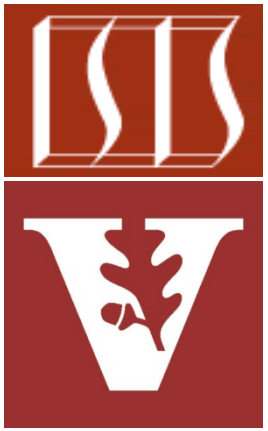
Introduction:
MOOC Organization & Topics

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 MOOC's structure & contents

Learning Objectives in this Part of the Module

- Understand the MOOC's structure & contents

Section 0: MOOC Introduction
- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies

Section 1: Android Services & Local IPC
- Part 1: Overview of Started & Bound Services
- Part 2: Programming Started Services
- Part 3: Android IntentService
- Part 4: Activity & Service Communication
- Part 5: Service to Activity Communication Using Messengers
- Part 6: Programming Bound Services with Messengers
- Part 7: Overview of Android Interface Definition Language (AIDL)
- Part 8: Programming Bound Services with AIDL

Section 3: Android Remote IPC
- Part 1: Overview of Hyper-Text Transfer Protocol (HTTP)
- Part 2: Designing Mobile Applications with HTTP Communication
- Part 3: Better Client-side Communication Abstractions for HTTP

Section 4: Communication Patterns in Android
- Part 1: Starting Services on Demand with the *Activator* Pattern
- Part 2: Passing Commands to Services with the *Command Processor* Pattern
- Part 3: Automating Marshaling & Demarshaling of Data with the *Proxy* Pattern
- Part 4: Supporting Object-Oriented Remote Method Calls with the *Broker* Pattern

Learning Objectives in this Part of the Module

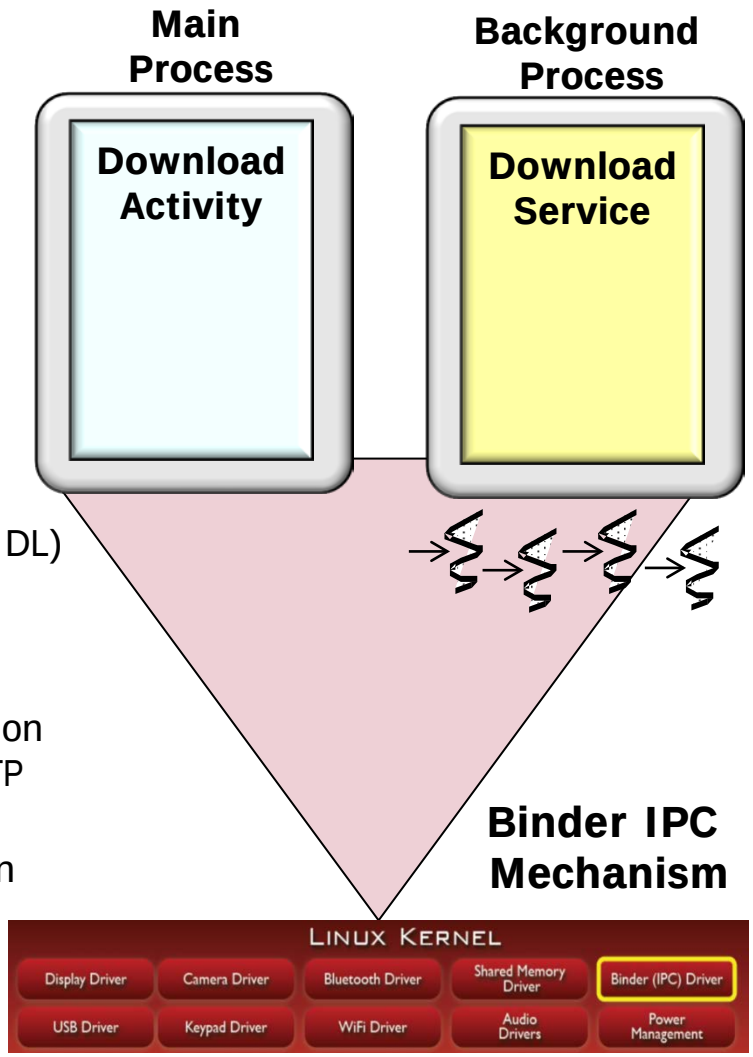
- Understand the MOOC's structure & contents

Section 0: MOOC Introduction
- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies

Section 1: Android Services & Local IPC
- Part 1: Overview of Started & Bound Services
- Part 2: Programming Started Services
- Part 3: Android IntentService
- Part 4: Activity & Service Communication
- Part 5: Service to Activity Communication Using Messengers
- Part 6: Programming Bound Services with Messengers
- Part 7: Overview of Android Interface Definition Language (AIDL)
- Part 8: Programming Bound Services with AIDL

Section 2: Android Remote IPC
- Part 1: Overview of Hyper-Text Transfer Protocol (HTTP)
- Part 2: Designing Mobile Applications with HTTP Communication
- Part 3: Better Client-side Communication Abstractions for HTTP

Section 3: Communication Patterns in Android
- Part 1: Starting Services on Demand with the *Activator* Pattern
- Part 2: Passing Commands to Services with the *Command Processor* Pattern
- Part 3: Automating Marshaling & Demarshaling of Data with the *Proxy* Pattern
- Part 4: Supporting Object-Oriented Remote Method Calls with the *Broker* Pattern



Learning Objectives in this Part of the Module

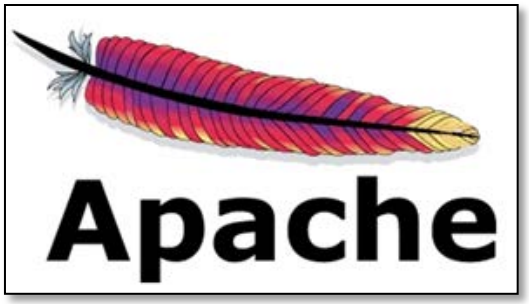
- Understand the MOOC's structure & contents

Section 0: MOOC Introduction
- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies

Section 1: Android Services & Local IPC
- Part 1: Overview of Started & Bound Services
- Part 2: Programming Started Services
- Part 3: Android IntentService
- Part 4: Activity & Service Communication
- Part 5: Service to Activity Communication Using Messengers
- Part 6: Programming Bound Services with Messengers
- Part 7: Overview of Android Interface Definition Language (AIDL)
- Part 8: Programming Bound Services with AIDL

Section 2: Android Remote IPC
- Part 1: Overview of Hyper-Text Transfer Protocol (HTTP)
- Part 2: Designing Mobile Applications with HTTP Communication
- Part 3: Better Client-side Communication Abstractions for HTTP

Section 3: Communication Patterns in Android
- Part 1: Starting Services on Demand with the *Activator* Pattern
- Part 2: Passing Commands to Services with the *Command Processor* Pattern
- Part 3: Automating Marshaling & Demarshaling of Data with the *Proxy* Pattern
- Part 4: Supporting Object-Oriented Remote Method Calls with the *Broker* Pattern



Learning Objectives in this Part of the Module

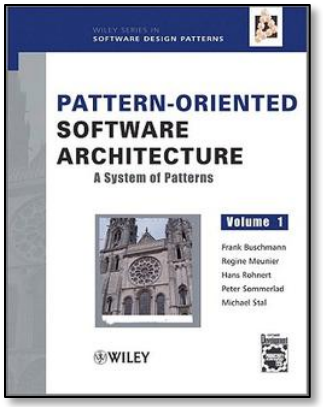
- Understand the MOOC's structure & contents

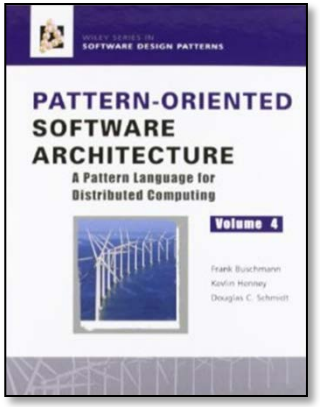
Section 0: MOOC Introduction
- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies

Section 1: Android Services & Local IPC
- Part 1: Overview of Started & Bound Services
- Part 2: Programming Started Services
- Part 3: Android IntentService
- Part 4: Activity & Service Communication
- Part 5: Service to Activity Communication Using Messengers
- Part 6: Programming Bound Services with Messengers
- Part 7: Overview of Android Interface Definition Language (AIDL)
- Part 8: Programming Bound Services with AIDL

Section 2: Android Remote IPC
- Part 1: Overview of Hyper-Text Transfer Protocol (HTTP)
- Part 2: Designing Mobile Applications with HTTP Communication
- Part 3: Better Client-side Communication Abstractions for HTTP

Section 3: Communication Patterns in Android
- Part 1: Starting Services on Demand with the *Activator* Pattern
- Part 2: Passing Commands to Services with the *Command Processor* Pattern
- Part 3: Automating Marshaling & Demarshaling of Data with the *Proxy* Pattern
- Part 4: Supporting Object-Oriented Remote Method Calls with the *Broker* Pattern





Learning Objectives in this Part of the Module

- Understand the MOOC's structure & contents

Section 0: MOOC Introduction
- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies

Section 1: Android Services & Local IPC
- Part 1: Overview of Started & Bound Services
- Part 2: Programming Started Services
- Part 3: Android IntentService
- Part 4: Activity & Service Communication
- Part 5: Service to Activity Communication Using Messengers
- Part 6: Programming Bound Services with Messengers
- Part 7: Overview of Android Interface Definition Language (AIDL)
- Part 8: Programming Bound Services with AIDL

Section 2: Android Remote IPC
- Part 1: Overview of Hyper-Text Transfer Protocol (HTTP)
- Part 2: Designing Mobile Applications with HTTP Communication
- Part 3: Better Client-side Communication Abstractions for HTTP

Section 3: Communication Patterns in Android
- Part 1: Starting Services on Demand with the *Activator* Pattern
- Part 2: Passing Commands to Services with the *Command Processor* Pattern
- Part 3: Automating Marshaling & Demarshaling of Data with the *Proxy* Pattern
- Part 4: Supporting Object-Oriented Remote Method Calls with the *Broker* Pattern

See github.com/douglascraigschmidt/POSA-15/wiki/POSA-15-FAQ item #26

Overview of the MOOC Topics in Section 1

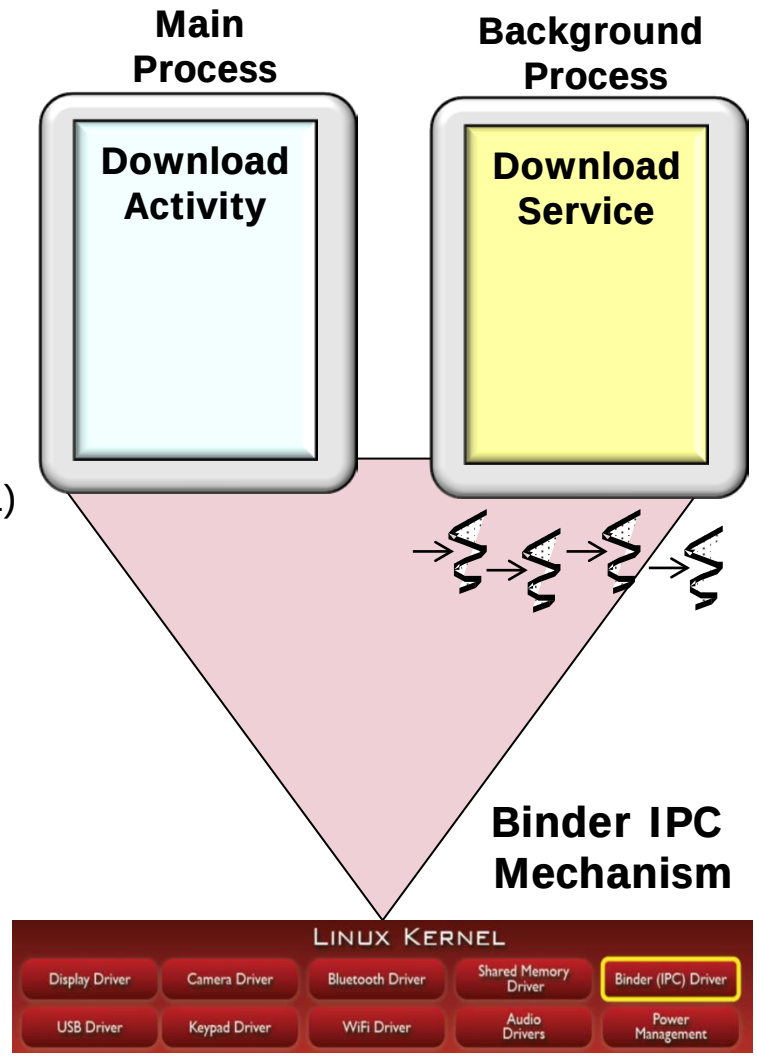
Overview of Topics Covered in Section 1

Section 0: MOOC Introduction
- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies

Section 1: Android Services & Local IPC
- Part 1: Overview of Started & Bound Services
- Part 2: Programming Started Services
- Part 3: Android IntentService
- Part 4: Activity & Service Communication
- Part 5: Service to Activity Communication Using Messengers
- Part 6: Programming Bound Services with Messengers
- Part 7: Overview of Android Interface Definition Language (AIDL)
- Part 8: Programming Bound Services with AIDL

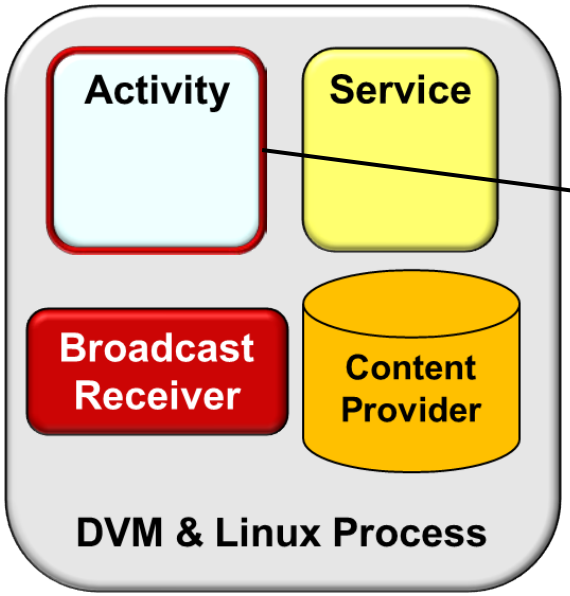
Section 2: Android Remote IPC
- Part 1: Overview of Hyper-Text Transfer Protocol (HTTP)
- Part 2: Designing Mobile Applications with HTTP Communication
- Part 3: Better Client-side Communication Abstractions for HTTP

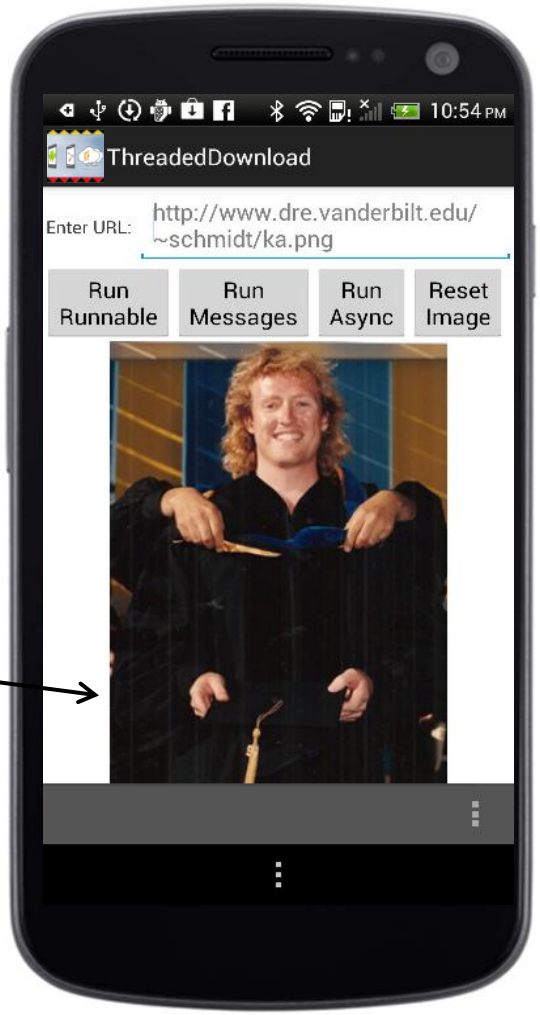
Section 3: Communication Patterns in Android
- Part 1: Starting Services on Demand with the *Activator* Pattern
- Part 2: Passing Commands to Services with the *Command Processor* Pattern
- Part 3: Automating Marshaling & Demarshaling of Data with the *Proxy* Pattern
- Part 4: Supporting Object-Oriented Remote Method Calls with the *Broker* Pattern



Overview of Topics Covered in Section 1

- **Section 1 – Android Services & Local IPC**
  - Android Services & Communication Frameworks

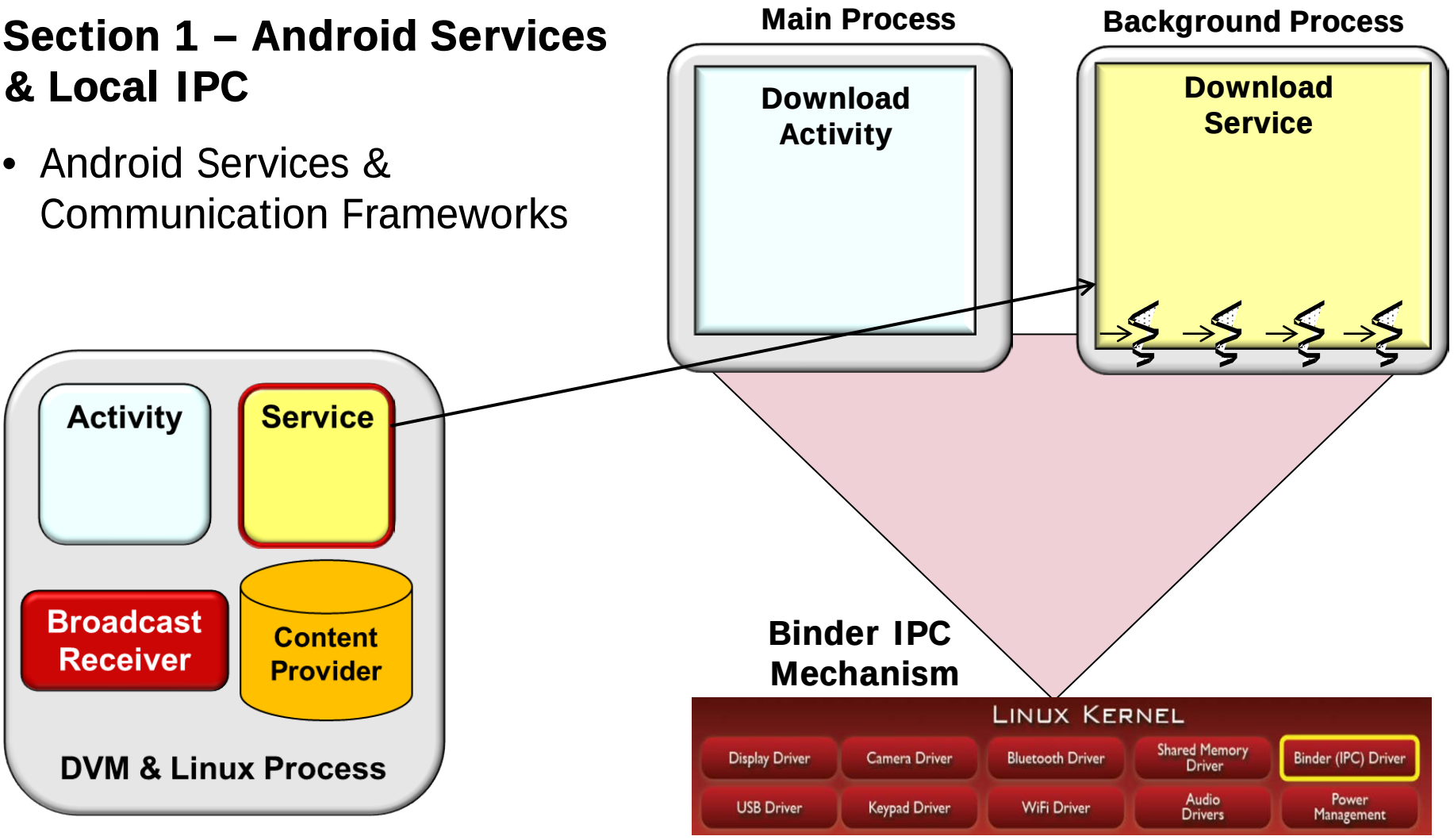




Activities handle user-facing operations

Overview of Topics Covered in Section 1

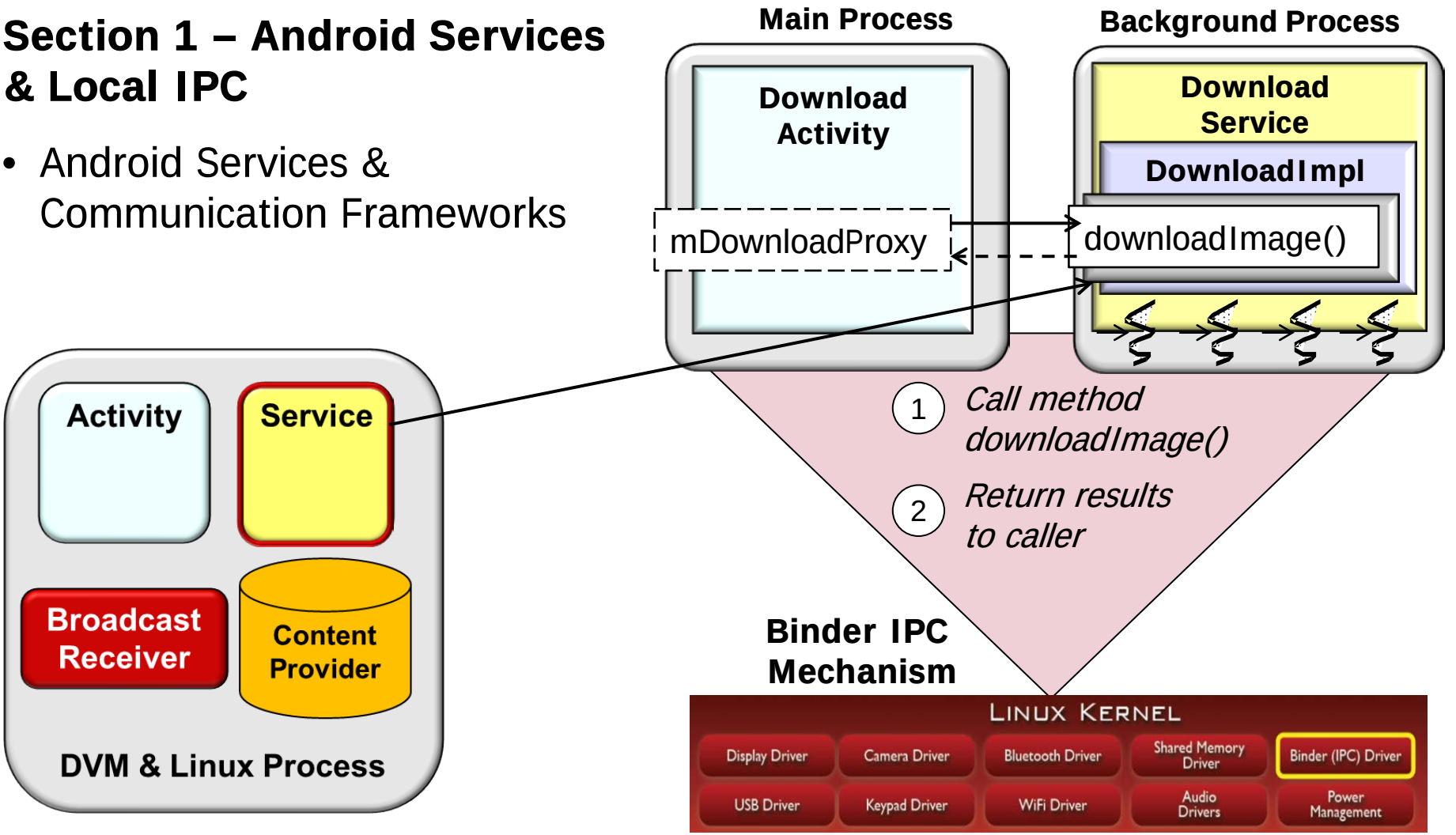
- **Section 1 – Android Services & Local IPC**
  - Android Services & Communication Frameworks



Services can handle long-duration operations that don't interact with the user directly

Overview of Topics Covered in Section 1

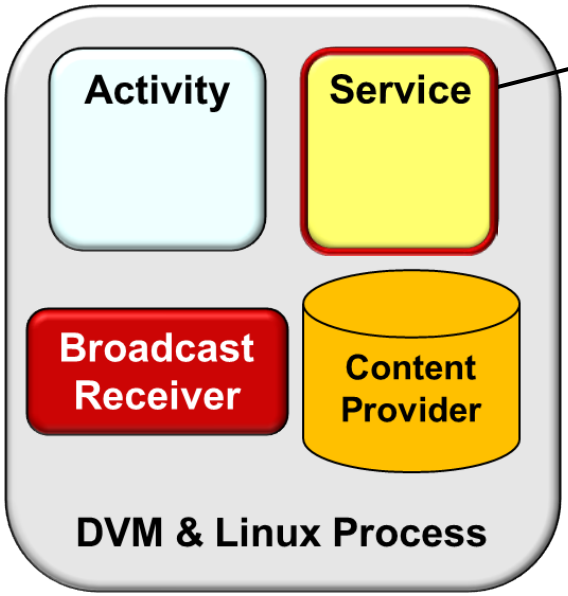
- **Section 1 – Android Services & Local IPC**
  - Android Services & Communication Frameworks

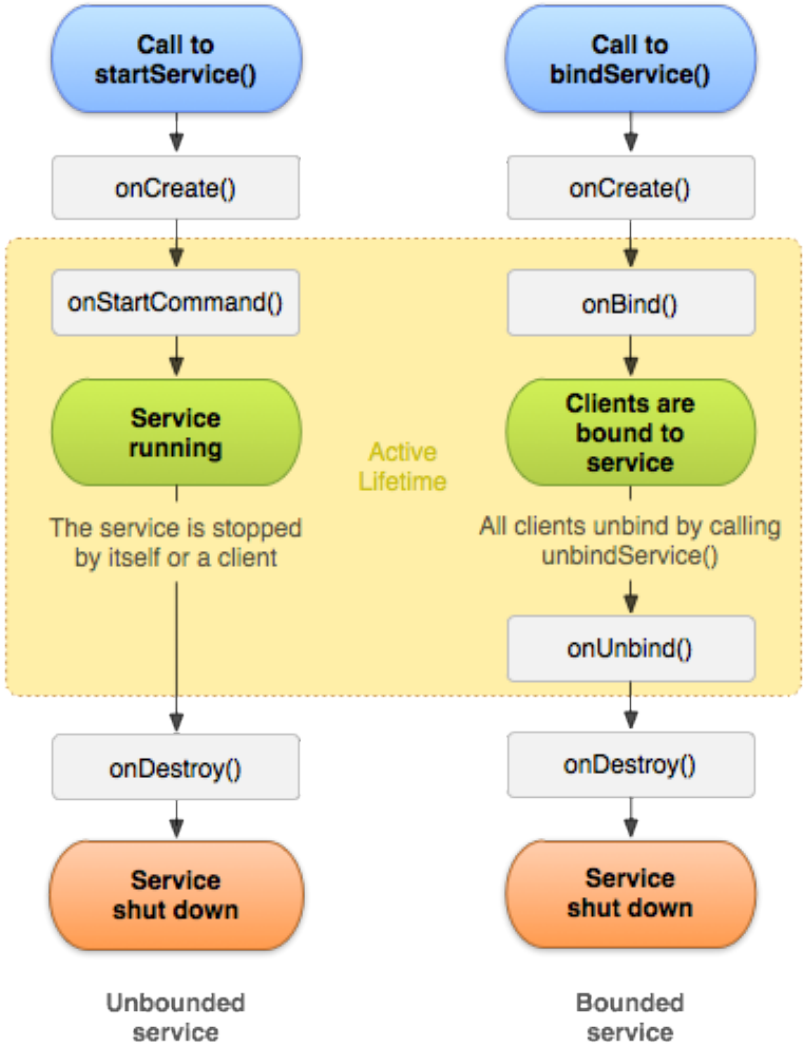


IPC mechanisms are used to facilitate Activity & Service communication

Overview of Topics Covered in Section 1

- **Section 1 – Android Services & Local IPC**
  - Android Services & Communication Frameworks

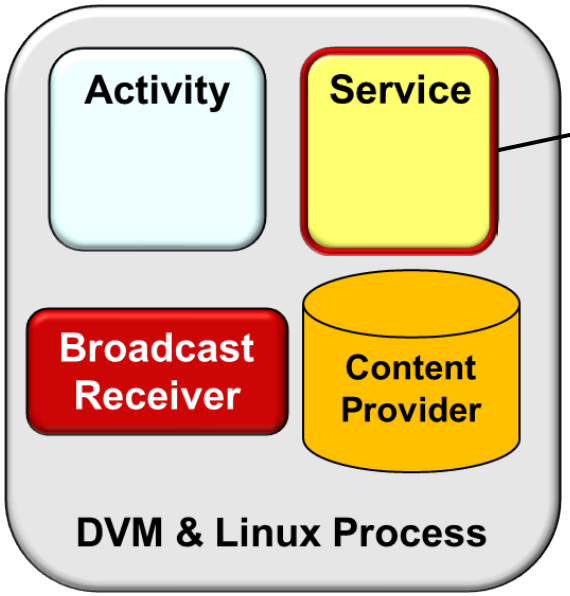


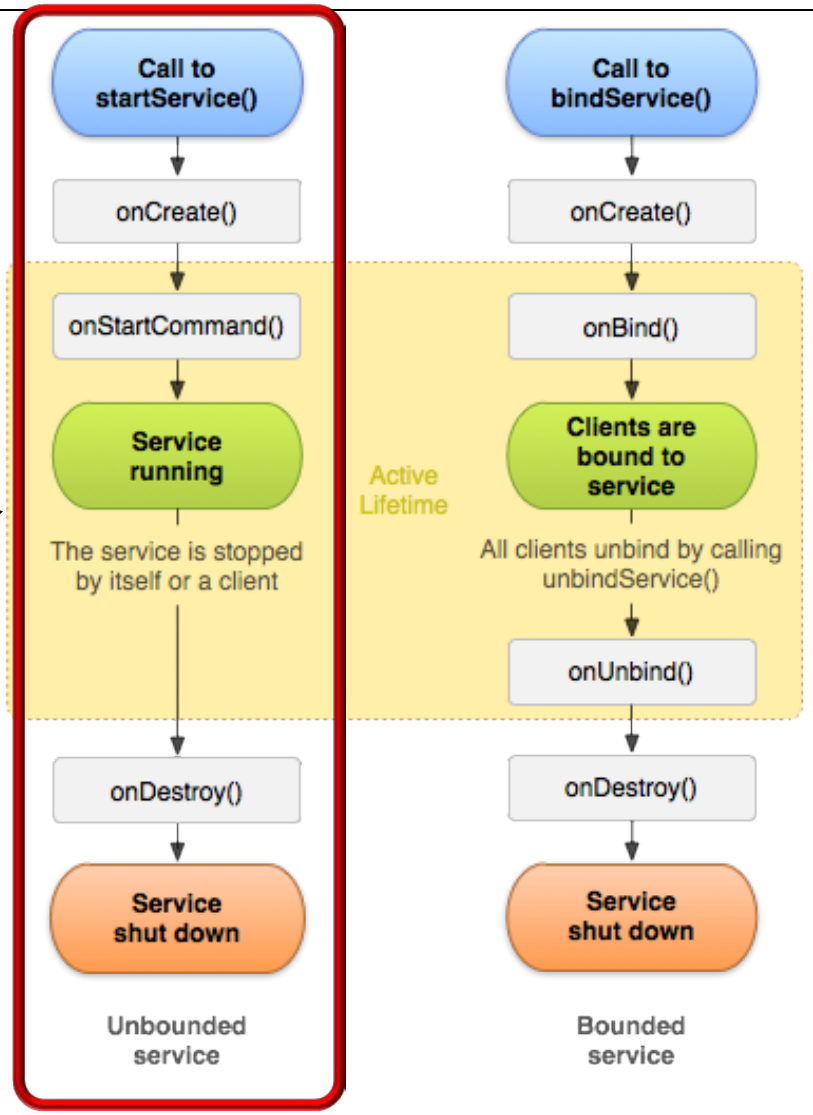


There are two types of Services

Overview of Topics Covered in Section 1

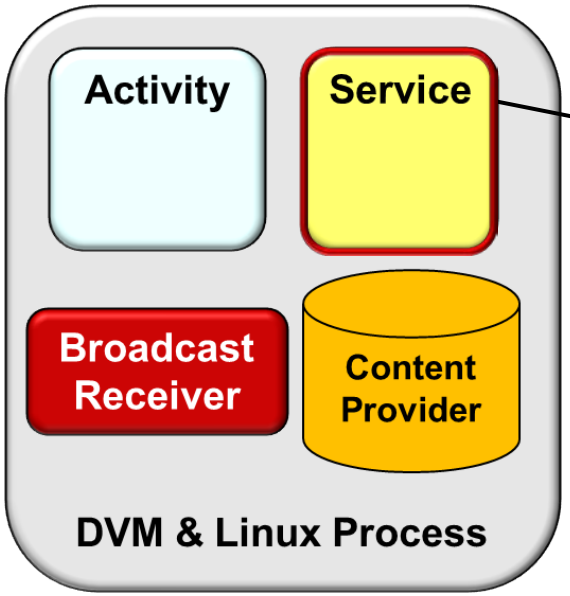
- **Section 1 – Android Services & Local IPC**
  - Android Services & Communication Frameworks

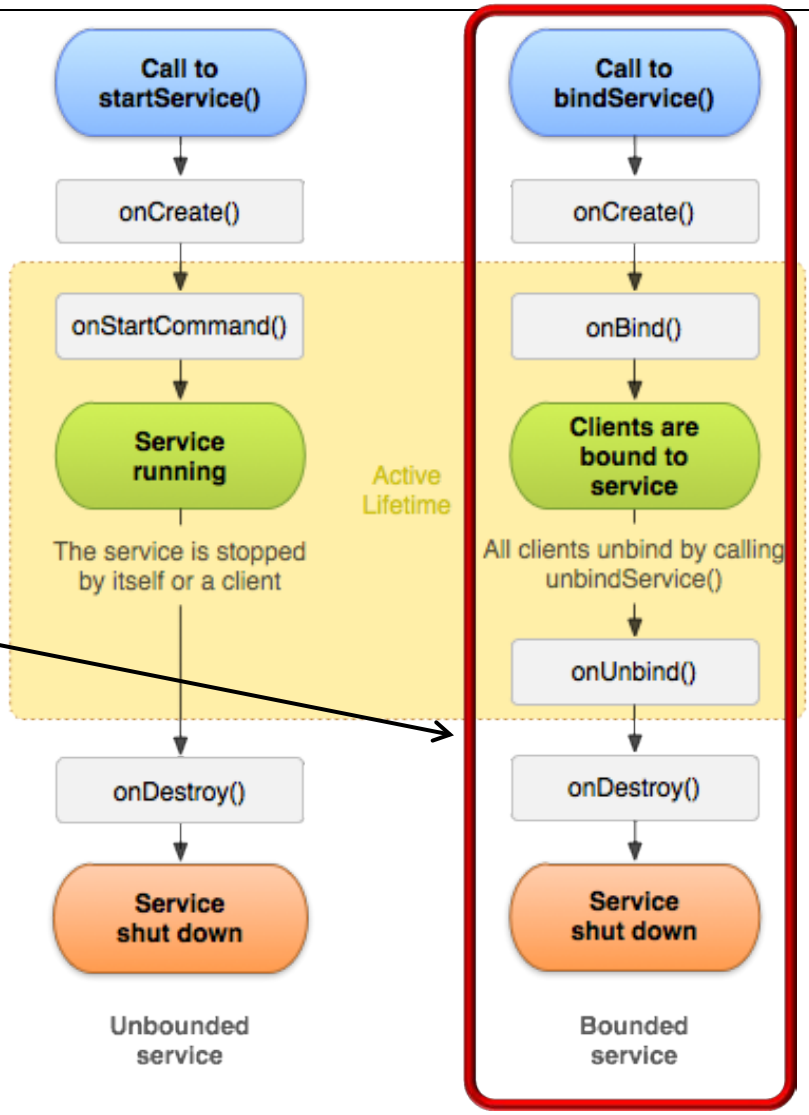




Overview of Topics Covered in Section 1

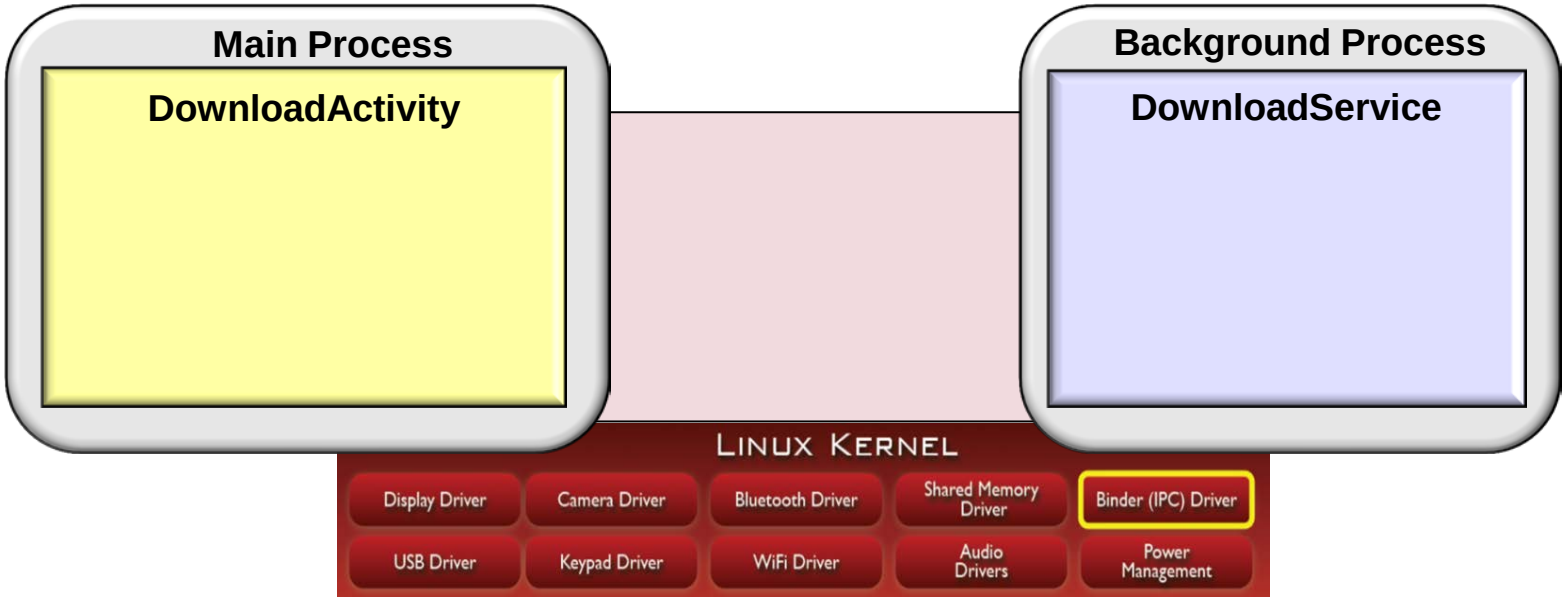
- **Section 1 – Android Services & Local IPC**
  - Android Services & Communication Frameworks





Overview of Topics Covered in Section 1

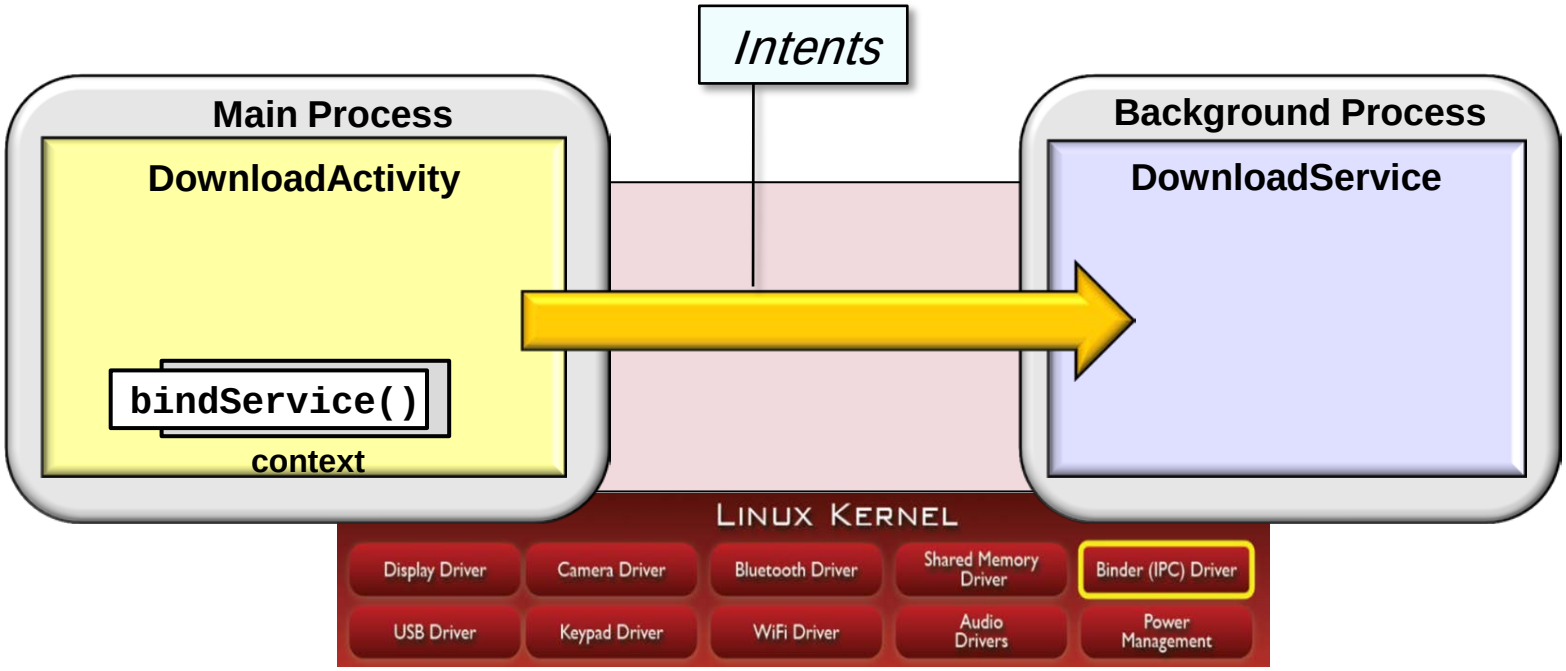
- **Section 1 – Android Services & Local IPC**
  - Android Services & Communication Frameworks



Android Linux provides mechanisms that are optimized for inter-process communication

Overview of Topics Covered in Section 1

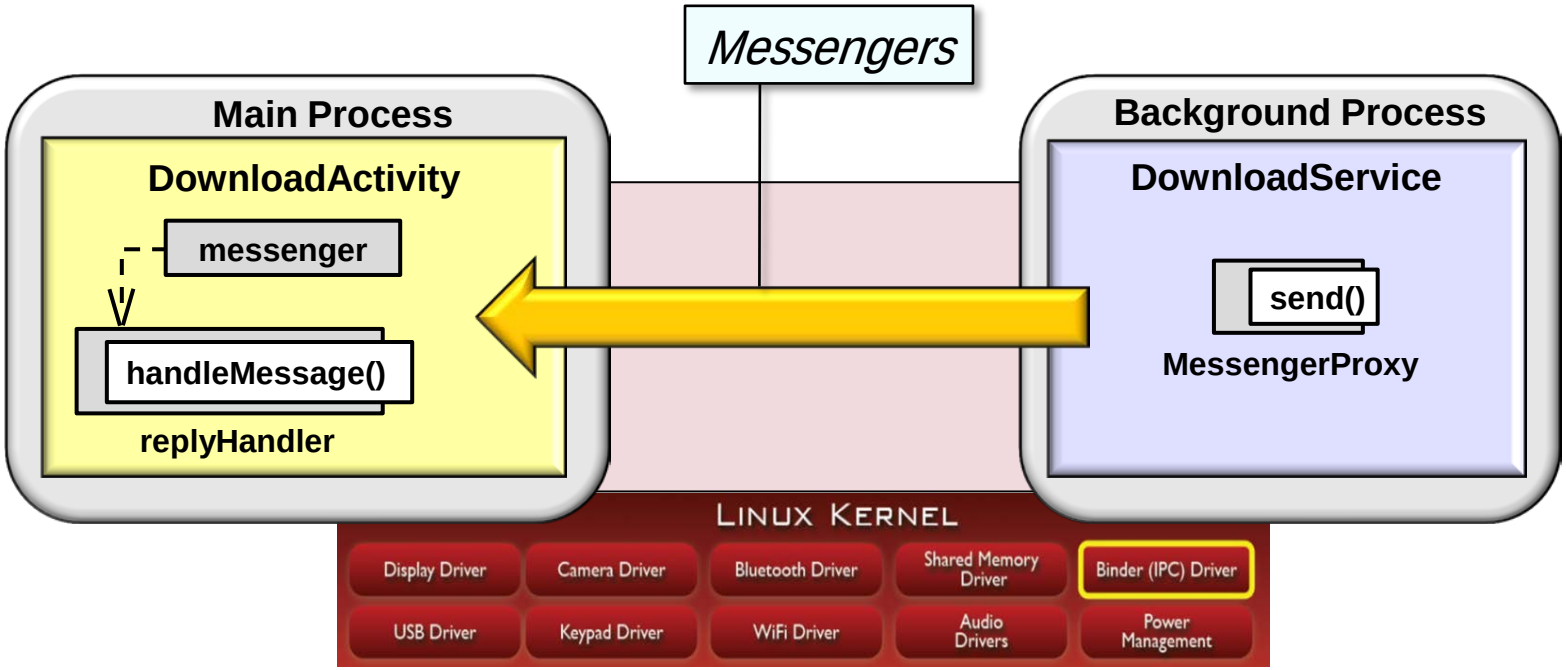
- **Section 1 – Android Services & Local IPC**
  - Android Services & Communication Frameworks



Intents are used to launch Services (& Activities) & can also contain data

Overview of Topics Covered in Section 1

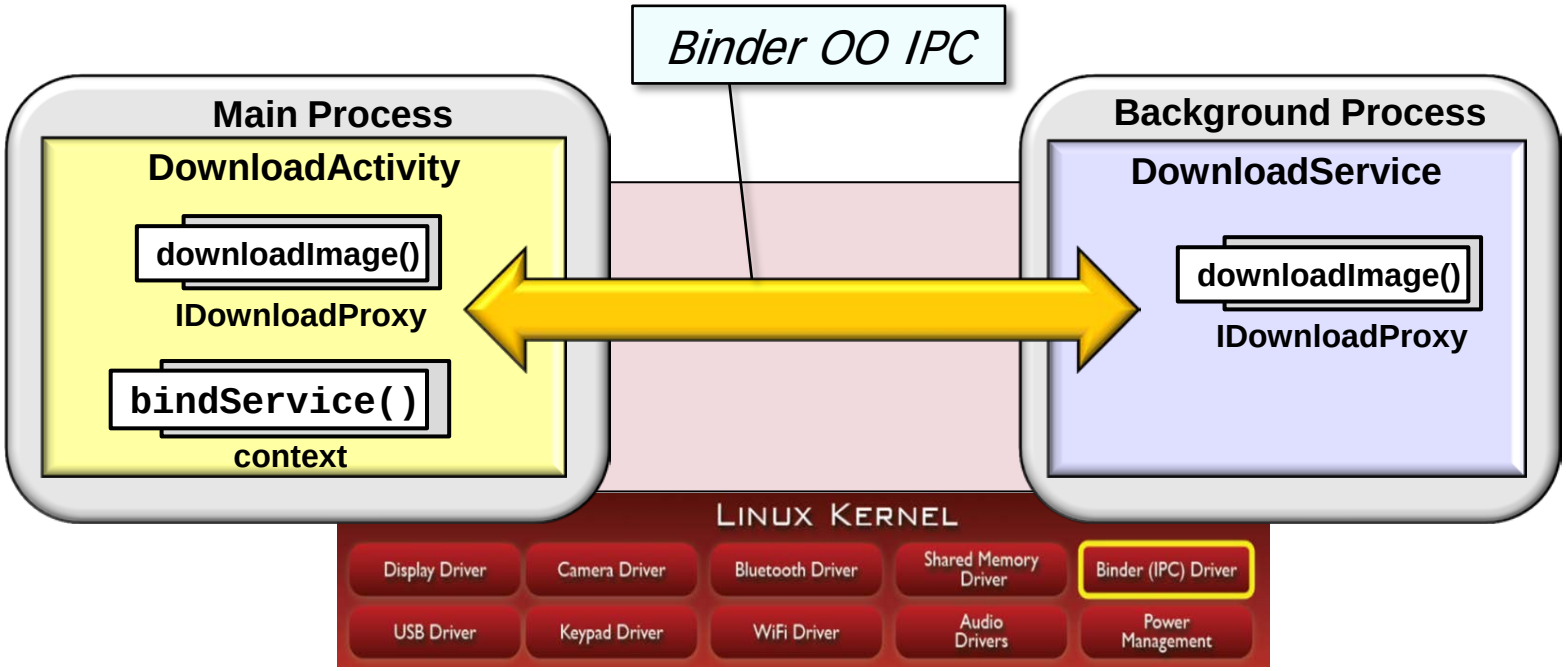
- **Section 1 – Android Services & Local IPC**
  - Android Services & Communication Frameworks



Messengers can be used to pass messages between processes

Overview of Topics Covered in Section 1

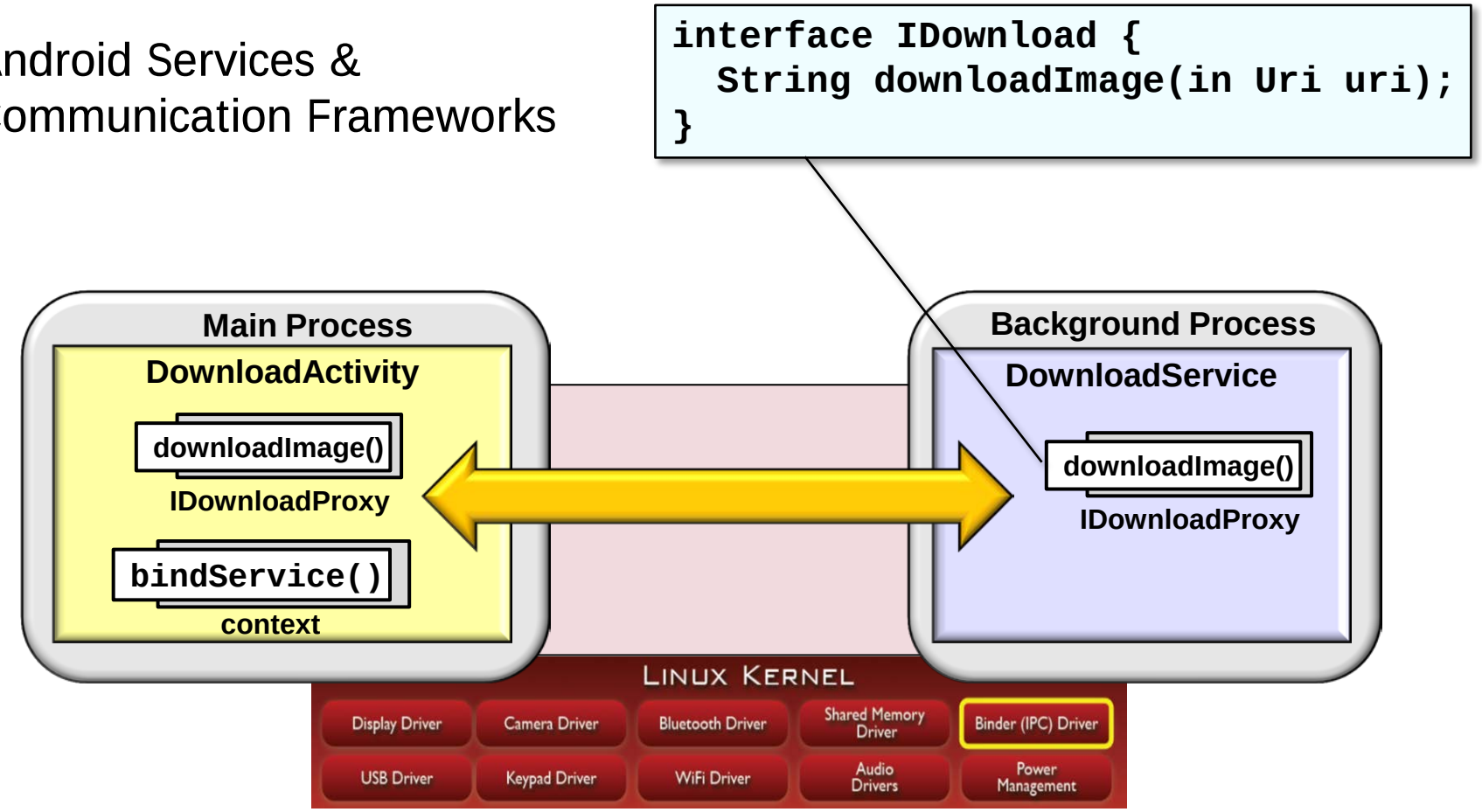
- **Section 1 – Android Services & Local IPC**
  - Android Services & Communication Frameworks



The Binder also provides an object-oriented IPC mechanism that can invoke methods on objects in other processes

Overview of Topics Covered in Section 1

- **Section 1 – Android Services & Local IPC**
  - Android Services & Communication Frameworks



The Android Interface Definition Language (AIDL) provides the means to enable strongly typed IPC across processes

Overview of the MOOC Topics in Section 2

Overview of Topics Covered in Section 2

Section 0: MOOC Introduction
- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies

Section 1: Android Services & Local IPC
- Part 1: Overview of Started & Bound Services
- Part 2: Programming Started Services
- Part 3: Android IntentService
- Part 4: Activity & Service Communication
- Part 5: Service to Activity Communication Using Messengers
- Part 6: Programming Bound Services with Messengers
- Part 7: Overview of Android Interface Definition Language (AIDL)
- Part 8: Programming Bound Services with AIDL

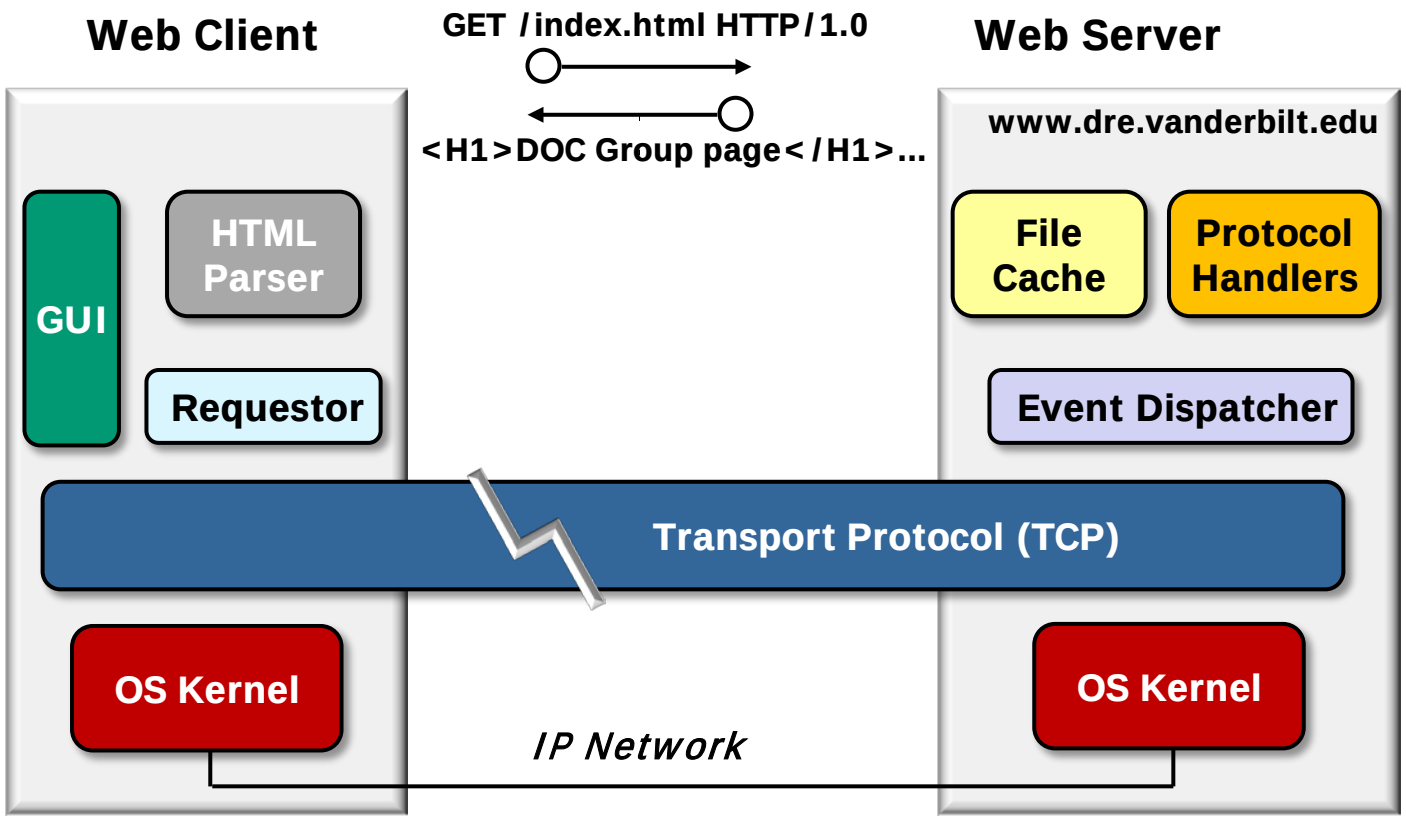
Section 2: Android Remote IPC
- Part 1: Overview of Hyper-Text Transfer Protocol (HTTP)
- Part 2: Designing Mobile Applications with HTTP Communication
- Part 3: Better Client-side Communication Abstractions for HTTP

Section 3: Communication Patterns in Android
- Part 1: Starting Services on Demand with the *Activator* Pattern
- Part 2: Passing Commands to Services with the *Command Processor* Pattern
- Part 3: Automating Marshaling & Demarshaling of Data with the *Proxy* Pattern
- Part 4: Supporting Object-Oriented Remote Method Calls with the *Broker* Pattern

java.net

Overview of Topics Covered in Section 2

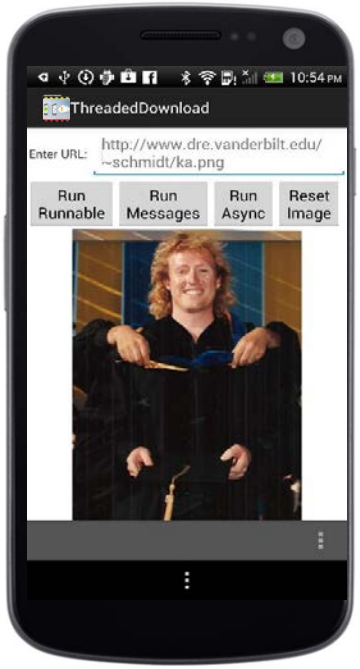
- **Section 2 – Android Remote IPC**
  - Overview of HTTP

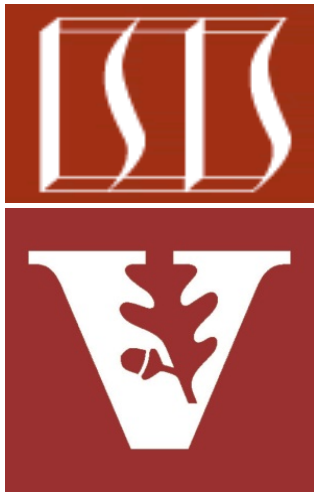


See en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Overview of Topics Covered in Section 2

- **Section 2 – Android Remote IPC**
  - Overview of HTTP
  - Writing clients that access web services via HTTP & JSon





Apache

Java java.net

See android-developers.blogspot.com/2011/09/androids-http-clients.html

Overview of the MOOC Topics in Section 3

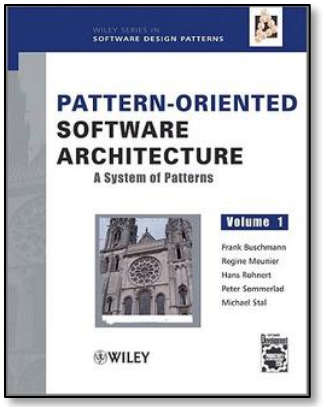
Overview of Topics Covered in Section 3

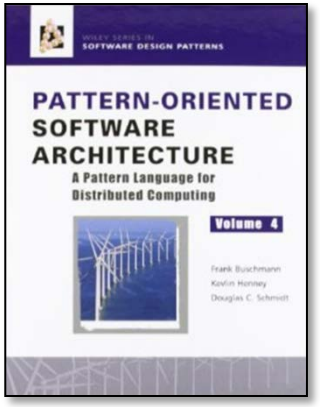
Section 0: MOOC Introduction
- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies

Section 1: Android Services & Local IPC
- Part 1: Overview of Started & Bound Services
- Part 2: Programming Started Services
- Part 3: Android IntentService
- Part 4: Activity & Service Communication
- Part 5: Service to Activity Communication Using Messengers
- Part 6: Programming Bound Services with Messengers
- Part 7: Overview of Android Interface Definition Language (AIDL)
- Part 8: Programming Bound Services with AIDL

Section 2: Android Remote IPC
- Part 1: Overview of Hyper-Text Transfer Protocol (HTTP)
- Part 2: Designing Mobile Applications with HTTP Communication
- Part 3: Better Client-side Communication Abstractions for HTTP

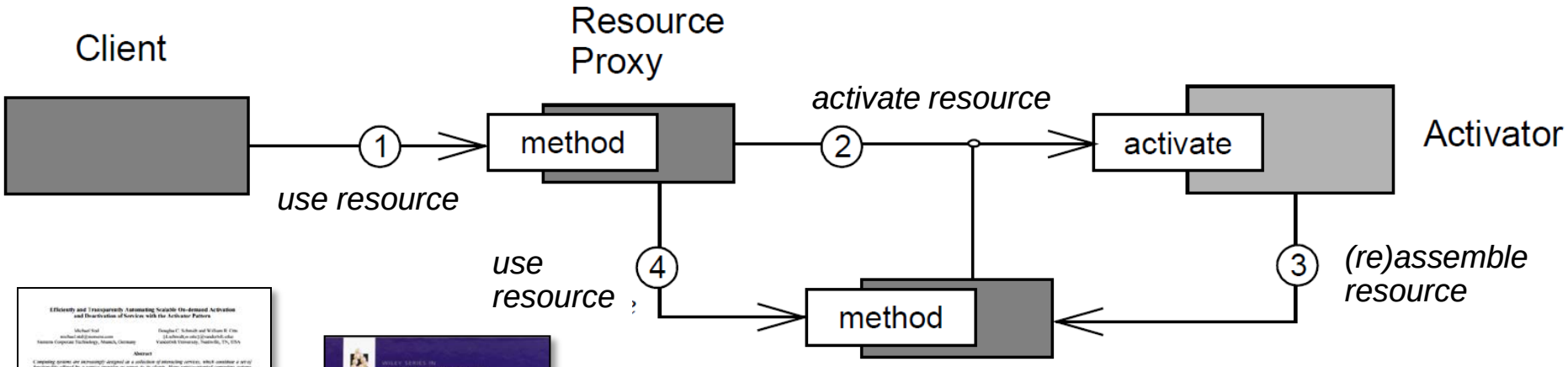
Section 3: Communication Patterns in Android
- Part 1: Starting Services on Demand with the *Activator* Pattern
- Part 2: Passing Commands to Services with the *Command Processor* Pattern
- Part 3: Automating Marshaling & Demarshaling of Data with the *Proxy* Pattern
- Part 4: Supporting Object-Oriented Remote Method Calls with the *Broker* Pattern

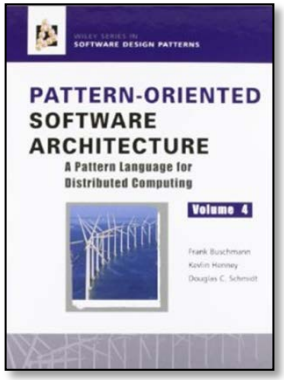




Overview of Topics Covered in Section 3

- **Section 3 – Communication Patterns Applied in Android**
  - *Activator*
    - Automates scalable on-demand activation & deactivation of service execution contexts to run services accessed by many clients without consuming excessive resources

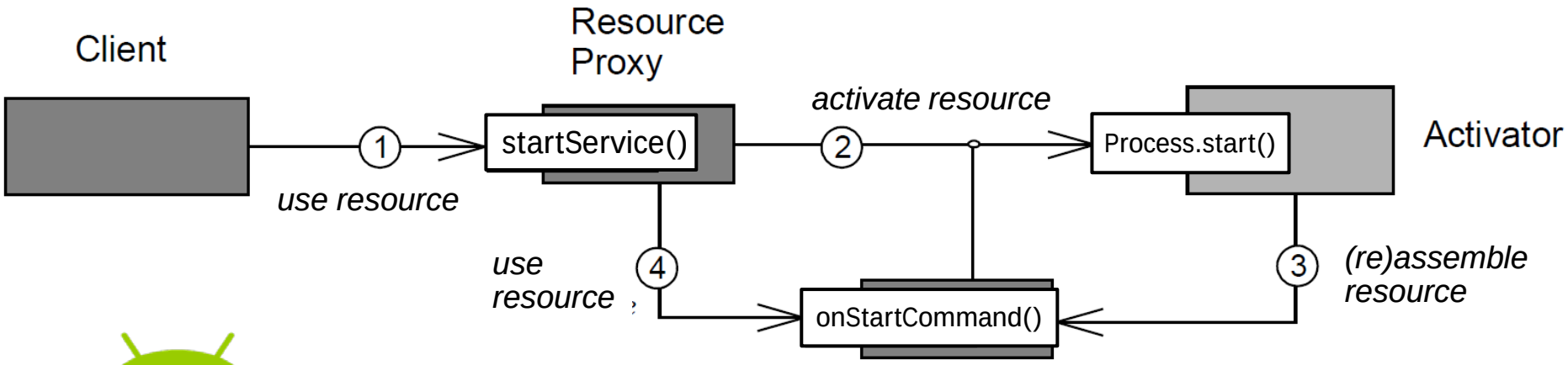




See www.dre.vanderbilt.edu/~schmidt/PDF/Activator.pdf

Overview of Topics Covered in Section 3

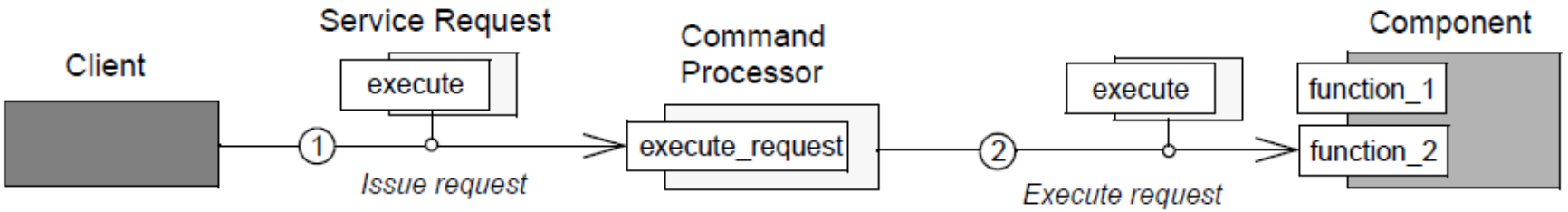
- **Section 3 – Communication Patterns Applied in Android**
  - *Activator*
    - Automates scalable on-demand activation & deactivation of service execution contexts to run services accessed by many clients without consuming excessive resources

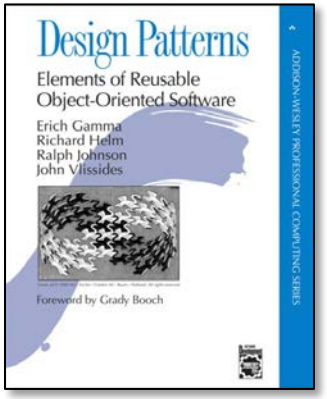


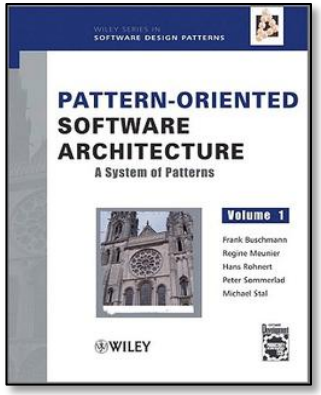
Android's Activity Manager Service applies *Activator* to launch Services (& Activities) on-demand

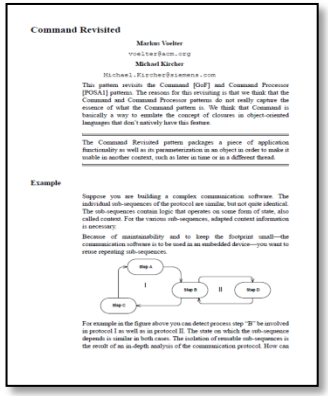
Overview of Topics Covered in Section 3

- **Section 3 – Communication Patterns Applied in Android**
  - *Command Processor*
    - Automates scalable on-demand activation & deactivation of service execution contexts to run services accessed by many clients without consuming excessive resources





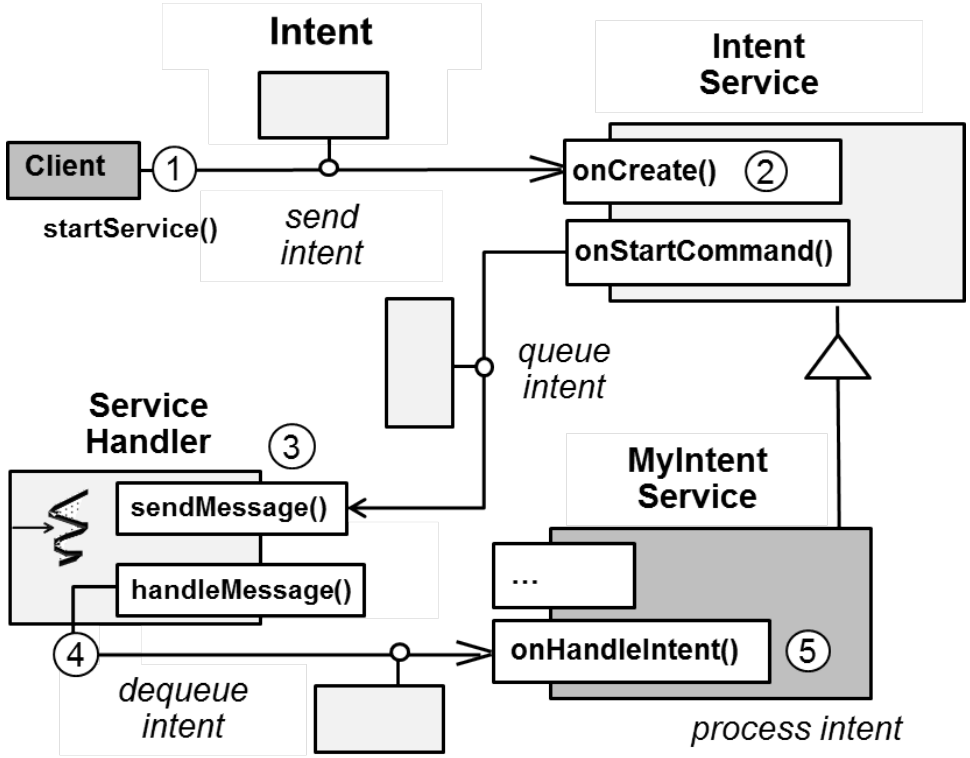




See www.dre.vanderbilt.edu/~schmidt/PDF/CommandRevisited.pdf

Overview of Topics Covered in Section 3

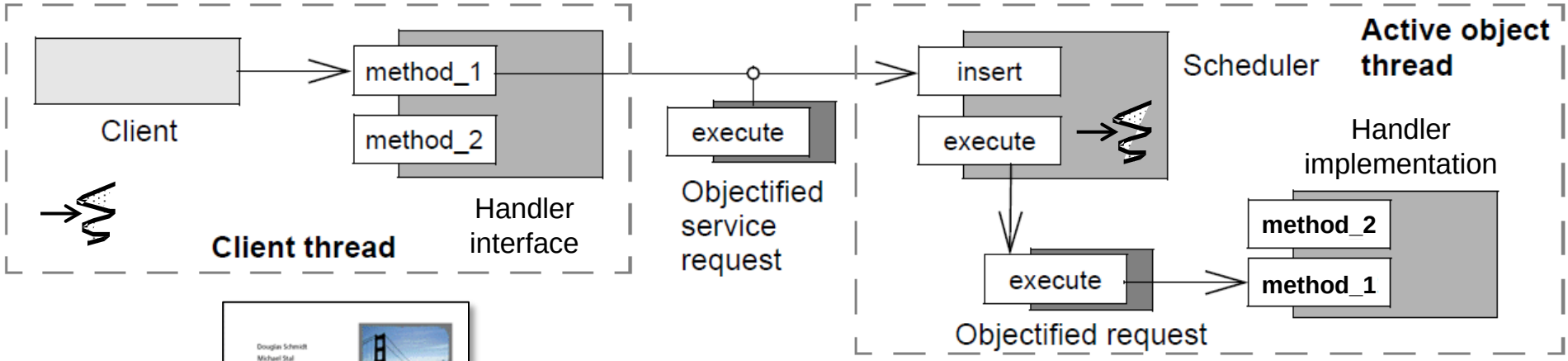
- **Section 3 – Communication Patterns Applied in Android**
  - *Command Processor*
    - Automates scalable on-demand activation & deactivation of service execution contexts to run services accessed by many clients without consuming excessive resources



Android's Activity Manager Service applies *Command Processor* to pass commands to Services

Overview of Topics Covered in Section 3

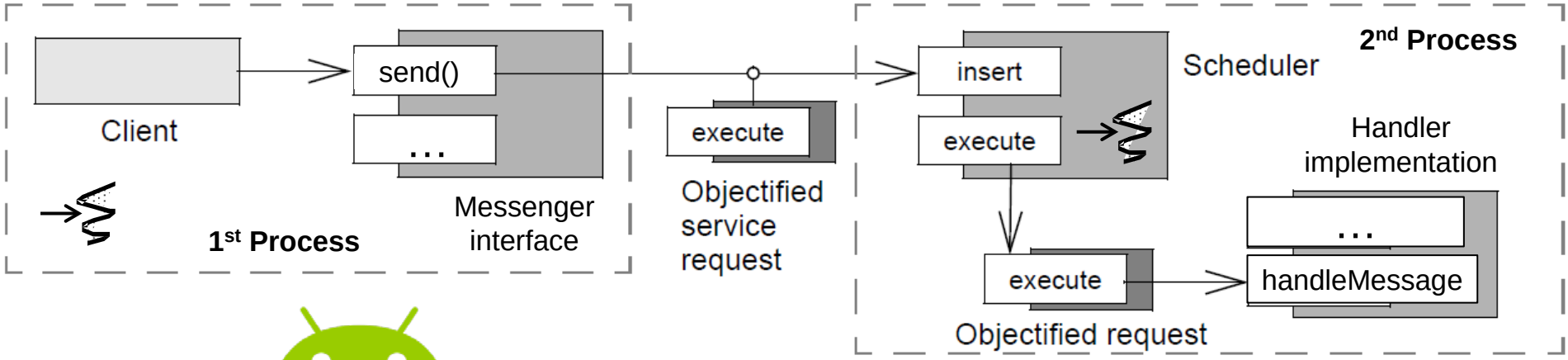
- **Section 3 – Communication Patterns Applied in Android**
  - *Active Object*
    - Define service requests on components as the unit of concurrency & run service requests on a component in different thread(s) from the requesting client thread



See en.wikipedia.org/wiki/Active_object

Overview of Topics Covered in Section 3

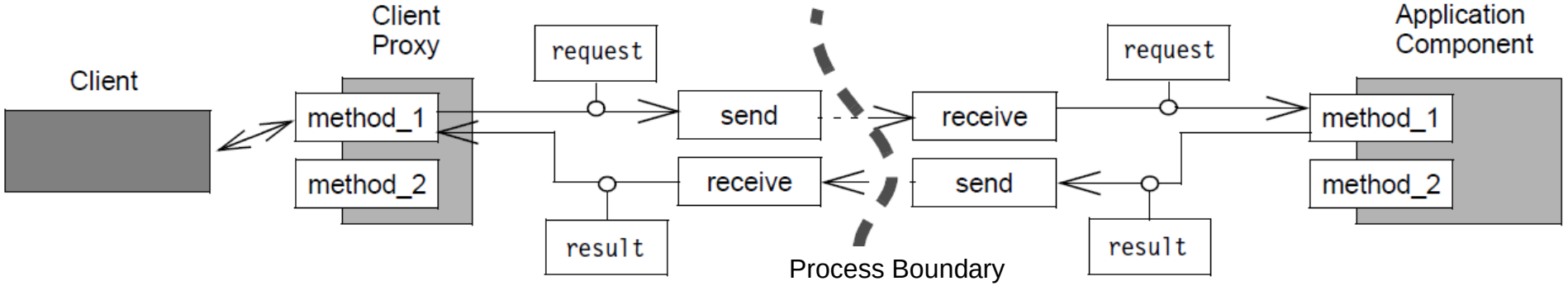
- **Section 3 – Communication Patterns Applied in Android**
  - *Active Object*
    - Define service requests on components as the unit of concurrency & run service requests on a component in different thread(s) from the requesting client thread

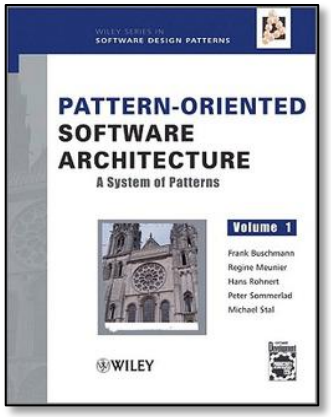


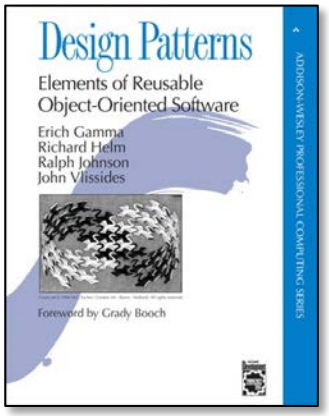
Android's Messenger IPC mechanism applies *Active Object* to send messages across processes

Overview of Topics Covered in Section 3

- **Section 3 – Communication Patterns Applied in Android**
  - *Proxy*
    - Provide a surrogate or placeholder for another object to control access to it



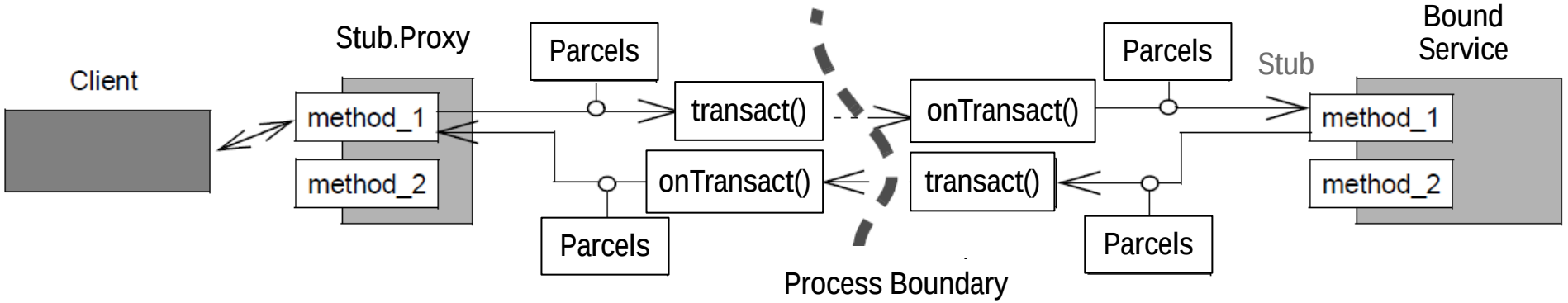




See en.wikipedia.org/wiki/Proxy_pattern

Overview of Topics Covered in Section 3

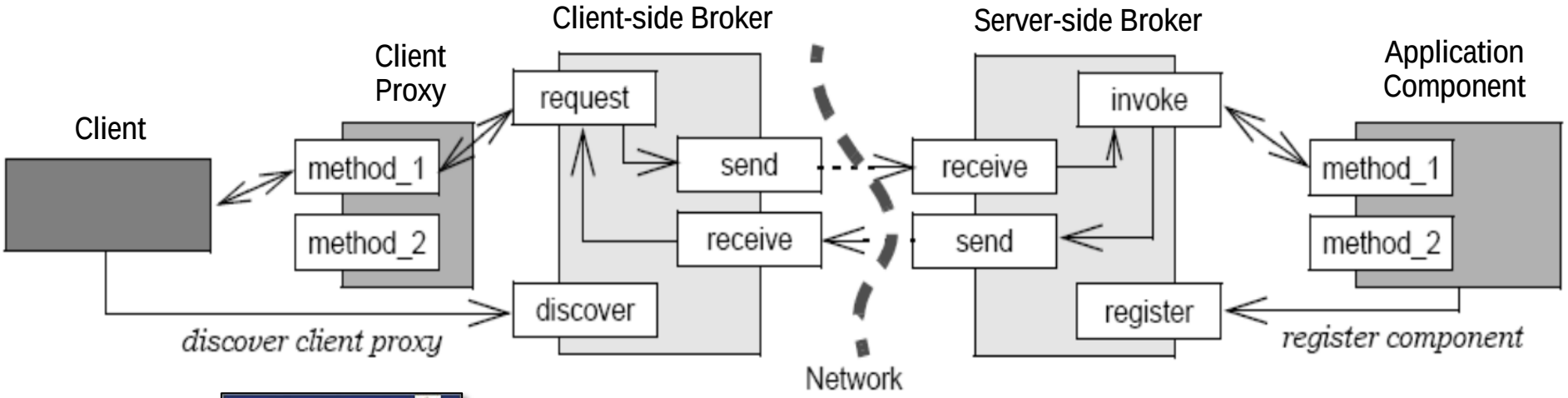
- **Section 3 – Communication Patterns Applied in Android**
  - *Proxy*
    - Provide a surrogate or placeholder for another object to control access to it

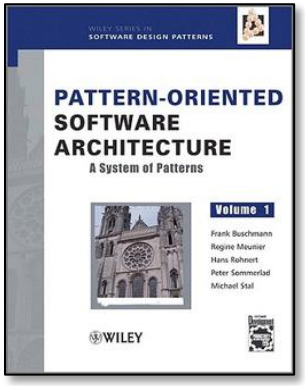


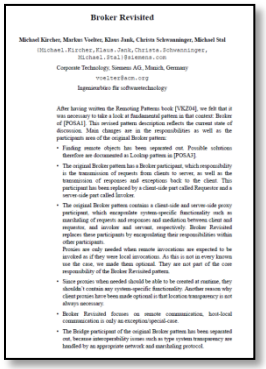
Android's Binder framework uses *Proxy* to support typed IPC

Overview of Topics Covered in Section 3

- **Section 3 – Communication Patterns Applied in Android**
  - *Broker*
    - Connect clients with remote objects by mediating invocations from clients to remote objects, while encapsulating details of local and/or remote IPC



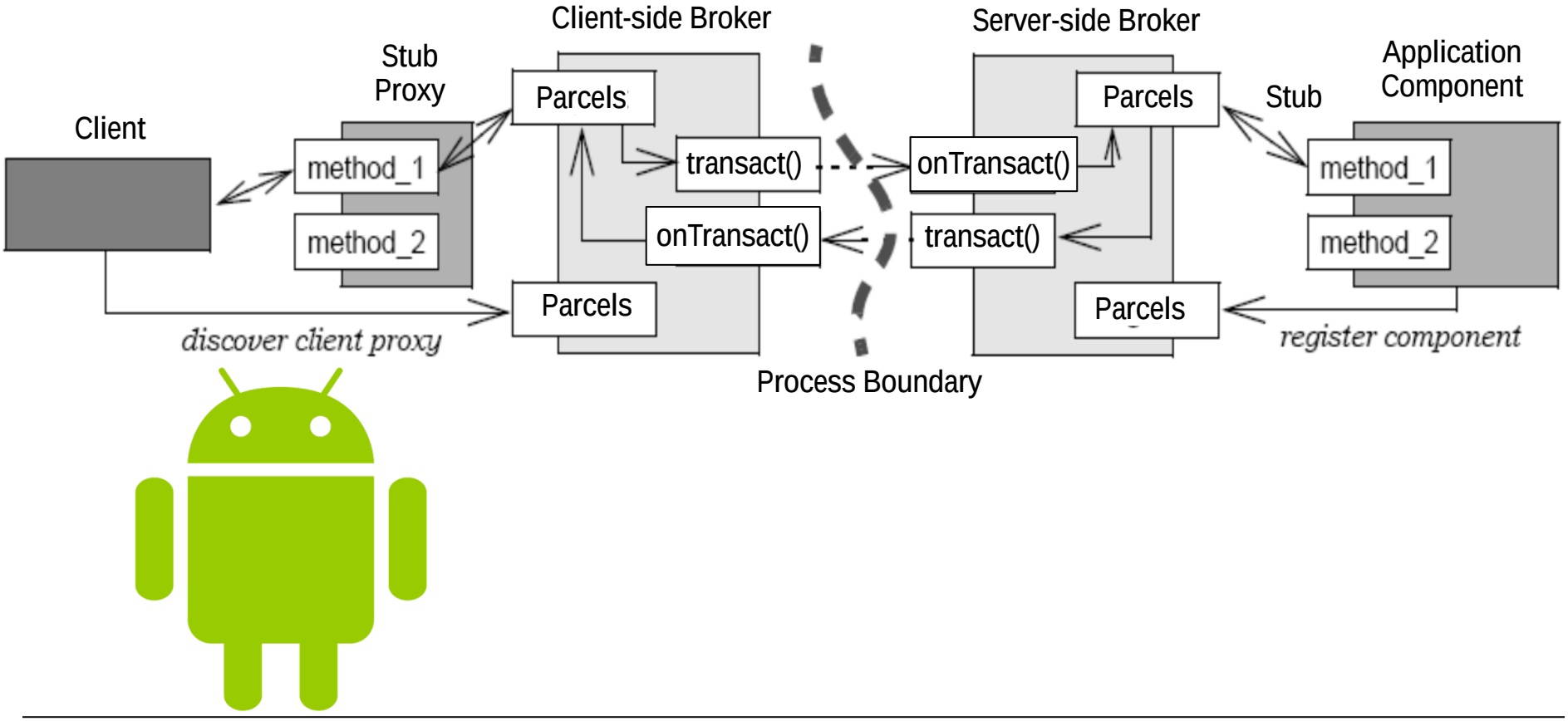




See www.kircher-schwanninger.de/michael/publications/BrokerRevisited.pdf

Overview of Topics Covered in Section 3

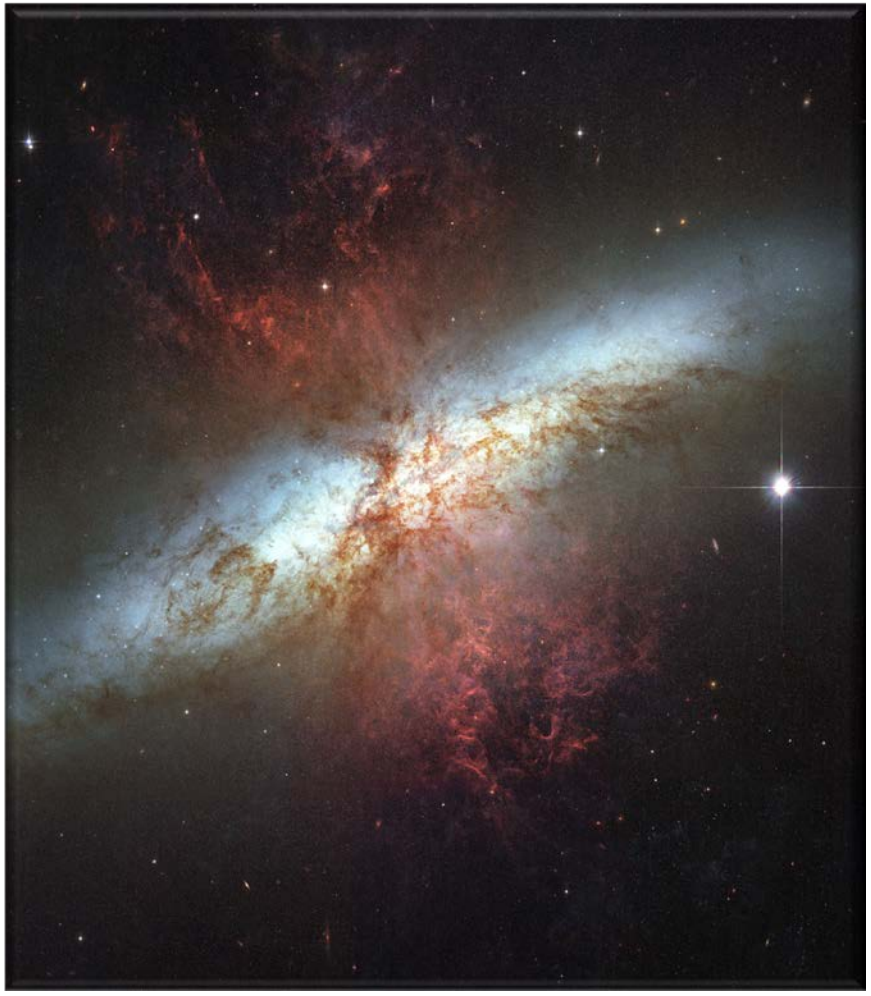
- **Section 3 – Communication Patterns Applied in Android**
  - *Broker*
    - Connect clients with remote objects by mediating invocations from clients to remote objects, while encapsulating details of local and/or remote IPC

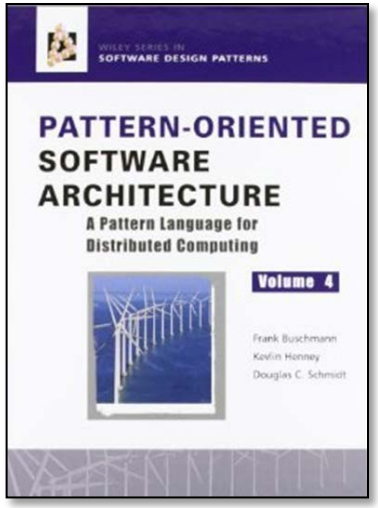


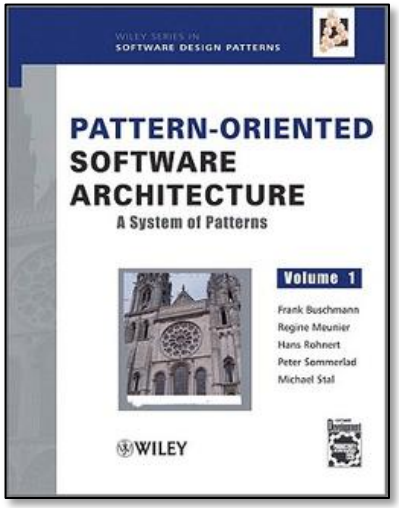
Android's Binder framework uses *Broker* to support object-oriented IPC

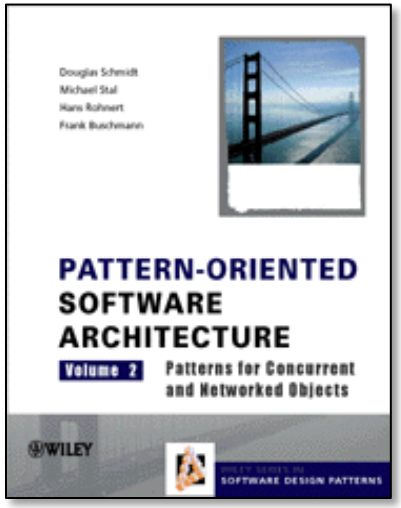
Overview of Topics Covered in Section 3

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







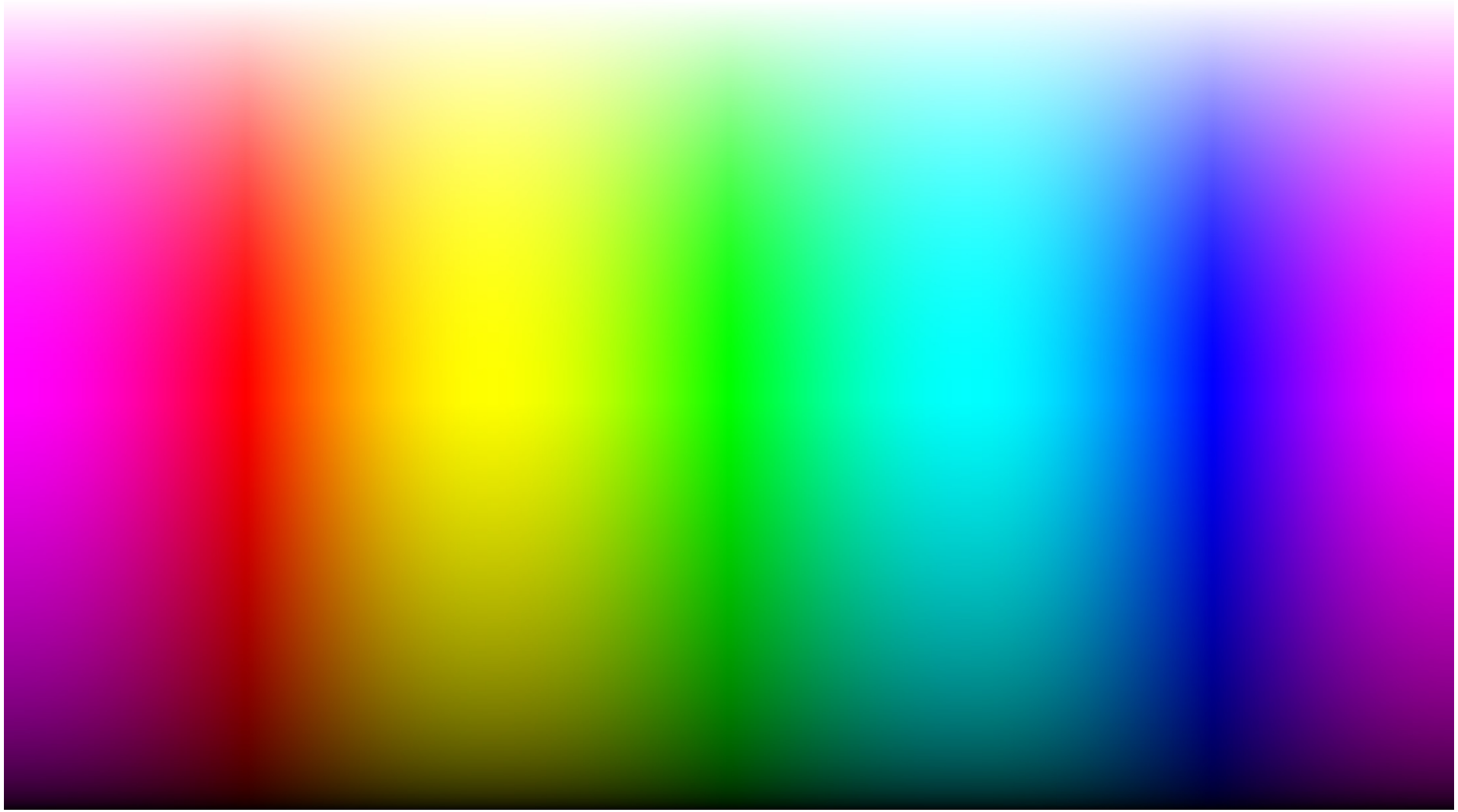


These patterns aren't limited to Android, Java, or mobile device programming

Summary

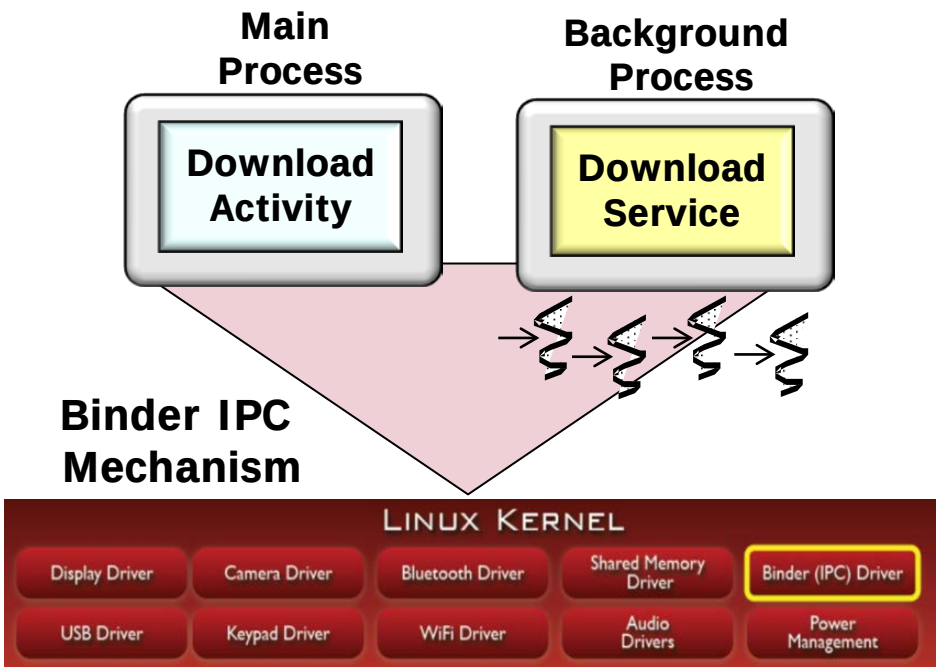
Summary

- This MOOC covers a spectrum of topics



Summary

- This MOOC covers a spectrum of topics



Summary

- This MOOC covers a spectrum of topics
- We analyze lots of Android software

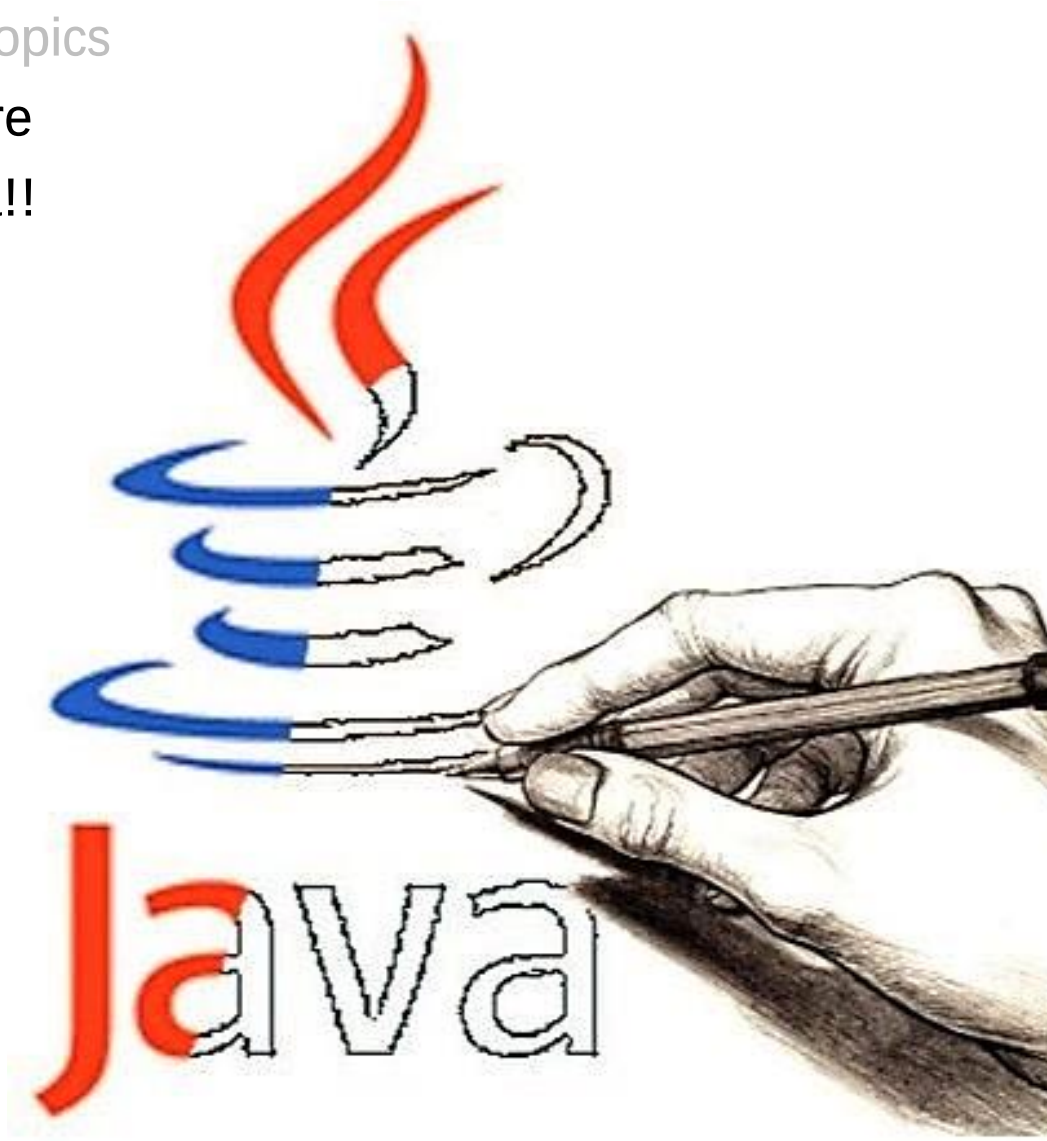
See github.com/douglascraigschmidt/POSA-15/wiki/POSA-15-FAQ item #25

Summary

- This MOOC covers a spectrum of topics
- We analyze lots of Android software
  - It's *essential* to understand Java!!

Summary

- This MOOC covers a spectrum of topics
- We analyze lots of Android software
  - It's *essential* to understand Java!!



See github.com/douglascraigschmidt/POSA-15/wiki/POSA-15-FAQ item #7