Developing Android REST Client Applications

Tomche Delev

Java User Group
Macedonia
www.jug.mk

December 18, 2011

REST Client Applications

- REST: A broadly adopted architecture style
- A large number of REST APIs are available
- Why develop them if mobile friendly web sites alreadyexist?

REST Client Applications

- REST: A broadly adopted architecture style
- A large number of REST APIs are available
- Why develop them if mobile friendly web sites alreadyexist?

REST Client Applications

- REST: A broadly adopted architecture style
- A large number of REST APIs are available
- Why develop them if mobile friendly web sites alreadyexist?

REST Client Applications

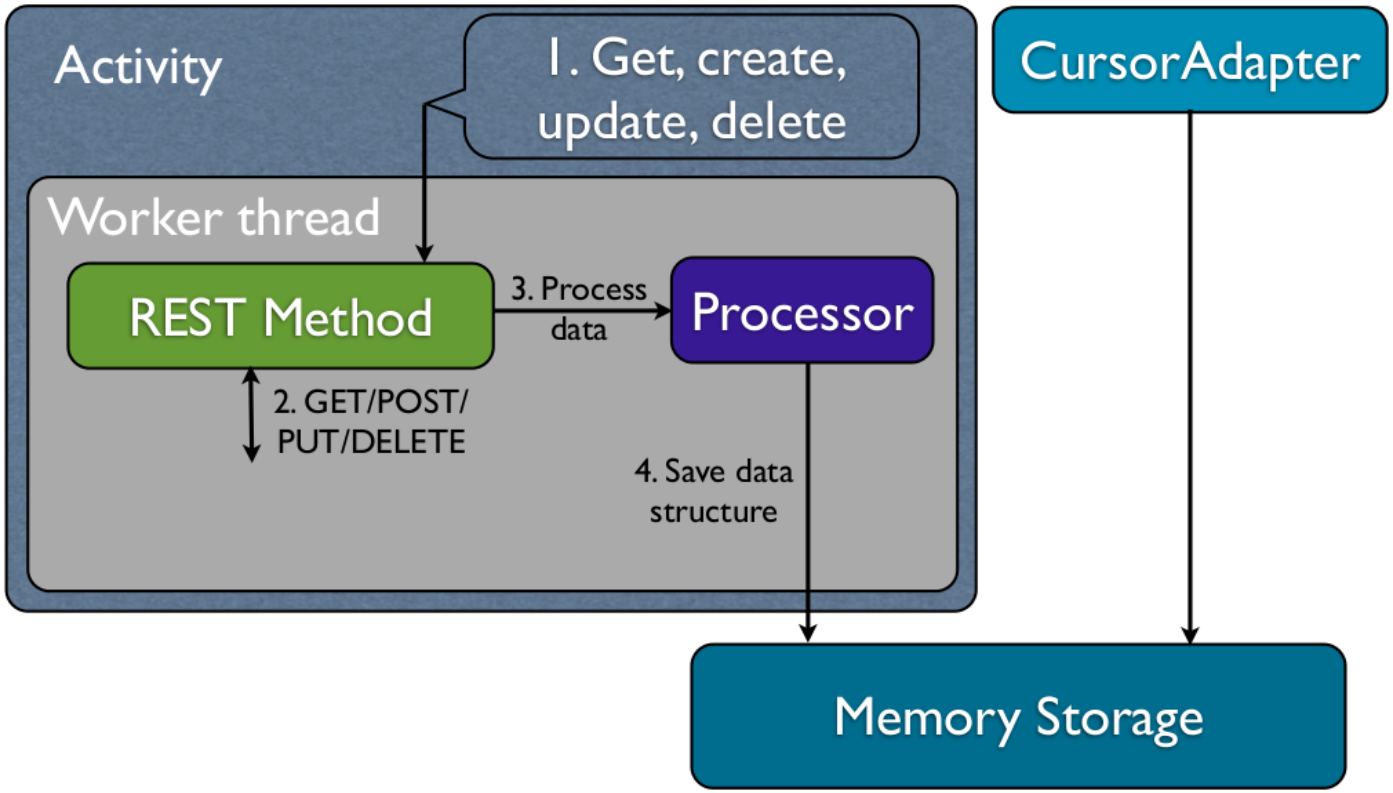
- REST: A broadly adopted architecture style
- A large number of REST APIs are available
- Why develop them if mobile friendly web sites alreadyexist?

Incorrect Implementation of REST Methods

"I haven't failed, I've found 10,000 ways that don't work."

- Thomas Alva Edison

The Incorrect Implementation of REST Methos



Whats wrong with this approach?

- The operating system may shut down the process
- Data is not persistently stored

Whats wrong with this approach?

- The operating system may shut down the process
- Data is not persistently stored

Whats wrong with this approach?

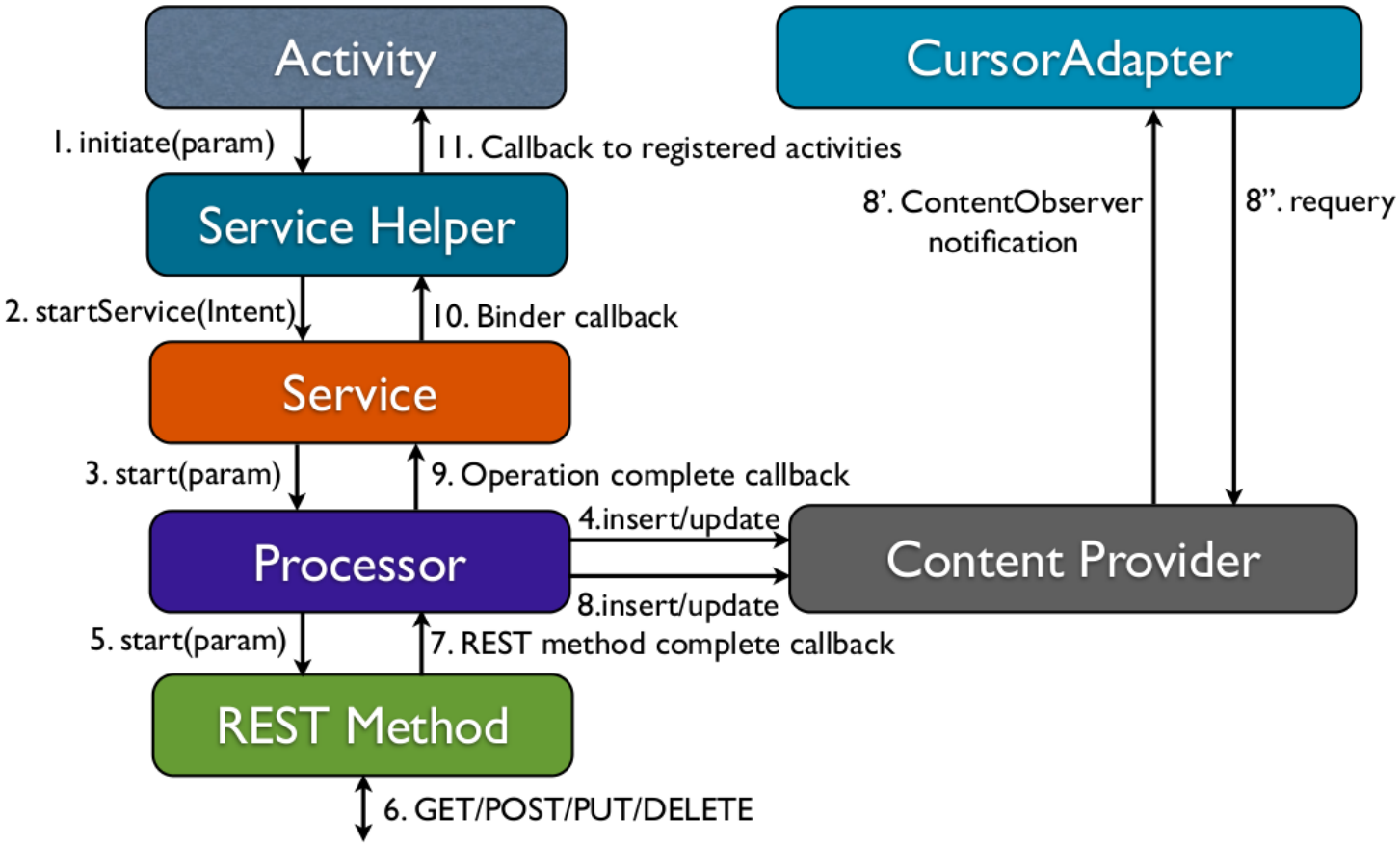
- The operating system may shut down the process
- Data is not persistently stored

Implementing REST Methods

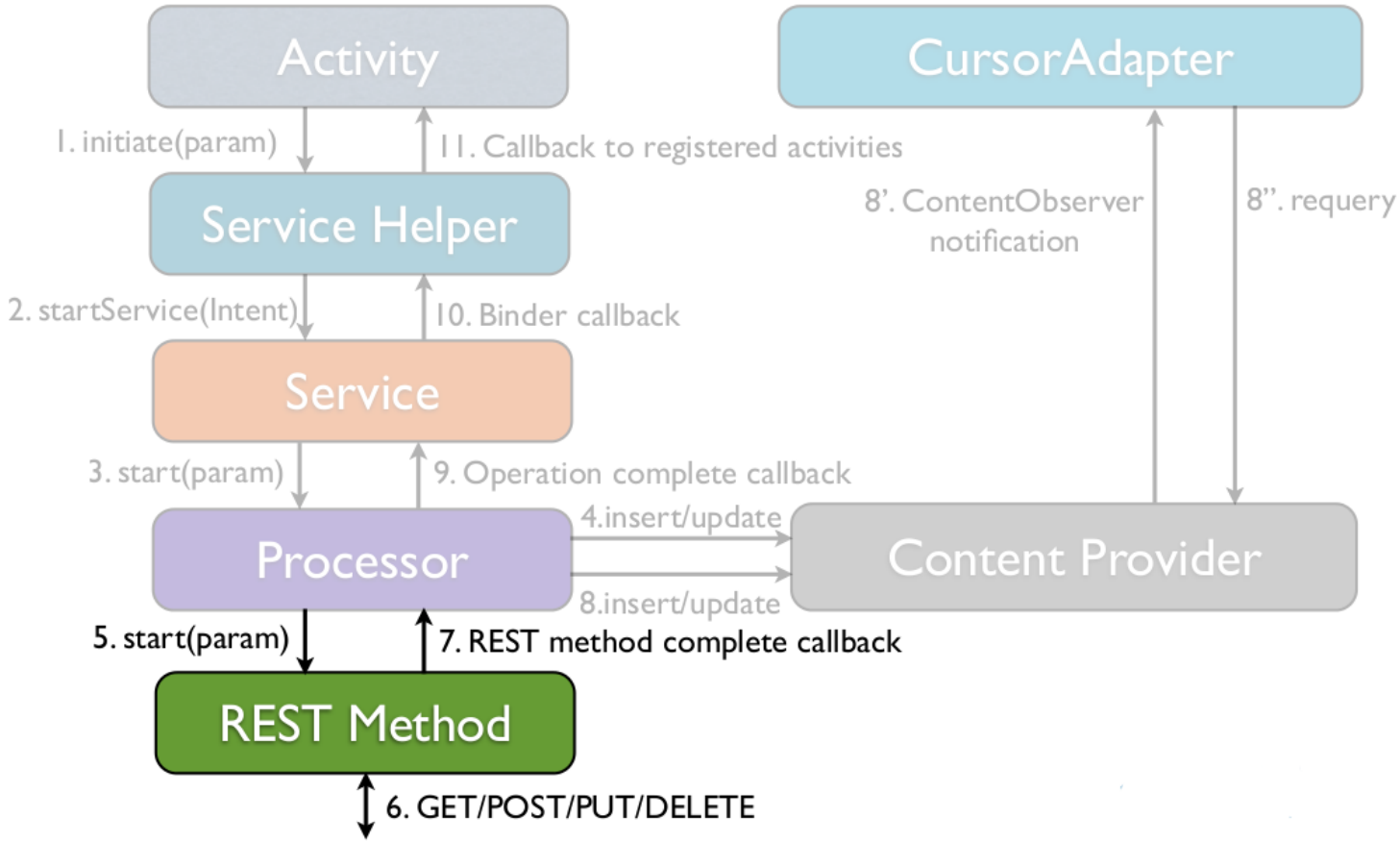
Theres a way to do it better ... find it.

- Thomas Alva Edison

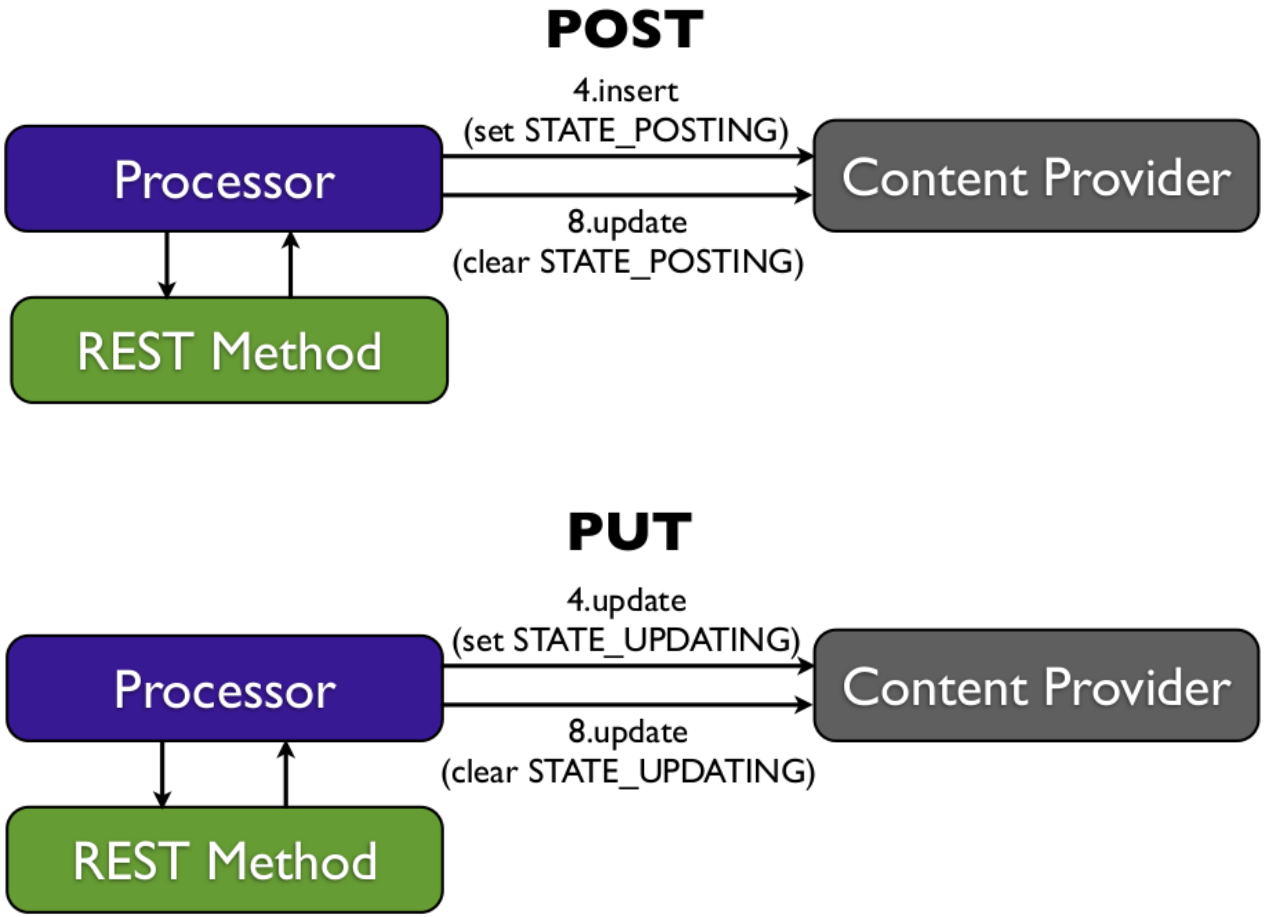
Implementing REST Methods using a Service API



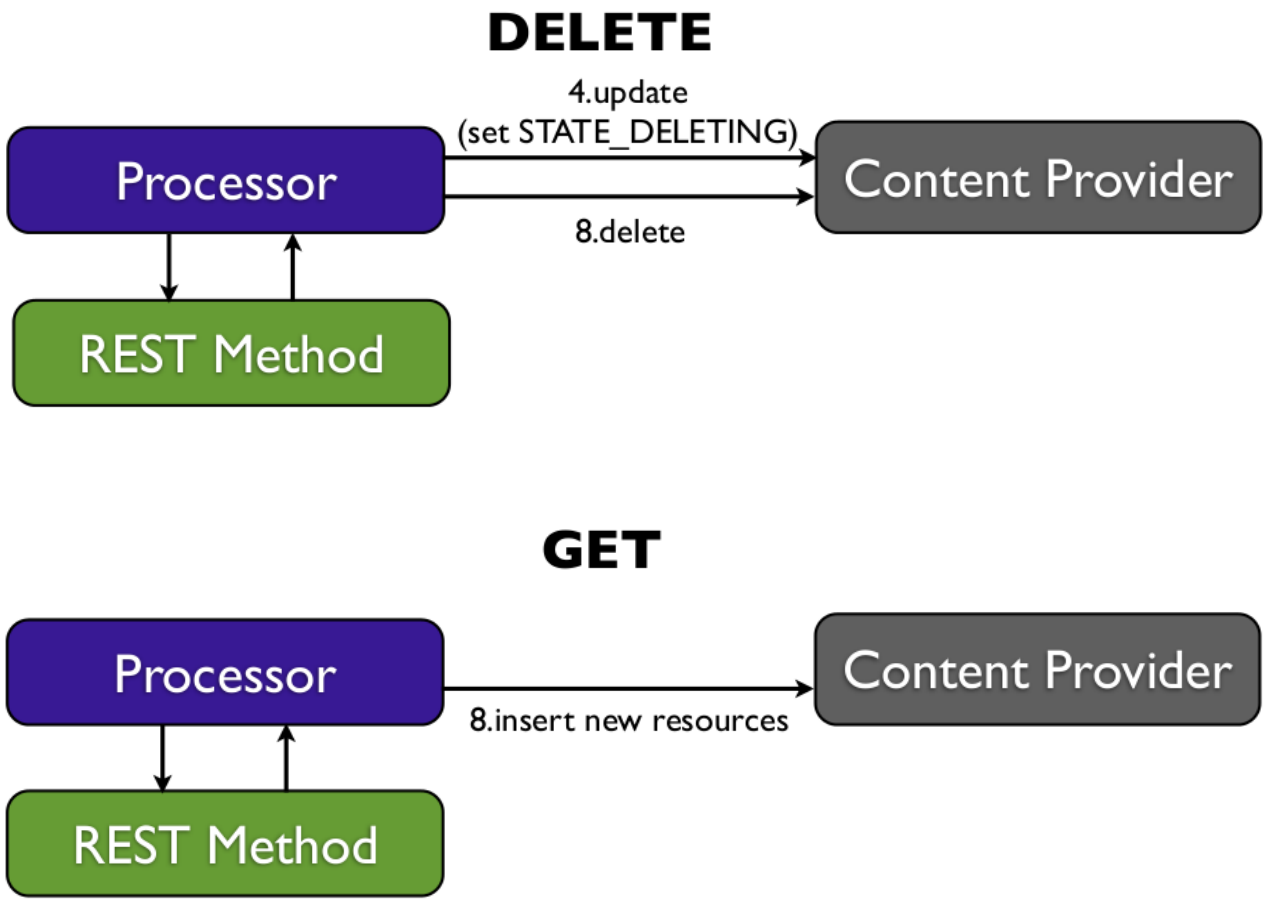
Implementing REST Methods using a Service API



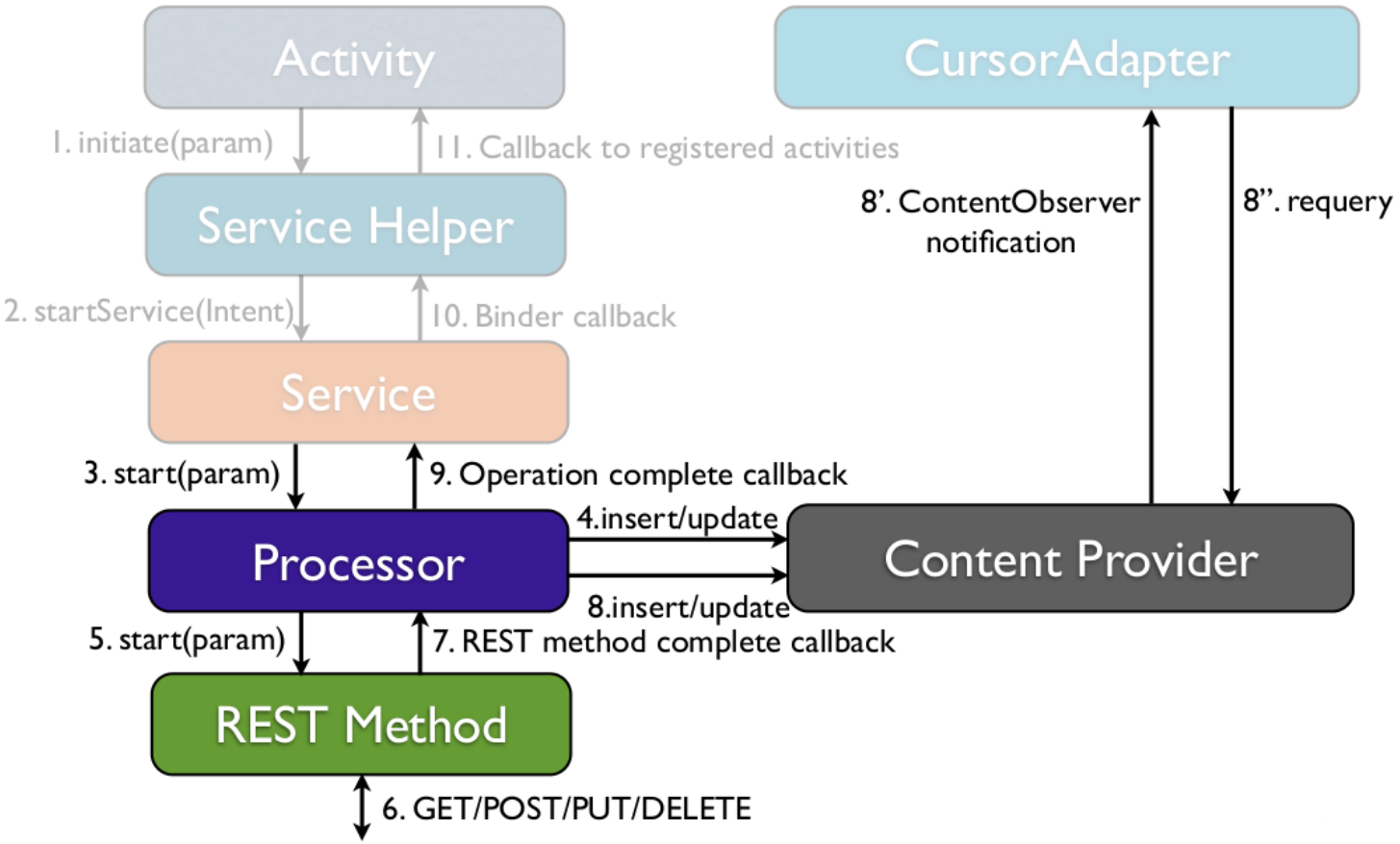
The Processor (POST & PUT)



The Processor (DELTE & GET)



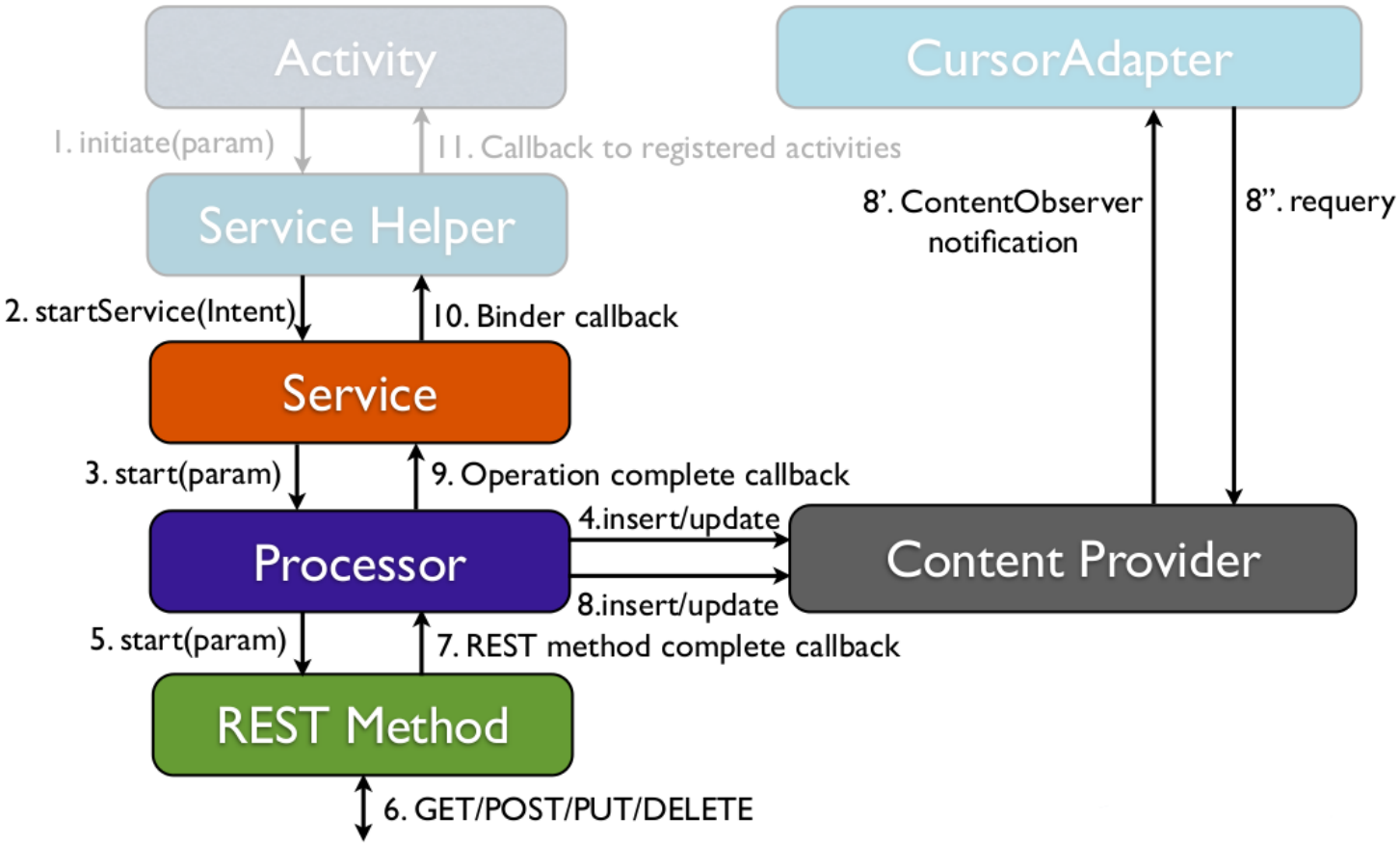
Implementing REST Methods using a Service API



The Service

- The role of the service
- Forward path: receives the Intent sent by the Service
- Helper and starts the corresponding REST Method
- Return path: handles the Processor callback and invokes the Service Helper binder callback
- It can implement a queue of downloads

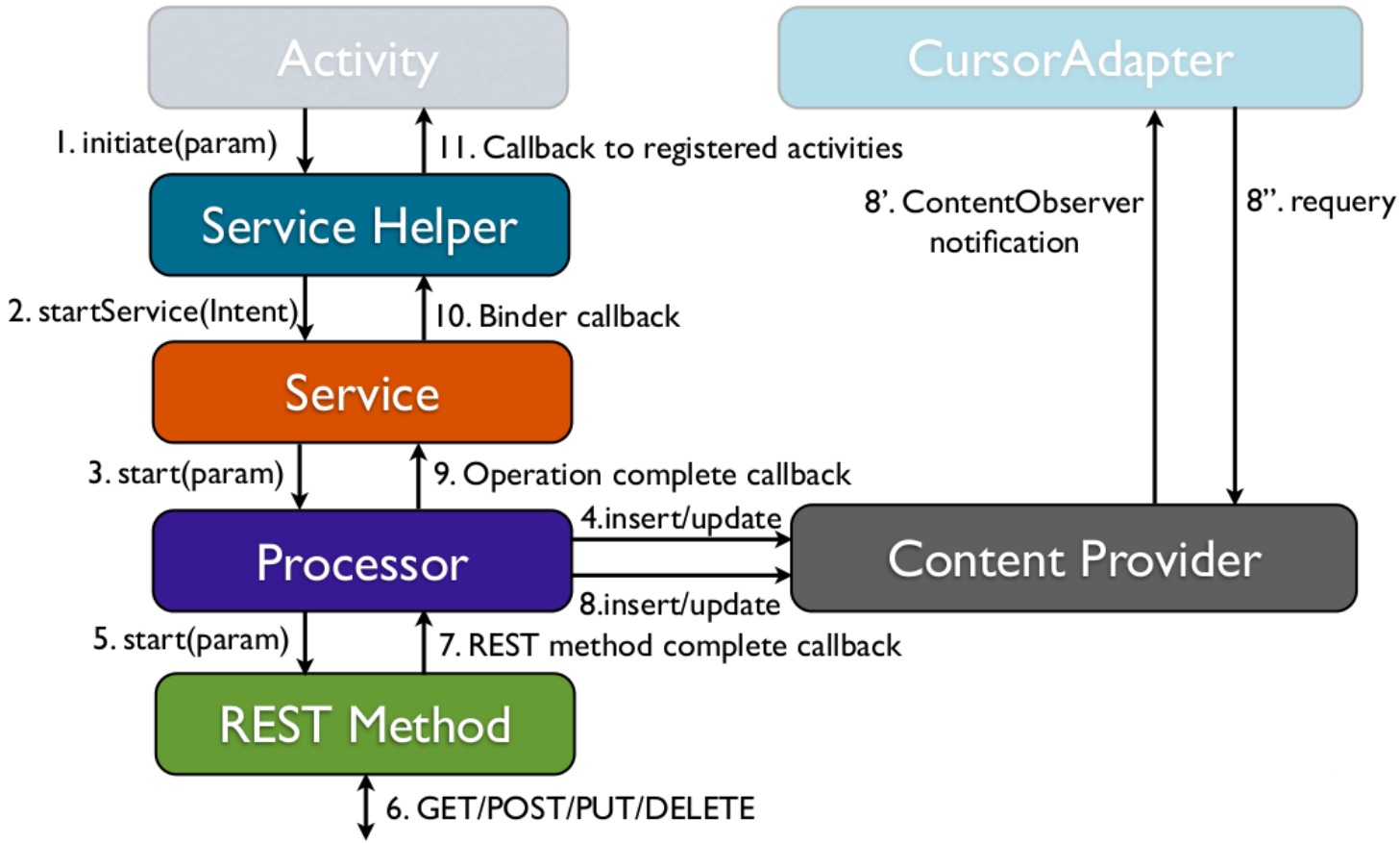
Implementing REST Methods using a Service API



The Service Helper

- Singleton which exposes a simple asynchronous API to be used by the user interface
- Prepare and send the Service request
  - Check if the method is already pending
  - Create the request Intent
  - Add the operation type and a unique request id
  - Add the method specific parameters
  - Add the binder callback
  - Call startService(Intent)
- Return the request id
- Handle the callback from the service
  - Dispatch callbacks to the user interface listeners

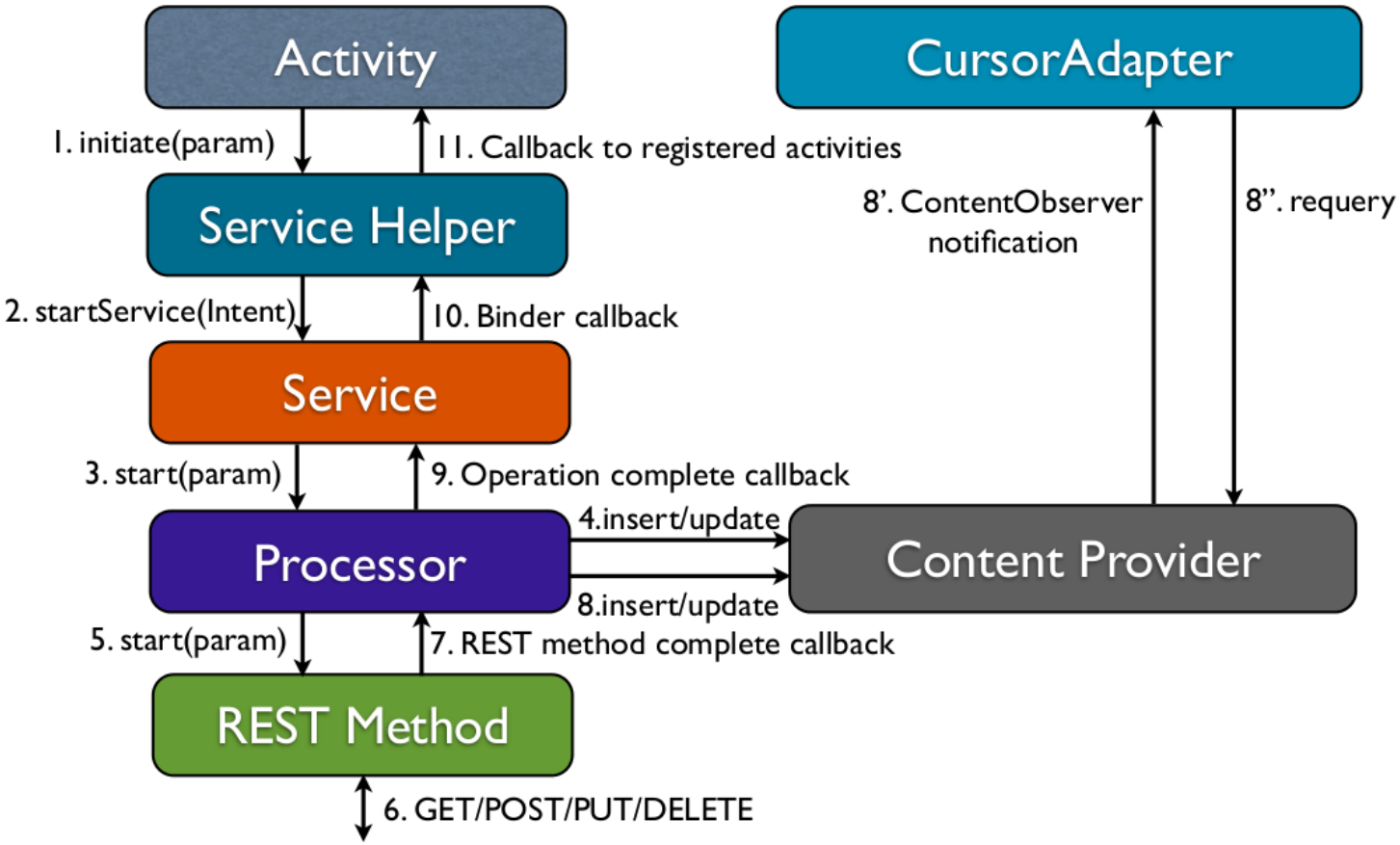
Implementing REST Methods using a Service API



Handling the REST Method in an Activity

- Add an operation listener in onResume and remove it in onPause
- Consider these cases:
  - The Activity is still active when the request completes
  - The Activity is paused then resumed and then the request completes
  - The Activity is paused when the request completes and then Activity is resumed
- The CursorAdapter handles the ContentProvider notification by implementing a ContentObserver

Implementing REST Methods using a Service API



Conclusions

"To have a great idea, have a lot of them."
&
"The value of an idea lies in the using of it."

- Thomas Alva Edison

Conclusions

- Do not implement REST methods inside Activities
- Start long running operations from a Service
- Persist early & persist often
- Minimize the network usage

Questions

Thanks for listenting. Questions?

POST questions HTTP/1.1
Host: jug.mk
User-Agent: JavaDay participant
Content-Length: reasonable
Content-Type: android-related/java