Blink Rendering

Rebuilding the Engine Mid-Flight

What is rendering？

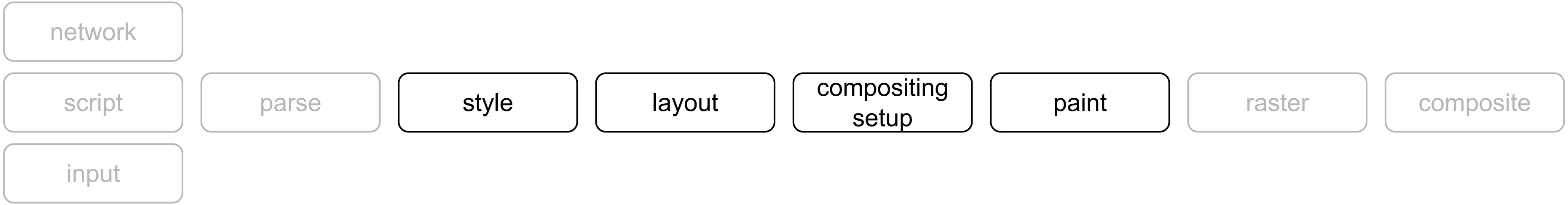
Turning DOM into pixels, 60 times per second.

blink/renderer/core/layout
blink/renderer/core/paint
blink/renderer/core/style

blink/renderer/platform/graphics
blink/renderer/platform/text

blink/renderer/core/frame
blink/renderer/core/page
...

How Rendering Happens



network

script

input

parse

style

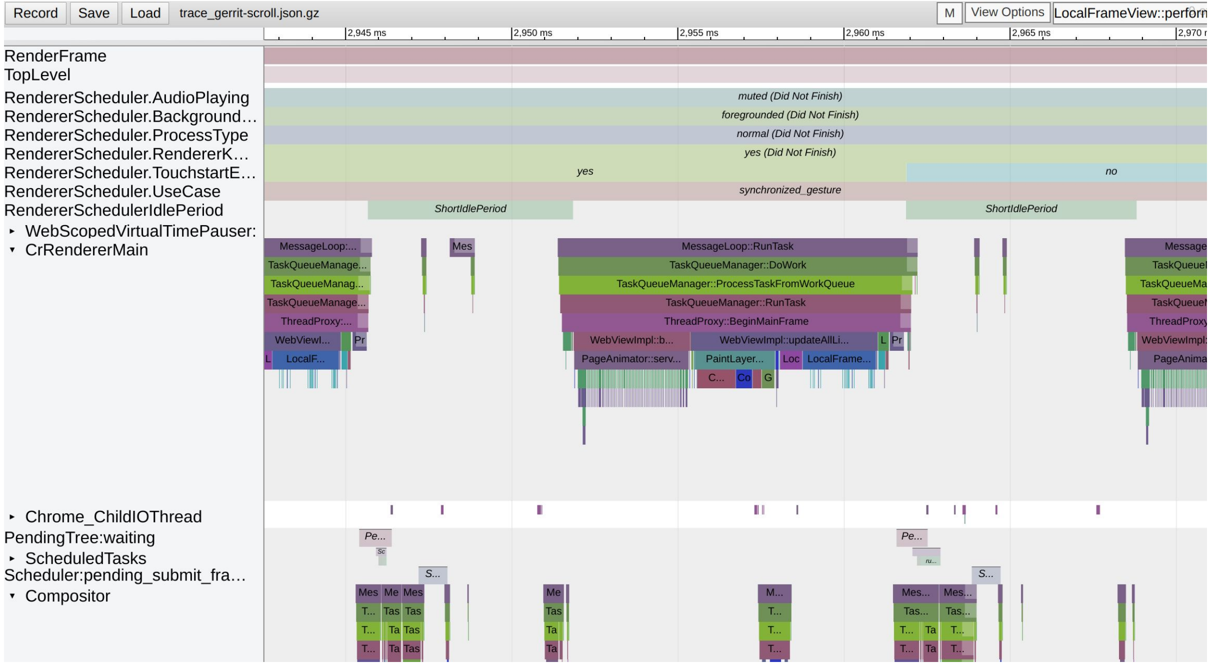
layout

compositing setup

paint

raster

composite


Record
Save
Load
trace_gerrit-scroll.json.gz
M
View Options
LocalFrameView::perforn
2,945 ms
2,950 ms
2,955 ms
2,960 ms
2,965 ms
2,970 r
RenderFrame
TopLevel
RendererScheduler.AudioPlaying
RendererScheduler.Background…
RendererScheduler.ProcessType
RendererScheduler.RendererK…
RendererScheduler.TouchstartE…
RendererScheduler.UseCase
RendererSchedulerIdlePeriod
WebScopedVirtualTimePauser:
CrRendererMain
Chrome_ChildIOThread
PendingTree:waiting
ScheduledTasks
Scheduler:pending_submit_fra…
Compositor
muted (Did Not Finish)
foregrounded (Did Not Finish)
normal (Did Not Finish)
yes (Did Not Finish)
yes
no
synchronized_gesture
ShortIdlePeriod
ShortIdlePeriod
MessageLoop::RunTask
TaskQueueManager::DoWork
TaskQueueManager::ProcessTaskFromWorkQueue
TaskQueueManager::RunTask
ThreadProxy::BeginMainFrame
WebViewImpl::b…
WebViewImpl::updateAllLi…
PageAnimator::serv…
PaintLayer…
Loc
LocalFrame…
C…
Co
G

network

script

parse

style

layout

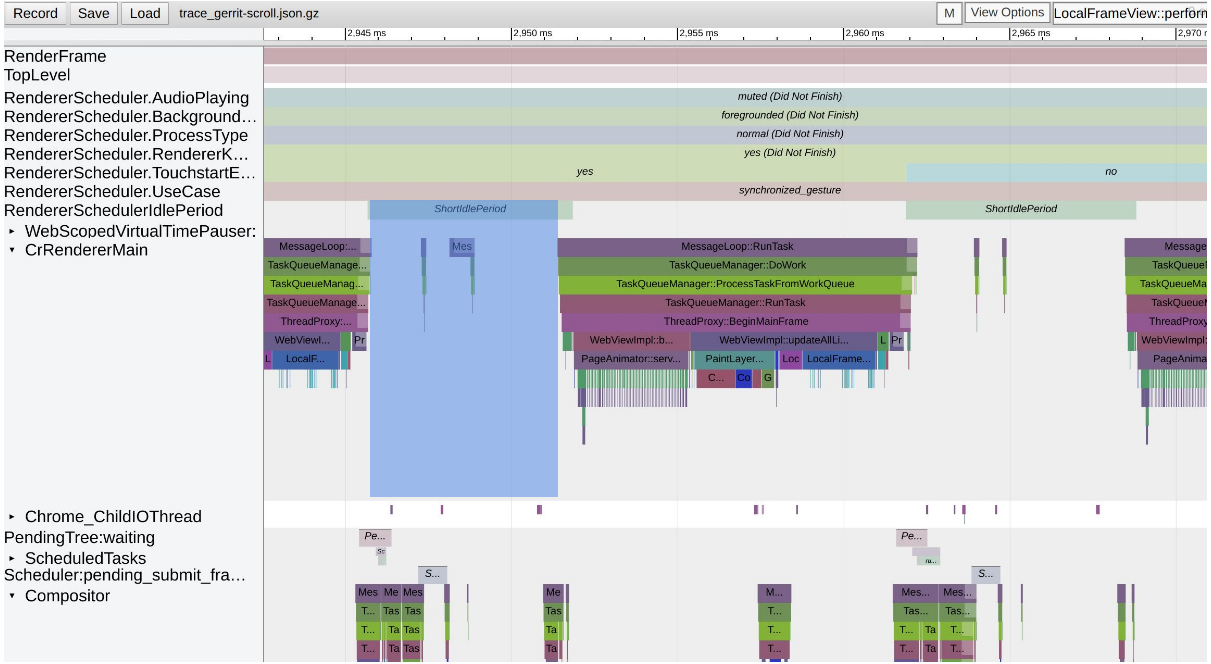
compositing setup

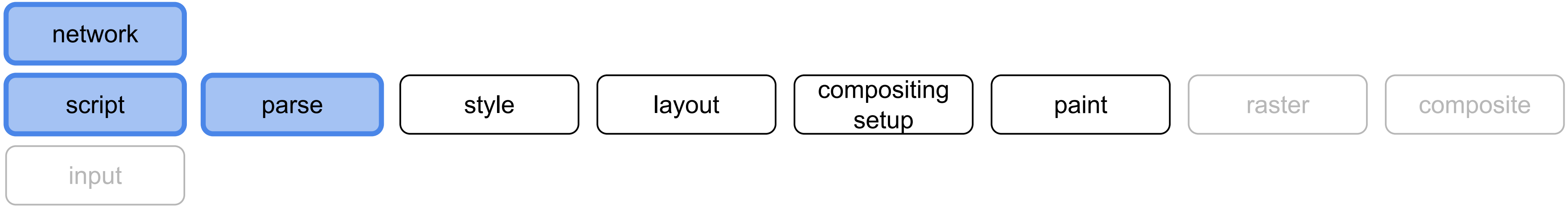
paint

raster

composite

input




network

script

parse

style

layout

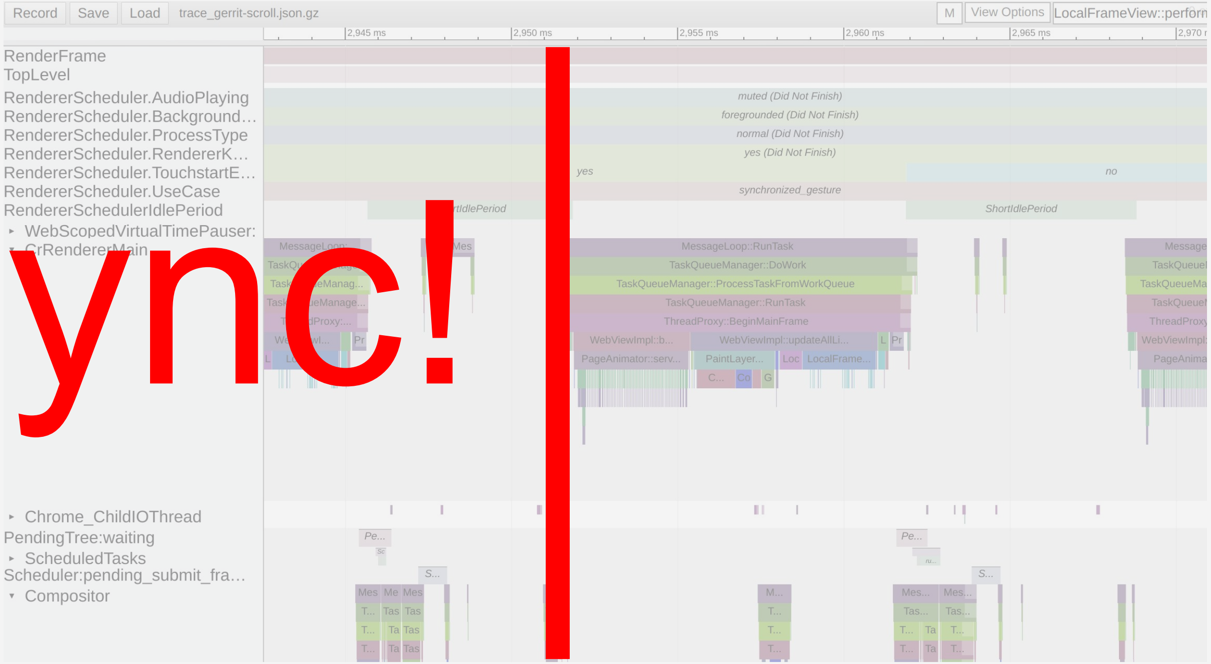
compositing setup

paint

raster

composite

input


Record
Save
Load
trace_gerrit-scroll.json.gz
M
View Options
LocalFrameView::perform
2,945 ms
2,950 ms
2,955 ms
2,960 ms
2,965 ms
2,970 ms
RenderFrame
TopLevel
RendererScheduler.AudioPlaying
RendererScheduler.Background…
RendererScheduler.ProcessType
RendererScheduler.RendererK…
RendererScheduler.TouchstartE…
RendererScheduler.UseCase
RendererSchedulerIdlePeriod
WebScopedVirtualTimePauser:
CrRendererMain
muted (Did Not Finish)
foregrounded (Did Not Finish)
normal (Did Not Finish)
yes (Did Not Finish)
yes
no
synchronized_gesture
ShortIdlePeriod
MessageLoop::RunTask
TaskQueueManager::DoWork
TaskQueueManager::ProcessTaskFromWorkQueue
TaskQueueManager::RunTask
ThreadProxy::BeginMainFrame
WebViewImpl::b...
WebViewImpl::updateAllLi...
PageAnimator::serv...
PaintLayer...
LocalFrame...
Chrome_ChildIOThread
PendingTree:waiting
ScheduledTasks
Scheduler:pending_submit_fra…
Compositor

vsync!

network

script

parse

style

layout

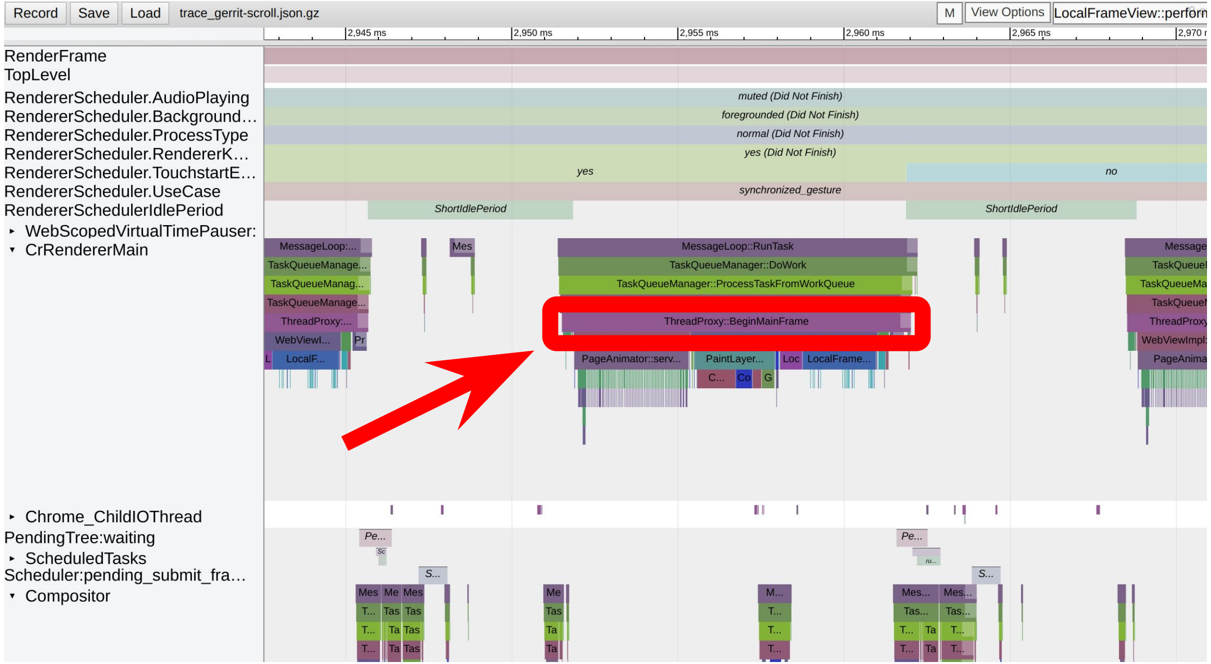
compositing setup

paint

raster

composite

input



network

script

parse

style

layout

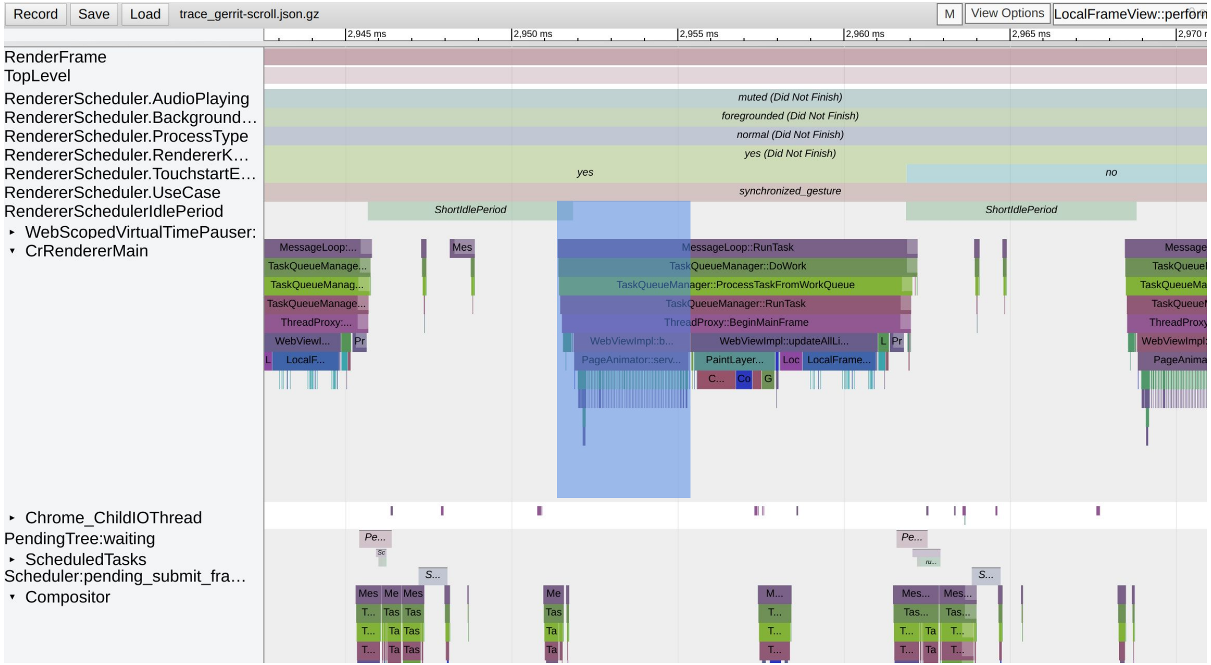
compositing setup

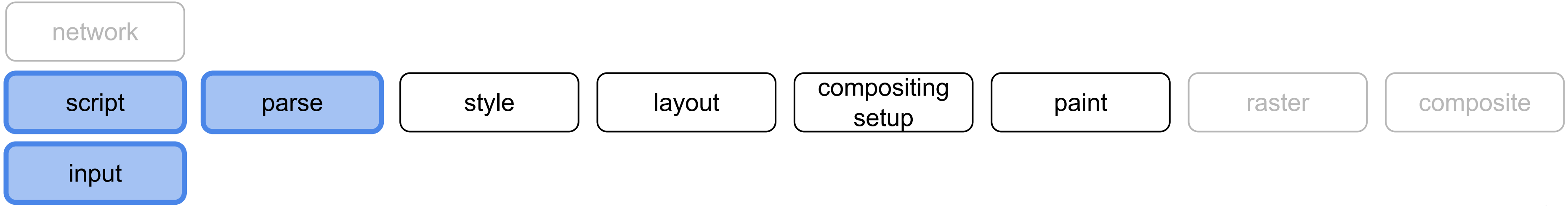
paint

raster

composite

input





network

script

parse

style

layout

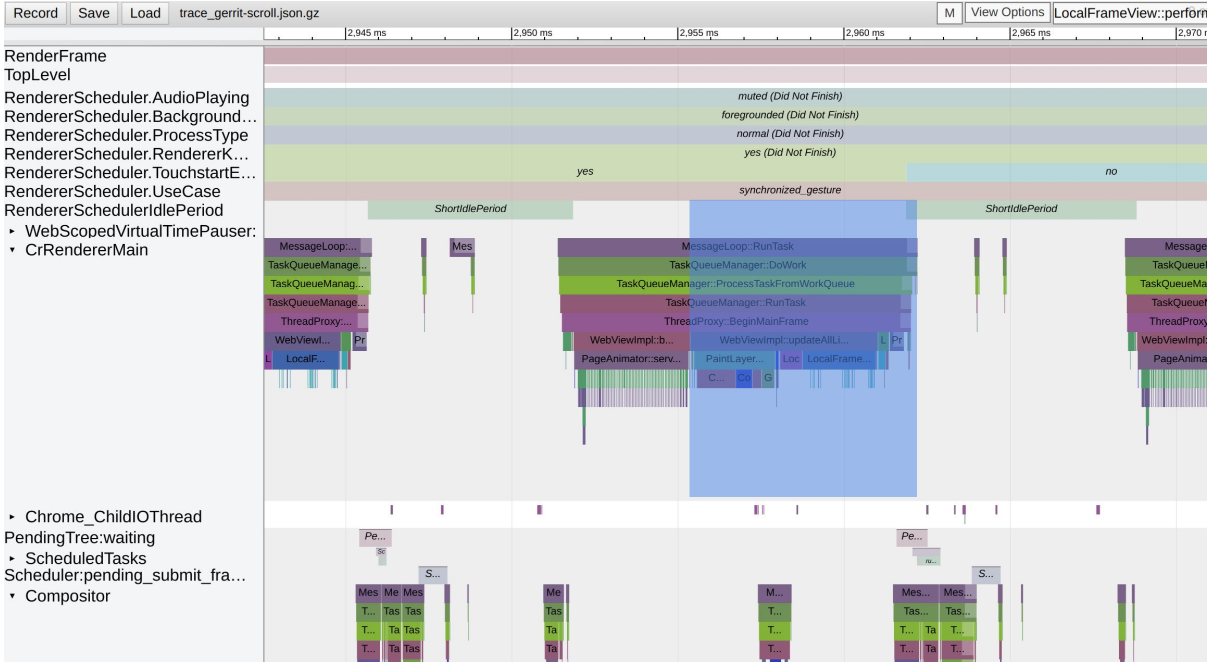
compositing setup

paint

raster

composite

input


Record
Save
Load
trace_gerrit-scroll.json.gz
M
View Options
LocalFrameView::perforn
2,945 ms
2,950 ms
2,955 ms
2,960 ms
2,965 ms
2,970 r
RenderFrame
TopLevel
RendererScheduler.AudioPlaying
RendererScheduler.Background…
RendererScheduler.ProcessType
RendererScheduler.RendererK…
RendererScheduler.TouchstartE…
RendererScheduler.UseCase
RendererSchedulerIdlePeriod
WebScopedVirtualTimePauser:
CrRendererMain
Chrome_ChildIOThread
PendingTree:waiting
ScheduledTasks
Scheduler:pending_submit_fra…
Compositor
muted (Did Not Finish)
foregrounded (Did Not Finish)
normal (Did Not Finish)
yes (Did Not Finish)
yes
no
synchronized_gesture
ShortIdlePeriod
ShortIdlePeriod
MessageLoop::RunTask
TaskQueueManager::DoWork
TaskQueueManager::ProcessTaskFromWorkQueue
TaskQueueManager::RunTask
ThreadProxy::BeginMainFrame
WebViewImpl::b...
WebViewImpl::updateAllLi...
PageAnimator::serv...
PaintLayer...
LocalFrame...

network

script

parse

style

layout

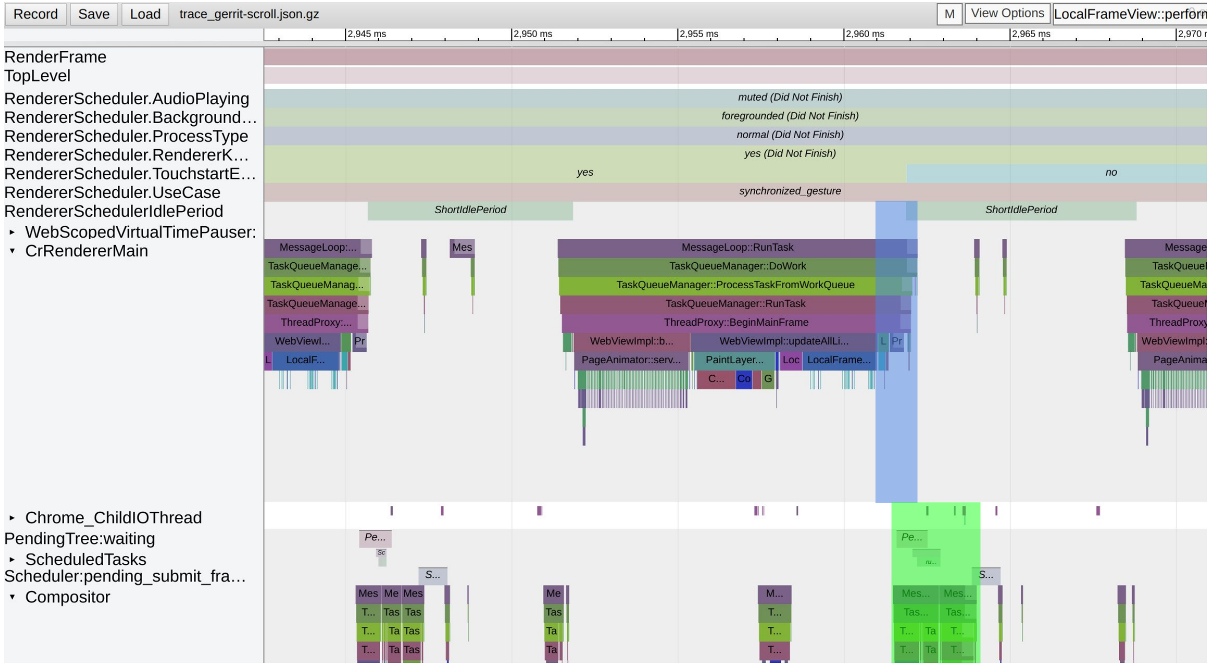
compositing setup

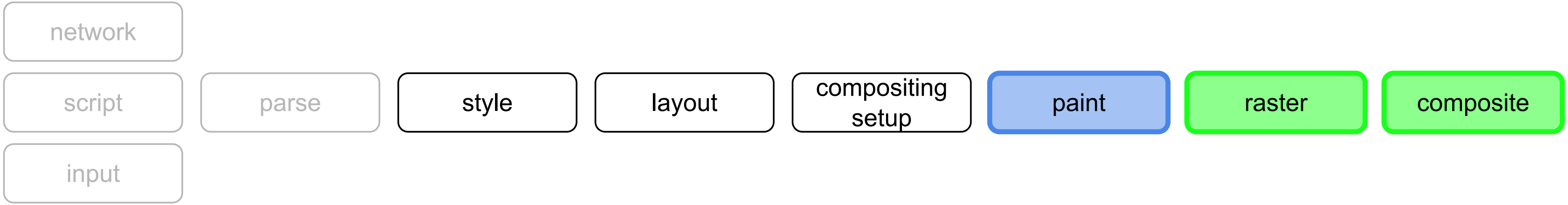
paint

raster

composite

input





network

script

parse

style

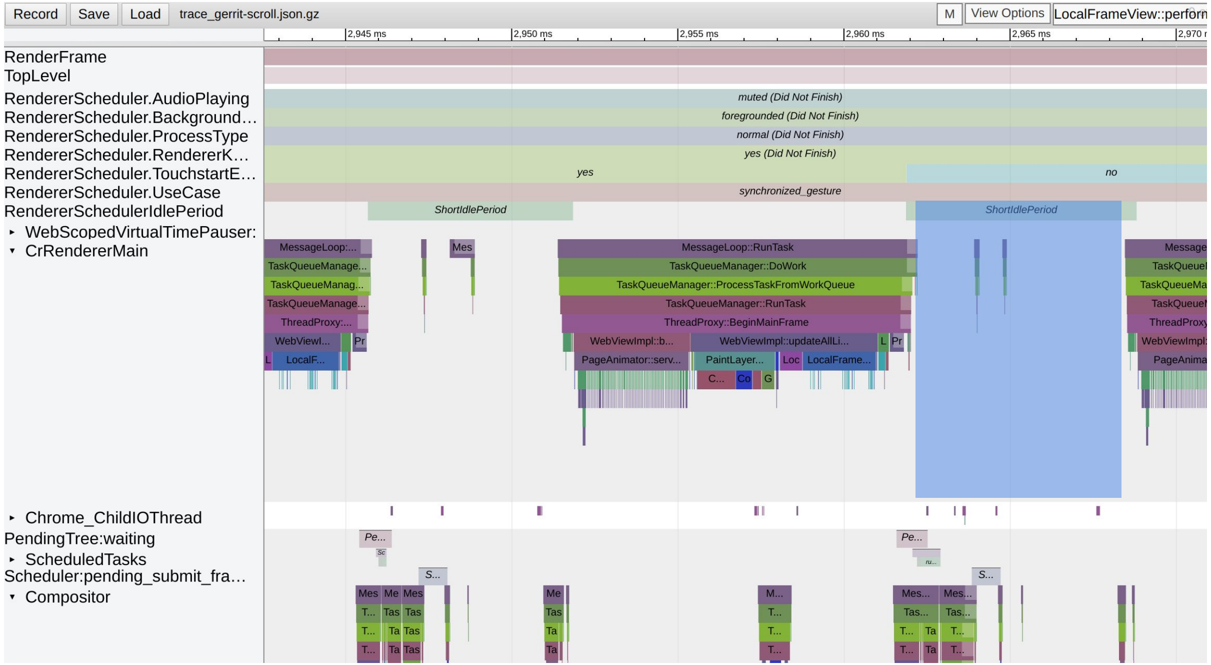
layout

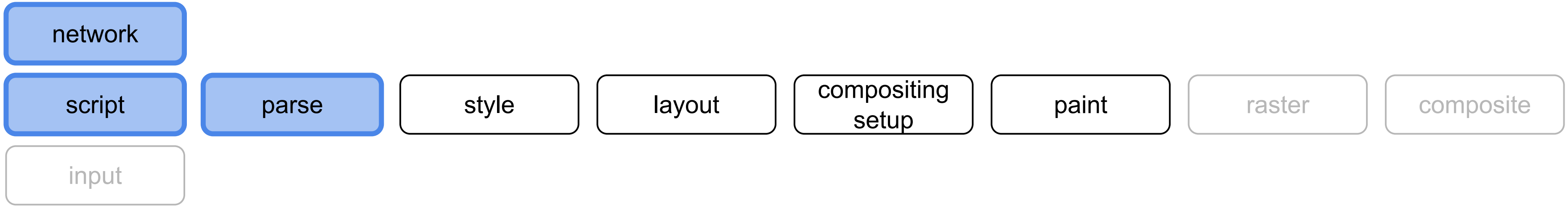
compositing setup

paint

raster

composite


Record
Save
Load
trace_gerrit-scroll.json.gz
M
View Options
LocalFrameView::perforn
2,945 ms
2,950 ms
2,955 ms
2,960 ms
2,965 ms
2,970 r
RenderFrame
TopLevel
RendererScheduler.AudioPlaying
RendererScheduler.Background…
RendererScheduler.ProcessType
RendererScheduler.RendererK…
RendererScheduler.TouchstartE…
RendererScheduler.UseCase
RendererSchedulerIdlePeriod
WebScopedVirtualTimePauser:
CrRendererMain
muted (Did Not Finish)
foregrounded (Did Not Finish)
normal (Did Not Finish)
yes (Did Not Finish)
yes
no
synchronized_gesture
ShortIdlePeriod
ShortIdlePeriod
MessageLoop::RunTask
TaskQueueManager::DoWork
TaskQueueManager::ProcessTaskFromWorkQueue
TaskQueueManager::RunTask
ThreadProxy::BeginMainFrame
WebViewImpl::b...
WebViewImpl::updateAllLi...
PageAnimator::serv...
PaintLayer...
LocalFrame...
Chrome_ChildIOThread
PendingTree:waiting
ScheduledTasks
Scheduler:pending_submit_fra…
Compositor



network

script

parse

style

layout

compositing setup

paint

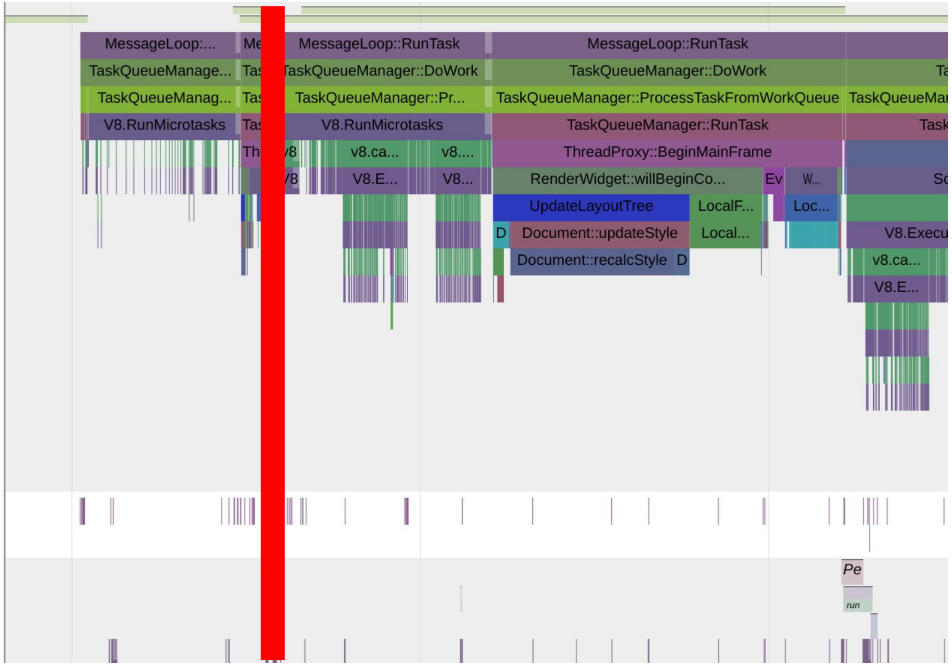
raster

composite

input

Rendering is Important, Rendering is Hard

- Rendered output is the foremost component of user experience.
- Excluding javascript, rendering is the biggest determinant of performance.
- Modern web pages feature lots of dynamic content.



network

script

parse

style

layout

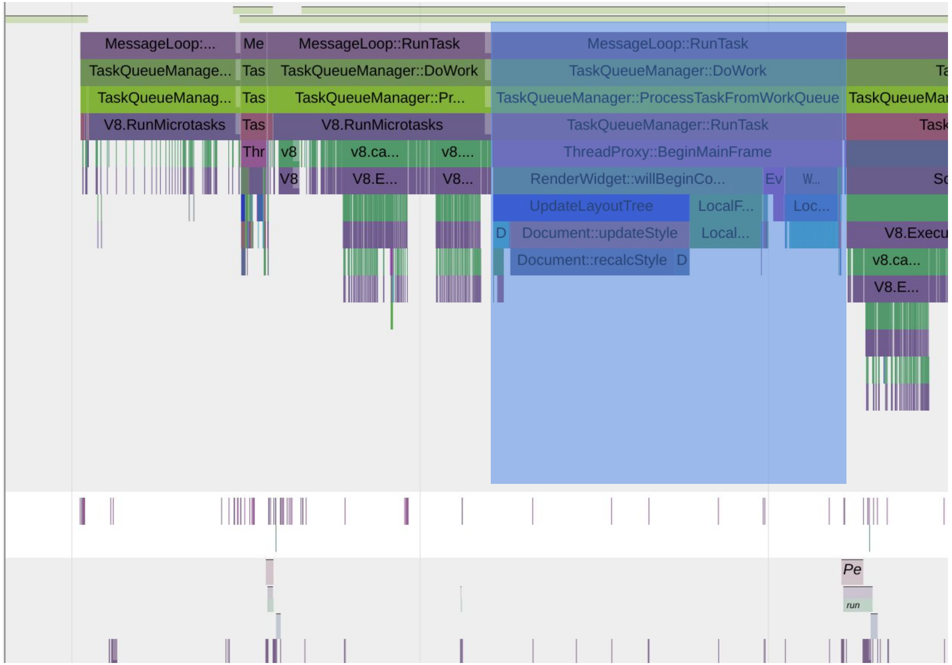
compositing setup

paint

raster

composite

input



network

script

parse

style

layout

compositing setup

paint

raster

composite

input

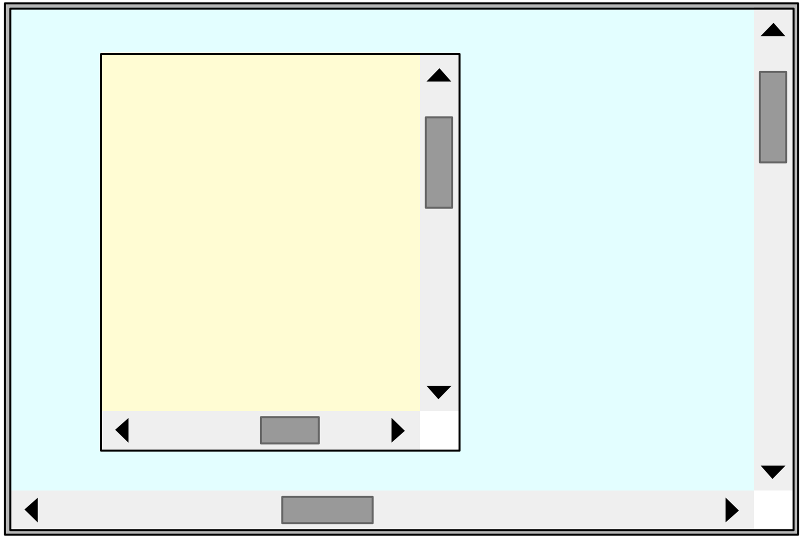
Rendering challenges

Scrolling

Paint & Compositing

Layout

Scrolling: A Brief History



Scrolling: Features, Optimizations, Complexity

Composited Scrolling

Threaded Scrolling

Custom Scrollbars

Scroll Event Handling

Touch/Fling Scrolling

Pinch Zoom

...

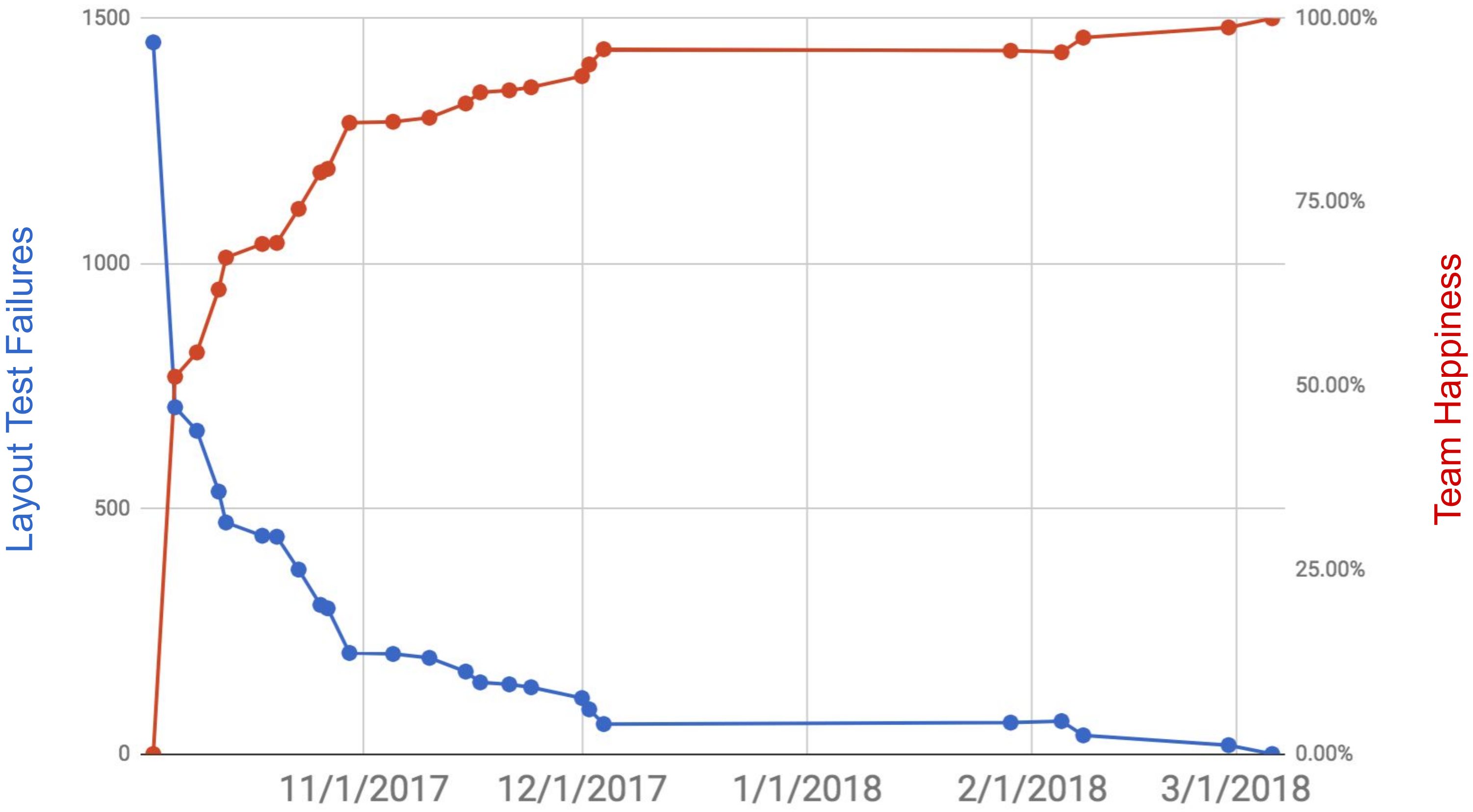
Root Layer Scrolling: One is Better than Two

- Initiated by Steve Kobes in 2014
- Goal: make document-level scrolls use the overflow scrolling code path
- Motivation: code health

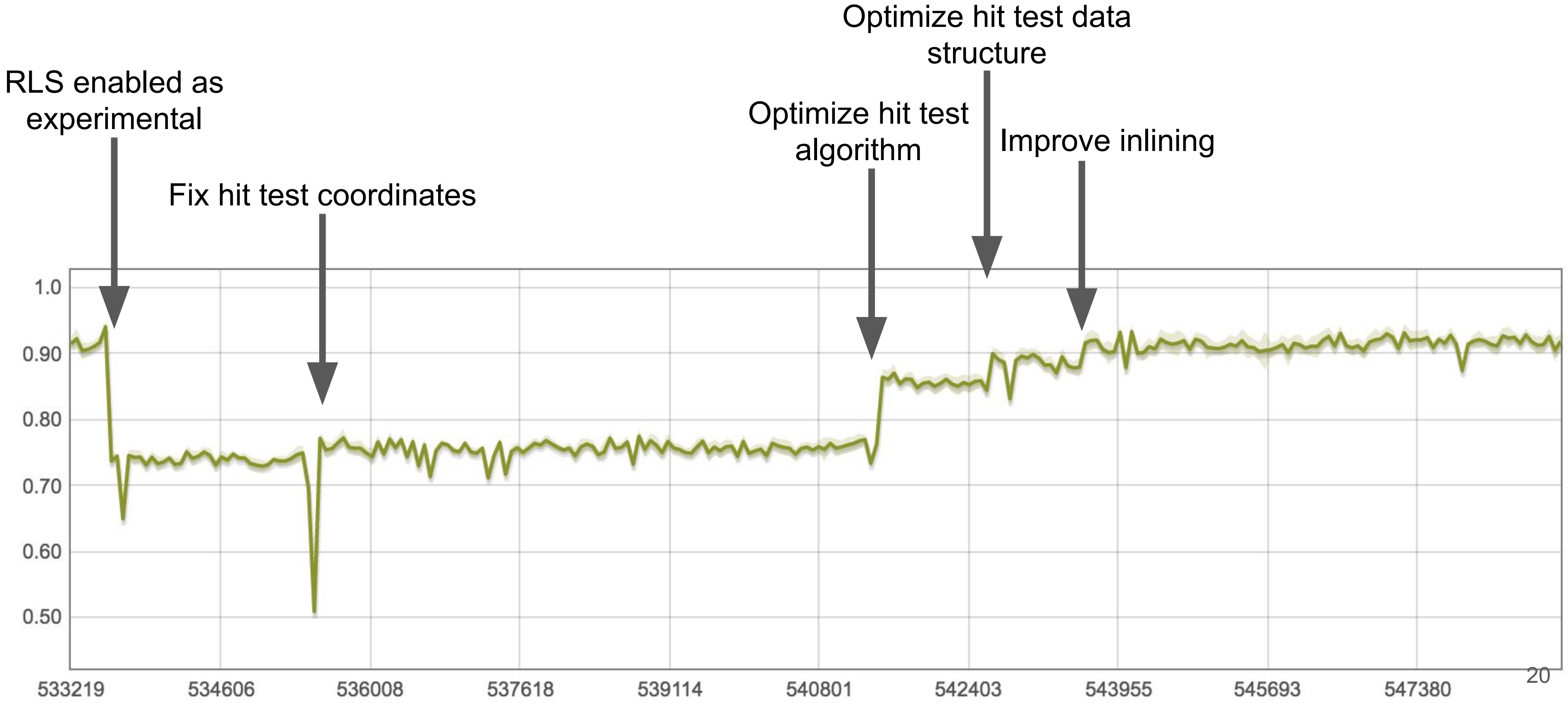
  ... but also resulted in many extra benefits

- Shipped in M66

Root layer scrolling



Root layer scrolling



RLS enabled as experimental

Fix hit test coordinates

Optimize hit test algorithm

Optimize hit test data structure

Improve inlining

Root layer scrolling

Steve Kobes (skobes@)

Stefan Zager (szager@)

Philip Rogers (pdr@)

David Bokan (bokan@)

Vladimir Levin (vmpstr@)

Chris Harrelson (chrishtr@)

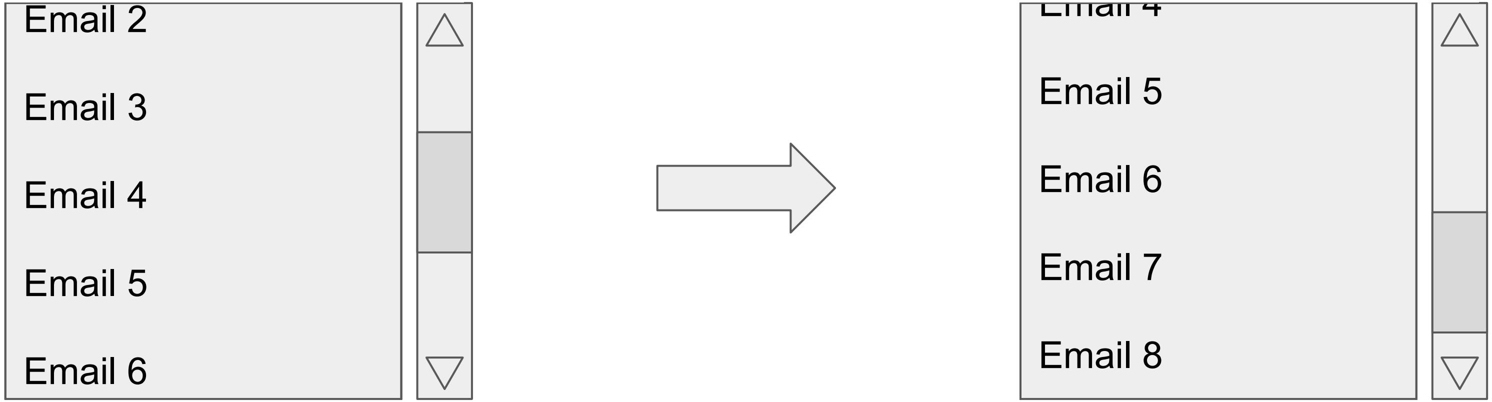
Rendering challenges

Scrolling

Paint and Compositing

Layout

History: naïve scrolling



Email 2

Email 3

Email 4

Email 5

Email 6

Email 4

Email 5

Email 6

Email 7

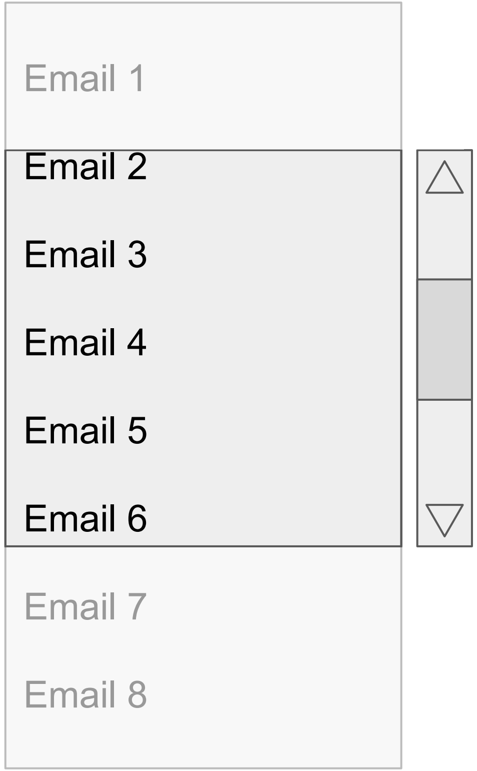
Email 8

History: composited, threaded scrolling

Composited: Scrolling becomes a blit*

Threaded: Don't have to wait for main thread

Amazing

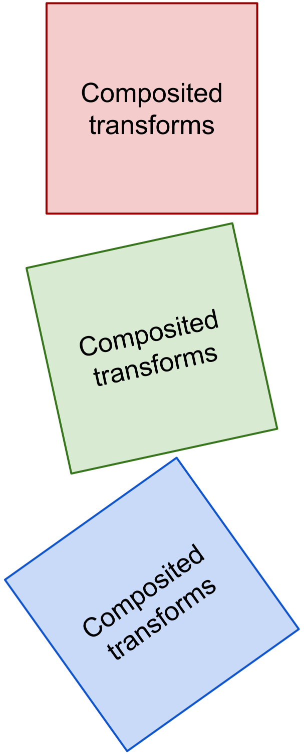


History: composited, threaded rendering

Works for more than scrolling

Opacity, transforms, animations, etc.

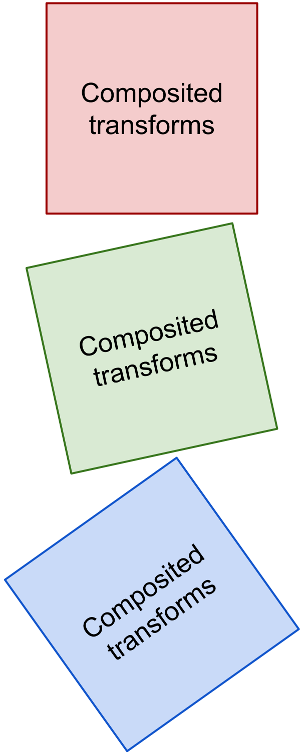
Spectacular



History: composited, threaded rendering

But...

Makes everything crazy!



Composited transforms

Composited transforms

Composited transforms

Current compositing architecture

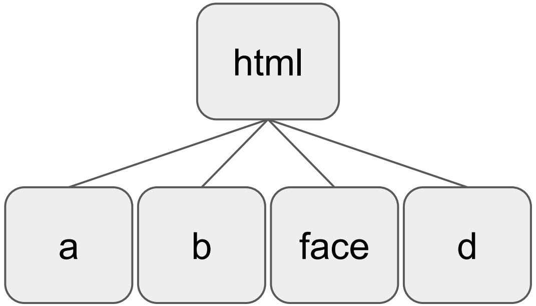
```
<html>
  <div>a</div>
  <div>b</div>
  <div>😀</div>
  <div>d</div>

  <style>
    😀: scrolls
  </style>
</html>
```

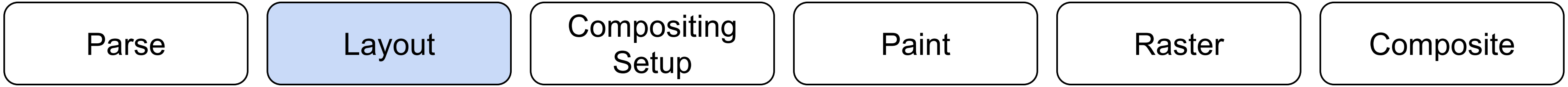
Parse | Layout | Compositing Setup | Paint | Raster | Composite

Current compositing architecture

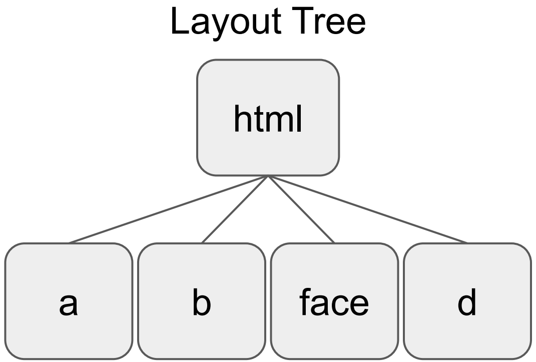
```
<html>
  <div>a</div>
  <div>b</div>
  <div>😀</div>
  <div>d</div>

  <style>
     😀: scrolls
  </style>
</html>
```

Layout Tree





Current compositing architecture



Parse

Layout

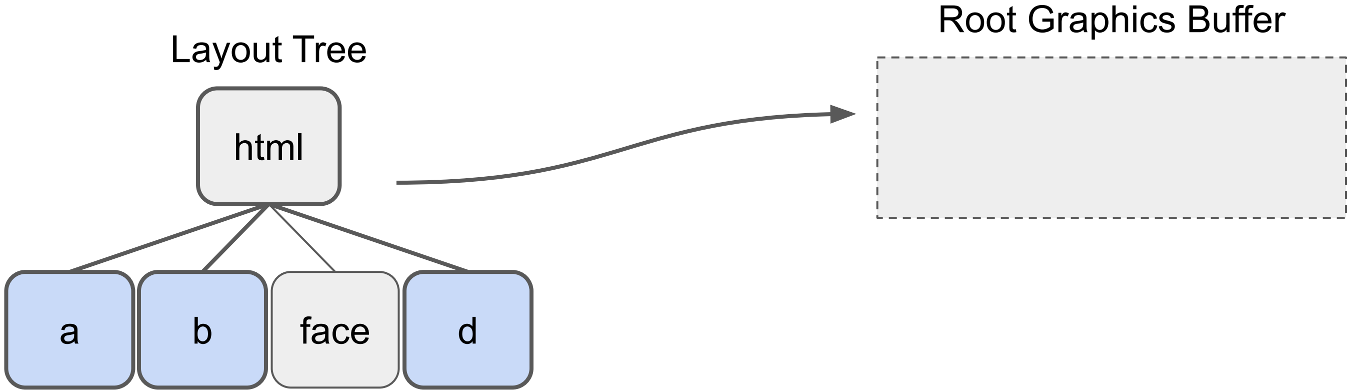
Compositing Setup

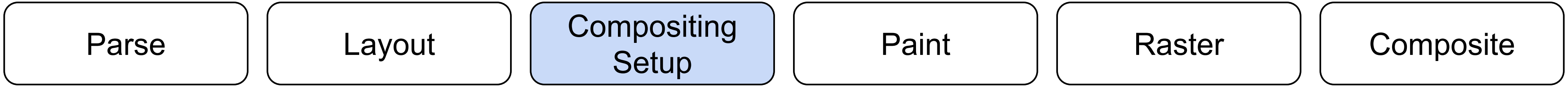
Paint

Raster

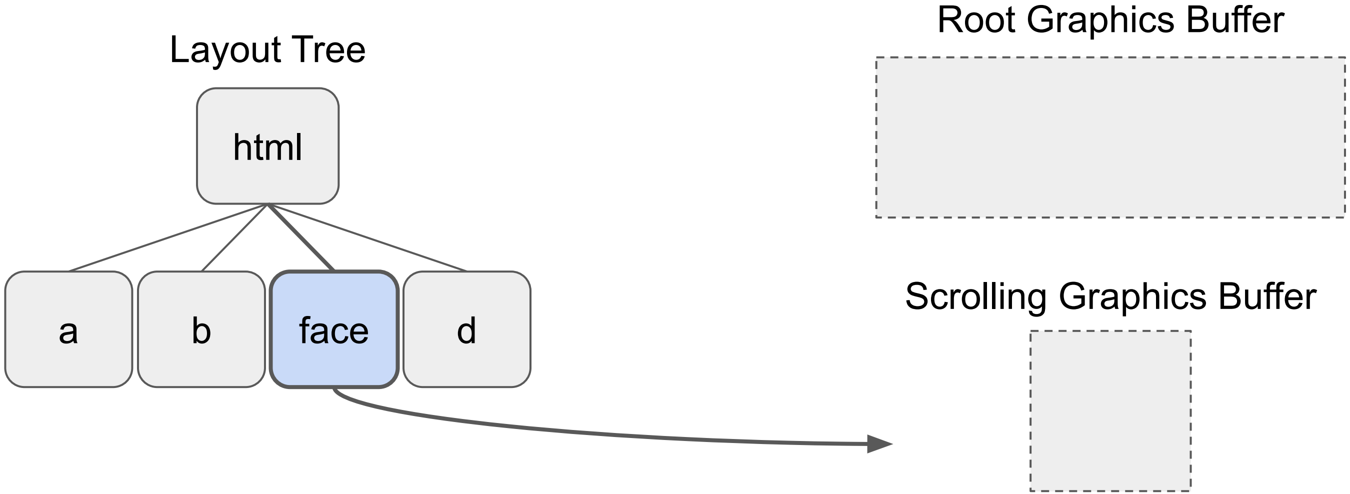
Composite

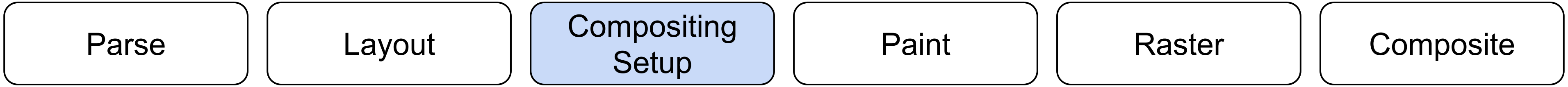
Current compositing architecture



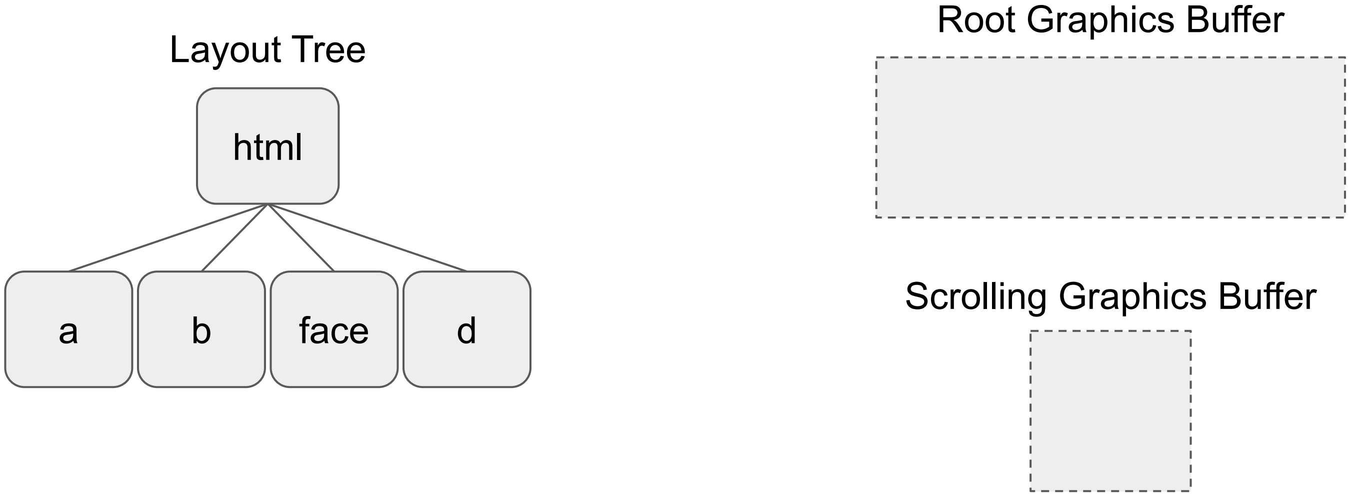


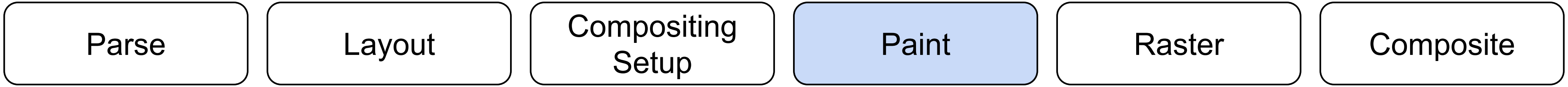
Current compositing architecture



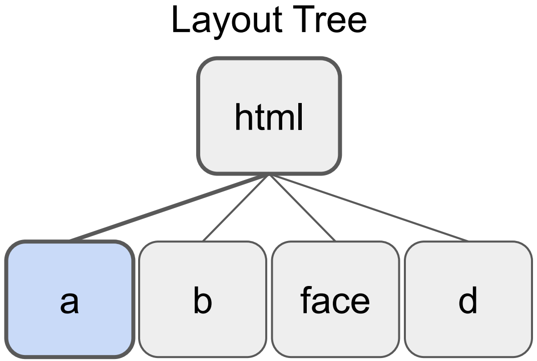


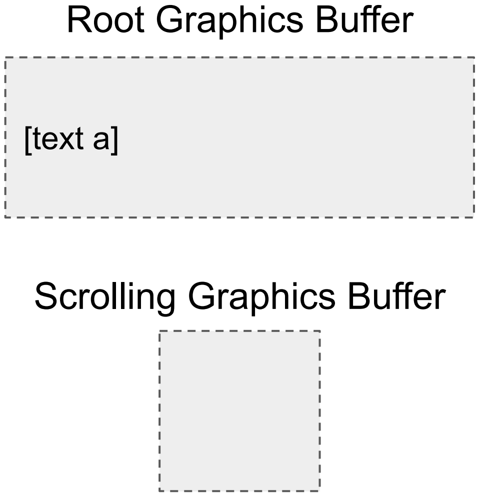
Current compositing architecture

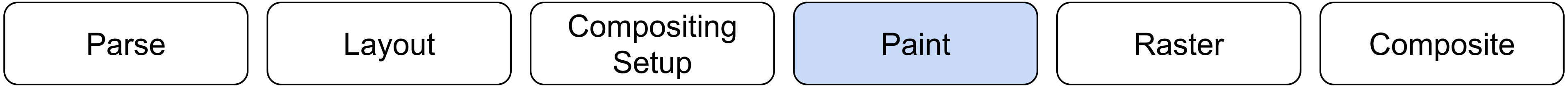




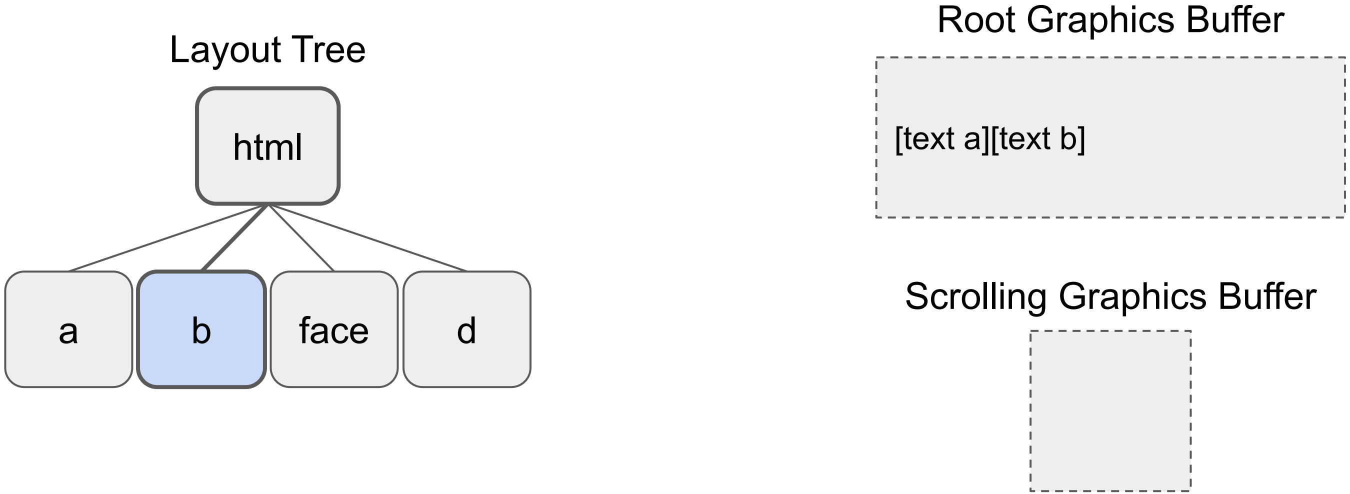
Current compositing architecture







Current compositing architecture



Layout Tree

html

a

b

face

d

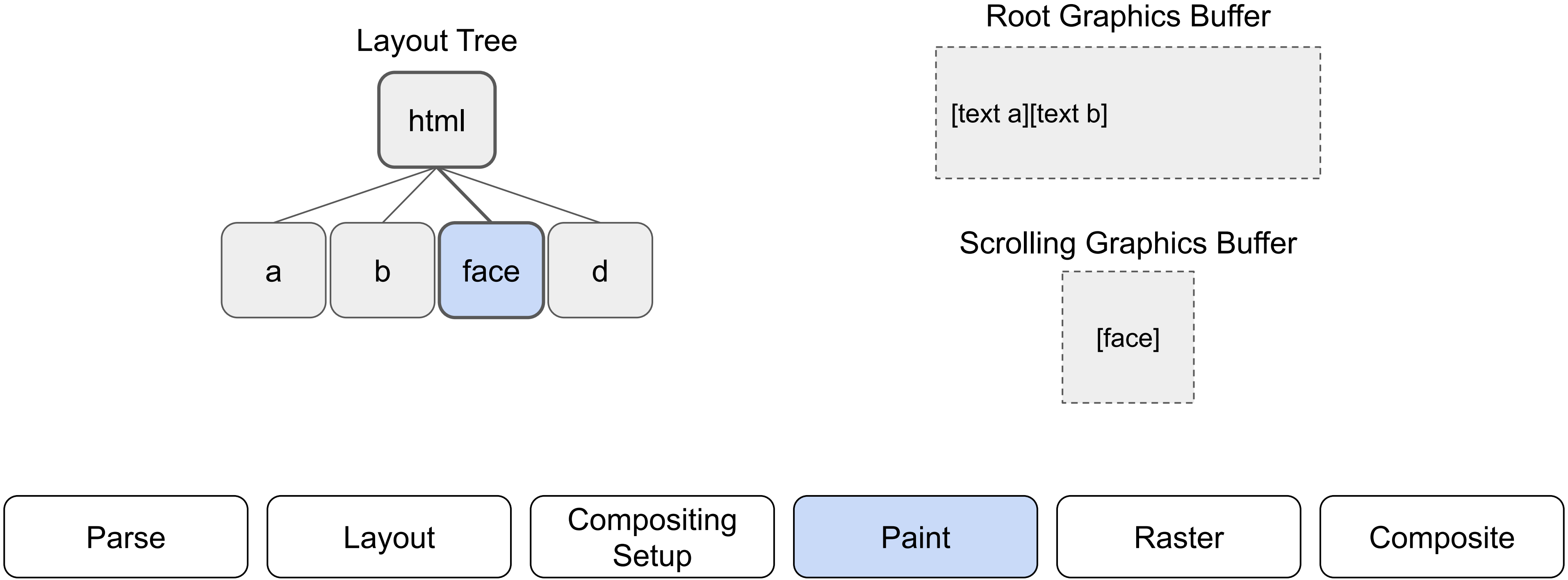
Root Graphics Buffer

[text a][text b]

Scrolling Graphics Buffer

Parse | Layout | Compositing Setup | Paint | Raster | Composite

Current compositing architecture



Layout Tree

html

a

b

face

d

Root Graphics Buffer

[text a][text b]

Scrolling Graphics Buffer

[face]

Parse

Layout

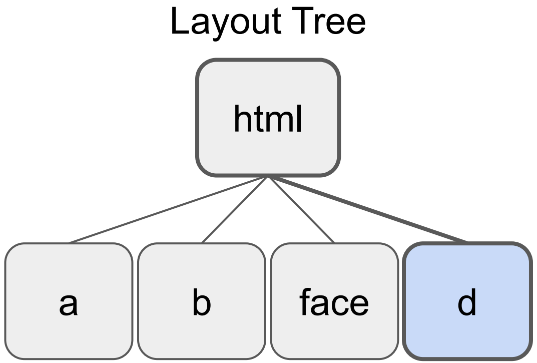
Compositing Setup

Paint

Raster

Composite

Current compositing architecture


Layout Tree

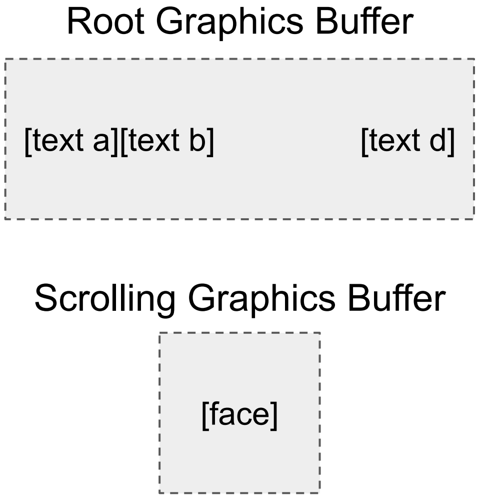
html

a

b

face

d


Root Graphics Buffer

[text a][text b]

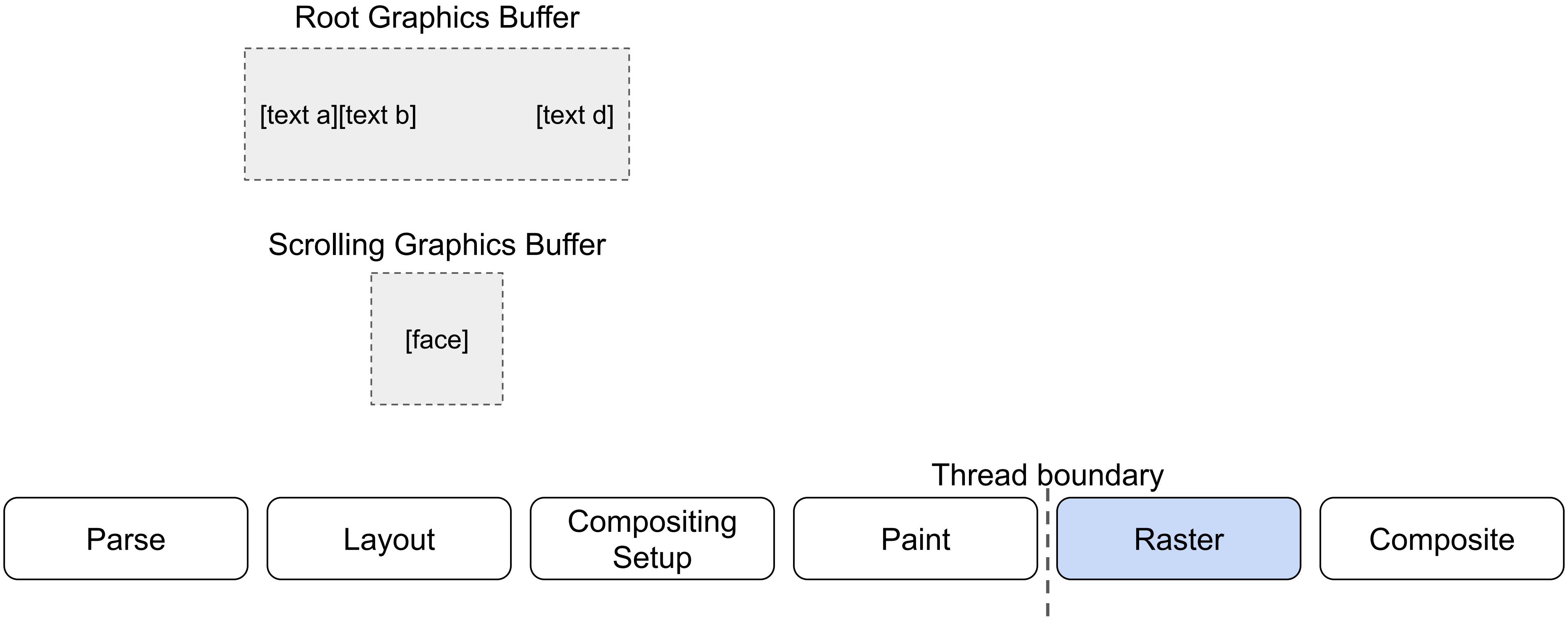
[text d]

Scrolling Graphics Buffer

[face]

Parse | Layout | Compositing Setup | **Paint** | Raster | Composite

Current compositing architecture



Root Graphics Buffer

[text a][text b]

[text d]

Scrolling Graphics Buffer

[face]

Thread boundary

Parse

Layout

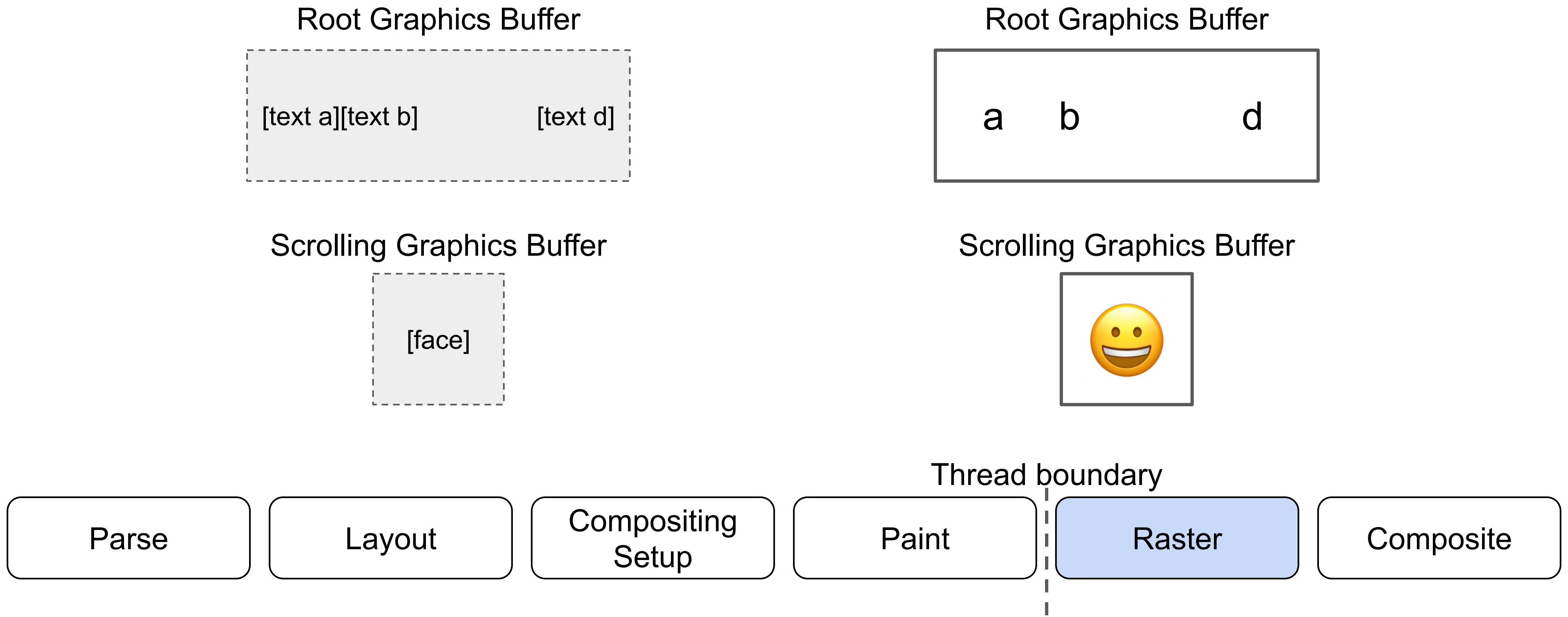
Compositing Setup

Paint

Raster

Composite

Current compositing architecture



Root Graphics Buffer

[text a][text b] [text d]

Scrolling Graphics Buffer

[face]

Root Graphics Buffer

a b d

Scrolling Graphics Buffer

😀

Thread boundary

Parse

Layout

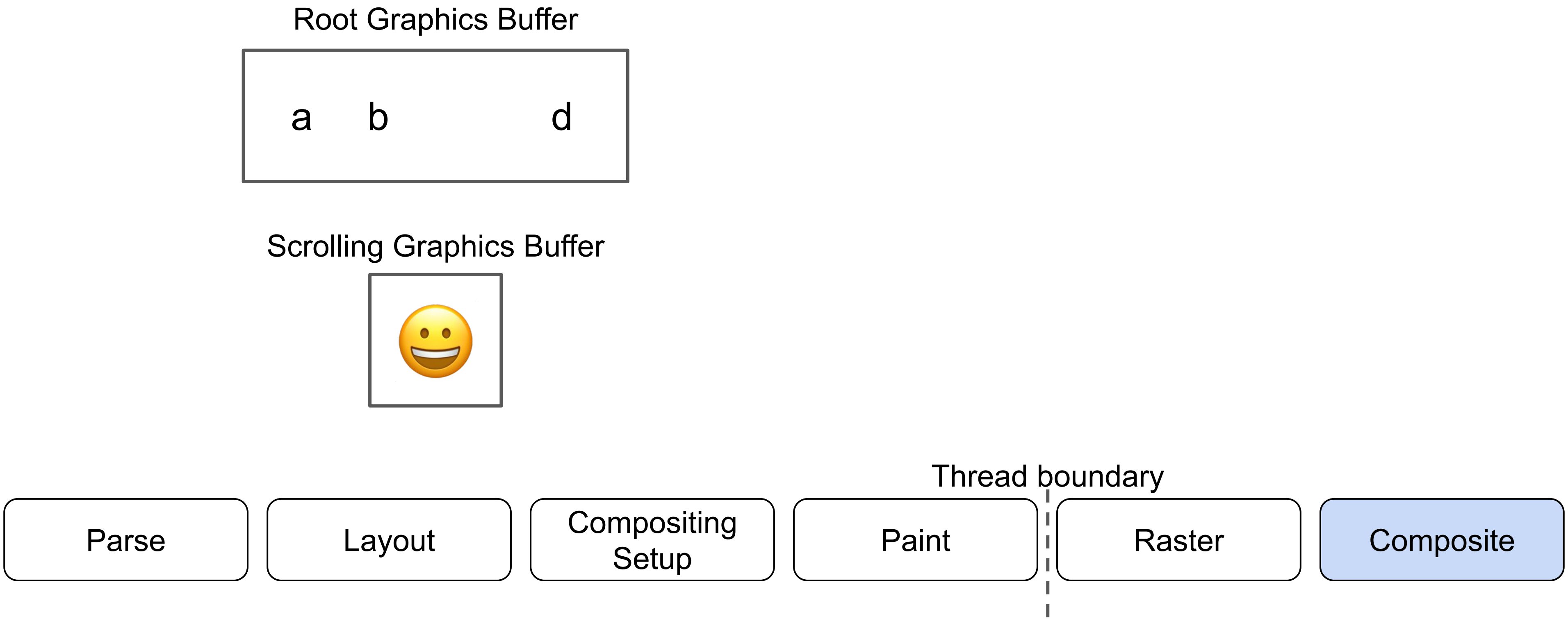
Compositing Setup

Paint

Raster

Composite

Current compositing architecture



Root Graphics Buffer

a b d

Scrolling Graphics Buffer

😀

Thread boundary

Parse

Layout

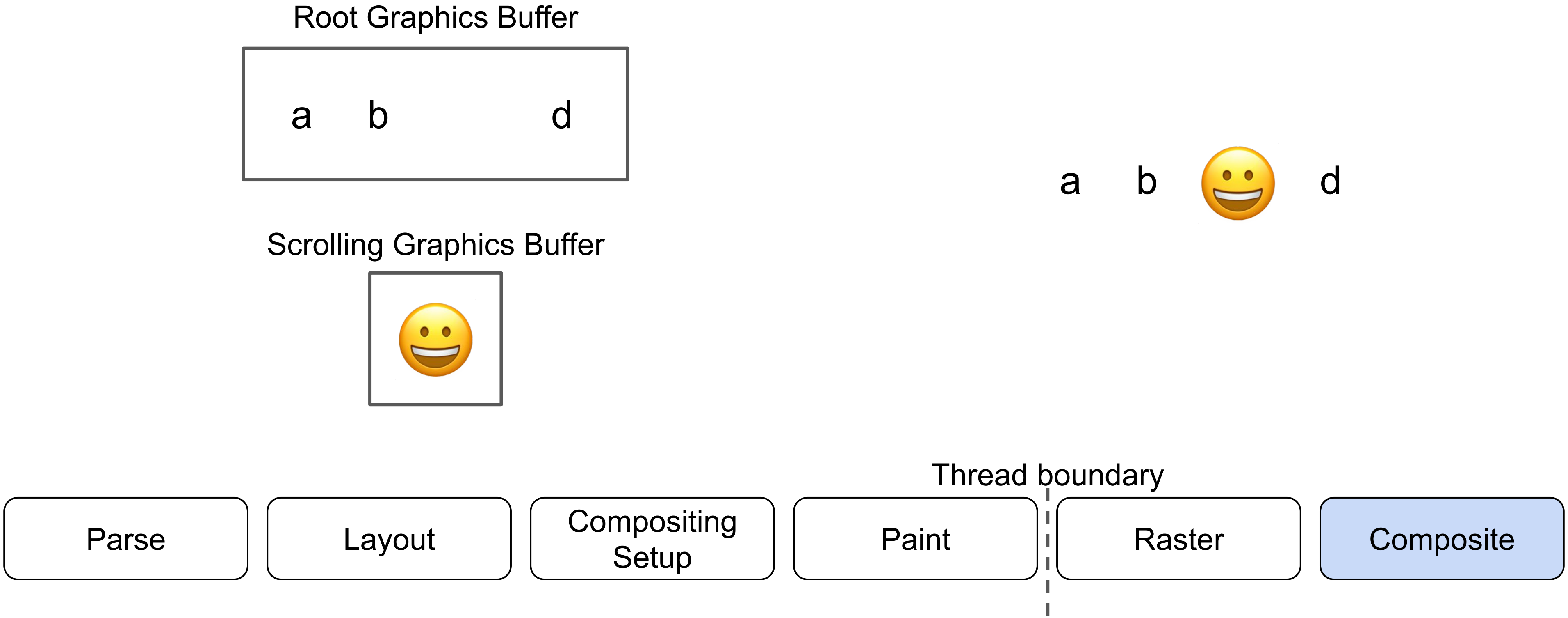
Compositing Setup

Paint

Raster

Composite

Current compositing architecture



Root Graphics Buffer

a b d

Scrolling Graphics Buffer

😀

a b 😀 d

Thread boundary

Parse

Layout

Compositing Setup

Paint

Raster

Composite

Current compositing architecture

- Compositing restricted to certain layout subtrees
  - Can not arbitrarily create graphics buffers, leads to the fundamental compositing bug

- Compositing setup before paint
  - Complex to do before paint: duplicate logic
  - Main thread
    - Suboptimal compositing decisions (one extra fullscreen buffer on 5k=60MB)
    - Difficult: threaded effects that change paint

Current compositing architecture

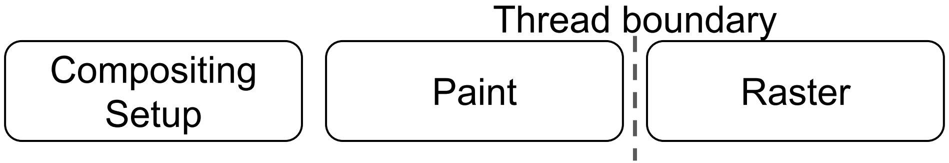
- Compositing restricted to certain layout subtrees
- Compositing setup before paint

New compositing architecture (Slimming Paint)

- Compositing allowed at any effect boundary
- Compositing setup after paint

Current compositing architecture

- Compositing restricted to certain layout subtrees
- Compositing setup before paint



Thread boundary

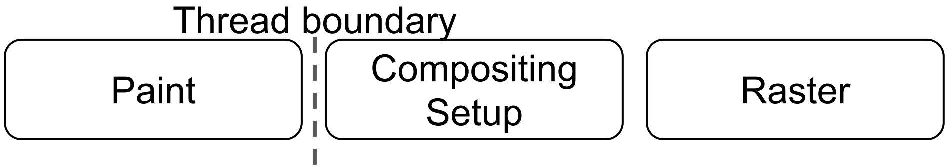
Compositing Setup

Paint

Raster

New compositing architecture (Slimming Paint)

- Compositing allowed at any effect boundary
- Compositing setup after paint



Thread boundary

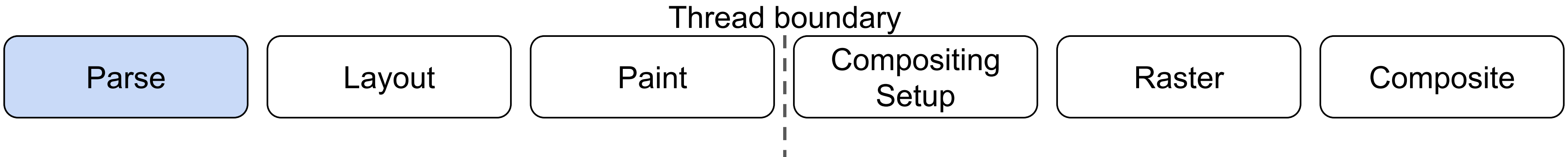
Paint

Compositing Setup

Raster

New compositing architecture (Slimming Paint)

```
<html>
  <div>a</div>
  <div>b</div>
  <div>😀</div>
  <div>d</div>

  <style>
    😀: scrolls
  </style>
</html>
```



Thread boundary

Parse

Layout

Paint

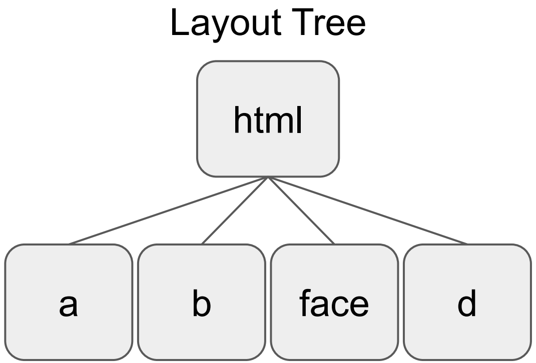
Compositing Setup

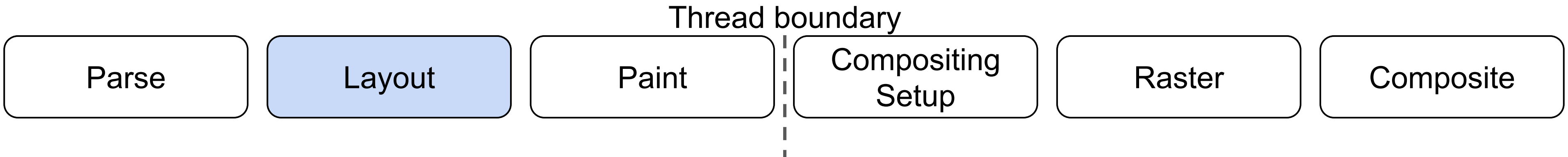
Raster

Composite

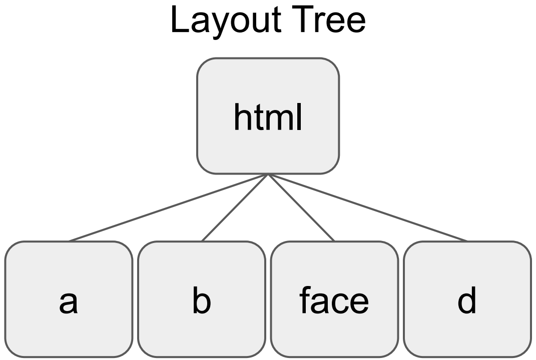
New compositing architecture (Slimming Paint)

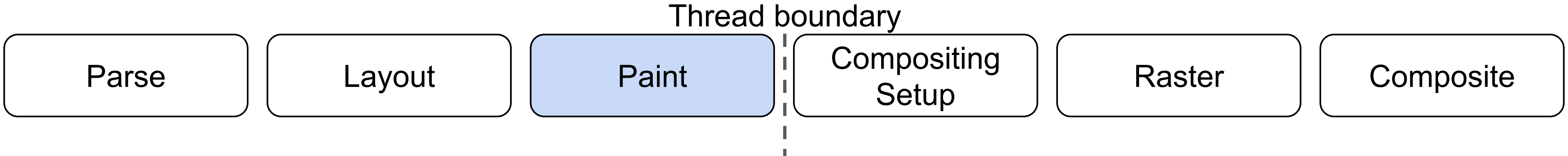
```
<html>
  <div>a</div>
  <div>b</div>
  <div>😀</div>
  <div>d</div>

  <style>
    😀: scrolls
  </style>
</html>
```



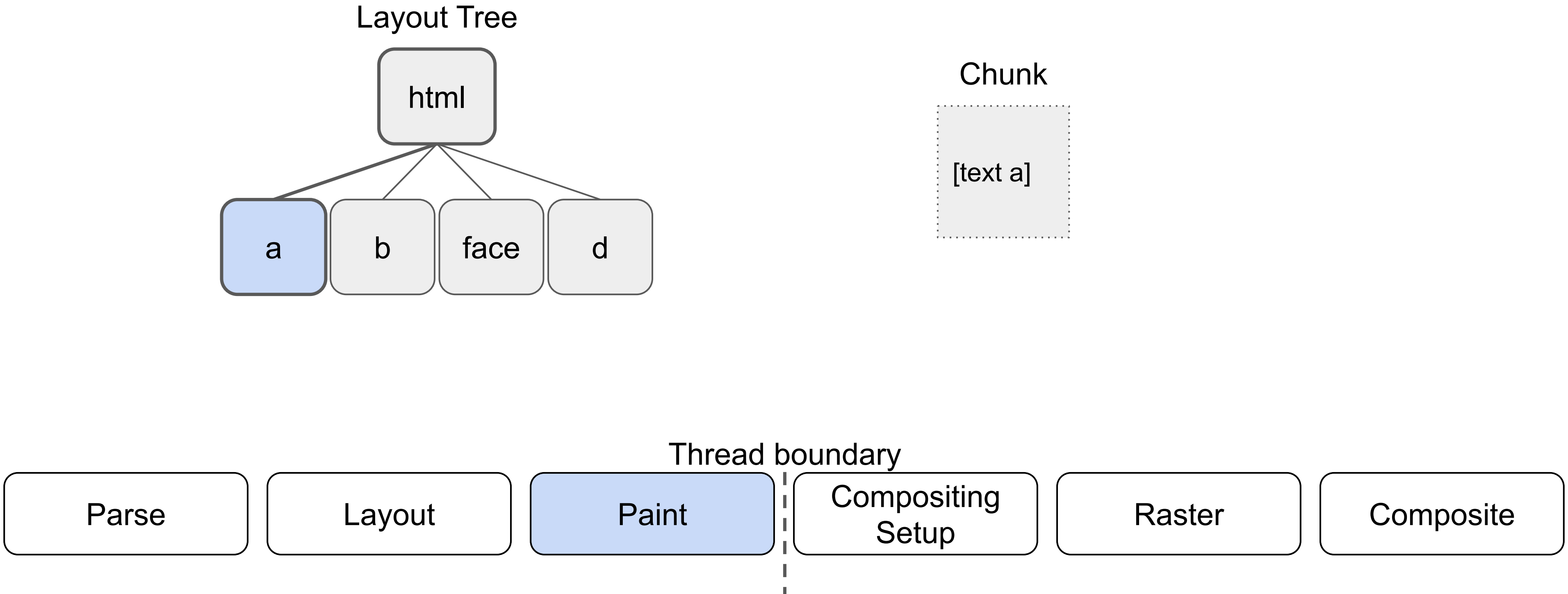


New compositing architecture (Slimming Paint)





New compositing architecture (Slimming Paint)



Layout Tree

html

a

b

face

d

Chunk

[text a]

Thread boundary

Parse

Layout

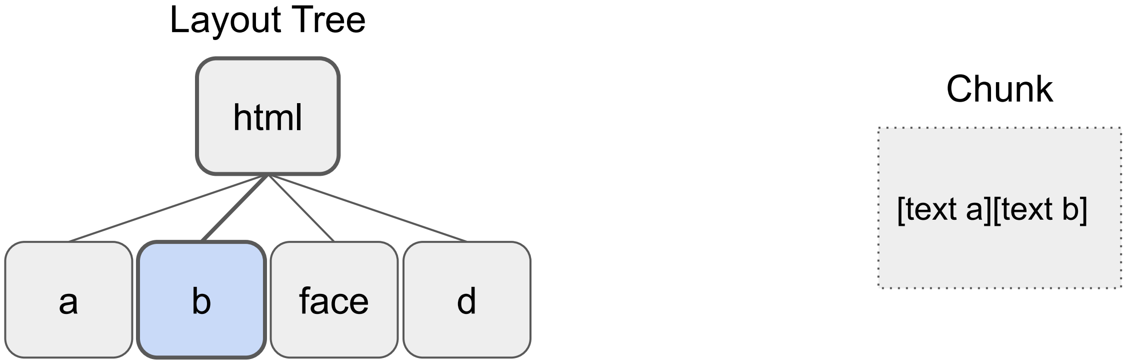
Paint

Compositing Setup

Raster

Composite

New compositing architecture (Slimming Paint)



Layout Tree

html

a

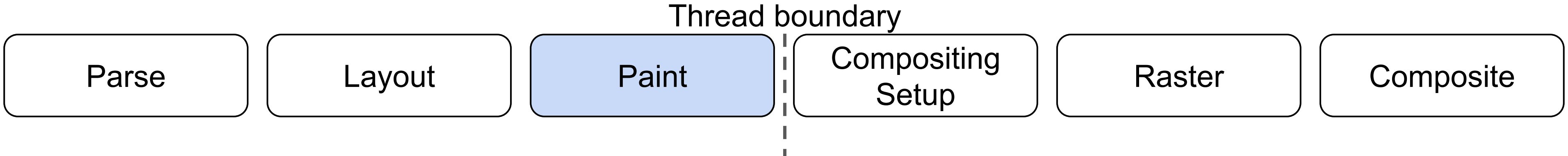
b

face

d

Chunk

[text a][text b]



Thread boundary

Parse

Layout

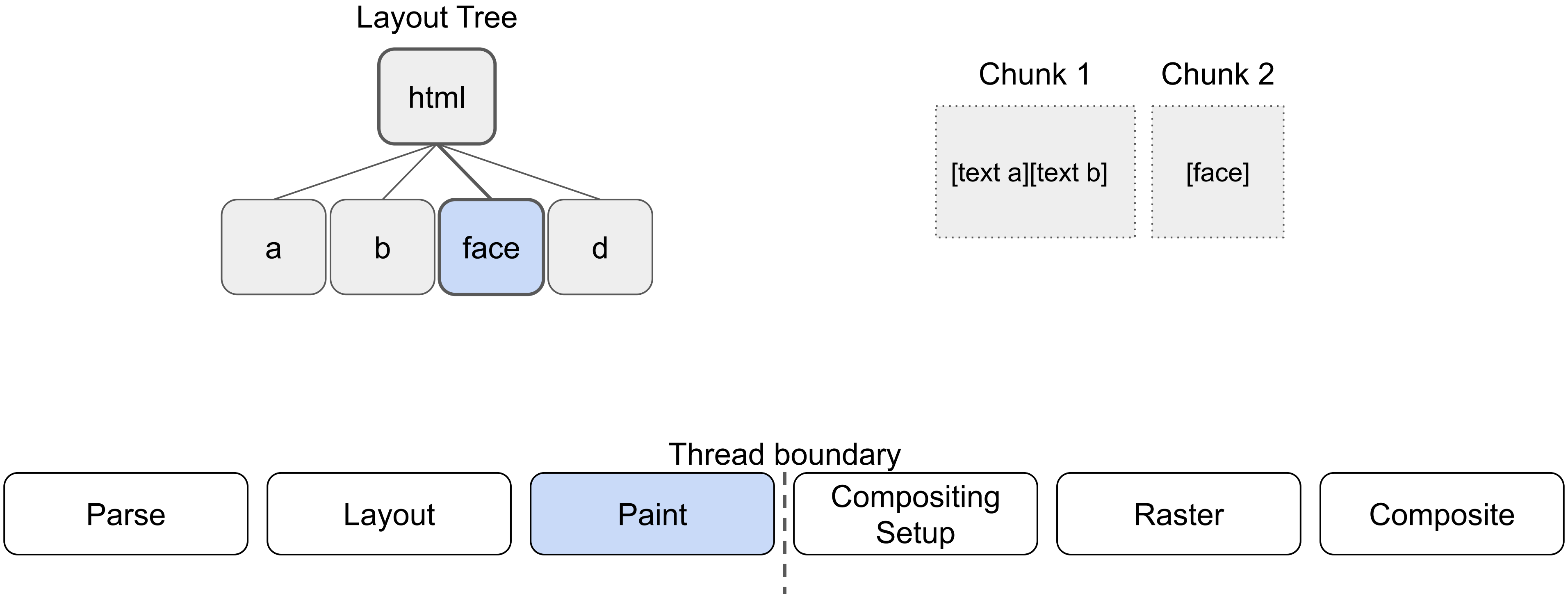
Paint

Compositing Setup

Raster

Composite

New compositing architecture (Slimming Paint)



Layout Tree

html

a

b

face

d

Chunk 1

[text a][text b]

Chunk 2

[face]

Thread boundary

Parse

Layout

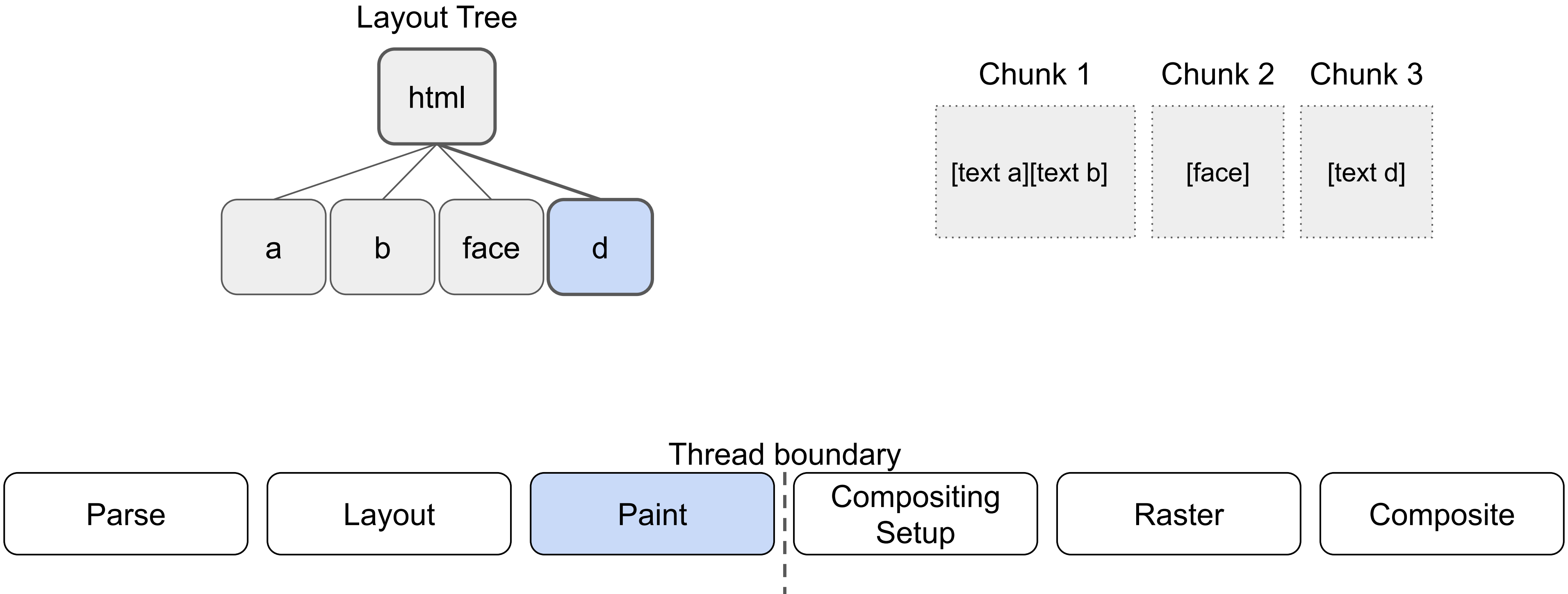
Paint

Compositing Setup

Raster

Composite

New compositing architecture (Slimming Paint)



Layout Tree

html

a

b

face

d

Chunk 1

[text a][text b]

Chunk 2

[face]

Chunk 3

[text d]

Thread boundary

Parse

Layout

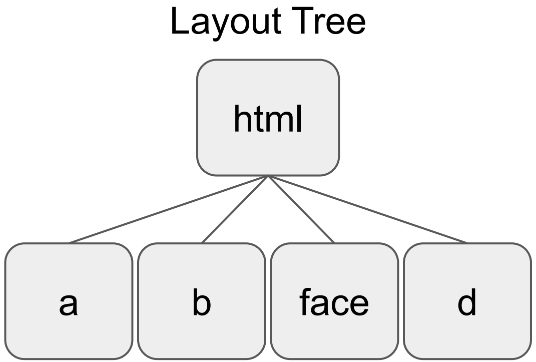
Paint

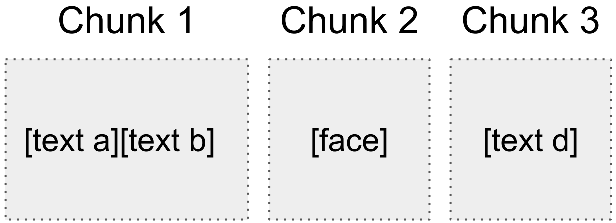
Compositing Setup

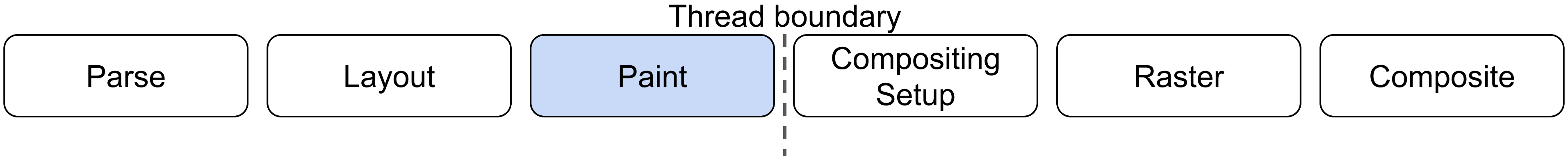
Raster

Composite

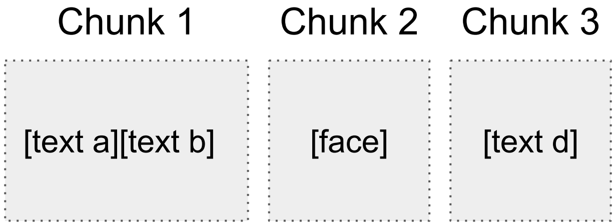
New compositing architecture (Slimming Paint)







New compositing architecture (Slimming Paint)



Chunk 1

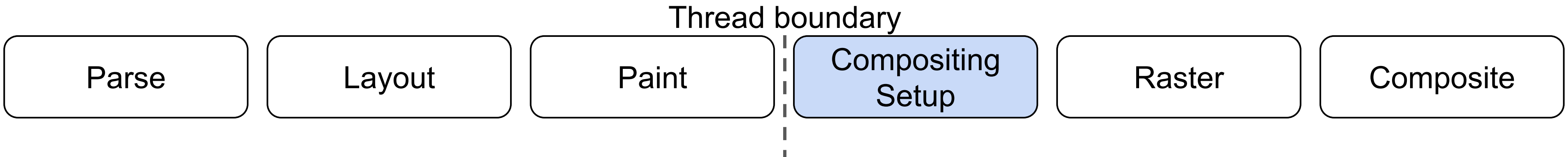
[text a][text b]

Chunk 2

[face]

Chunk 3

[text d]



Thread boundary

Parse

Layout

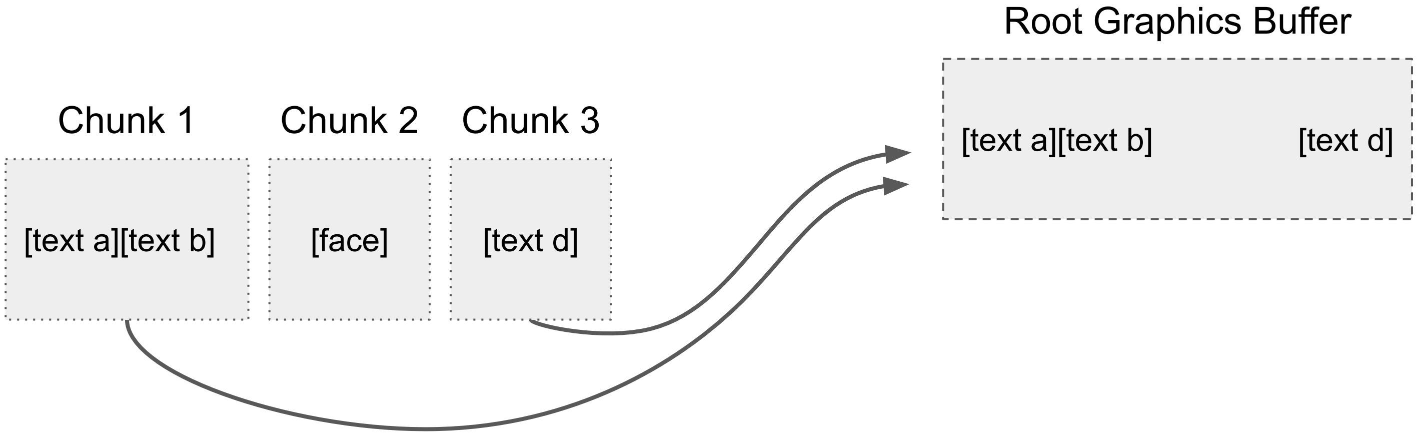
Paint

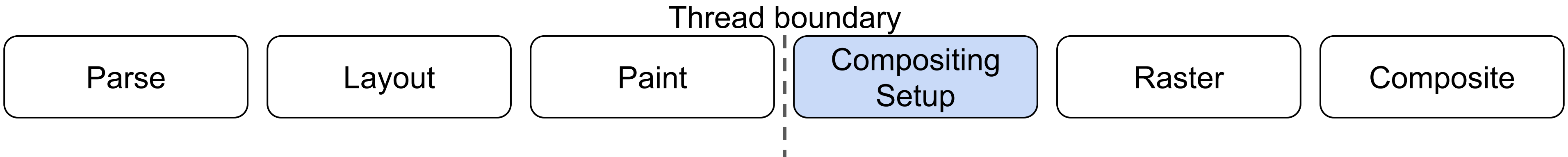
Compositing Setup

Raster

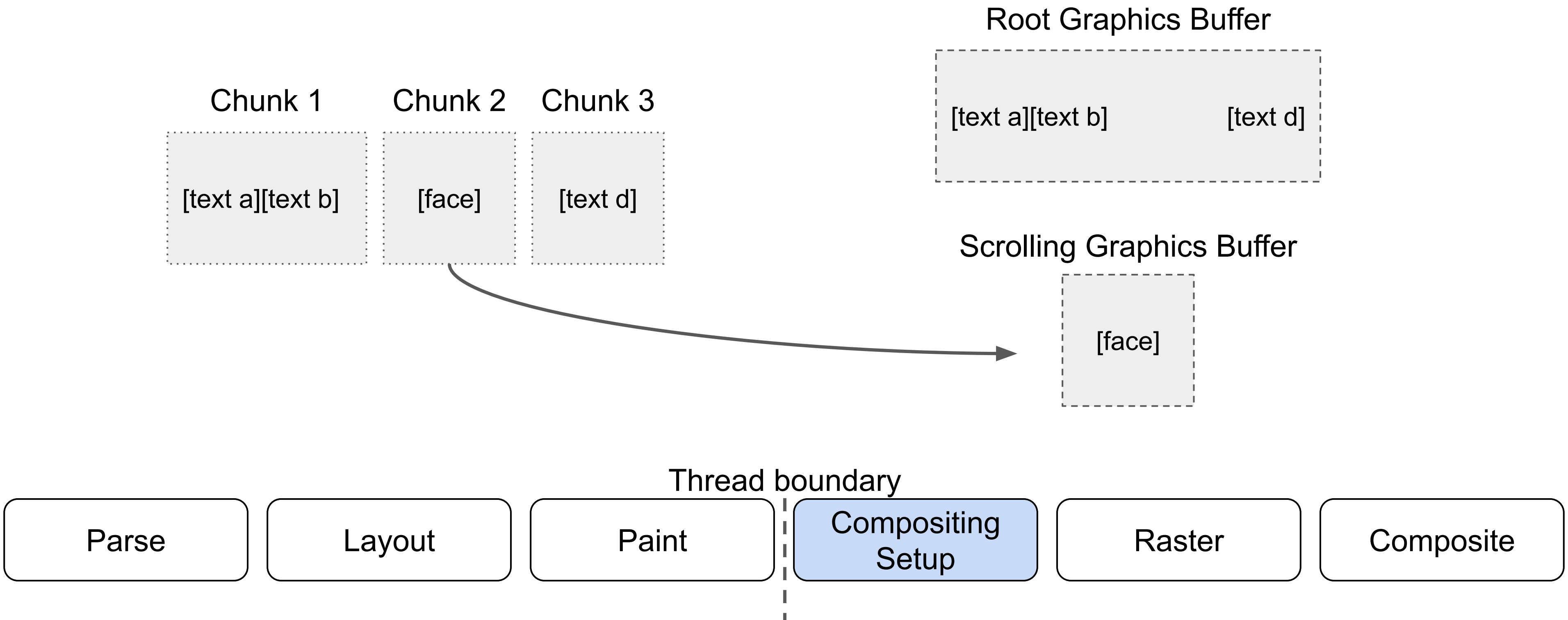
Composite

New compositing architecture (Slimming Paint)

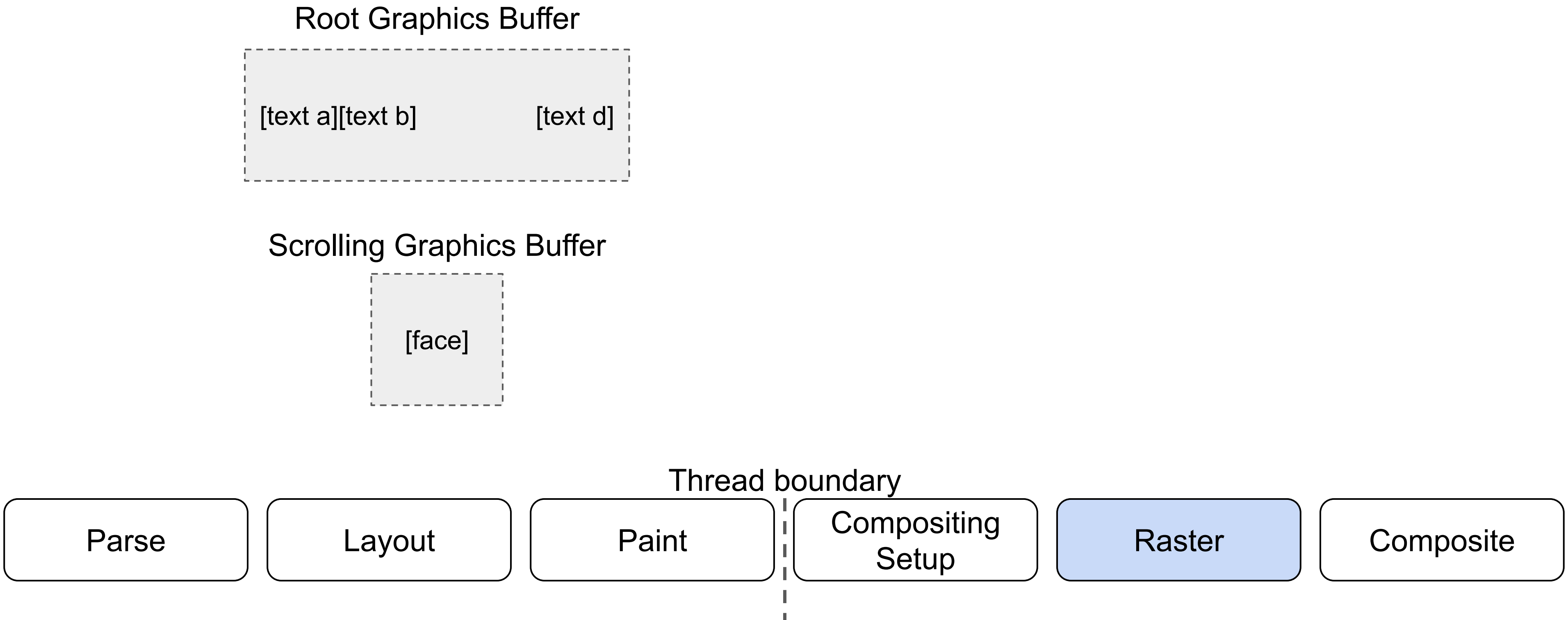




New compositing architecture (Slimming Paint)



New compositing architecture (Slimming Paint)



Root Graphics Buffer

[text a][text b]

[text d]

Scrolling Graphics Buffer

[face]

Thread boundary

Parse

Layout

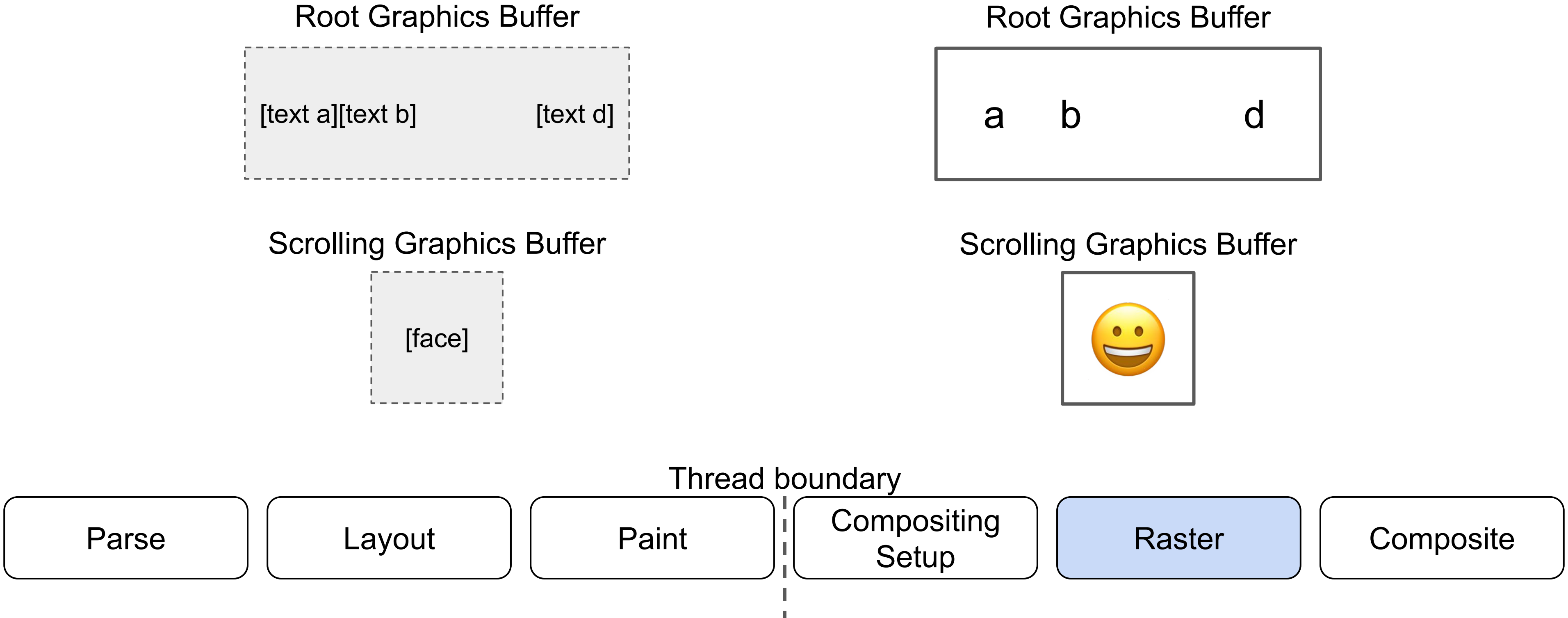
Paint

Compositing Setup

Raster

Composite

New compositing architecture (Slimming Paint)



Root Graphics Buffer

[text a][text b] [text d]

Scrolling Graphics Buffer

[face]

Root Graphics Buffer

a b d

Scrolling Graphics Buffer

😀

Thread boundary

Parse

Layout

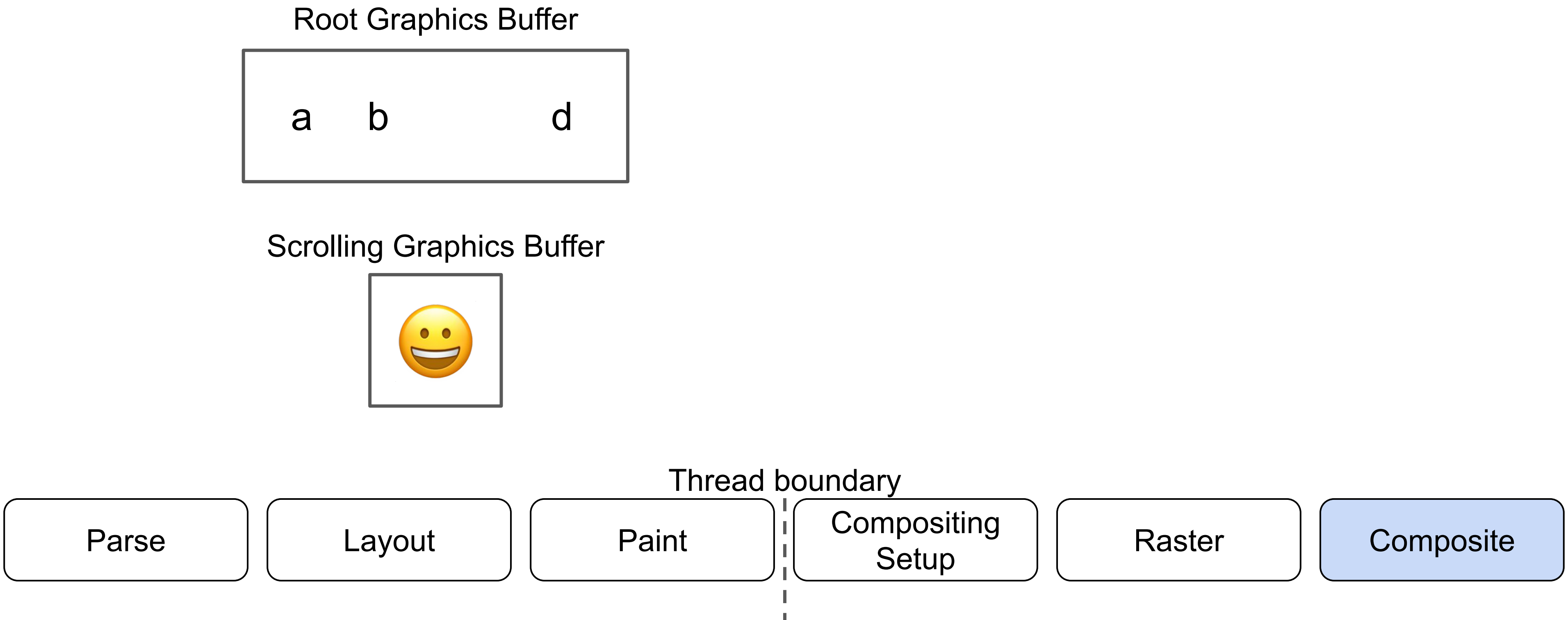
Paint

Compositing Setup

Raster

Composite

New compositing architecture (Slimming Paint)



Root Graphics Buffer

a b d

Scrolling Graphics Buffer

Thread boundary

Parse

Layout

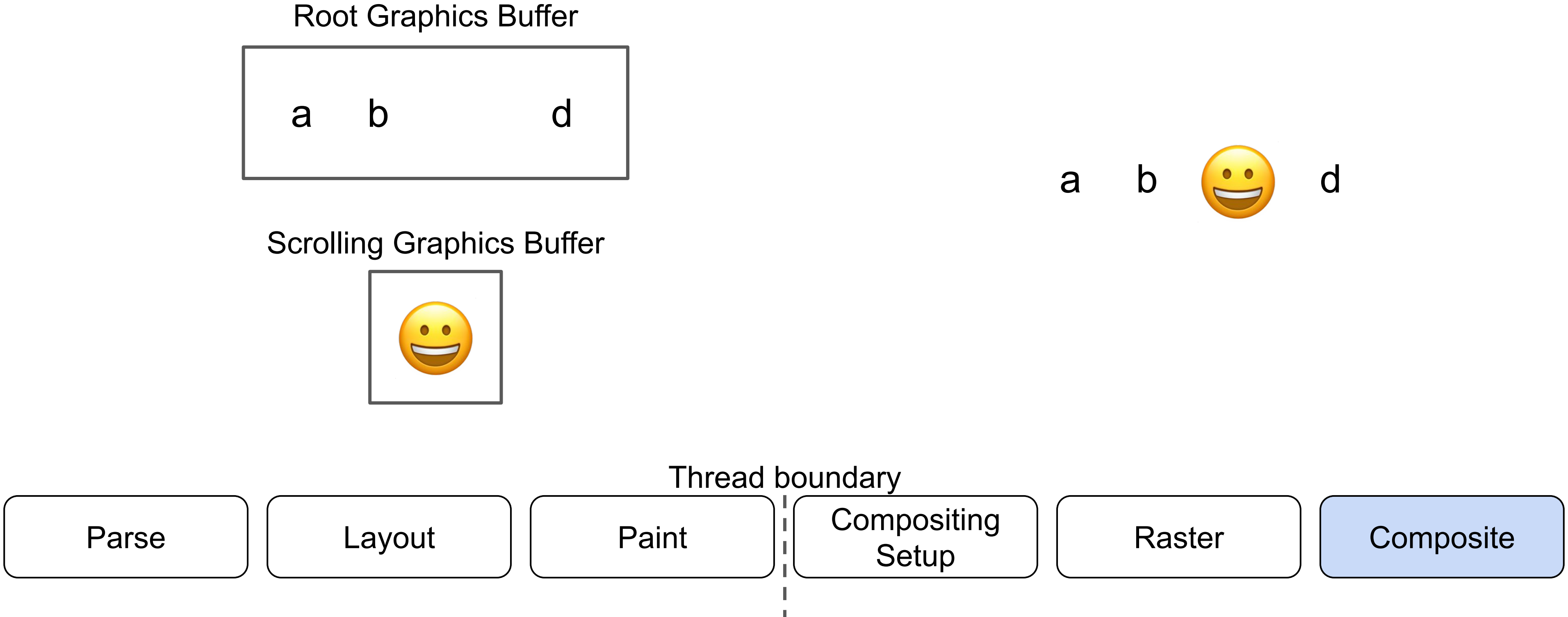
Paint

Compositing Setup

Raster

Composite

New compositing architecture (Slimming Paint)



Root Graphics Buffer

a b d

Scrolling Graphics Buffer

😀

a b 😀 d

Thread boundary

Parse

Layout

Paint

Compositing Setup

Raster

Composite

New compositing architecture (Slimming Paint)

- Can composite at any effect boundary
  - Paint chunks fix fundamental compositing bug

- Compositing setup after paint
  - Better code health
  - Easier to optimize memory/performance trade-offs
  - Decouples threading from compositing
  - Moves work off the main thread

Slimming Paint: do not let engineers name projects

- Slimming Paint V1 (Sept 2015, M45)
  - Moved paint out of layout, added display items to cache paint
  - Led to -25% paint time, -25% raster time
- Slimming Paint V1.5 (June 2017, M59)
  - Added property trees, geometry mapper, simpler paint invalidation
  - Faster paint invalidation, -10% @ 75th, -6% @ 95th
- **Slimming Paint V1.75 (launching now, M67)**
  - Paint using chunks, but with existing compositing decisions
  - Fixes top (77-star) paint bug (771852): transformed HTML in SVG (foreignObject)
  - Improved raster invalidation, 3% fewer tiles rastered (go/spv175finch)
- Slimming Paint V2 (later in 2018)
  - Move compositing setup after paint

Slimming Paint: do not let engineers name projects

Cache paint

- Slimming Paint V1 (Sept 2015, M45)
  - Moved paint out of layout, added display items to cache paint
  - Led to -25% paint time, -25% raster time
- Slimming Paint V1.5 (June 2017, M59)
  - Added property trees, geometry mapper, simpler paint invalidation
  - Faster paint invalidation, -10% @ 75th, -6% @ 95th

Composite after paint

- **Slimming Paint V1.75 (launching now, M67)**
  - Paint using chunks, but with existing compositing decisions
  - Fixes top (77-star) paint bug (771852): transformed HTML in SVG (foreignObject)
  - Improved raster invalidation, 3% fewer tiles rastered (go/spv175finch)
- Slimming Paint V2 (later in 2018)
  - Move compositing setup after paint

Slimming Paint 1.75 Launch (M67)

Launching now, M67

- Launches big part of SPV2 using existing compositing setup
  - De-risks final SPV2 launch

- Adds paint chunks: new display list grouping
  - Will become potential composited layers in SPV2
  - Cleaner code factoring
    - Fixes top (77-star) paint bug (771852): transformed HTML in SVG
  - Raster invalidation moved from before-paint to after-paint
    - 3% fewer tiles rastered (go/spv175finch)

Slimming Paint 1.75 Launch (M67)

Xianzhu Wang (wangxianzhu@)

Tien-Ren Chen (trchen@)

Philip Rogers (pdr@)

Chris Harrelson (chrishtr@)

Rendering challenges

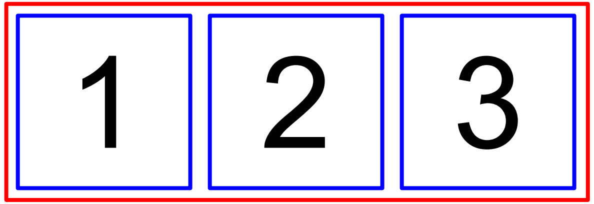
Scrolling

Paint & Compositing

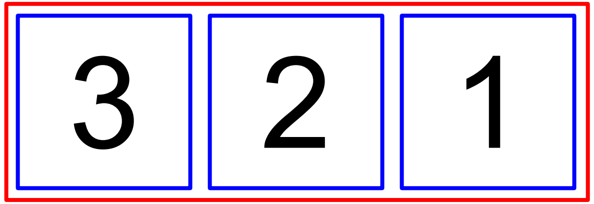
Layout

Layout: The Combinatorial Problem

```
.red {
  display: flex;
}
```

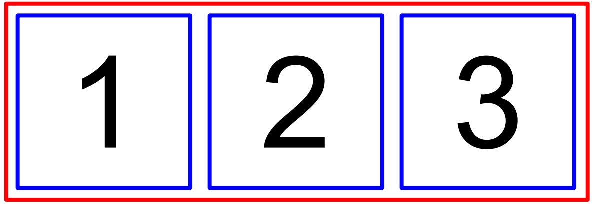


Layout: The Combinatorial Problem



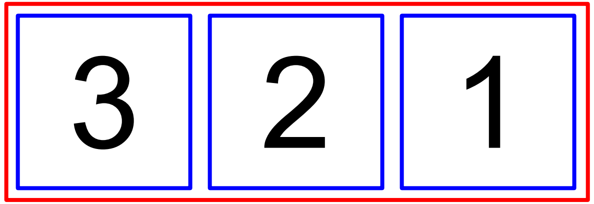
```
.red {
  display: flex;
  direction: rtl;
}
```

Layout: The Combinatorial Problem



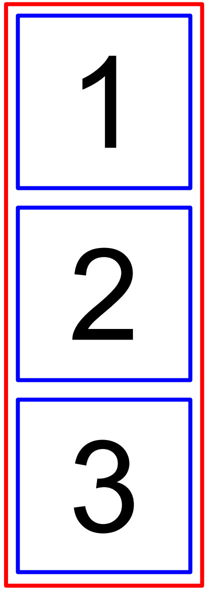
```
.red {
  display: flex;
  direction: rtl;
  flex-direction: row-reverse;
}
```

Layout: The Combinatorial Problem



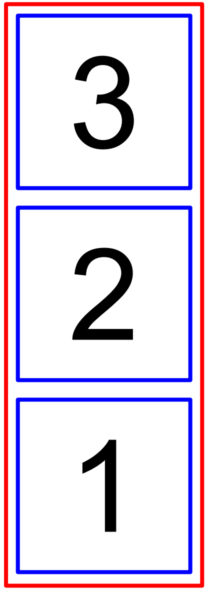
```
.red {
  display: flex;

  flex-direction: row-reverse;
}
```

Layout: The Combinatorial Problem



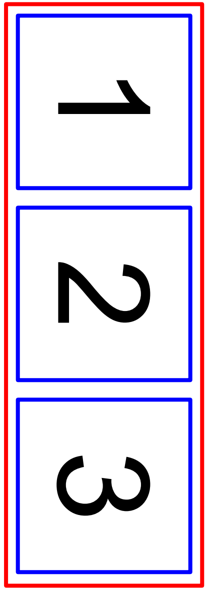
```
.red {
  display: flex;

  flex-direction: column;
}
```

Layout: The Combinatorial Problem



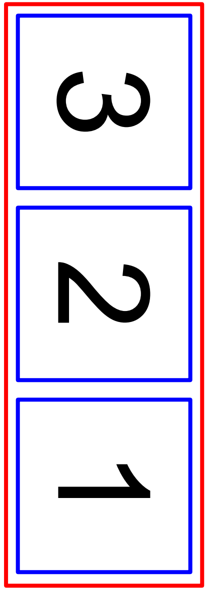
```
.red {
  display: flex;

  flex-direction: column-reverse;
}
```

Layout: The Combinatorial Problem



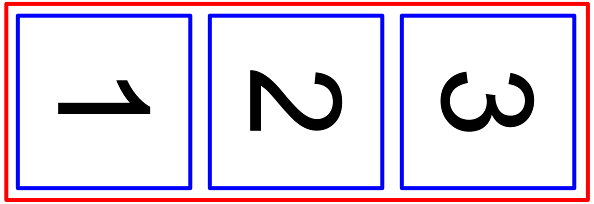
```
.red {
  display: flex;

  flex-direction: row;
  writing-mode: vertical-lr;
}
```

Layout: The Combinatorial Problem

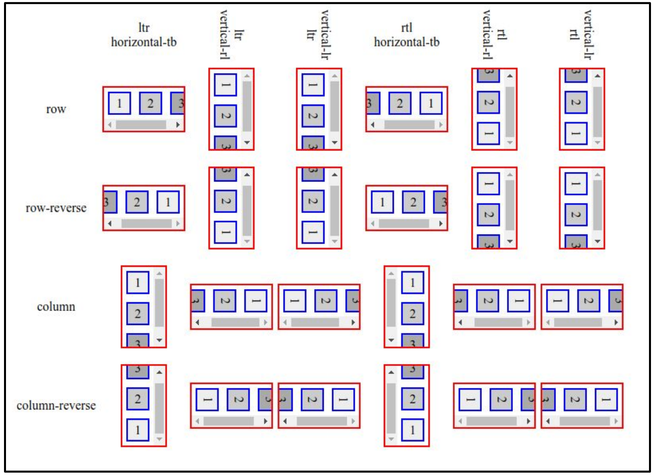


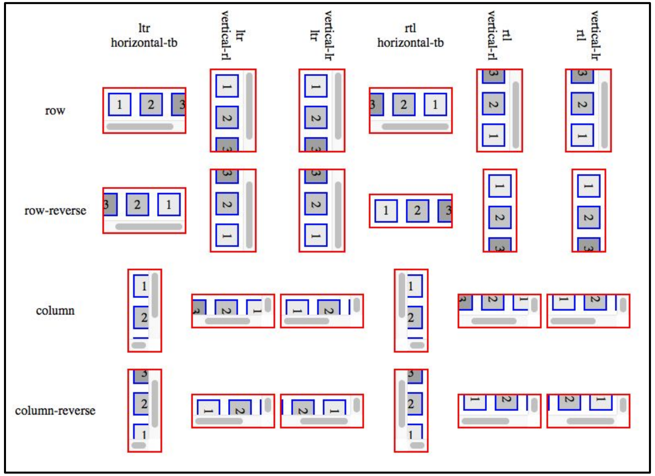
```
.red {
  display: flex;

  flex-direction: row-reverse;
  writing-mode: vertical-lr;
}
```

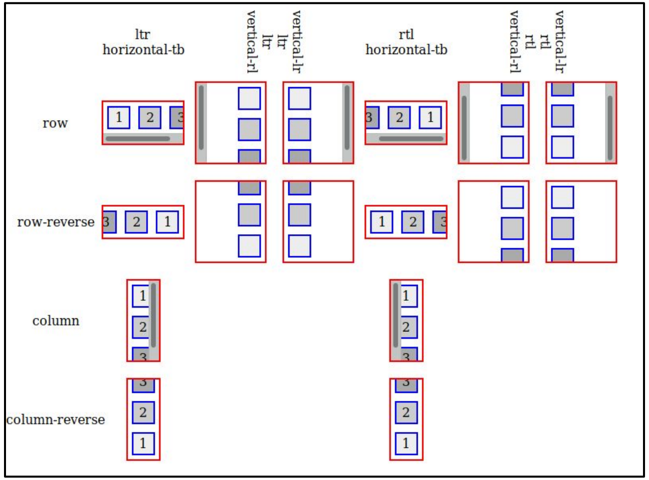
Layout: The Combinatorial Problem

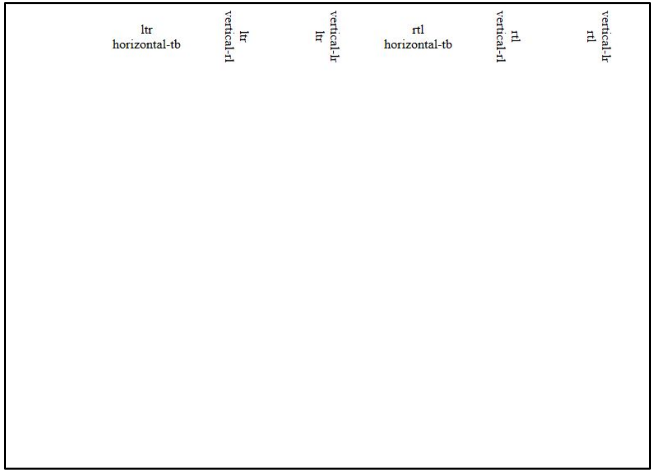


```
.red {
  display: flex;

  flex-direction: column;
  writing-mode: vertical-lr;
}
```


ltr horizontal-tb
ltr vertical-rl
ltr vertical-lr
rtl horizontal-tb
rtl vertical-rl
rtl vertical-lr
row
row-reverse
column
column-reverse


ltr horizontal-tb
ltr vertical-rl
ltr vertical-lr
rtl horizontal-tb
rtl vertical-rl
rtl vertical-lr
row
row-reverse
column
column-reverse


ltr horizontal-tb
ltr vertical-rl
ltr vertical-lr
rtl horizontal-tb
rtl vertical-rl
rtl vertical-lr
row
row-reverse
column
column-reverse

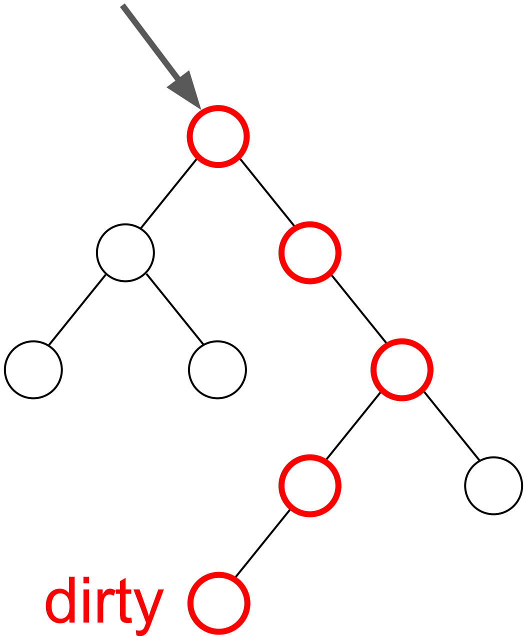

ltr horizontal-tb
ltr vertical-rl
ltr vertical-lr
rtl horizontal-tb
rtl vertical-rl
rtl vertical-lr

Layout: Existing Code

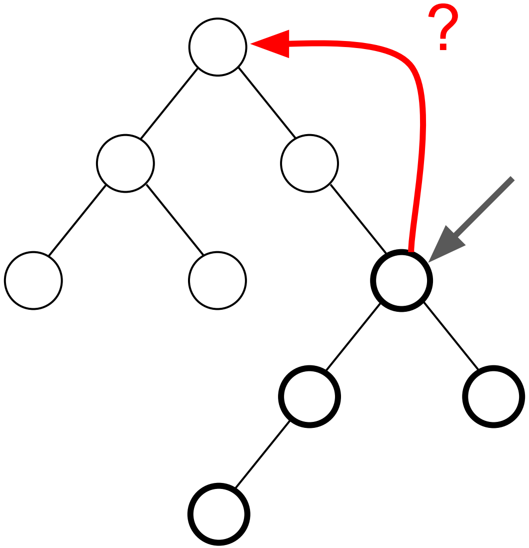
This is some of the oldest code in blink; much of it can be traced to KHTML.

- Monolithic
- Non-encapsulated
- Not reentrant
- Not thread-safe

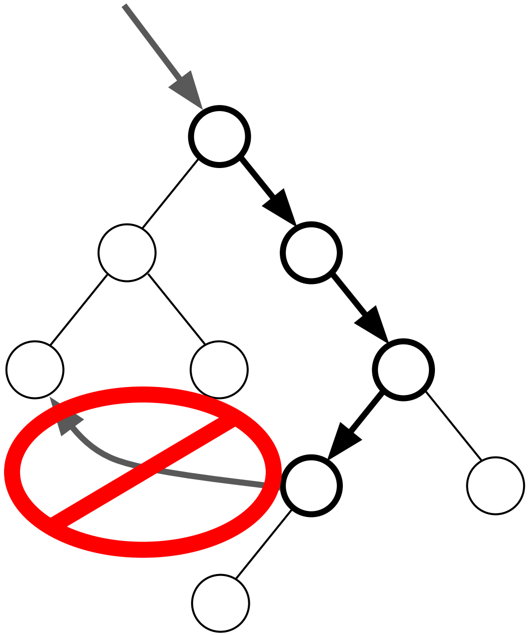
Layout: Monolithic



Layout: Non-encapsulated



Layout: Non-Reentrant



Layout NG and Custom Layout

CSS Custom Layout (aka **Houdini**) side-steps the combinatorial problem.

Layout NG is a ground-up re-architecture of Blink’s layout code.

Layout NG, in a Nutshell

“Constraint-driven” layout, i.e., encapsulation.

Immutable inputs (layout tree) and outputs (fragment tree).

Phase 1 (most of block flow layout) to ship in Q4/Q1.

Layout NG Questions/Comments/Complaints

layout-dev@chromium.org

Rendering. So Hot Right Now.

Big things are happening!

Stay tuned.