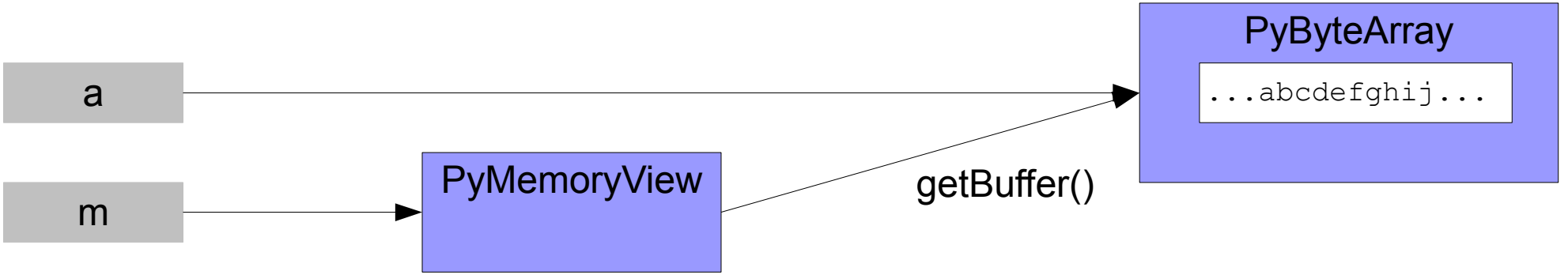
Jython Buffer Protocol

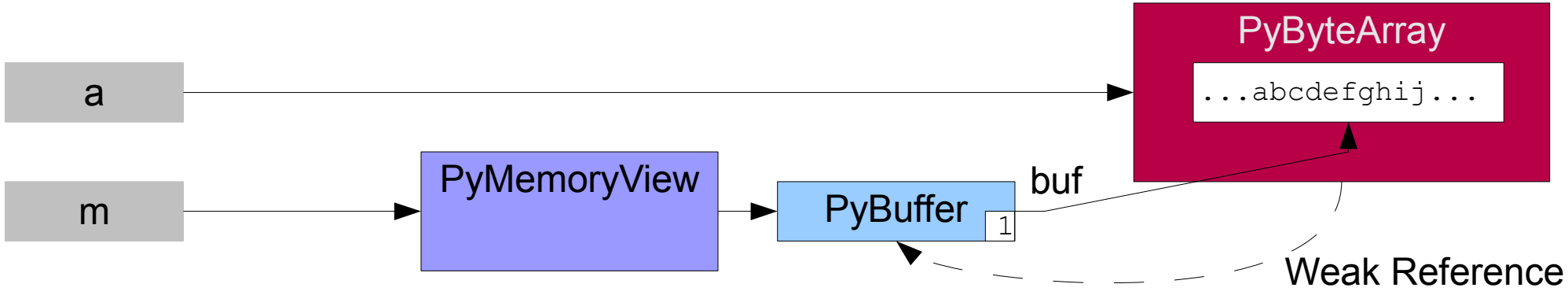
A complex example in simple pictures

```
a = bytearray(b'abcdefghij')
```

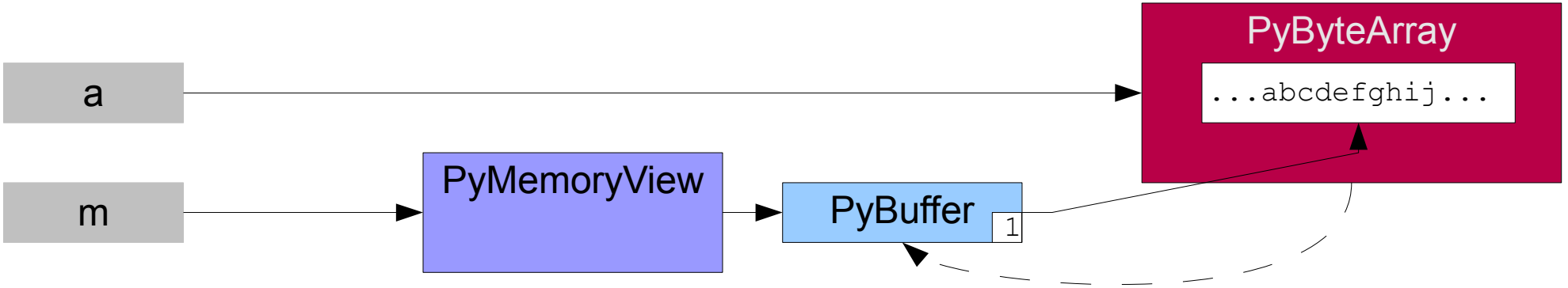
```
m = memoryview(a)
```



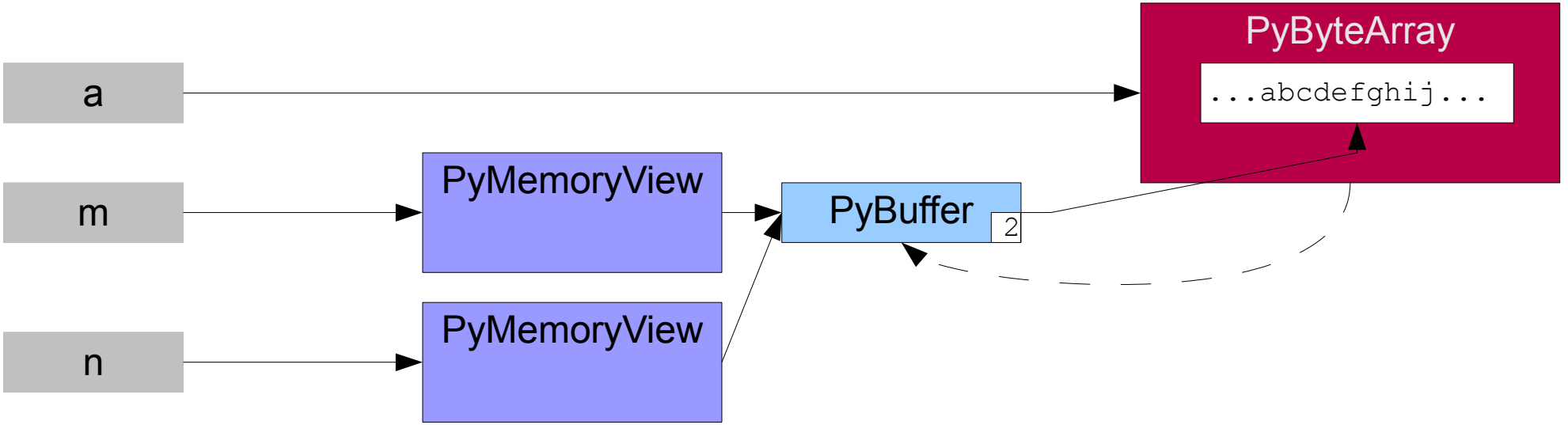
```
m = memoryview(a) # continued
```



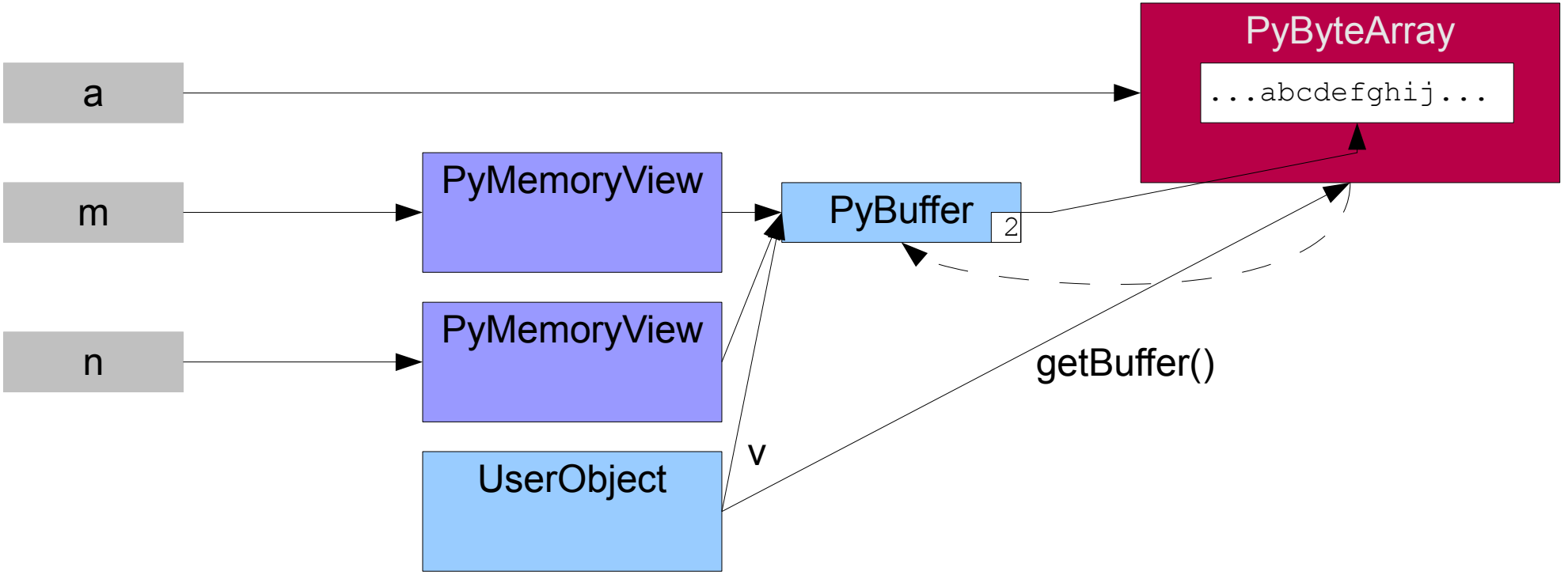
```
a.extend(0) # raises BufferError
```



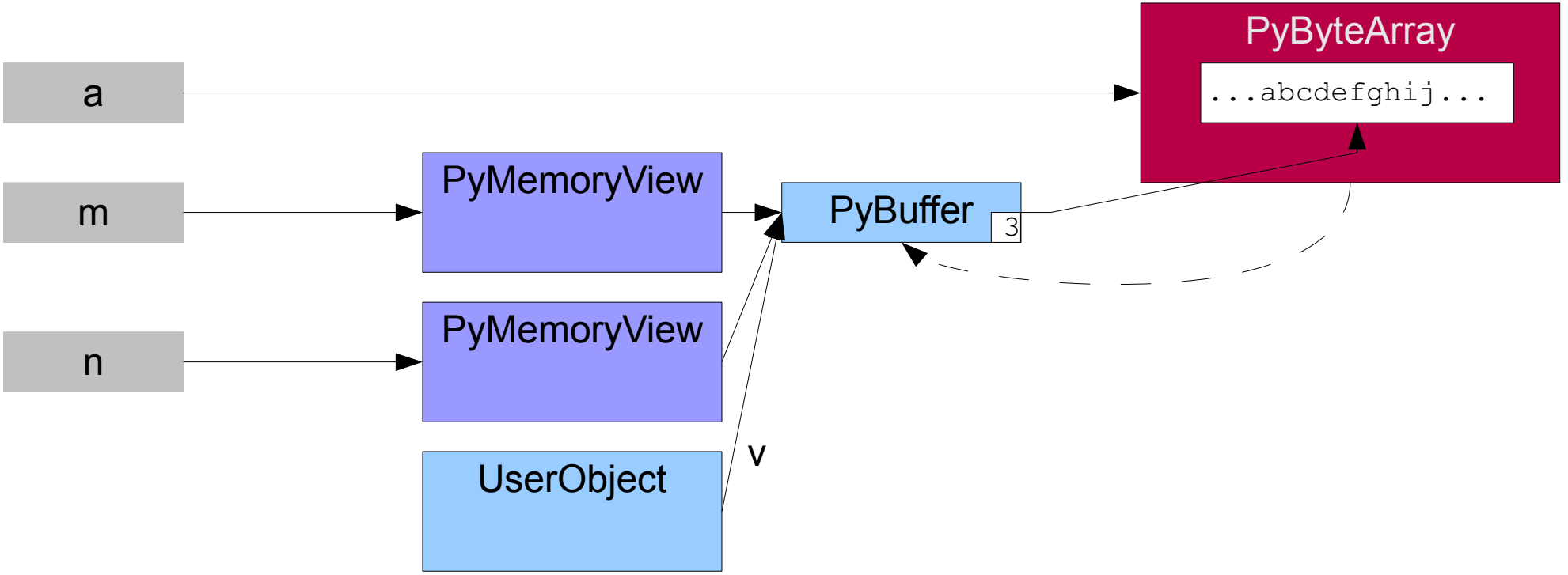
```
n = memoryview(m)
```



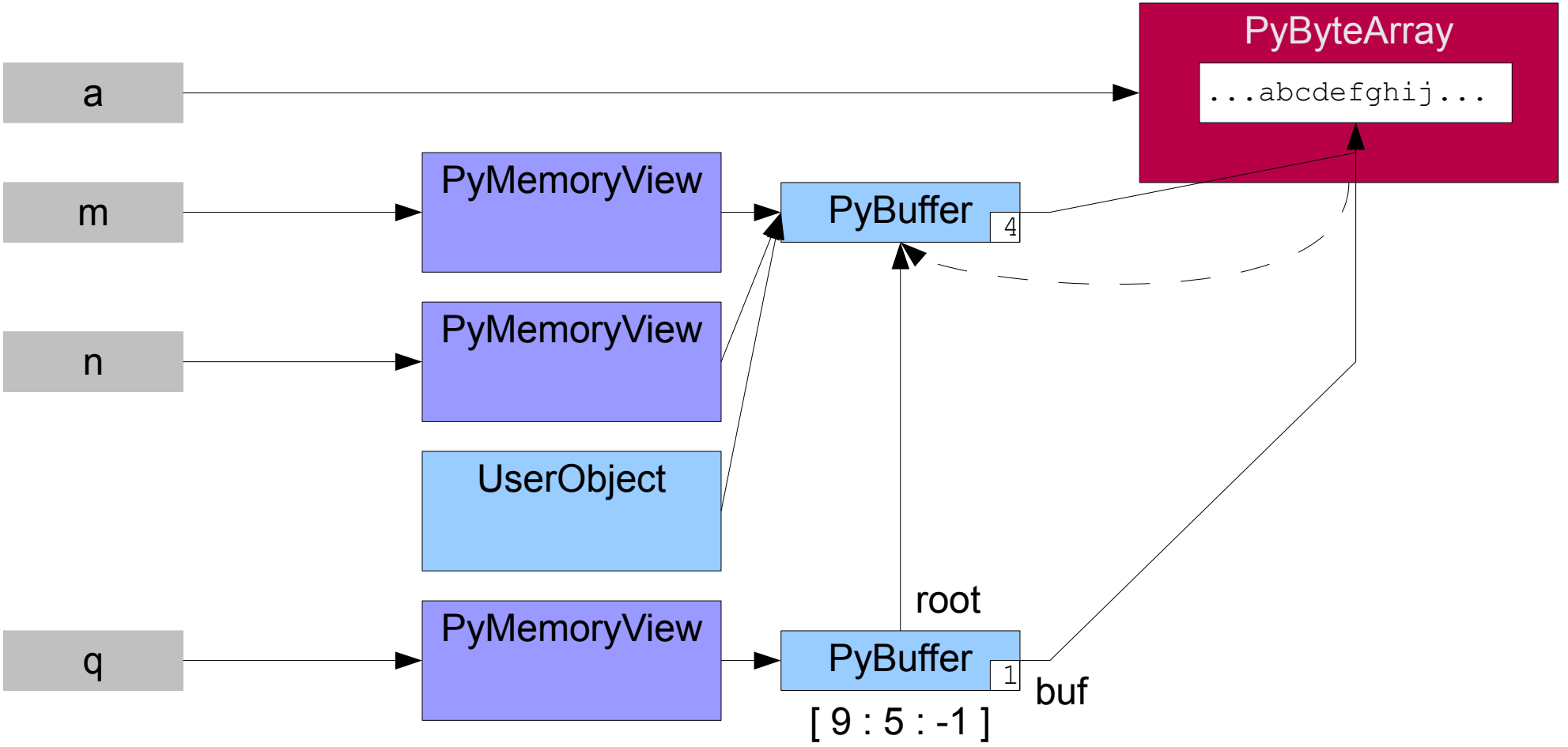
```
v = ((BufferProtocol)a).getBuffer(f)
```



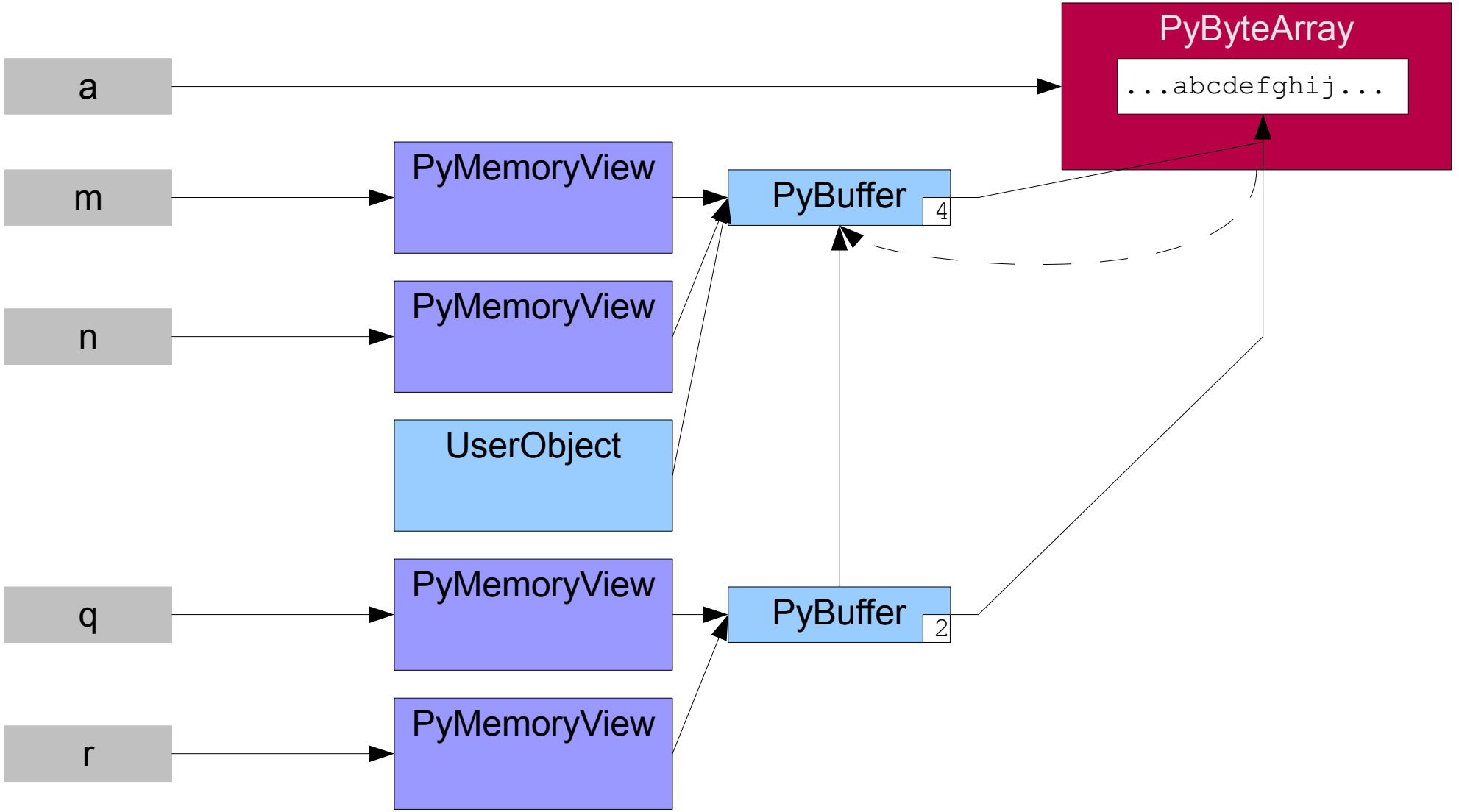
```
byte value = v.byteAt(i)
```



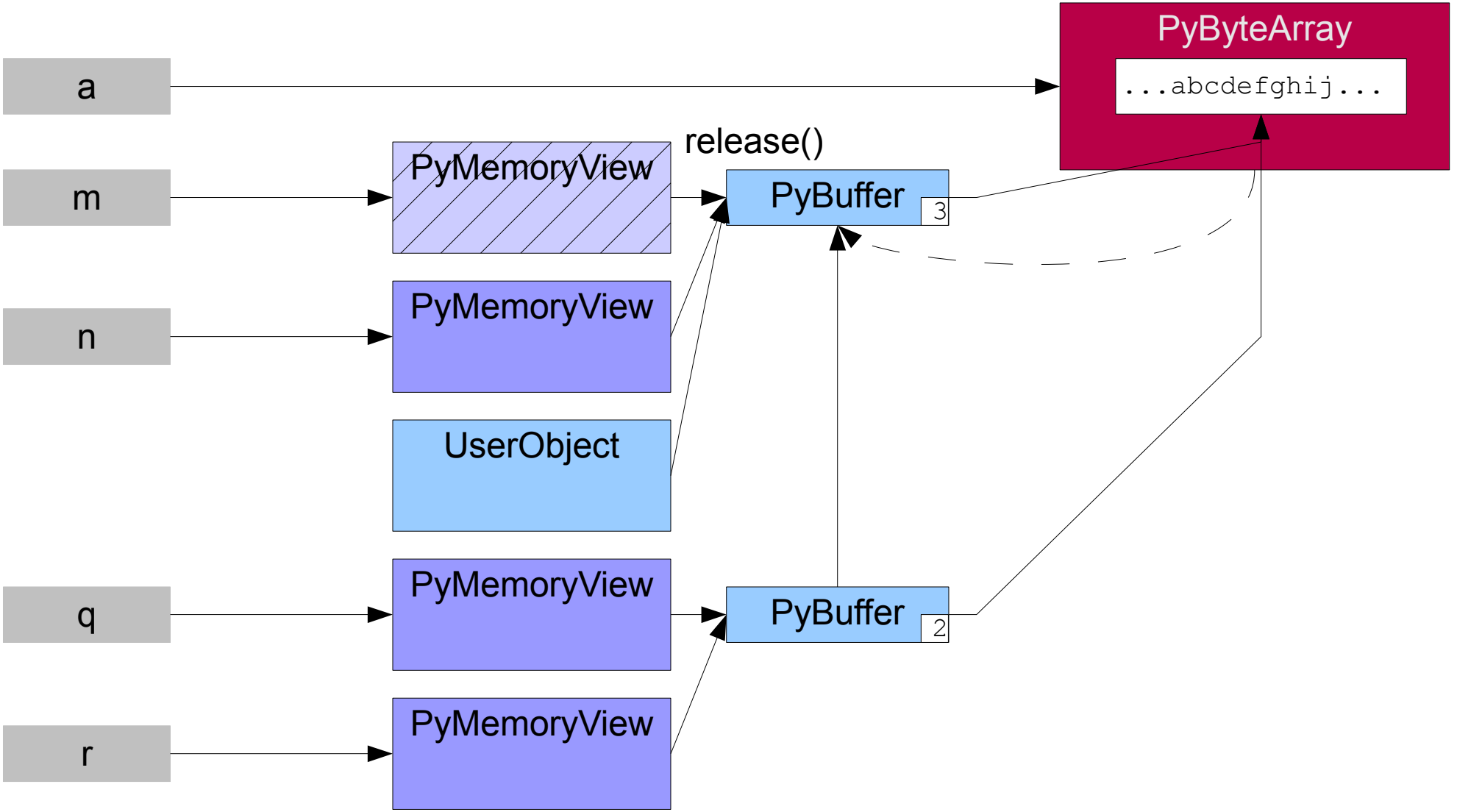
```
q = n[:5:-1] # = jihg
```


PyByteArray
...abcdefghij...
a
m
n
q
PyMemoryView
PyMemoryView
UserObject
PyMemoryView
PyBuffer
4
PyBuffer
1
root
buf
[9 : 5 : -1]

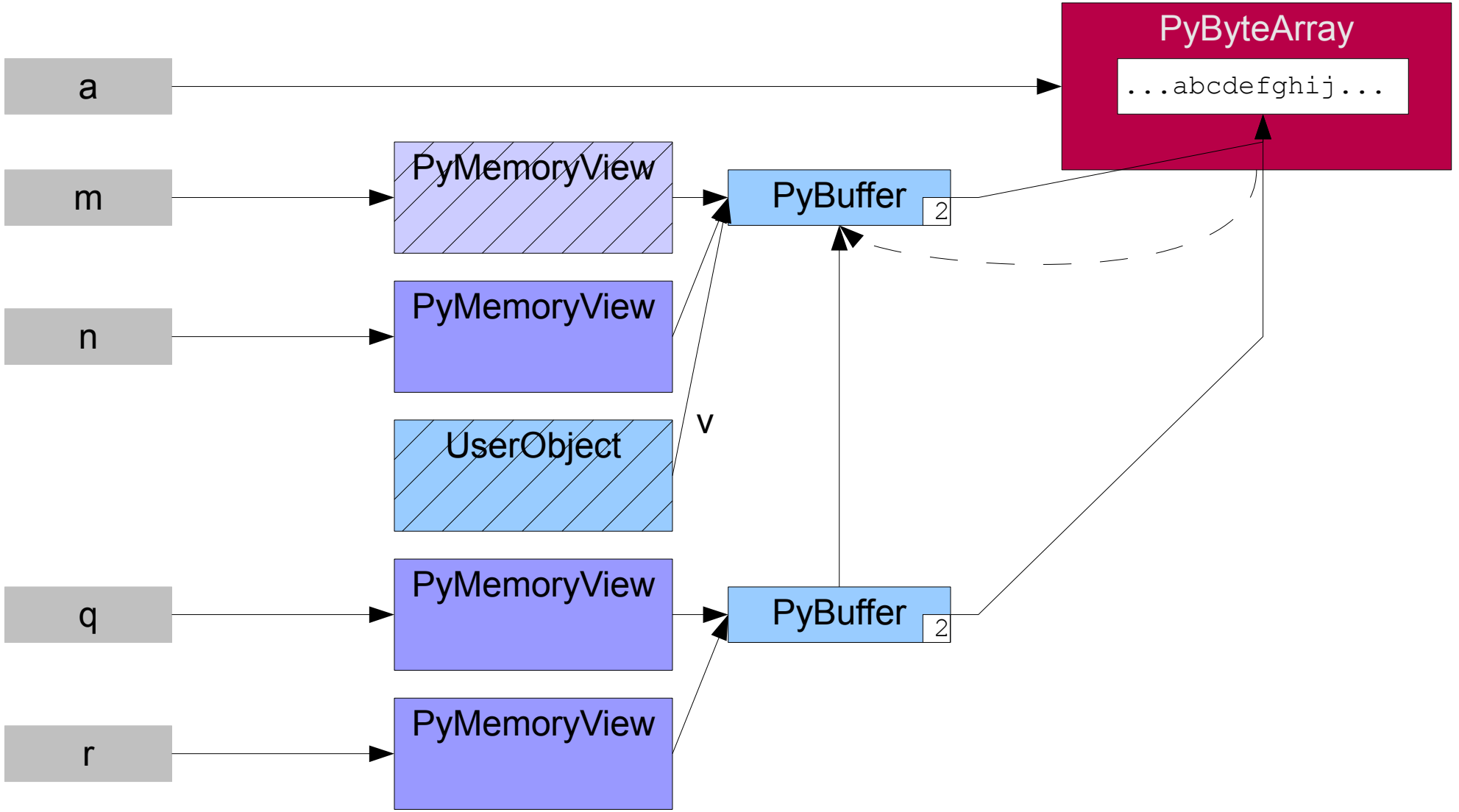
```
r = memoryview(q)
```



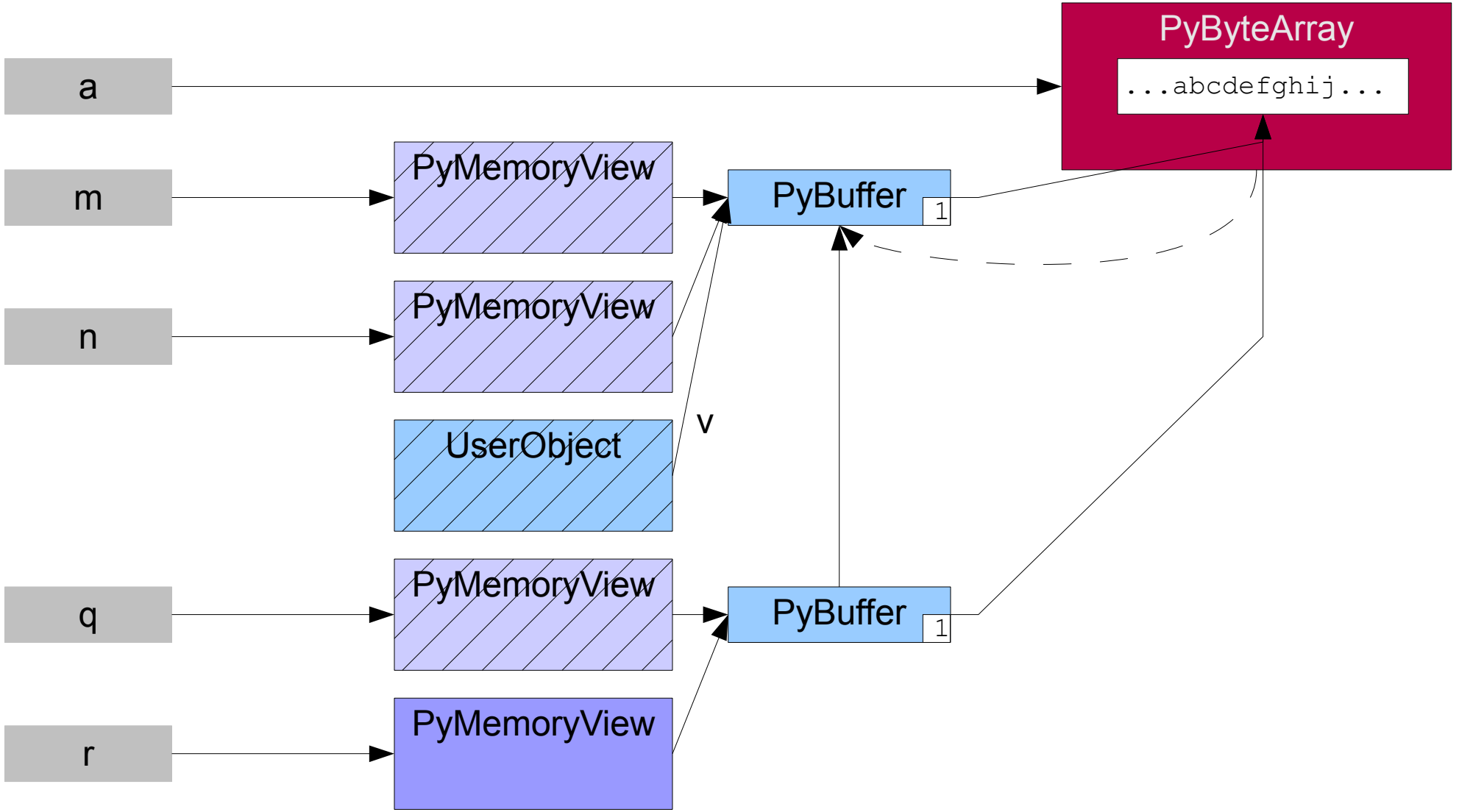
m.release()


PyByteArray
...abcdefghij...
a
m
n
q
r
PyMemoryView
PyMemoryView
UserObject
PyMemoryView
PyMemoryView
release()
PyBuffer
3
PyBuffer
2

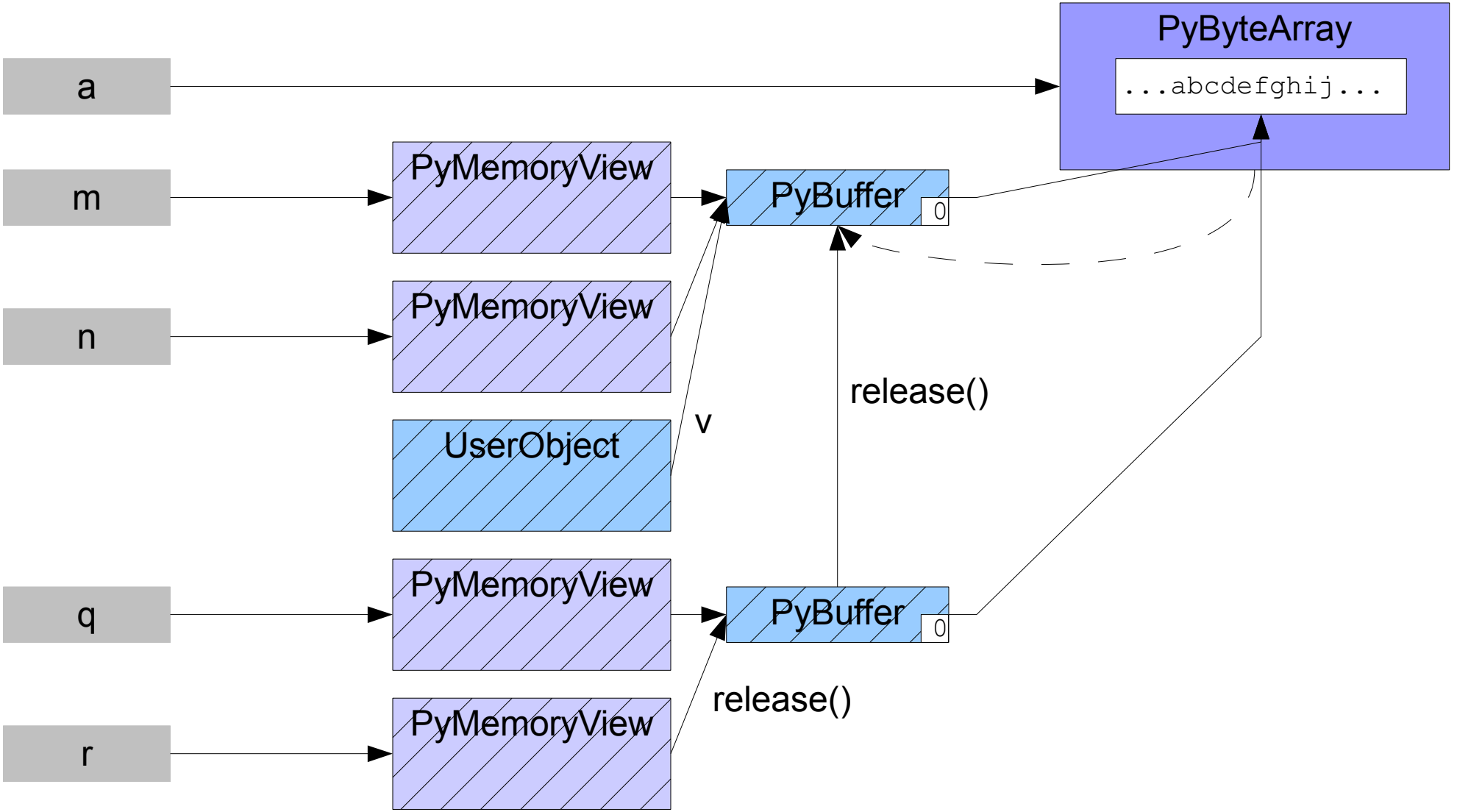
v.release()// in Java



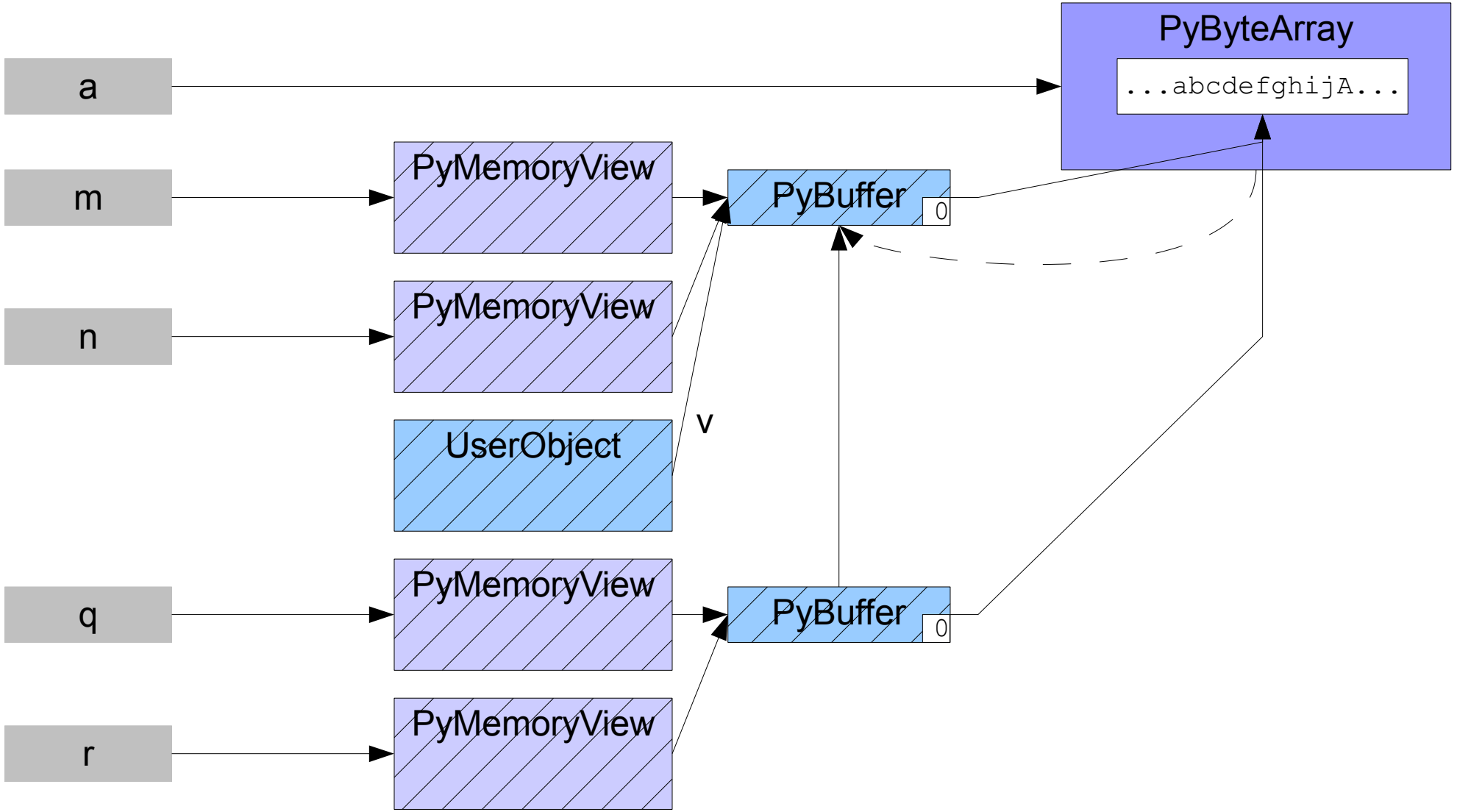
```
q.release()
n.release()
```



r.release()


PyByteArray
...abcdefghij...
a
m
n
q
r
PyMemoryView
PyMemoryView
UserObject
PyMemoryView
PyMemoryView
PyBuffer
0
PyBuffer
0
v
release()
release()

```
a.extend(ord('A')) # succeeds
```

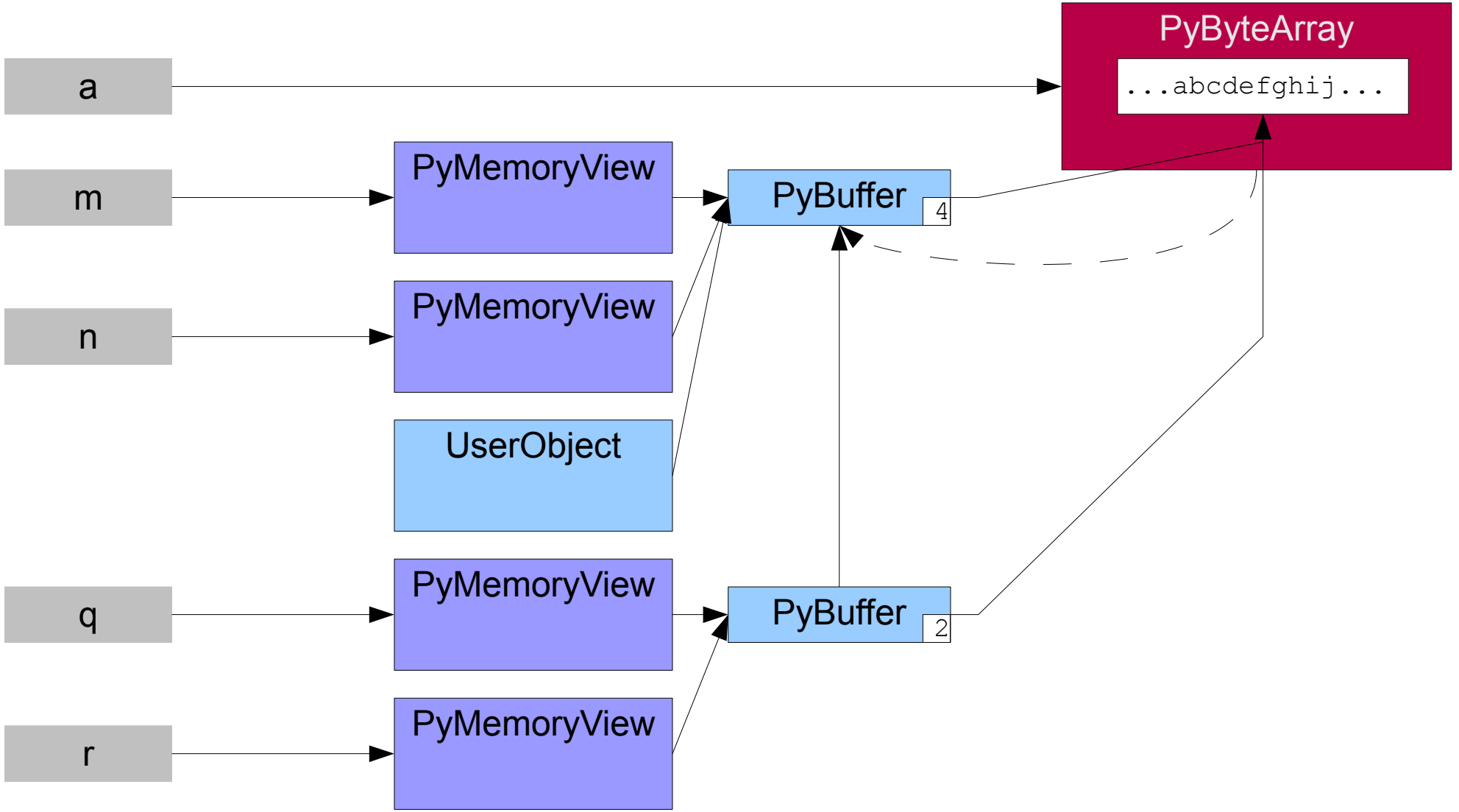


Java-style release()

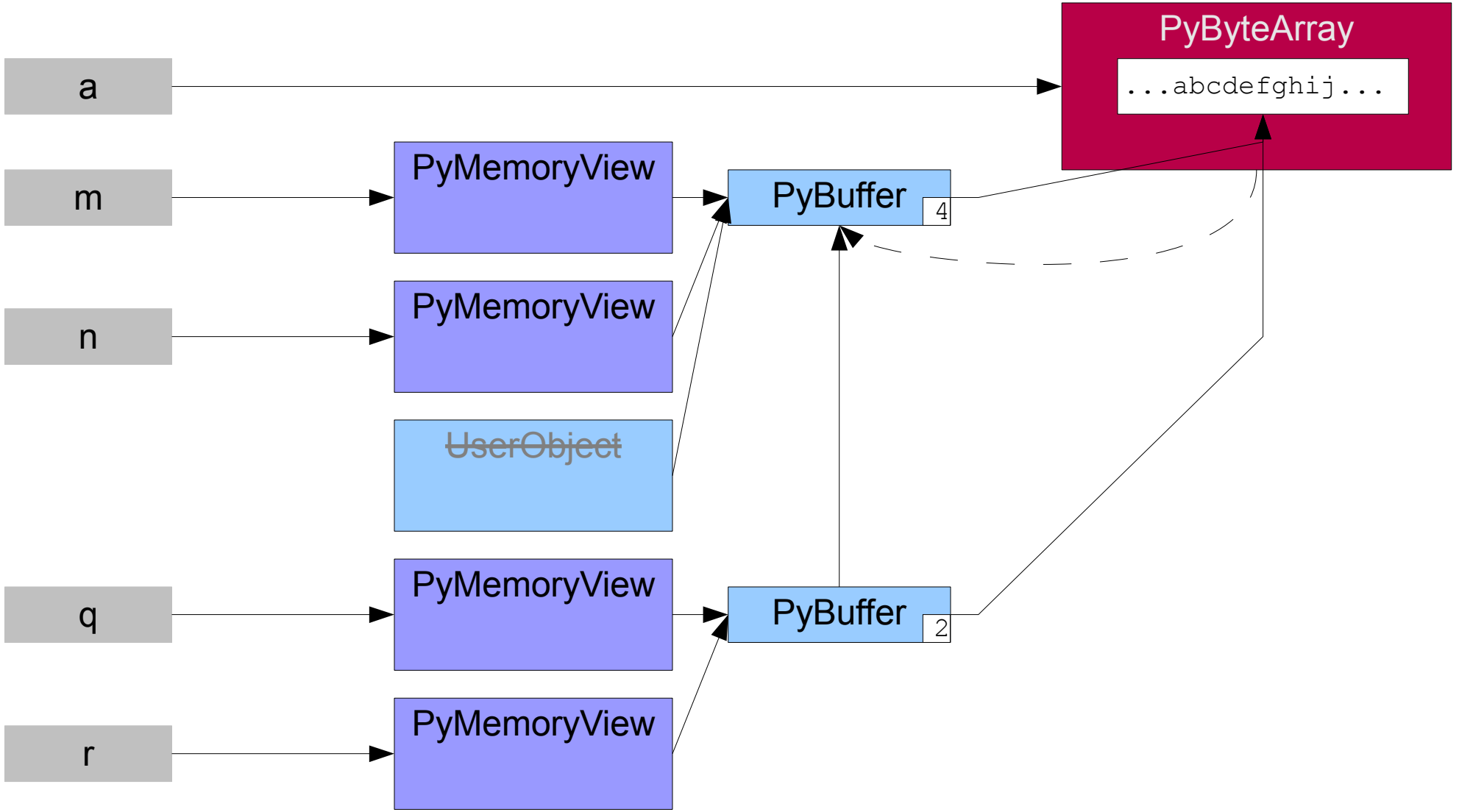
Some Python objects (bytearray) change their behaviour while exporting a buffer. To resume normal behaviour depends on explicit early release, or implicit release on object destruction.

Java programmers do not expect to manage object lifetimes explicity and are likely simply to drop references to a buffer. The Jython buffer protocol can recover from this treatment if the Java garbage collector runs.

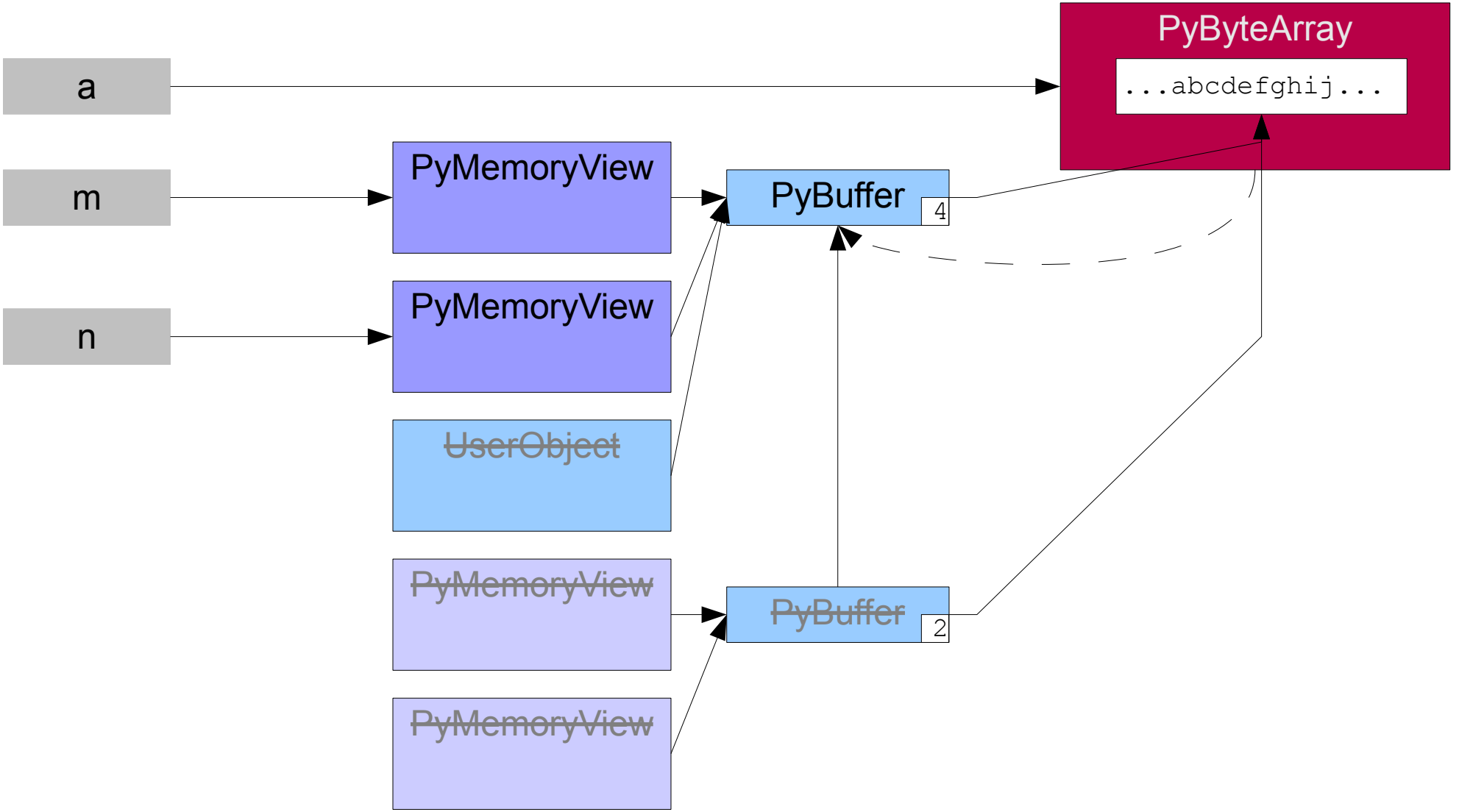
What if we simply dereference?



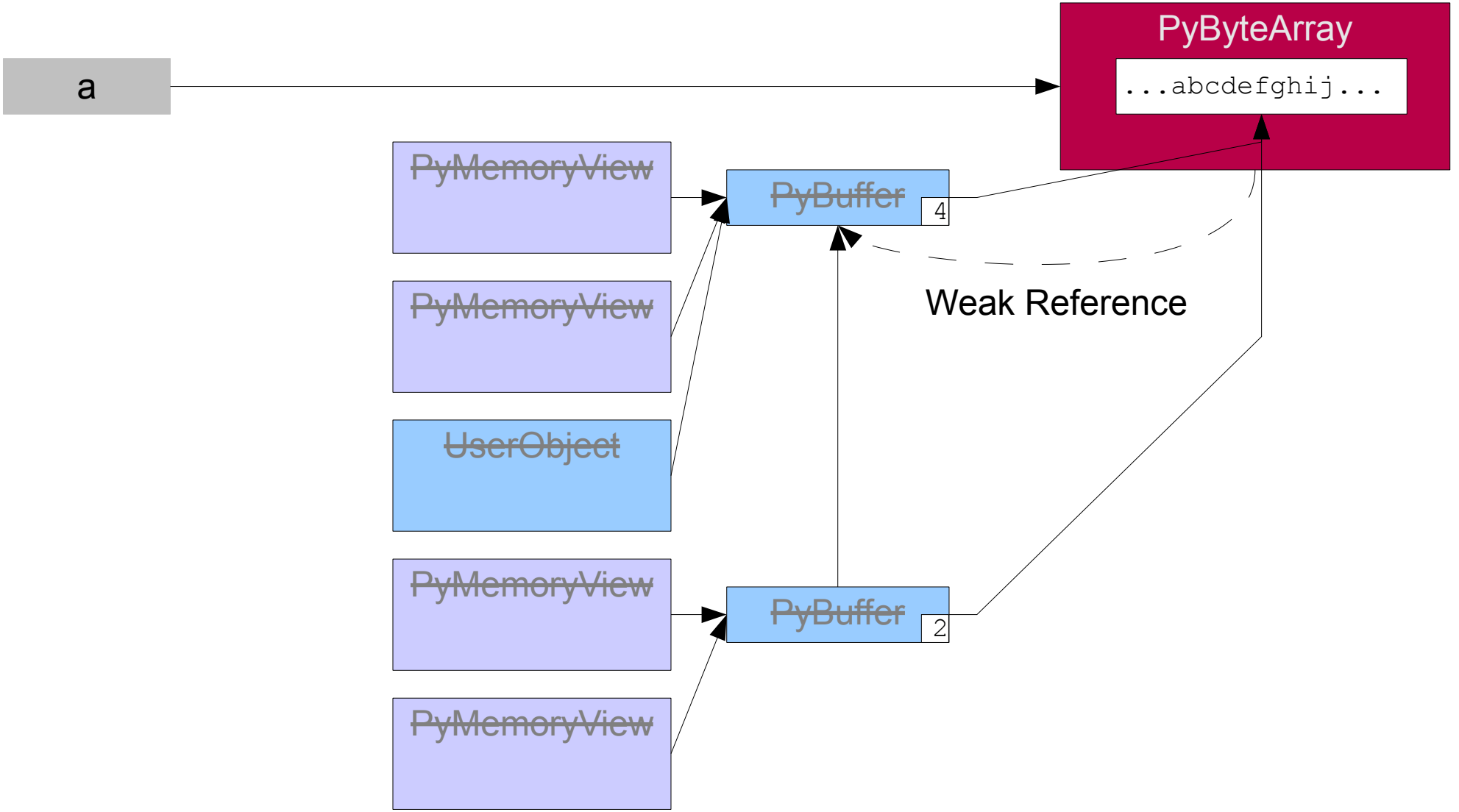
Java UserObject goes out of scope



```
del q, r
```


a
m
n
PyByteArray
...abcdefghij...
PyMemoryView
PyMemoryView
UserObject
PyMemoryView
PyMemoryView
PyBuffer
4
PyBuffer
2

```
del m, n
```


a
PyByteArray
...abcdefghij...
PyMemoryView
PyMemoryView
UserObject
PyMemoryView
PyMemoryView
PyBuffer
4
PyBuffer
2
Weak Reference

Java garbage collection runs

```
a.extend(ord('A')) # succeeds
```