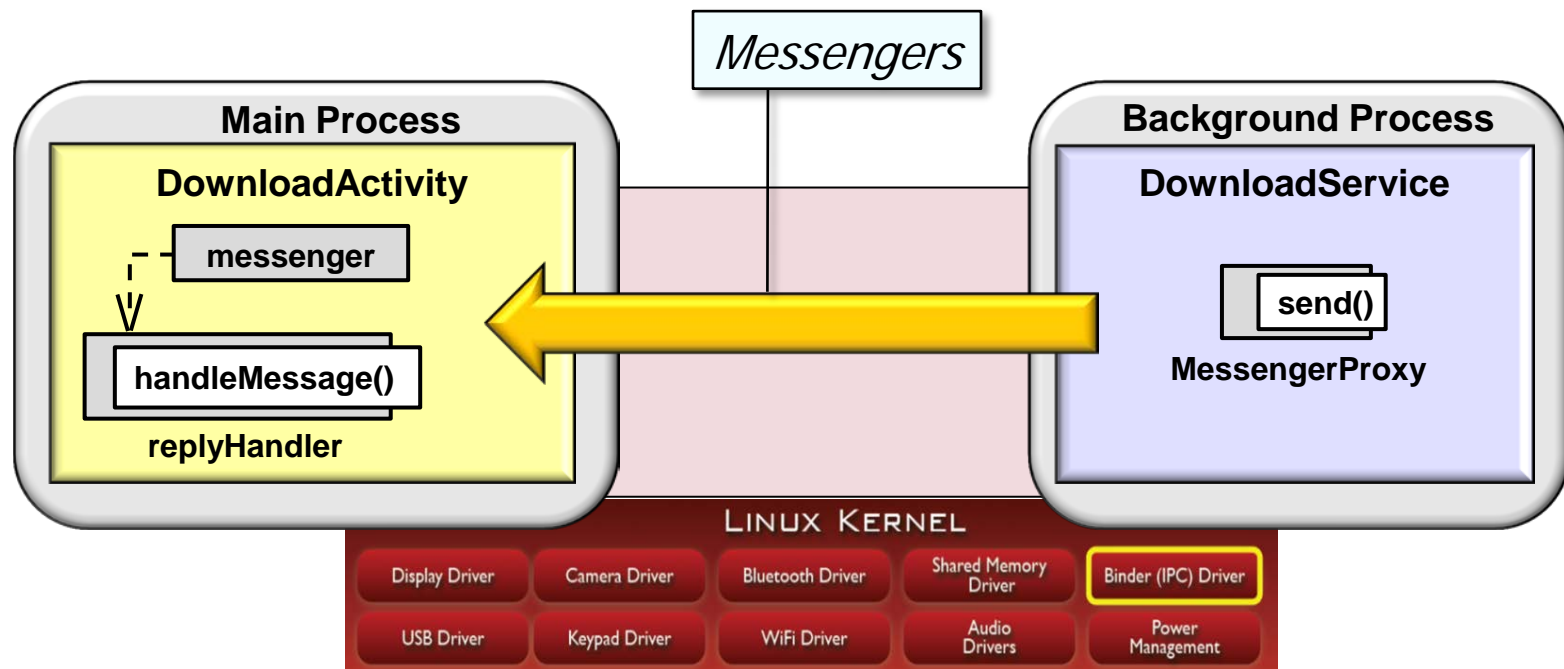
Overview of Topics Covered in Section 2

- **Section 2 – Android IPC & Security**
 - Android Services & Local IPC Mechanisms



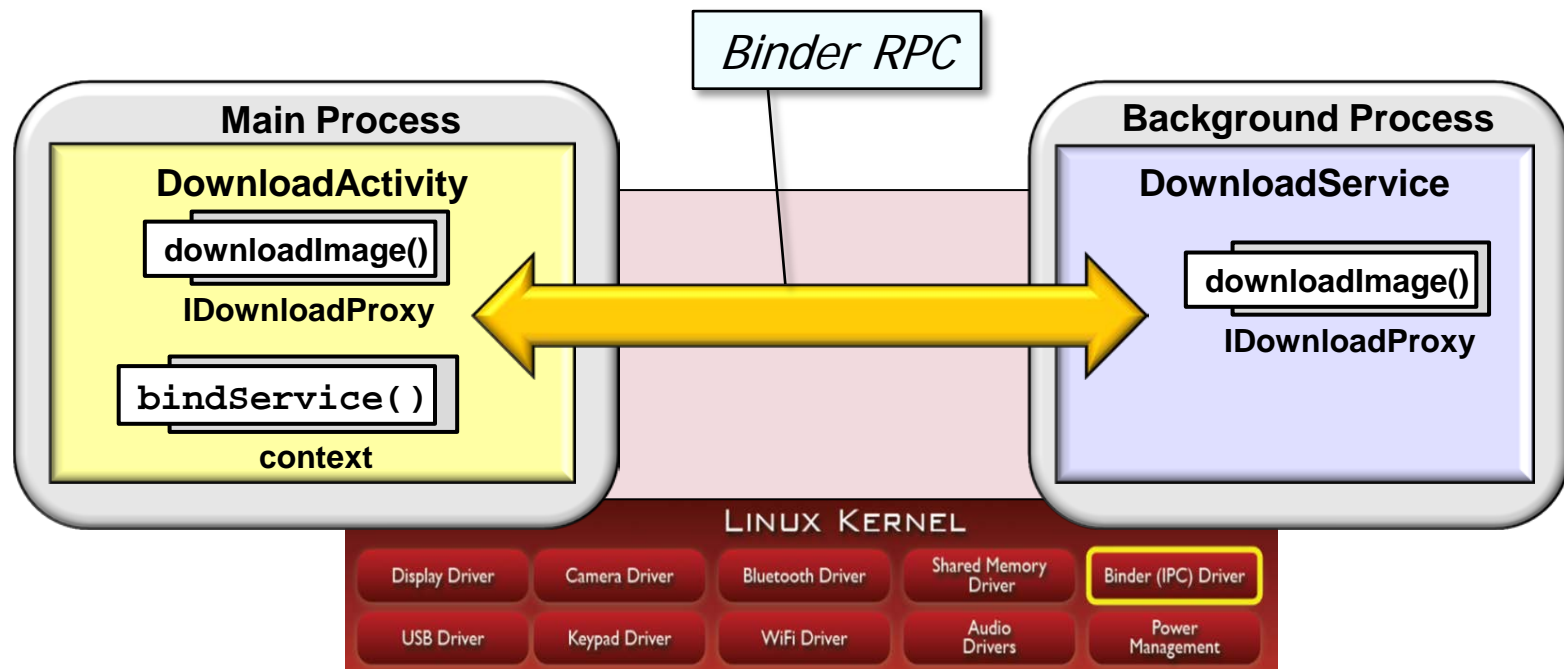
Overview of Topics Covered in Section 2

- **Section 2 – Android IPC & Security**
 - Android Services & Local IPC Mechanisms



Overview of Topics Covered in Section 2

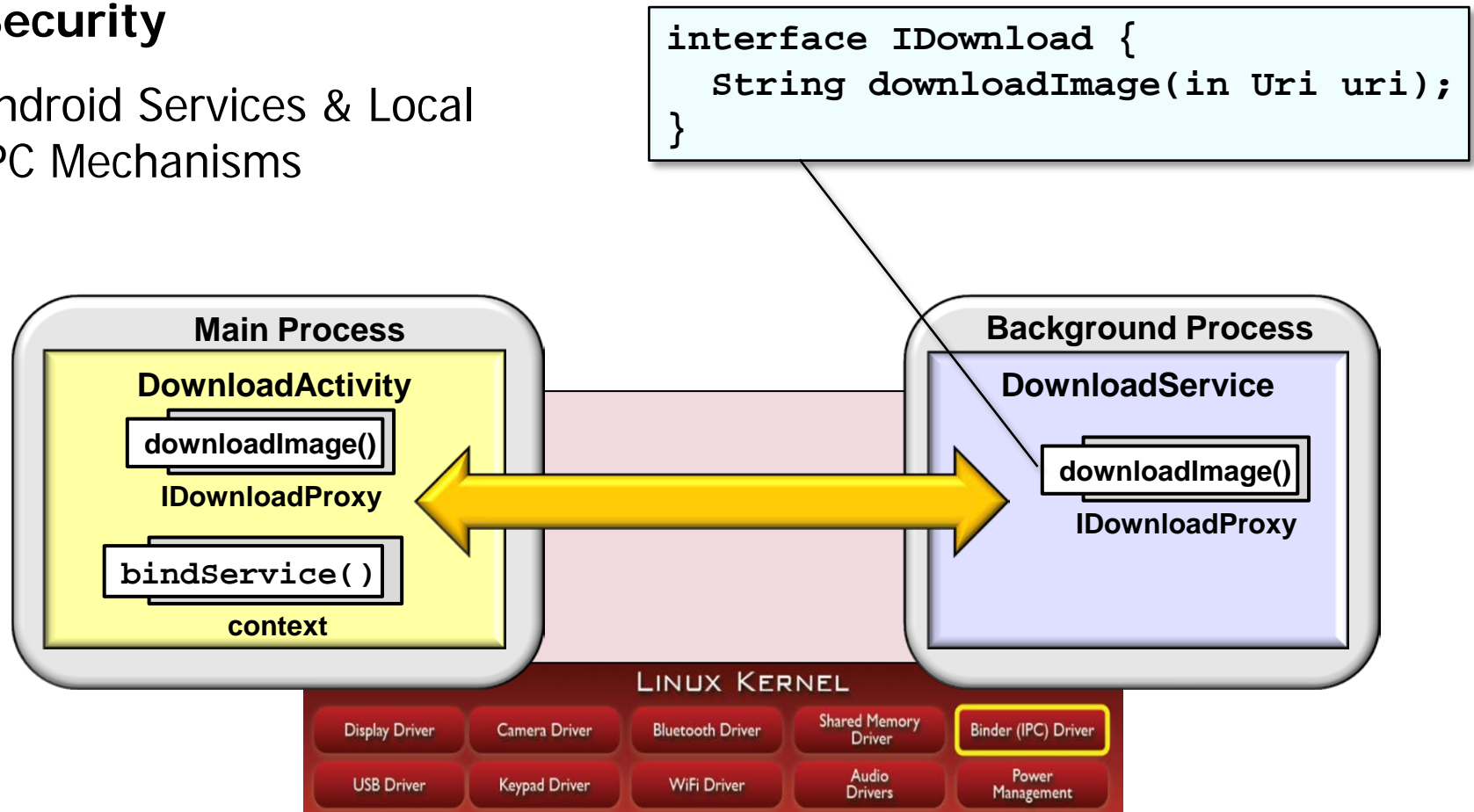
- **Section 2 – Android IPC & Security**
 - Android Services & Local IPC Mechanisms



Overview of Topics Covered in Section 2

- Section 2 – Android IPC & Security

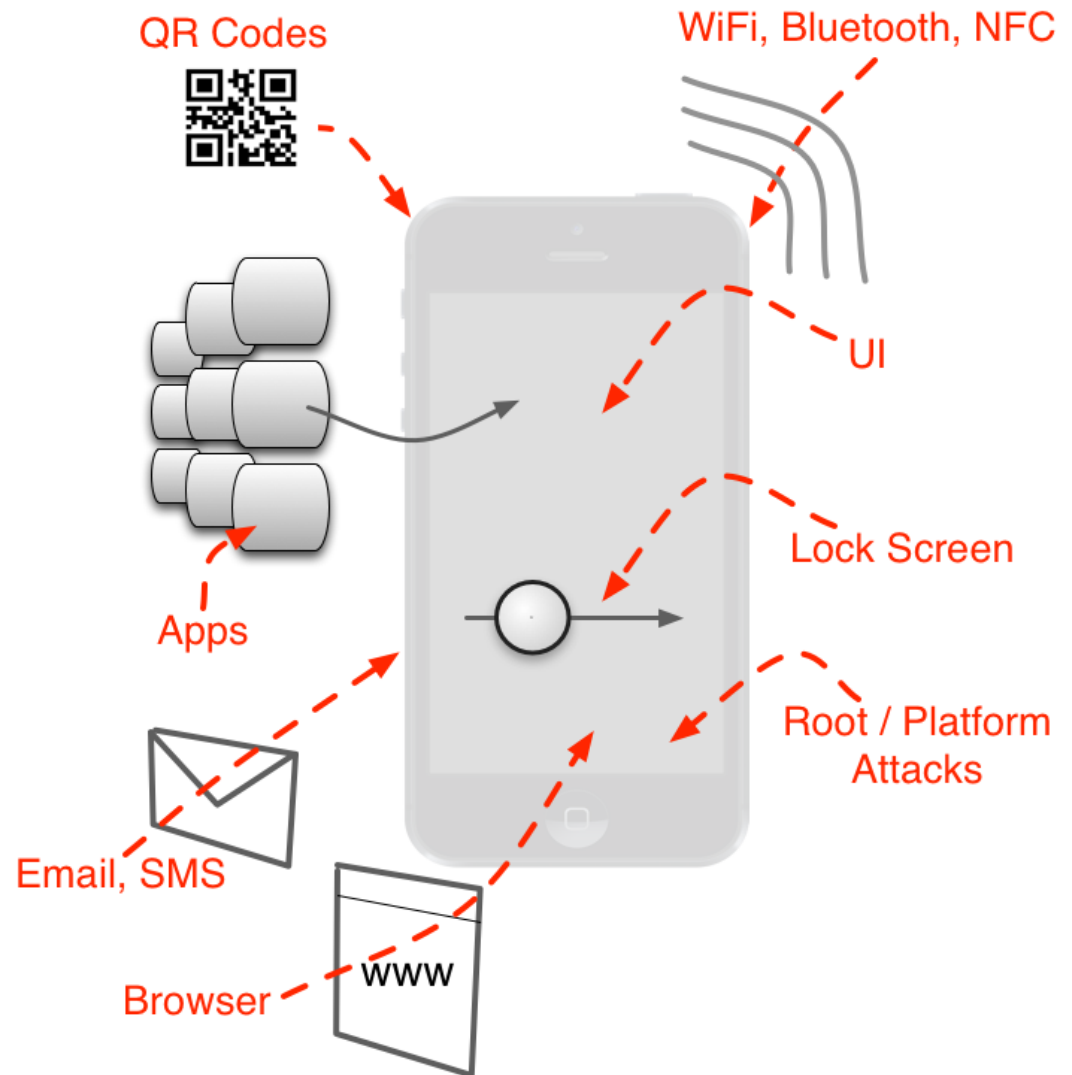
- Android Services & Local IPC Mechanisms



Overview of Topics Covered in Section 2

- **Section 2 – Android IPC & Security**

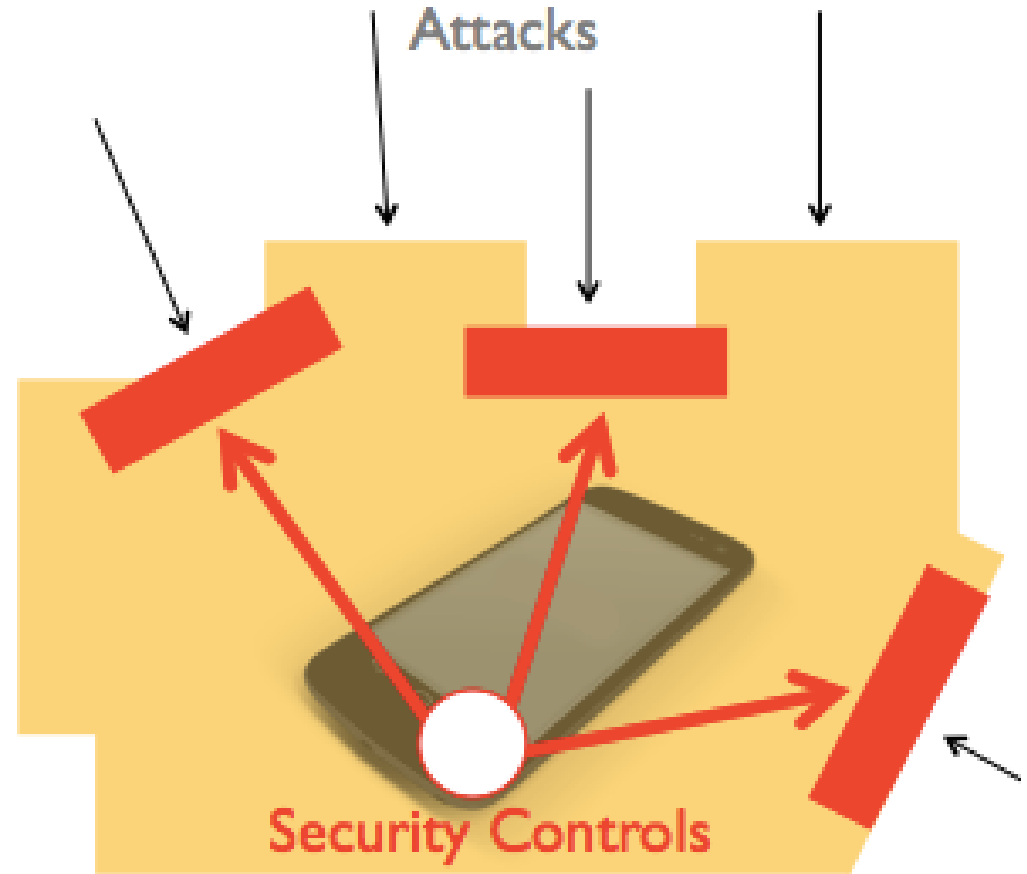
- Android Services & Local IPC Mechanisms
- Android App Security & Risks



Overview of Topics Covered in Section 2

- **Section 2 – Android IPC & Security**

- Android Services & Local IPC Mechanisms
- Android App Security & Risks
- Building More Secure Android Applications

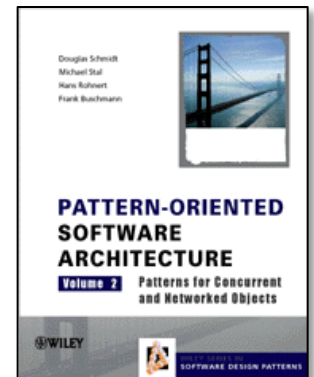
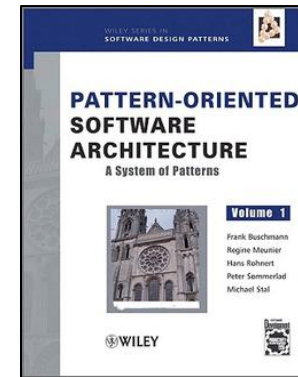


Overview of the Course Topics in Section 3

Overview of Topics Covered in Section 3

• Section 3: Concurrency & Communication Patterns Applied in Android

- *Module 1: Coordinating Concurrent Access to Shared Data*
 - Part 1: the *Monitor Object* Pattern (Part 1)
 - Part 2: the *Monitor Object* Pattern (Part 2)
- *Module 2: Ensuring Only One Looper Per Thread*
 - Part 1: the *Thread-Specific Storage* Pattern (Part 1)
 - Part 2: the *Thread-Specific Storage* Pattern (Part 2)
- *Module 3: Passing Commands to Threads & Services*
 - Part 1: the *Command Processor* Pattern (Part 1)
 - Part 2: the *Command Processor* Pattern (Part 2)
- *Module 4: Passing Message Requests to Threads*
 - Part 1: the *Active Object* Pattern (Part 1)
 - Part 2: the *Active Object* Pattern (Part 2)
- *Module 5: Decoupling Synchronous & Asynchronous Processing*
 - Part 1: the *Half-Sync/Half-Async* Pattern (Part 1)
 - Part 2: the *Half-Sync/Half-Async* Pattern (Part 2)
- *Module 6: Supporting Object-Oriented Remote Method Calls*
 - Part 1: the *Broker* Pattern (Part 1)
 - Part 2: the *Broker* Pattern (Part 2)

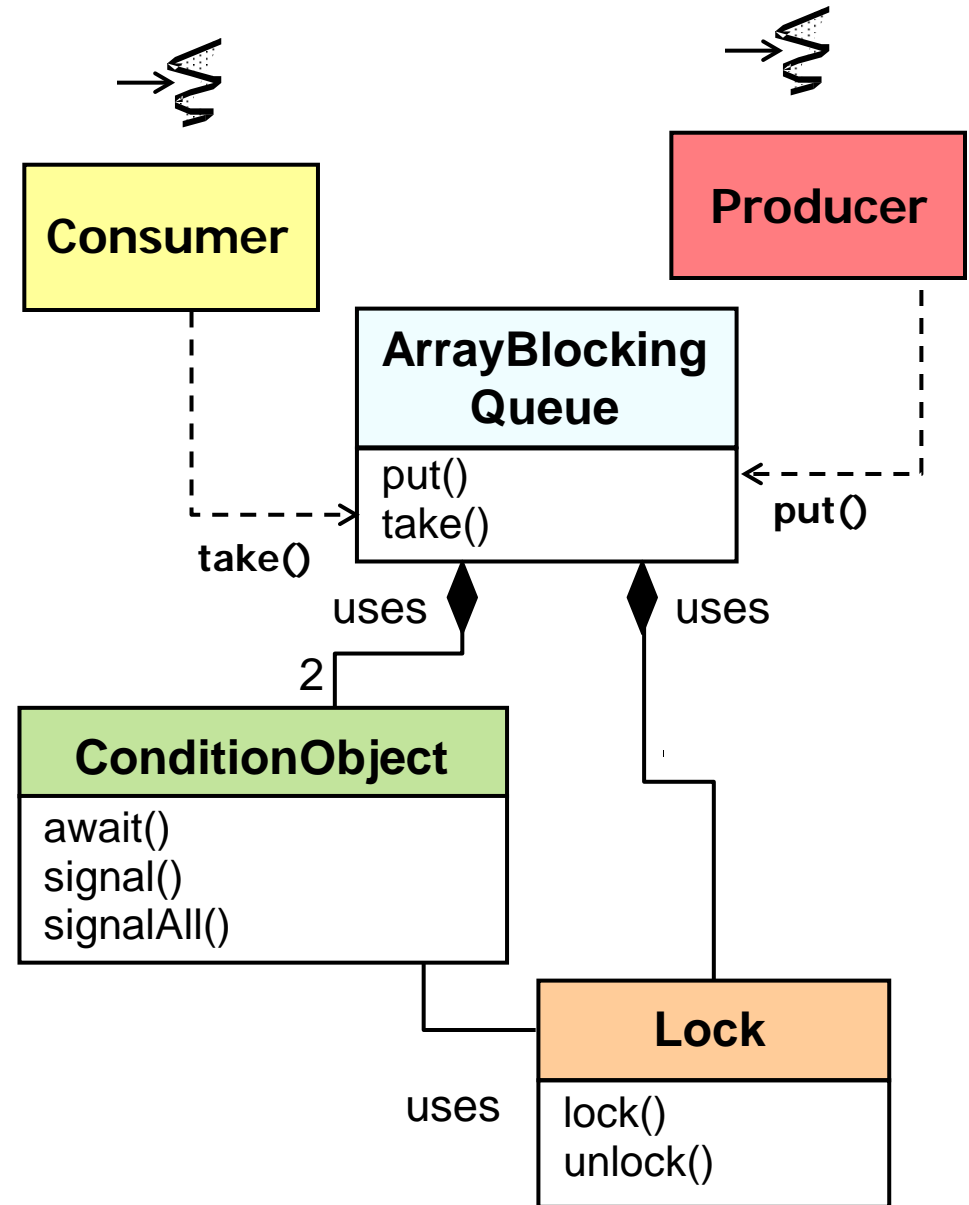


Overview of Topics Covered in Section 3

- **Section 3 – Concurrency & Communication Patterns Applied in Android**

- *Monitor Object*

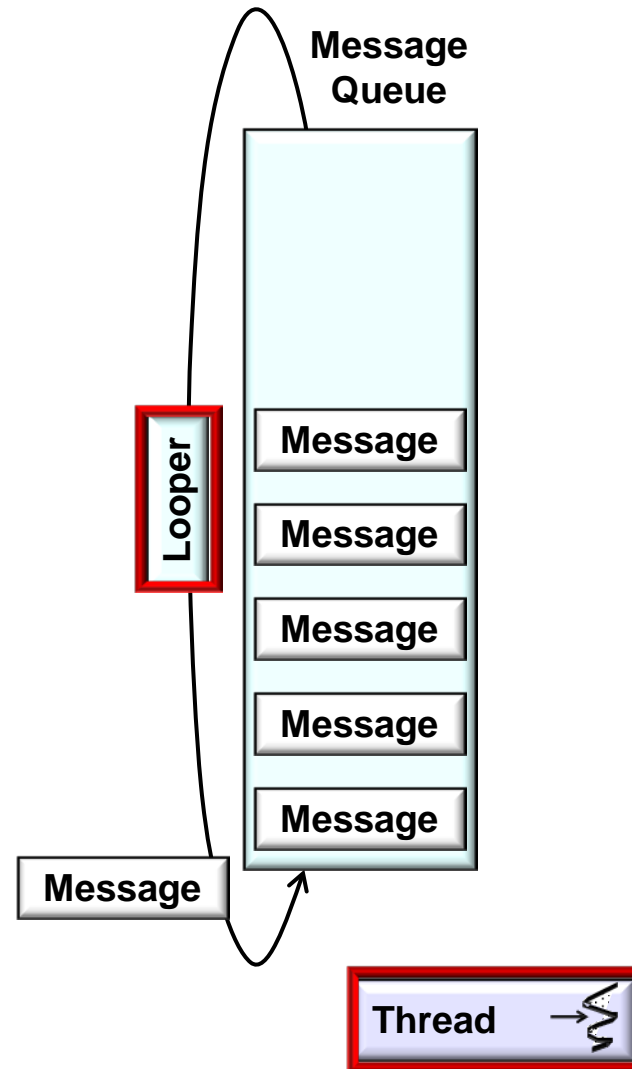
- Coordinate concurrent access to shared data



Overview of Topics Covered in Section 3

- **Section 3 – Concurrency & Communication Patterns Applied in Android**

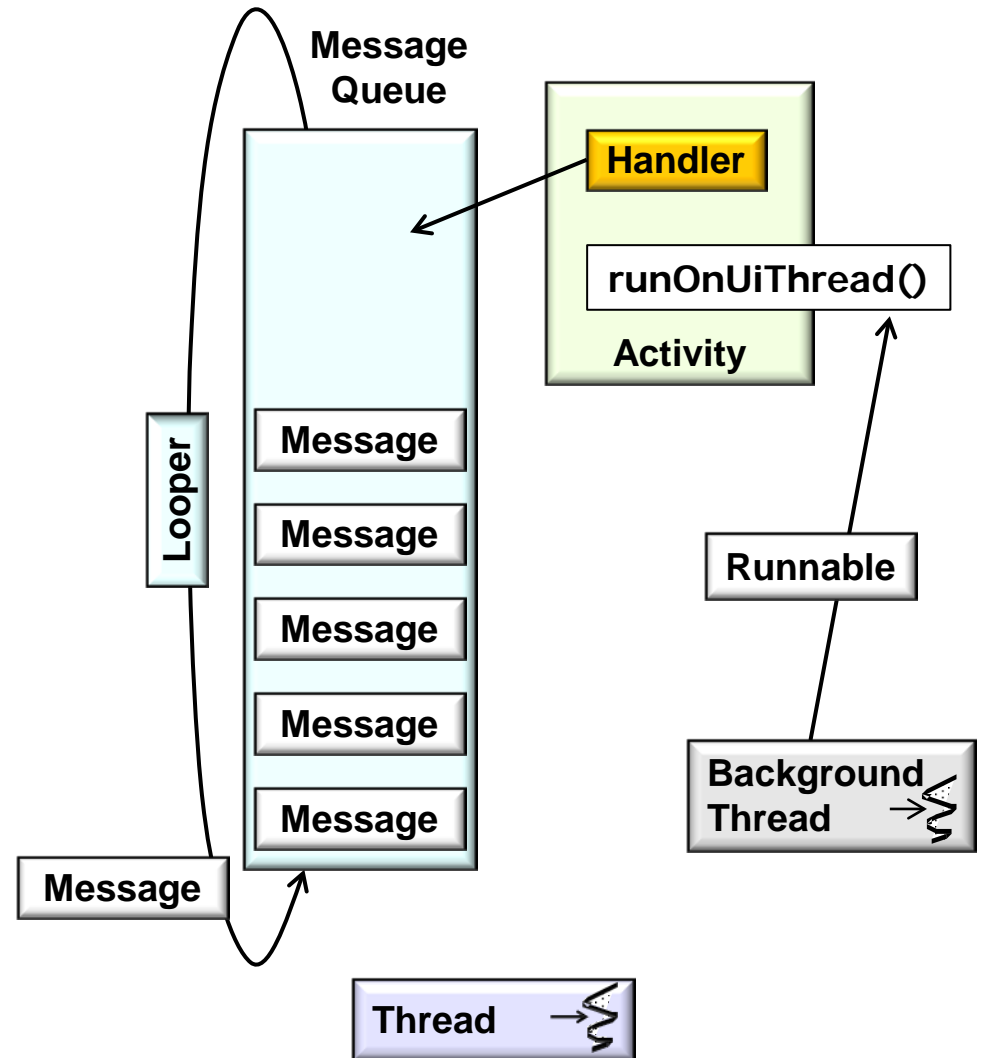
- *Monitor Object*
- *Thread-Specific Storage*
 - Ensure only one Looper per Thread



Overview of Topics Covered in Section 3

- **Section 3 – Concurrency & Communication Patterns Applied in Android**

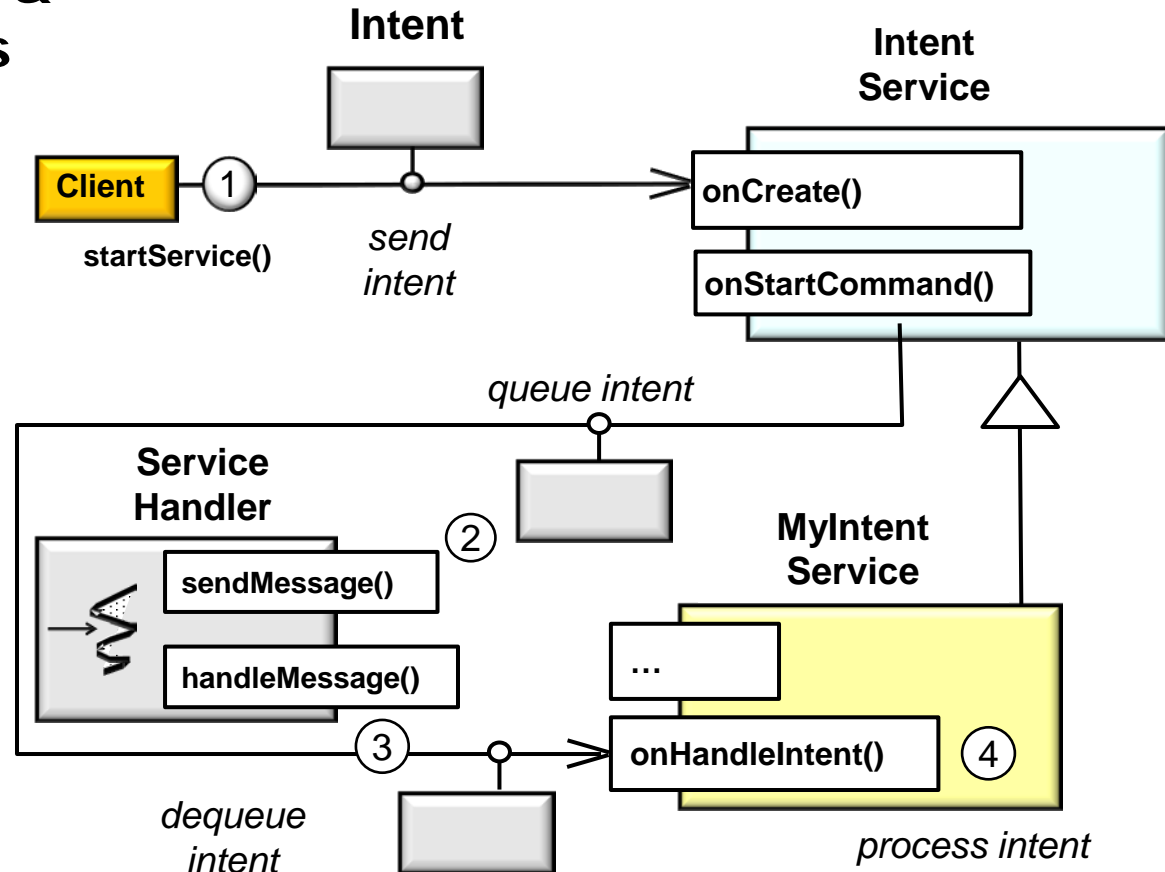
- *Monitor Object*
- *Thread-Specific Storage*
- *Command Processor*
 - Pass commands to Threads & Services



Overview of Topics Covered in Section 3

- Section 3 – Concurrency & Communication Patterns Applied in Android

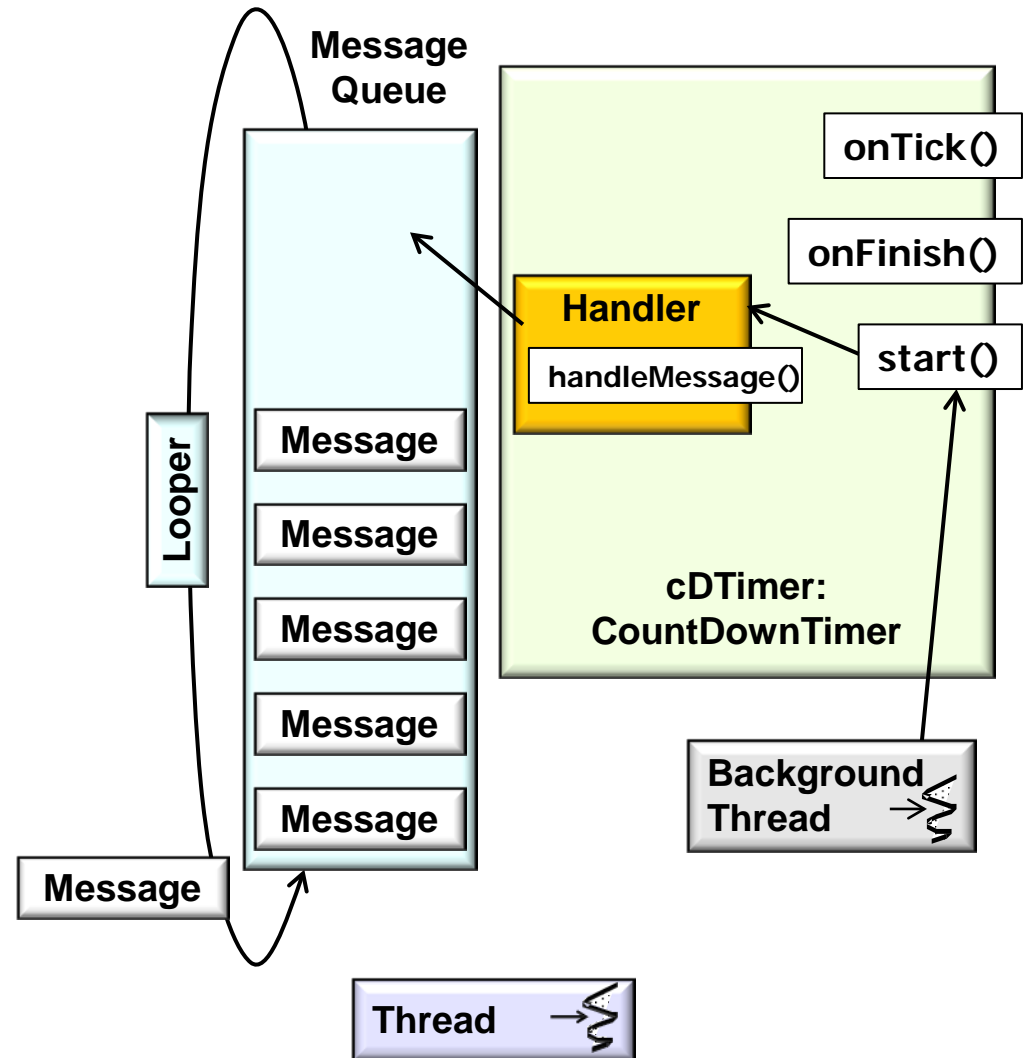
- Monitor Object*
- Thread-Specific Storage*
- Command Processor*
 - Pass commands to Threads & Services



Overview of Topics Covered in Section 3

- **Section 3 – Concurrency & Communication Patterns Applied in Android**

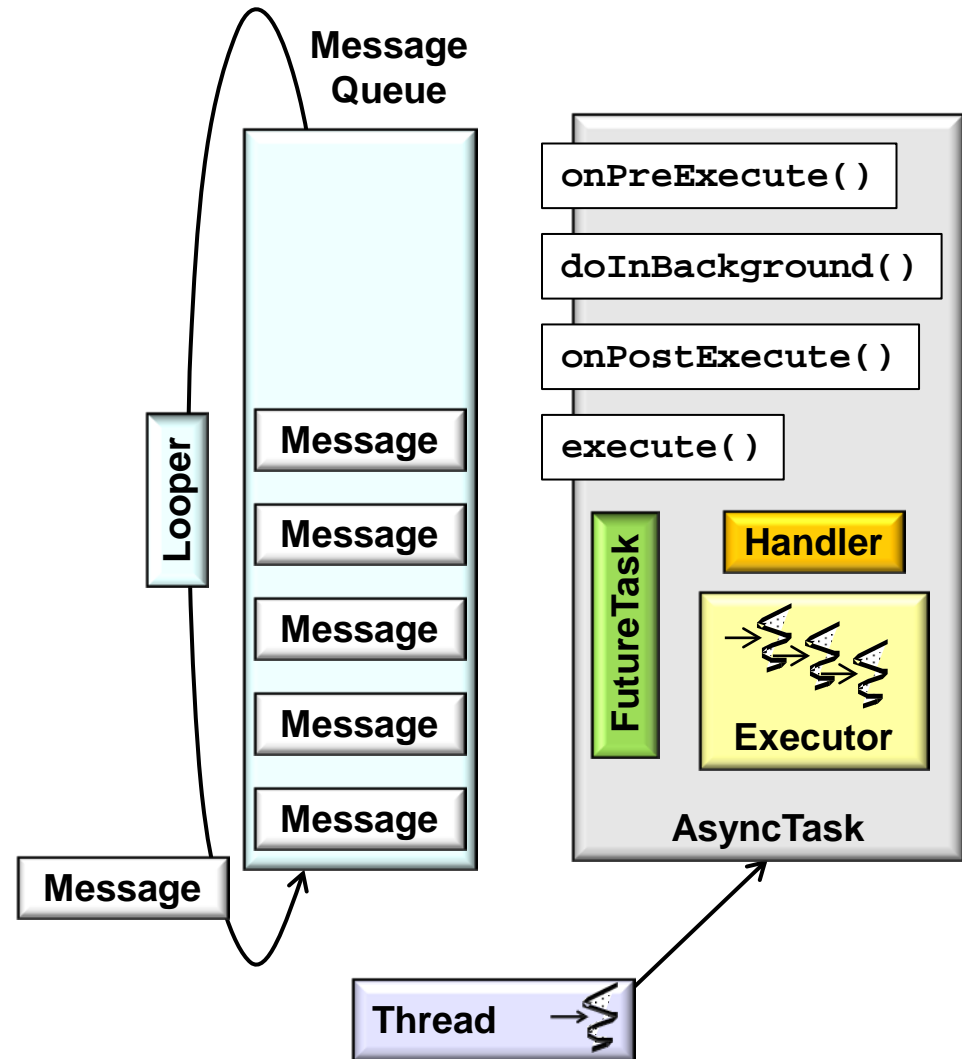
- *Monitor Object*
- *Thread-Specific Storage*
- *Command Processor*
- *Active Object*
 - Pass Message requests to Threads or Processes



Overview of Topics Covered in Section 3

- **Section 3 – Concurrency & Communication Patterns Applied in Android**

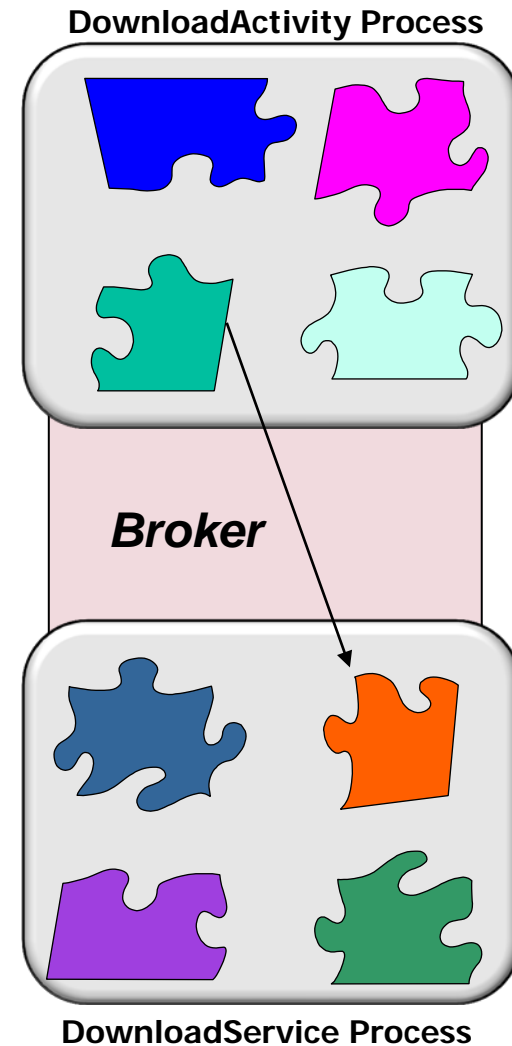
- *Monitor Object*
- *Thread-Specific Storage*
- *Command Processor*
- *Active Object*
- *Half-Sync/Half-Async*
 - Decouple synchronous & asynchronous processing



Overview of Topics Covered in Section 3

- **Section 3 – Concurrency & Communication Patterns Applied in Android**

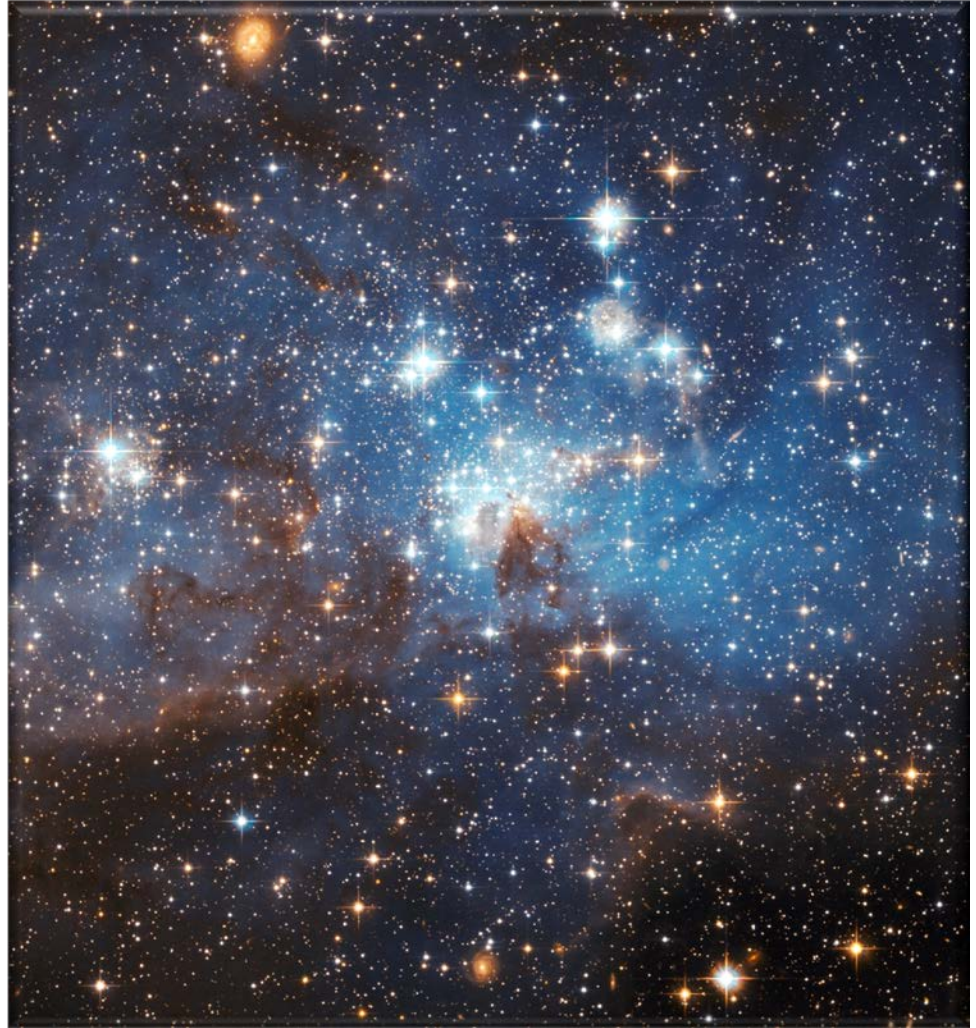
- *Monitor Object*
- *Thread-Specific Storage*
- *Command Processor*
- *Active Object*
- *Half-Sync/Half-Async*
- *Broker*
 - Support object-oriented remote method calls



Overview of Topics Covered in Section 3

- **Section 3 – Concurrency & Communication Patterns Applied in Android**

- *Monitor Object*
- *Thread-Specific Storage*
- *Command Processor*
- *Active Object*
- *Half-Sync/Half-Async*
- *Broker*



These patterns are not limited to Android or mobile device programming

Summary



Summary

- This MOOC covers a wide range of topics related to Android

[Edit Course Description](#) [Edit Session Descriptions▾](#) [Edit Session Materials▾](#)



Pattern-Oriented Software Architectures: Programming Mobile Services for Android Handheld Systems

Part of the "Mobile Cloud Computing with Android" Specialization »

In this course--the second in a trans-institution sequence of MOOCs on Mobile Cloud Computing with Android--we will learn how to apply patterns, pattern languages, and frameworks to alleviate the complexity of developing concurrent and networked services on mobile devices running Android that connect to popular cloud computing platforms.

[Preview Lectures](#)



Summary

- This MOOC covers a wide range of topics related to Android




Summary

- This MOOC covers a wide range of topics related to Android

VANDERBILT UNIVERSITY

Pattern-Oriented Software Architecture
by Dr. Douglas C. Schmidt, Dr. C. Jules White



COURSE

Discussion Forums

Announcements

Video Lectures

Source Code By Week

Source Code By Week

We've created a GitHub repository for the 2014 offering of the POSA MOOC (see www.coursera.org/course/posa for more information).

When the 2014 POSA MOOC officially starts (likely in the fall), you can also get various parts of the Android 4.0 (Ice Cream Sandwich) source code.

bionic
Page : https://github.com/android/platform_bionic
D/L : https://github.com/android/platform_bionic

dalvik
Page : https://github.com/android/platform_dalvik/tree/android-4.0.4_r2.1

douglascraigschmidt / POSA-14 Unwatch 150 ★

This repository contains assignments and examples for the 2014 offering of the POSA MOOC (see www.coursera.org/course/posa for more information) — Edit

23 commits 1 branch 0 releases 2 contributors

branch: master **POSA-14**

Fixed a few bugs.

douglascraigschmidt authored 3 days ago latest commit f380ad8b5f6

assignments	Fixed a few bugs.	3 days ago
ex	Some light editing and reformatting.	4 days ago
.gitattributes	Some Improvements for the ThreadedDownloads example	5 days ago
.gitignore	Some Improvements for the ThreadedDownloads example	5 days ago
README.md	Create README.md	21 days ago

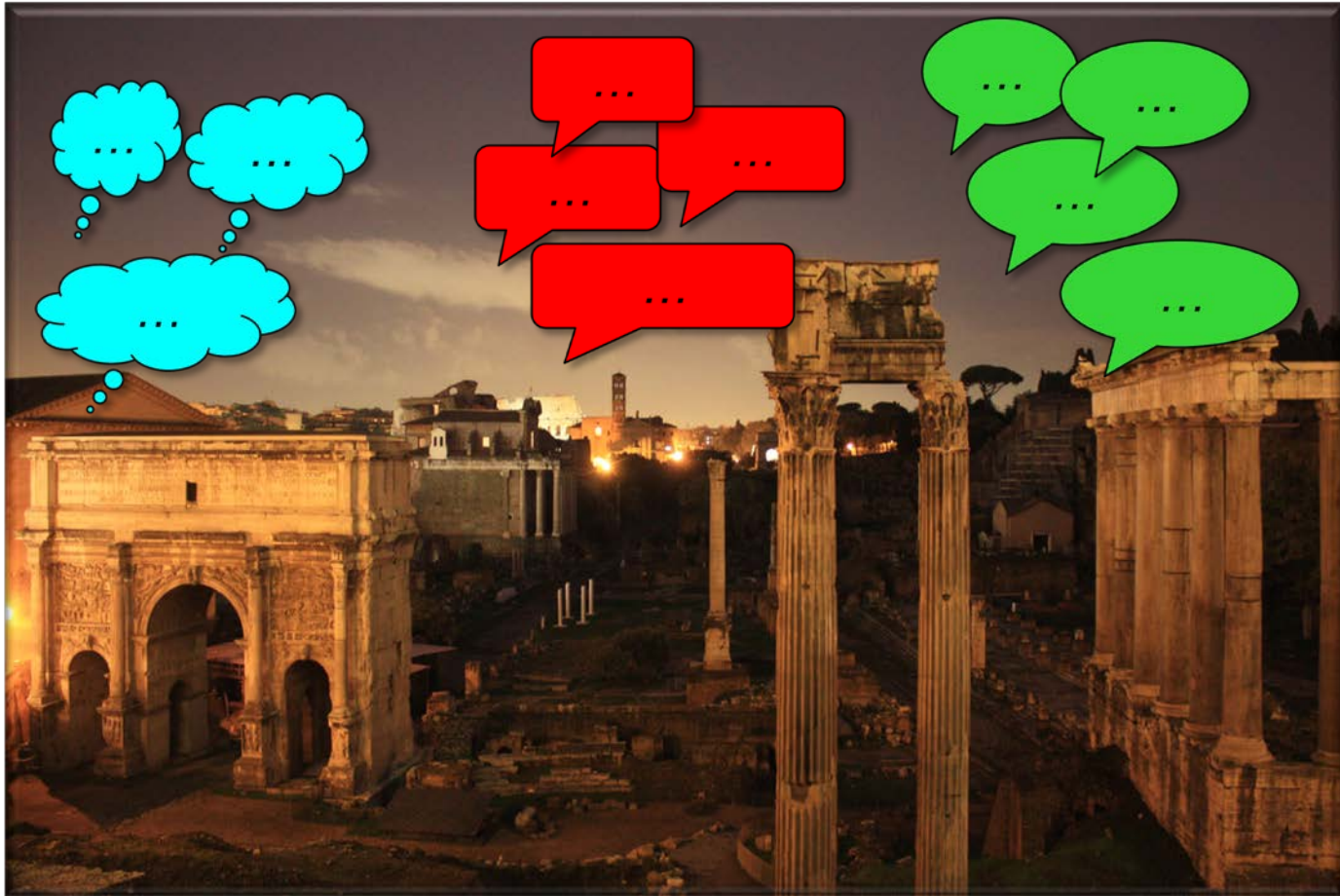
README.md

POSA-14

This repository contains assignments and examples for the 2014 offering of the POSA MOOC (see www.coursera.org/course/posa for more information)

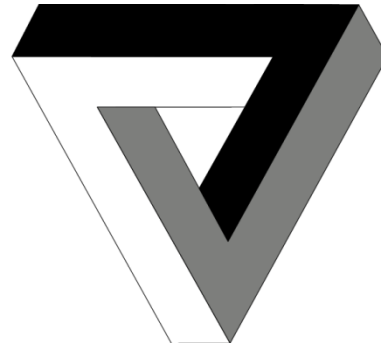
Summary

- This MOOC covers a wide range of topics related to Android



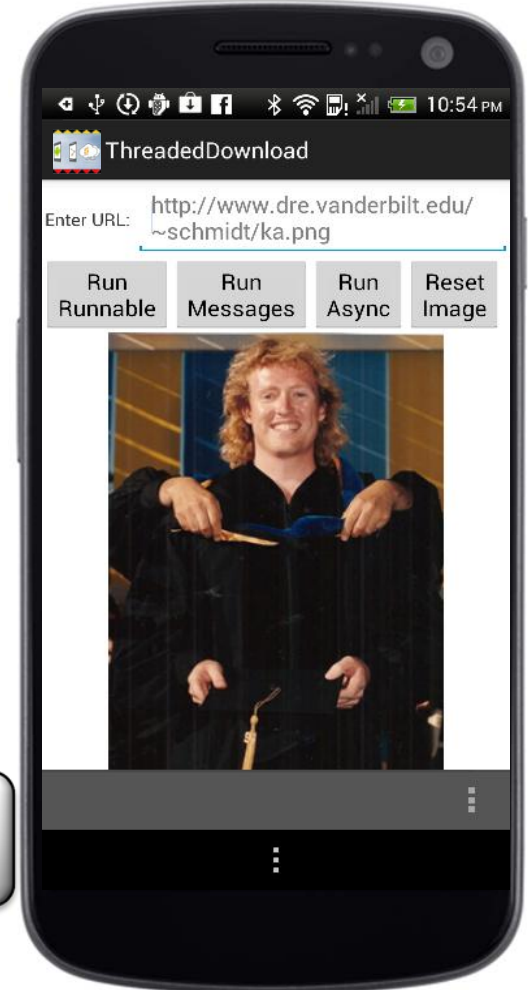
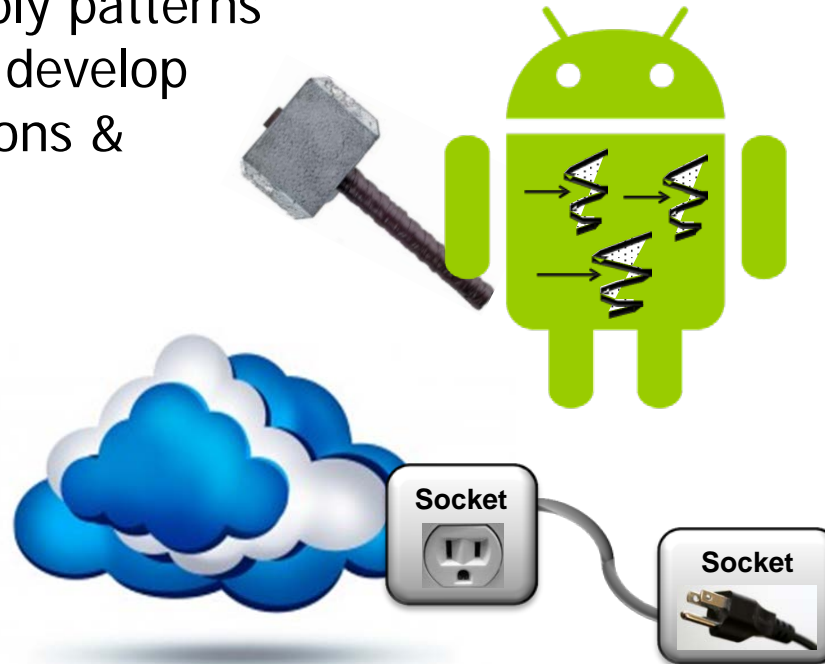
Summary

- This MOOC covers a wide range of topics related to Android
- Several student outcomes
 - Recognize inherent & accidental complexities




Summary

- This MOOC covers a wide range of topics related to Android
- Several student outcomes
 - Recognize inherent & accidental complexities
 - Learn how to apply patterns & frameworks to develop Android applications & services




Summary

- This MOOC covers a wide range of topics related to Android
- Several student outcomes
 - Recognize inherent & accidental complexities
 - Learn how to apply patterns & frameworks to develop Android applications & services
- Know where to find other sources of information

 **VANDERBILT UNIVERSITY**

Pattern-Oriented Software Architectures: Programming Mobile Serv..
by Dr. Douglas C. Schmidt, Dr. C. Jules White

[Join Signature Track](#)
3 weeks and 6 days left



Q

COURSE

[Discussion Forums](#)

[Announcements](#)

[Video Lectures](#)

[Source Code By Week](#)

[Installing and Using Eclipse](#)

EXERCISES

[Quizzes](#)

[Peer Assessments](#)

[Programming Assignments](#)

[Surveys](#)

ABOUT THE COURSE

[FAQ](#)

[Syllabus](#)

[Statements of Accomplishment](#)

[About Us](#)

[Instructor's Website](#)

COMMUNITY

[Course Wiki](#)

Frequently Asked Questions

[Help](#)

1. What are the course objectives?
Upon completing this course, students should be able to:

- Recognize the inherent and accidental complexities involved with developing concurrent and networked software.
- Understand how pattern-oriented software architecture techniques can and cannot help to alleviate this complexity.
- Apply key pattern-oriented software architecture techniques to develop reusable concurrent and networked Android Apps using the Java object-oriented programming language and Android middleware.
- Understand advanced Android middleware systems programming mechanisms and use them effectively to develop concurrent and networked software Apps and Services on Android.
- Know where to find additional sources of information on how to successfully apply pattern-oriented software architecture techniques to concurrent and networked software on Android and other infrastructure platforms.

2. How does this MOOC compare/contrast with courses at Vanderbilt?
This MOOC is heavily based on senior-level undergraduate courses we teach at Vanderbilt, such as the course Systems Programming for Android (CS 282). CS 282 focuses on teaching mobile software development at both a conceptual level (e.g., an understanding of software patterns, object-oriented design, and frameworks) and a practical level (e.g., experience programming Android services and applications using Java and Eclipse). Students in this course are expected to be familiar with Java and Eclipse and are capable of learning new material without significant hand-holding by the teachers and course staff. The lecture material in CS 282 is similar to the material for this MOOC (in fact, you can watch earlier versions of this material that I presented live in CS 282 via a [YouTube channel](#)). The quizzes, programming assignments, and level of feedback for the Vanderbilt courses are different, however, since the courses at Vanderbilt have many fewer students, so there's significantly more personalized guidance from the professor and TAs that can't (yet) be replicated via a MOOC.

3. What are your assumptions about--and expectations for--students taking this MOOC?
As mentioned above, this MOOC is based on material we teach to senior-level undergraduate students at Vanderbilt. The lecture material is therefore intended for self-motivated students who already know Java and Eclipse (or can learn them quickly on their own) and who want to understand both the concepts and practice of mobile software development. Moreover, this material is targeted at students who want to know (1) how to program mobile software applications and services and (2) how the Android software stack is designed and implemented. Therefore, students who are looking for vocational training (e.g., having an instructor provide solutions directly or simply walk through application code projects step-by-step in the Android development environment) may not find this MOOC suitable for their needs since our goal is to help students learn how to find a solution, not simply provide a solution. Moreover, this MOOC covers a wide range of topics (especially patterns and frameworks) associated with quality mobile software development. For students who just want training we recommend others sources, such as the [Android Application Development Tutorials](#) on YouTube.

4. What is the most effective way to learn material covered in the course and to successfully complete the programming assignments?
We recommend watching the videos multiple times, looking for different levels of meaning in the diagrams and the examples. It's particularly important to carefully watch all the videos referenced in the programming assignment descriptions since they provide many relevant tips and insights. Likewise, we recommend reading the material pointed to by links in the slides, as well as material from the (optional) recommended reading. Naturally, participating in the online discussion forum (and ideally, a meetup group if one is available in your area) will help make the course material more engaging and personalized. Additional discussions of the programming assignment goals, requirements, and (ultimately) their solutions will be covered in "Virtual Office Hours" (see the FAQ entry on this topic below).

Summary

- This MOOC covers a wide range of topics related to Android
- Several student outcomes
- We analyze lots of Android software

USE THE
SOURCE LUKETEM



Summary

- This MOOC covers a wide range of topics related to Android
- Several student outcomes
- We analyze lots of Android software
 - It's essential to understand Java!!

USE THE
SOURCE LUKE
SOURCE LUKE

