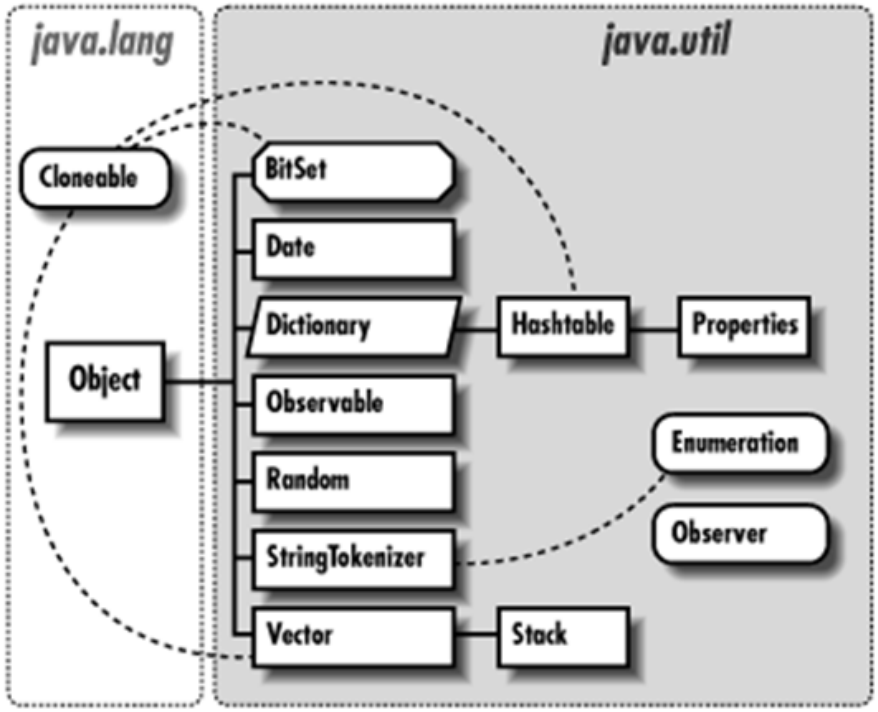
Overview of the Android Runtime: Core Java Libraries

Android Runtime

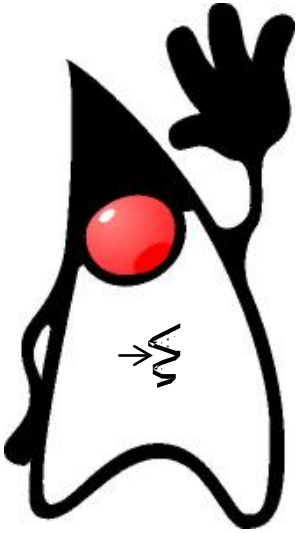
- Supports concurrently executing Java apps on mobile devices
  - Virtual Machine (VM)
  - **Core Libraries**
    - Core Java classes
      - java.*, javax.*



See en.wikipedia.org/wiki/Comparison_of_Java_and_Android_API

Android Runtime

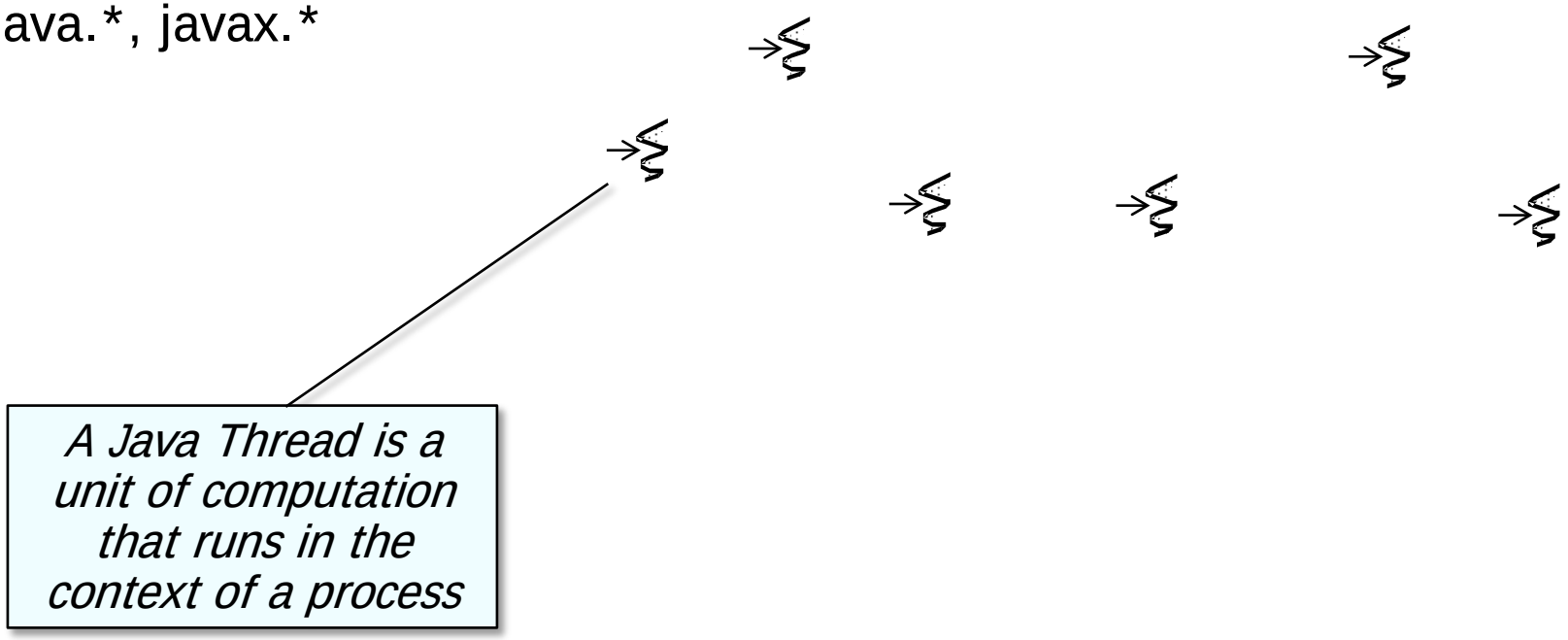
- Supports concurrently executing Java apps on mobile devices
  - Virtual Machine (VM)
  - **Core Libraries**
    - Core Java classes
      - java.*, javax.*



See developer.android.com/reference/java/lang/Thread.html

Android Runtime

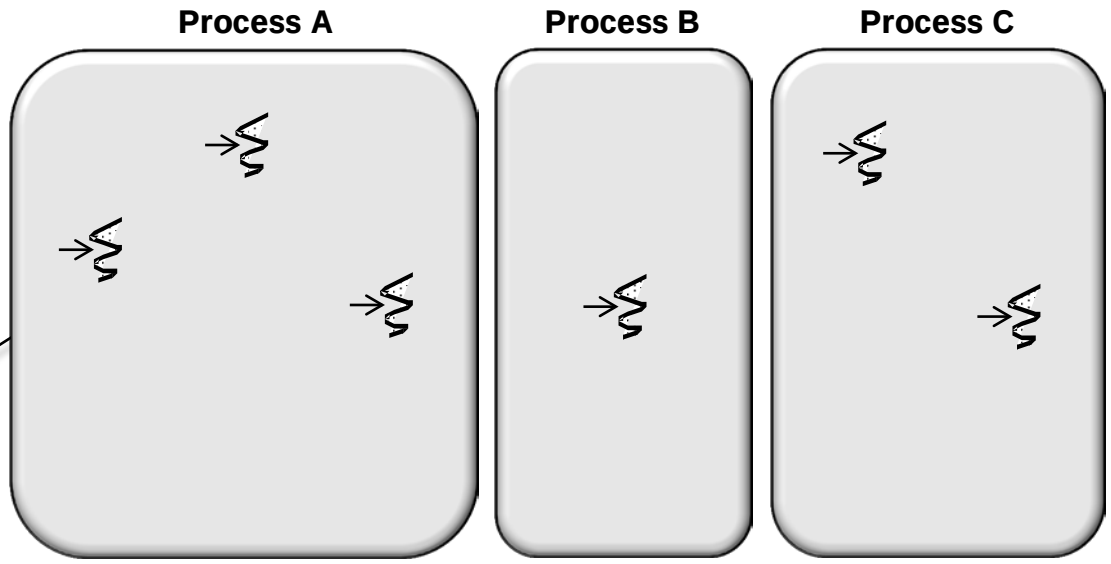
- Supports concurrently executing Java apps on mobile devices
  - Virtual Machine (VM)
  - **Core Libraries**
    - Core Java classes
      - java.*, javax.*



See en.wikipedia.org/wiki/Thread_(computing)

Android Runtime

- Supports concurrently executing Java apps on mobile devices
  - Virtual Machine (VM)
  - **Core Libraries**
    - Core Java classes
      - java.*, javax.*

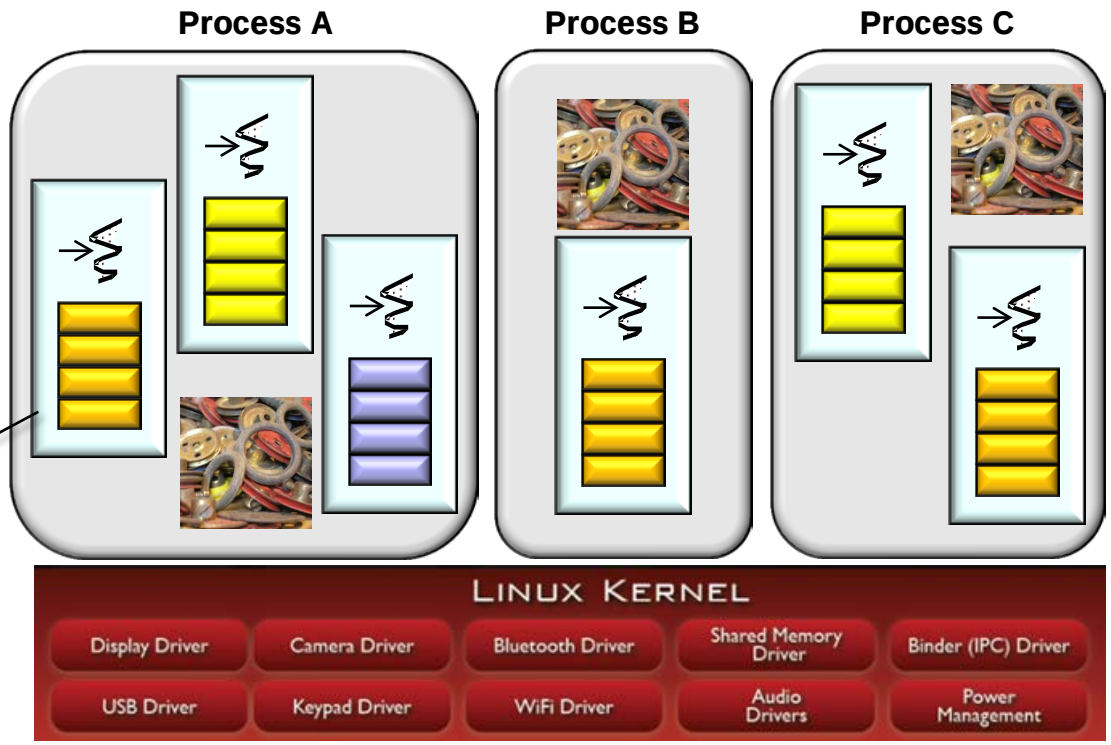


Threads running in a process can communicate with each other via shared objects or message passing

See developer.android.com/reference/java/util/concurrent/package-summary.html

Android Runtime

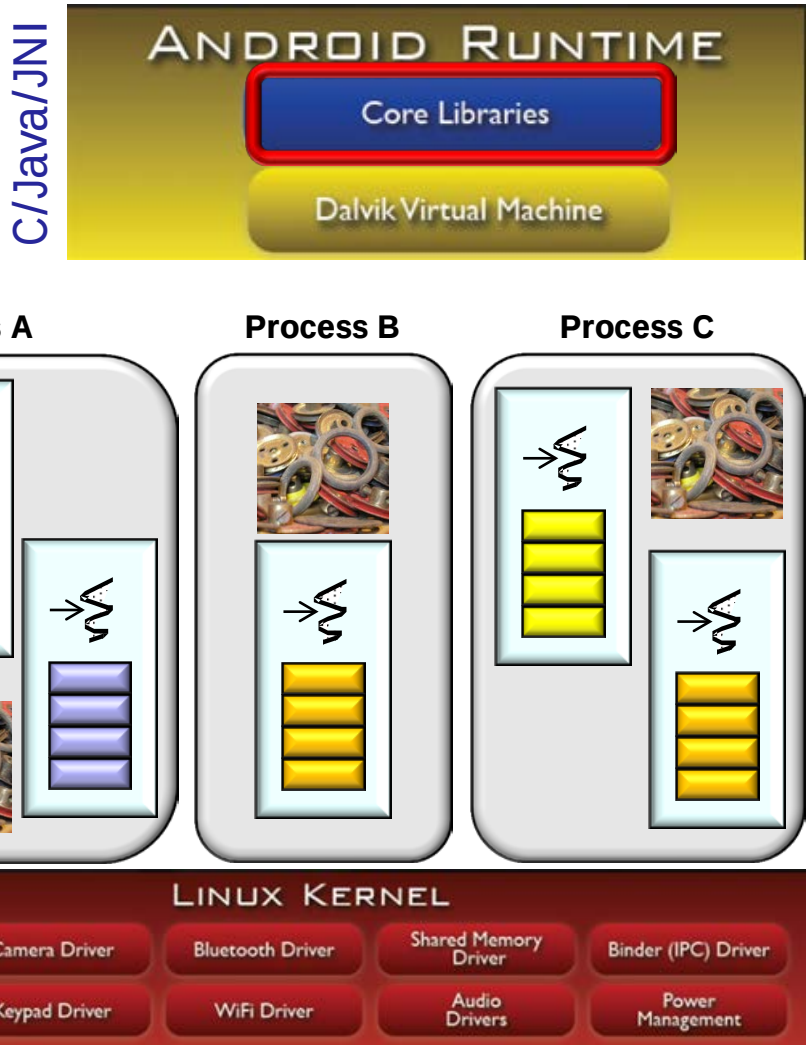
- Supports concurrently executing Java apps on mobile devices
  - Virtual Machine (VM)
  - **Core Libraries**
    - Core Java classes
      - java.*, javax.*



Each Java Thread has a stack, a program counter, & other registers (unique "state")

Android Runtime

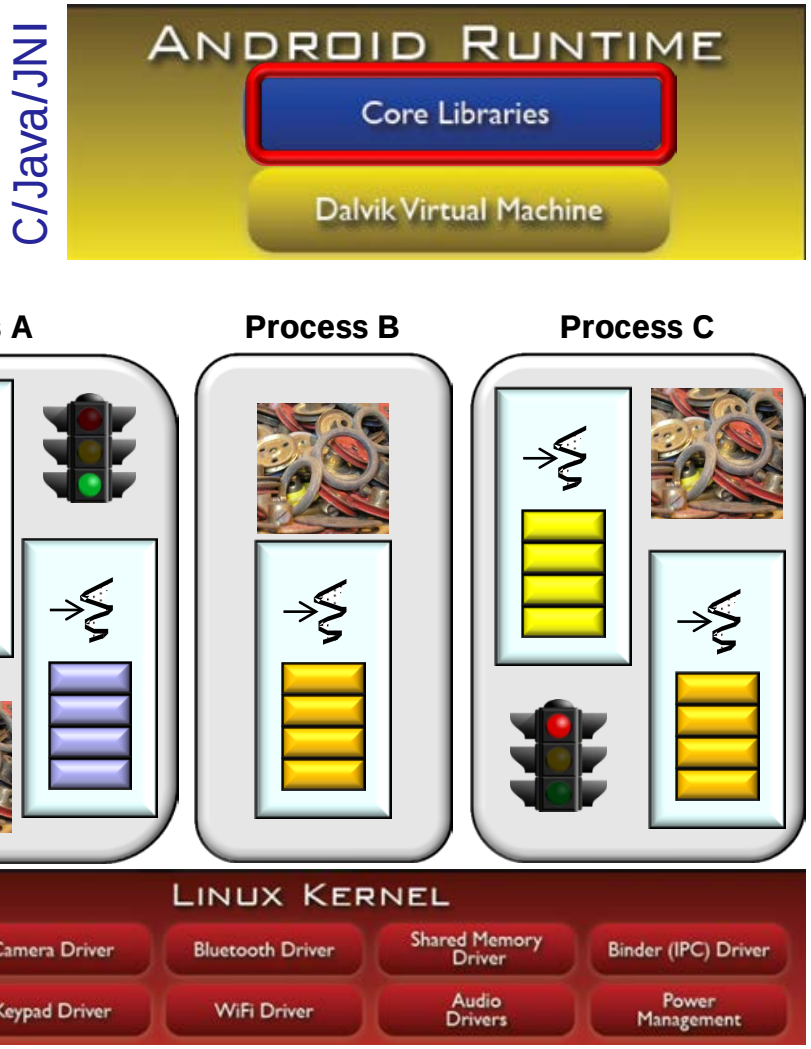
- Supports concurrently executing Java apps on mobile devices
  - Virtual Machine (VM)
  - **Core Libraries**
    - Core Java classes
      - java.*, javax.*



The heap & static areas are shared across Threads (common "state")

Android Runtime

- Supports concurrently executing Java apps on mobile devices
  - Virtual Machine (VM)
  - **Core Libraries**
    - Core Java classes
      - java.*, javax.*



Java synchronization mechanisms can be used to prevent race conditions

See en.wikipedia.org/wiki/Race_condition