Tech Talk

android

Topic

- ★ Handler
- ★ Message
- ★ Message Queue
- ★ Looper

Handler

It can be invoked from both the producer and Consumer thread and, typically used for :

- Creating Message
- Inserting Message
- Processing Message on consumer thread
- Managing message in queue

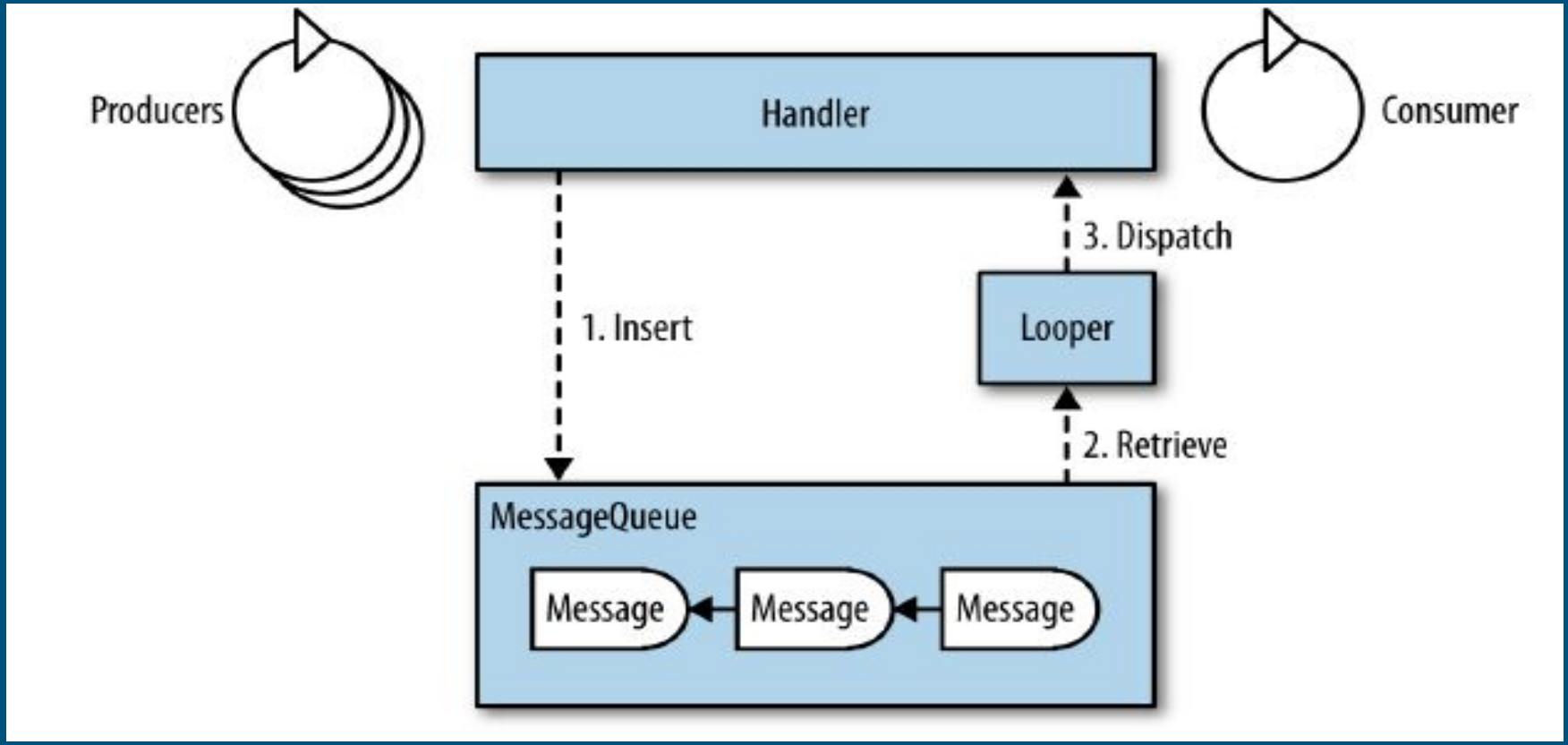


Fig. 1 Message passing Mechanism between multiple producer and one consumer thread

Message

- Each Item in MessageQueue is of android.os.Message class.
- This is a container object carrying either a data item or a task, never both.

Data Message

- Data message has multiple parameters that can be handed off to the consumer thread as shown:

Parameter name	Type	Usage
`what`	`int`	Message identifier. Communicates intention of the message.
`arg1, arg2`	`int`	Simple data values to handle the common use case of handing over integers. If a maximum of two integer values are to be passed to the consumer, these parameters are more efficient than allocating a `Bundle`, as explained under the `data` parameter.
`obj`	`Object`	Arbitrary object. If the object is handed off to a thread in another process, it has to implement `Parcelable`.
`data`	`Bundle`	Container of arbitrary data values.
`replyTo`	`Messenger`	Reference to `Handler` in some other process. Enables interprocess message communication, as described in "Two-Way Communication"
`callback`	`Runnable`	Task to execute on a thread. This is an internal instance field that holds the Runnable object from the `Handler.post` methods in "Handler"

Task Message

- The Task is represented by a java.lang.Runnable object to be executed on consumer thread, no data.
- The processing to be executed is defined in run() method of runnable, which is automatically executed without invoking Handler.handle(Message msg) method.

Message Queue

- A MQ can contain any combination of data and task message.
- Consumer Thread process them sequentially irrespective of their type.

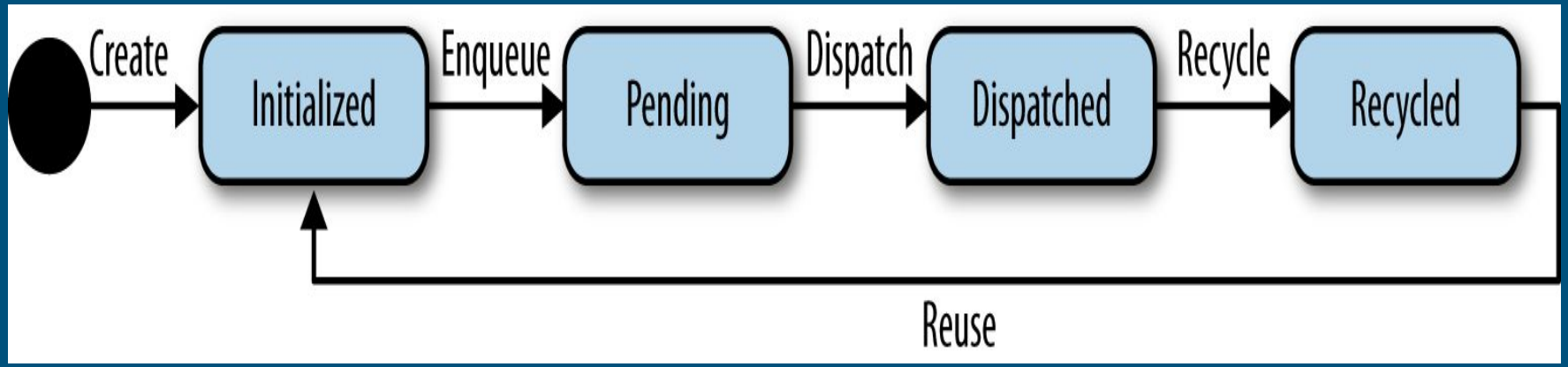
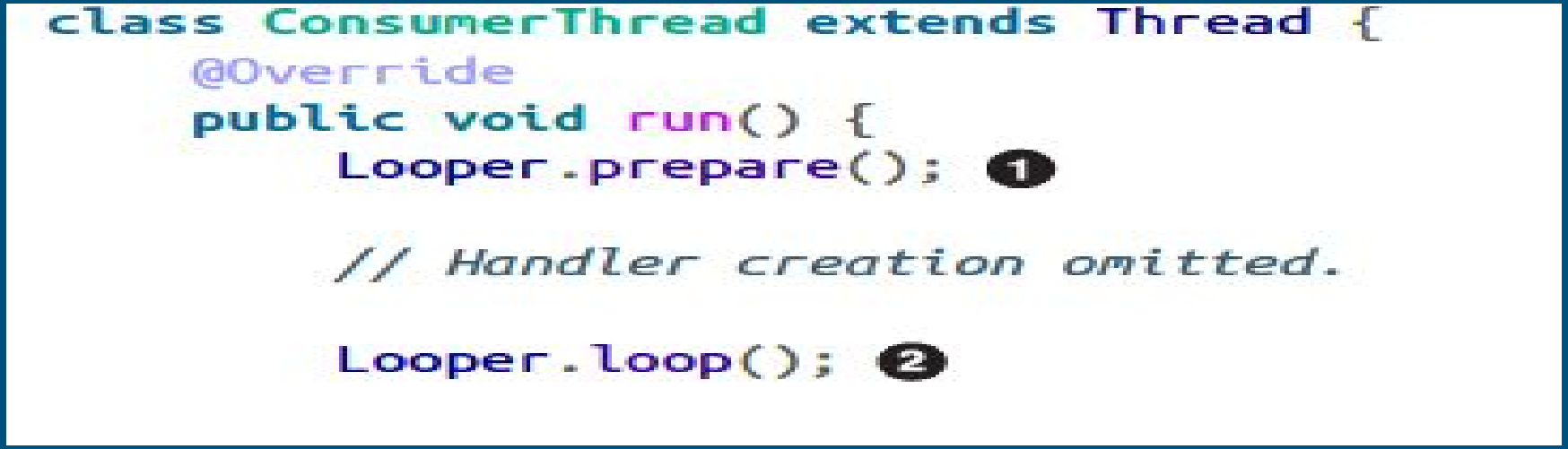


Fig. 2 Life cycle of message

Looper

- It manages the MQ and facilitates the dispatch of messages to consumer thread.
- Consumer thread does not directly interact with MQ, Instead a MQ is added to thread when a looper has been attached.
- By default only UI thread has looper.
- Thread created in application needs to gets a looper explicitly



1. The first step create the looper, i.e it create the MQ and associate it with thread.
2. This is a blocking method that ensures run() is not finished, while run() block Lopper dispatches the messages to consumer thread for processing.