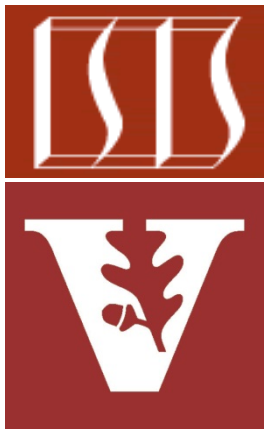
Android Concurrency: Overview of Patterns in Android Communication Frameworks

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt



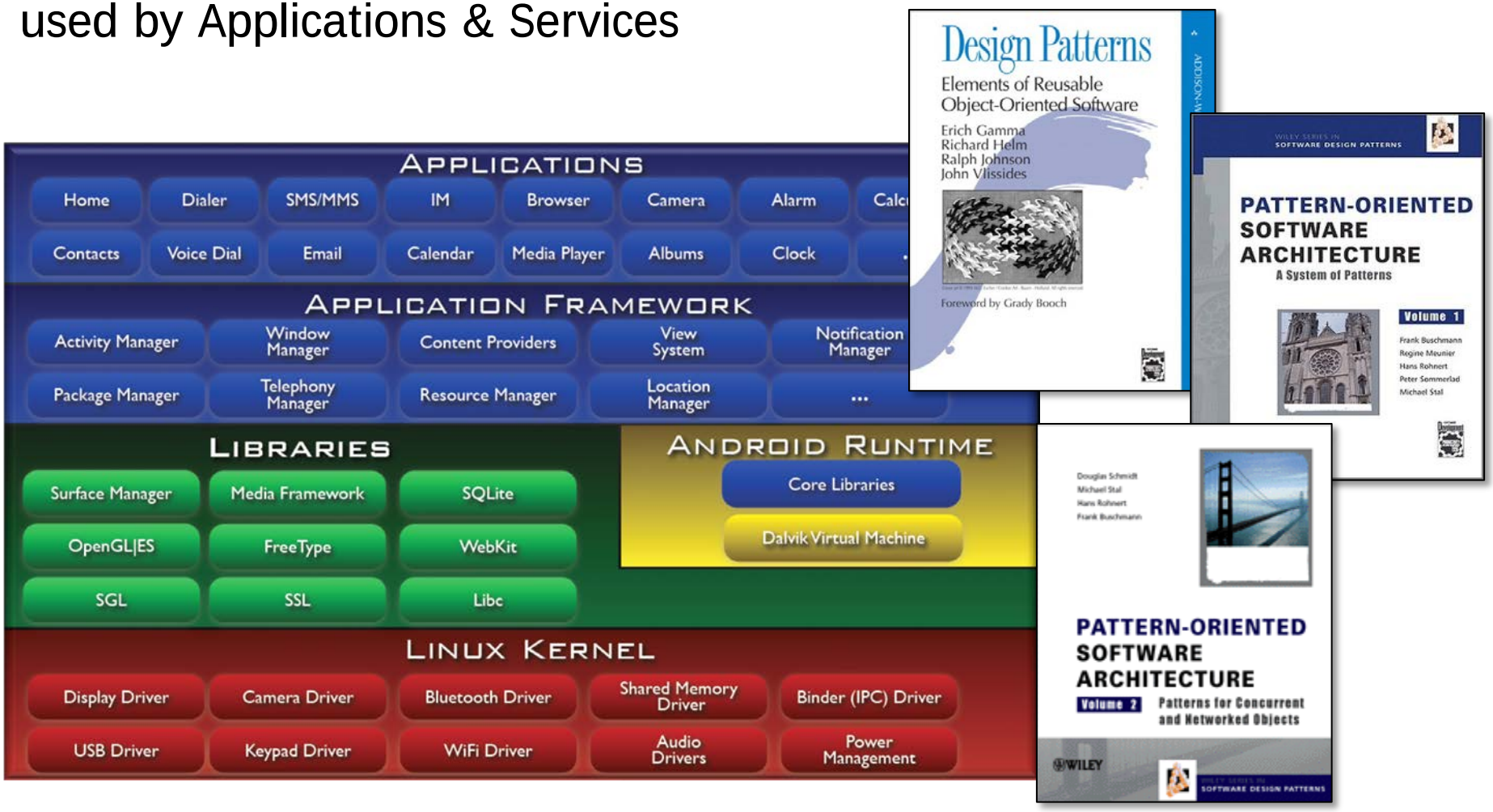
Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Module

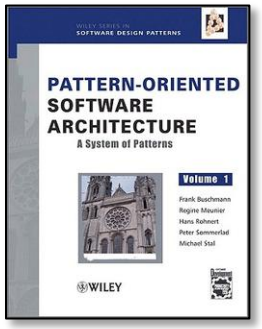
- Understand how *patterns* & *frameworks* help improve the structure & functionality of Android's concurrency & communication middleware used by Applications & Services

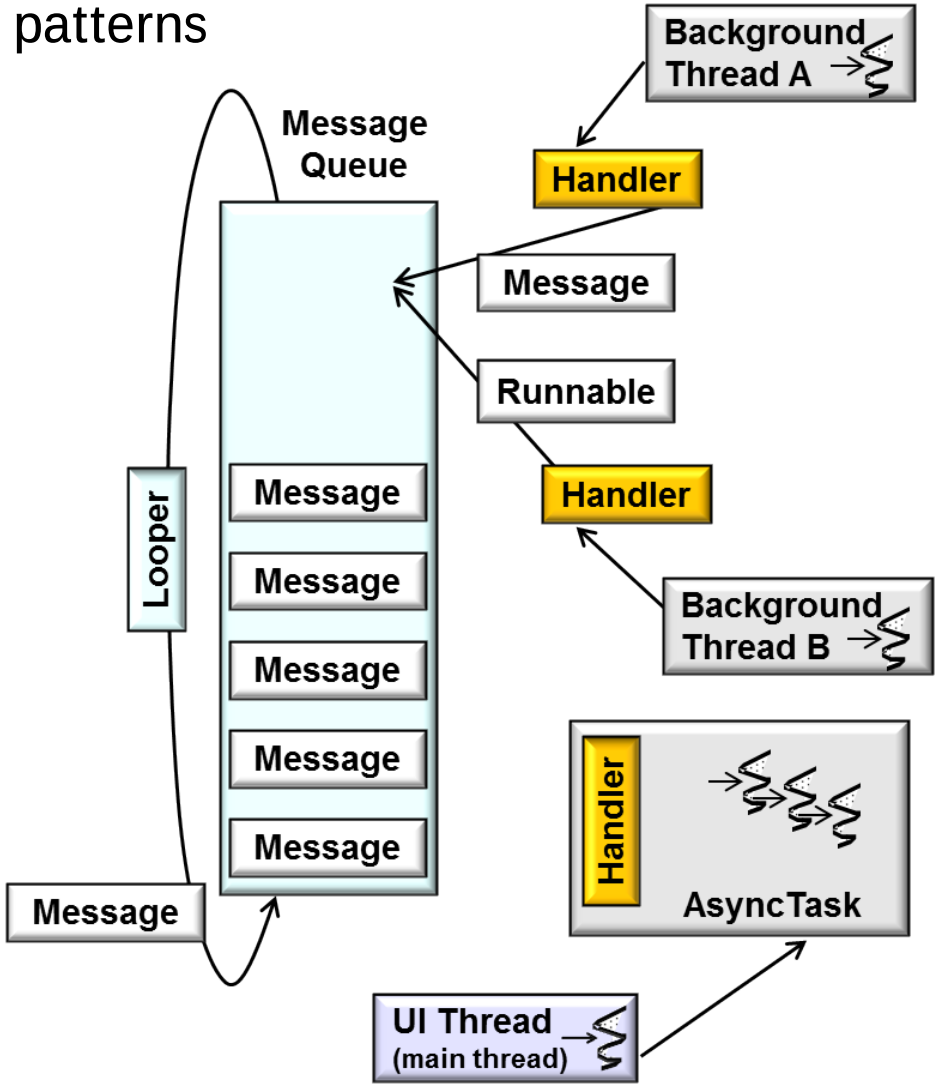


Patterns in Android Communication Frameworks (Part 1)

Patterns in Android Communication Frameworks

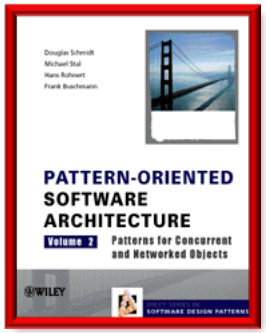
- Android's communication frameworks are designed, implemented, & integrated in accordance with many patterns

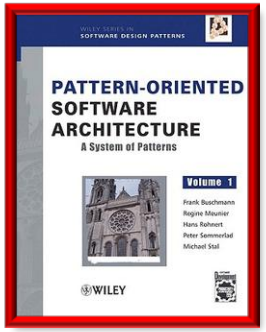


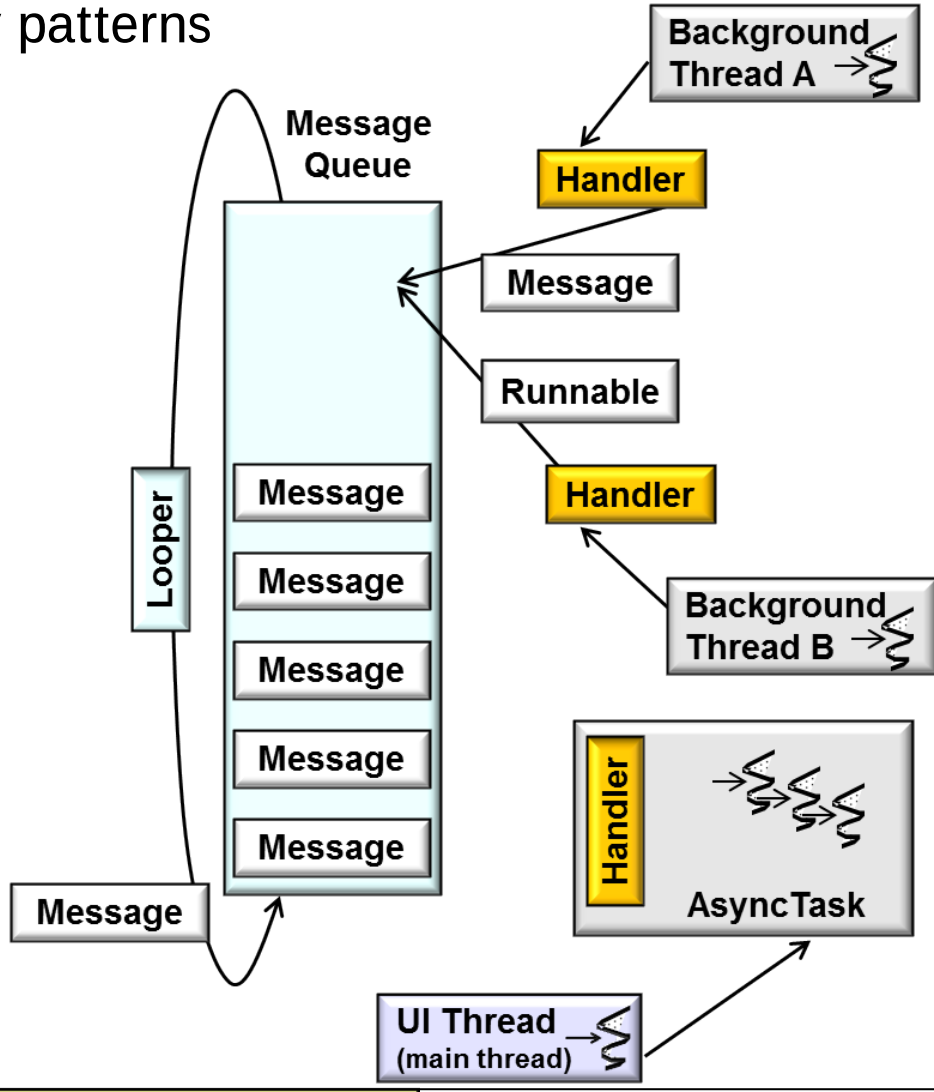


Patterns in Android Communication Frameworks

- Android's communication frameworks are designed, implemented, & integrated in accordance with many patterns



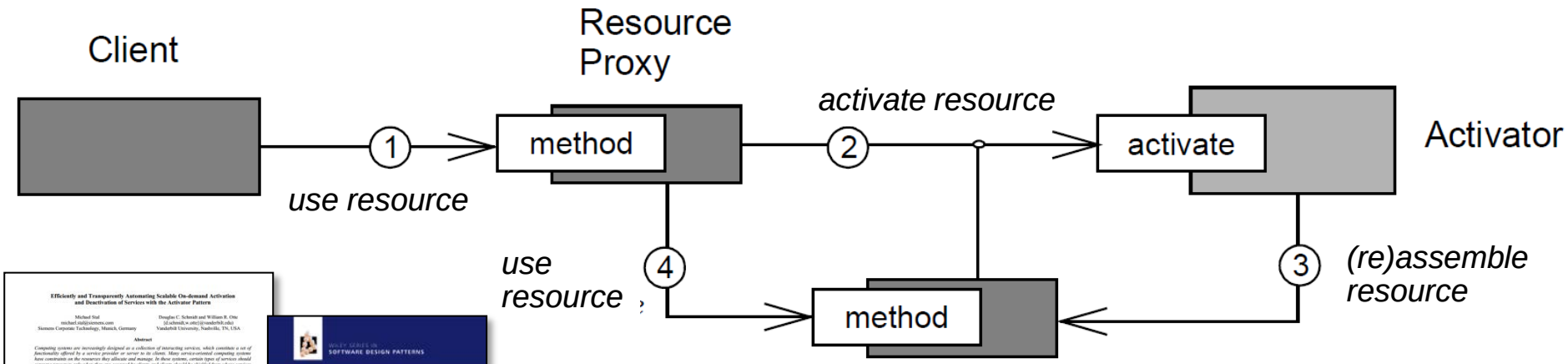




Many patterns are based on material in the POSA books

Patterns in Android Communication Frameworks

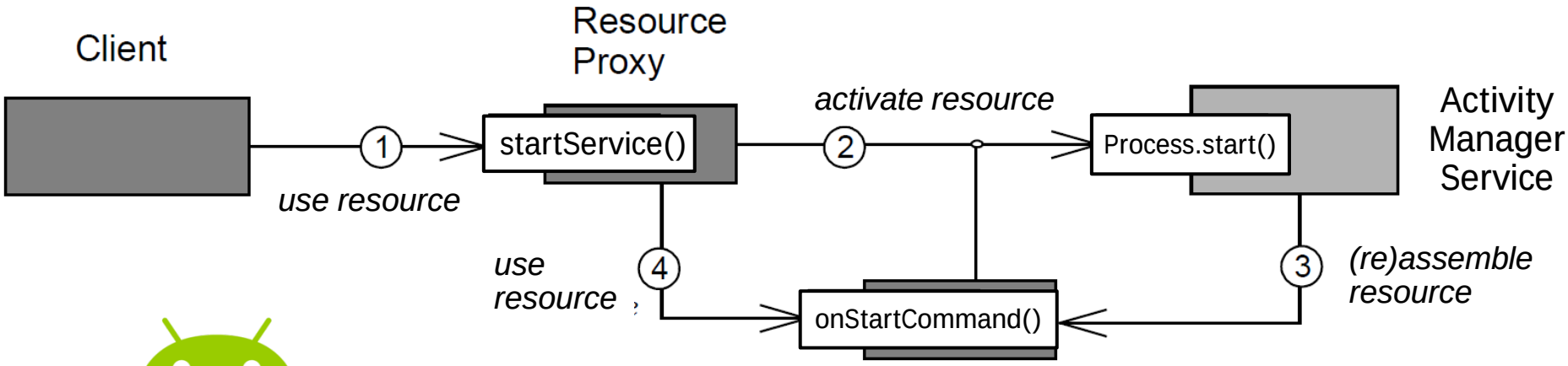
- Android's communication frameworks are designed, implemented, & integrated in accordance with many patterns
  - *Activator* – Automates scalable on-demand activation & deactivation of service execution contexts to run services accessed by many clients without consuming excessive resources



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

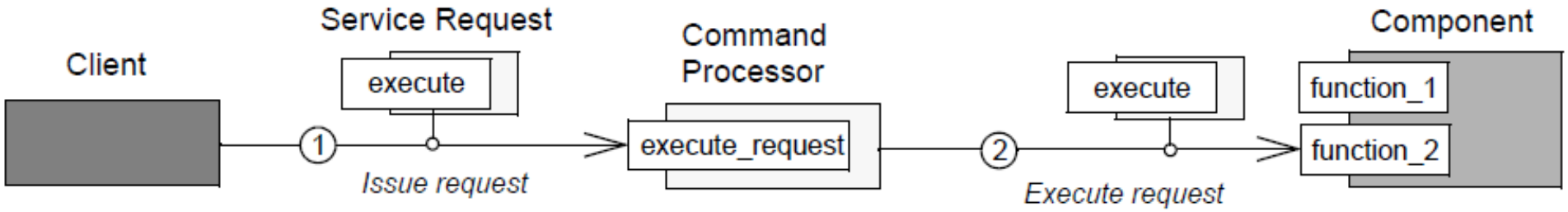
Patterns in Android Communication Frameworks

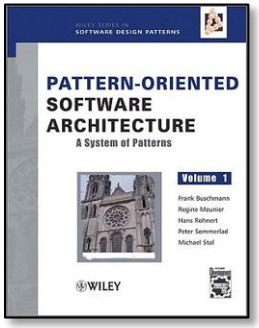
- Android's communication frameworks are designed, implemented, & integrated in accordance with many patterns
  - *Activator* – Automates scalable on-demand activation & deactivation of service execution contexts to run services accessed by many clients without consuming excessive resources



Patterns in Android Communication Frameworks

- Android's communication frameworks are designed, implemented, & integrated in accordance with many patterns
  - *Command Processor* – Package a piece of application functionality—as well as its parameterization in an object—to execute it in another context
    - e.g., at a later point in time, in a different process or thread, etc.

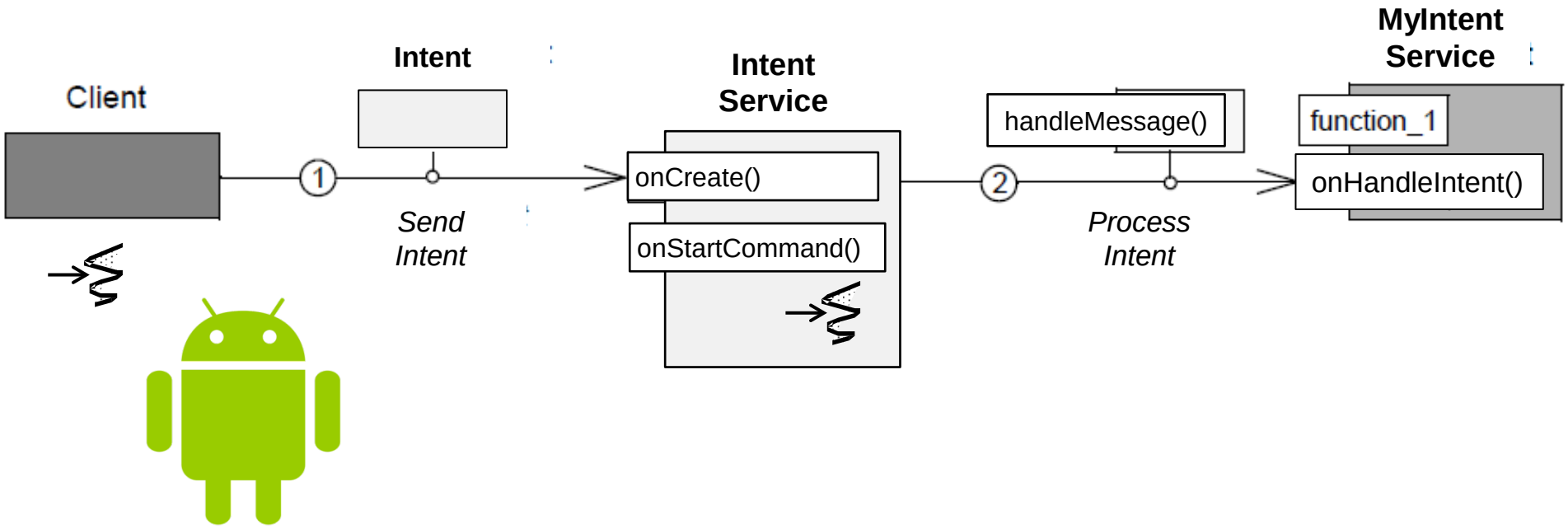




See www.dre.vanderbilt.edu/~schmidt/PDF/CommandRevisited.pdf

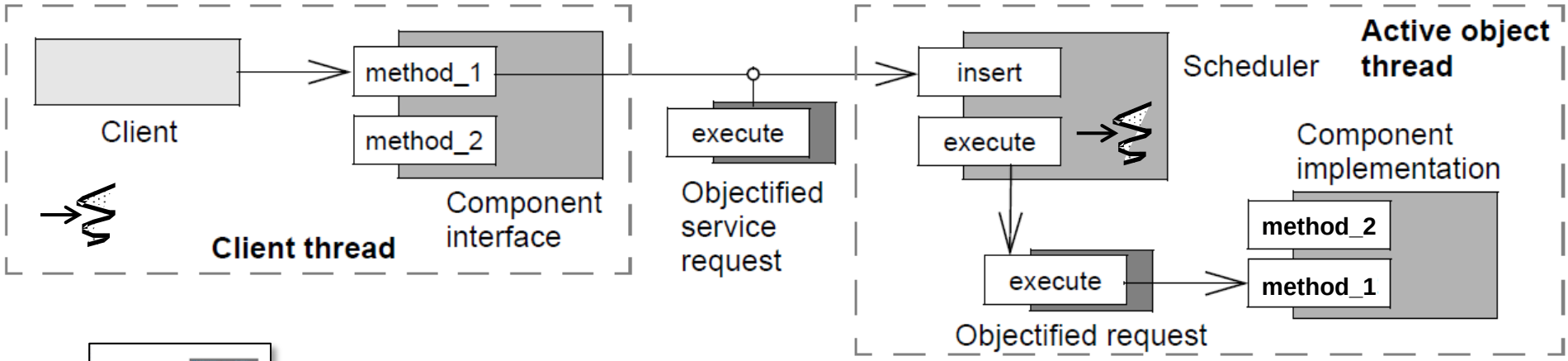
Patterns in Android Communication Frameworks

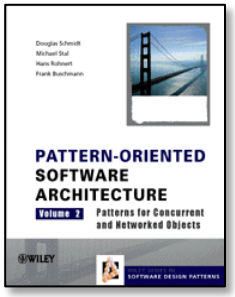
- Android's communication frameworks are designed, implemented, & integrated in accordance with many patterns
  - *Command Processor* – Package a piece of application functionality—as well as its parameterization in an object—to execute it in another context
    - e.g., at a later point in time, in a different process or thread, etc.



Patterns in Android Communication Frameworks

- Android's communication frameworks are designed, implemented, & integrated in accordance with many patterns
  - *Active Object* – Define service requests on components as the units of concurrency & run service requests on a component in different thread(s) from the requesting client thread

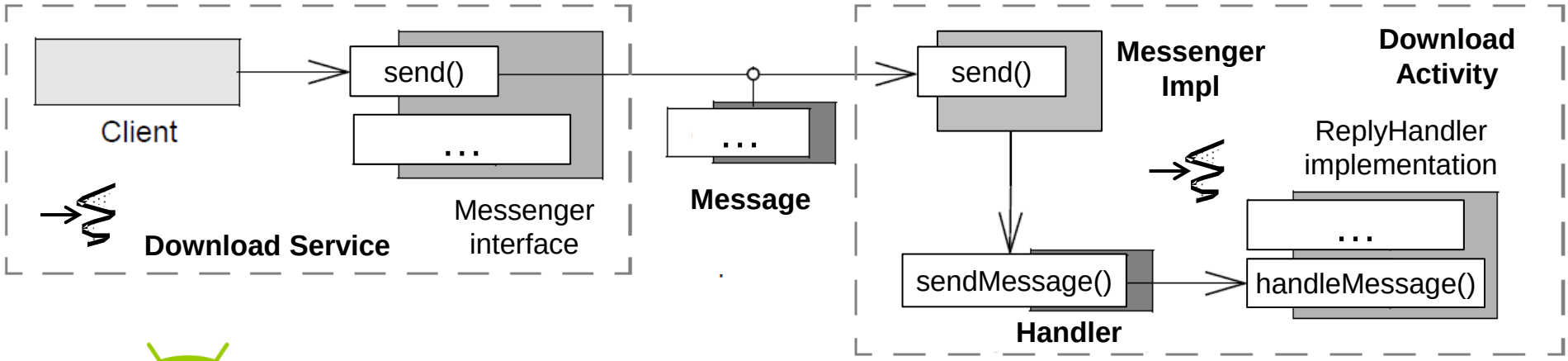




See en.wikipedia.org/wiki/Active_object

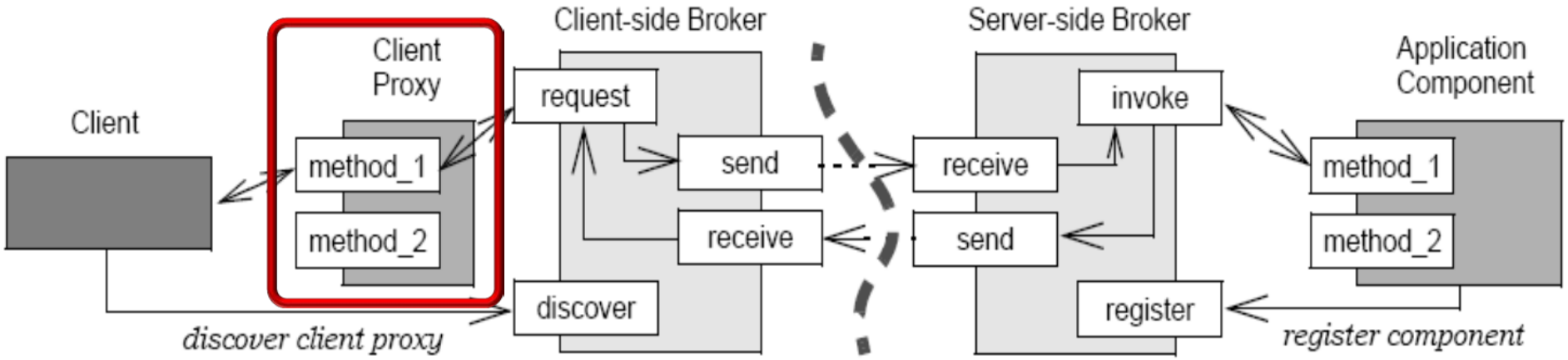
Patterns in Android Communication Frameworks

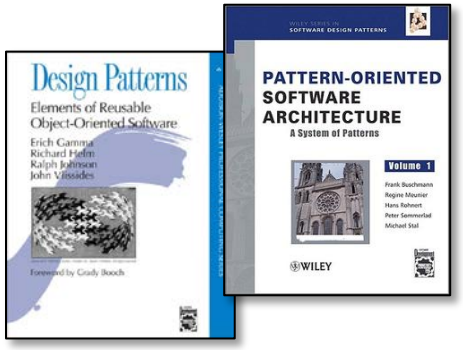
- Android's communication frameworks are designed, implemented, & integrated in accordance with many patterns
  - *Active Object* – Define service requests on components as the units of concurrency & run service requests on a component in different thread(s) from the requesting client thread



Patterns in Android Communication Frameworks

- Android's communication frameworks are designed, implemented, & integrated in accordance with many patterns
  - *Proxy* – Provide a surrogate or placeholder for another object to control access to it

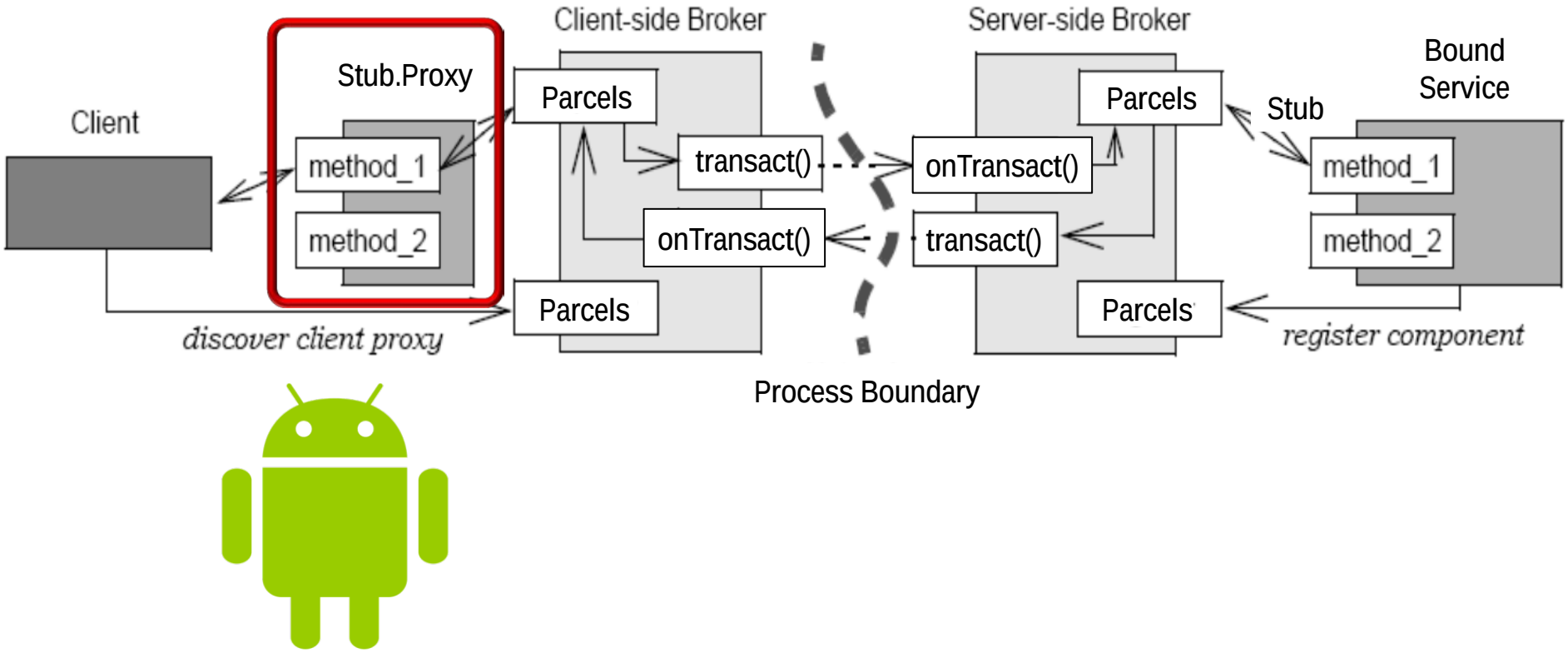




See en.wikipedia.org/wiki/Proxy_pattern

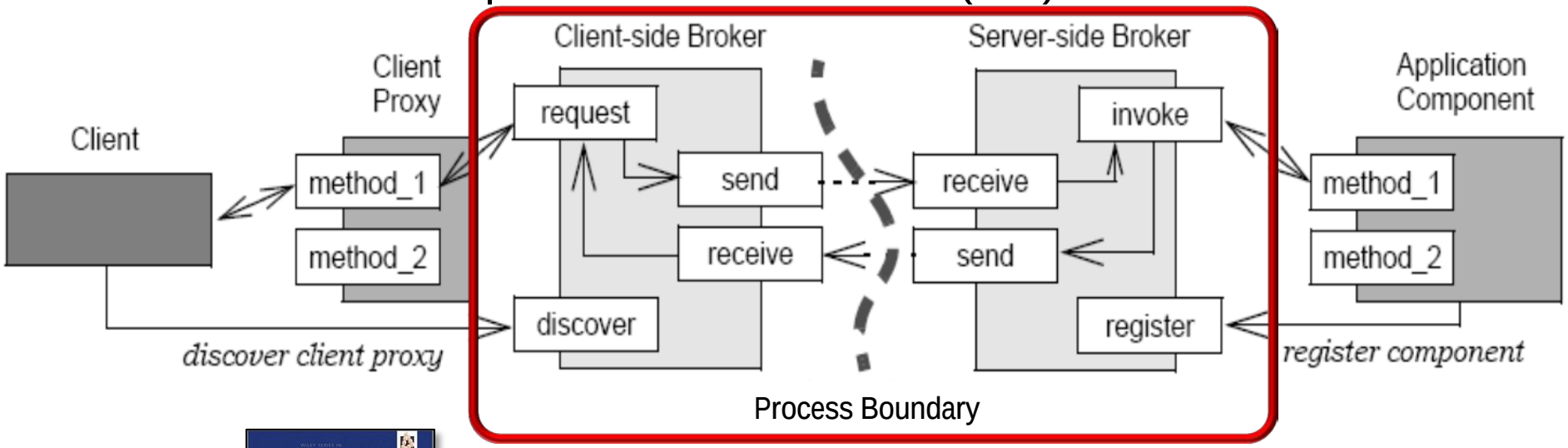
Patterns in Android Communication Frameworks

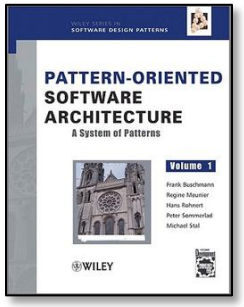
- Android's communication frameworks are designed, implemented, & integrated in accordance with many patterns
  - *Proxy* – Provide a surrogate or placeholder for another object to control access to it



Patterns in Android Communication Frameworks

- Android's communication frameworks are designed, implemented, & integrated in accordance with many patterns
  - *Broker* – Connect clients with remote objects by mediating invocations from clients to remote objects, while encapsulating the details of local and/or remote inter-process communication (IPC)





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

Patterns in Android Communication Frameworks

- Android's communication frameworks are designed, implemented, & integrated in accordance with many patterns
  - *Broker* – Connect clients with remote objects by mediating invocations from clients to remote objects, while encapsulating the details of local and/or remote inter-process communication (IPC)