

LIFE OF A pixel

Steve Kobes
skobes@chromium.org

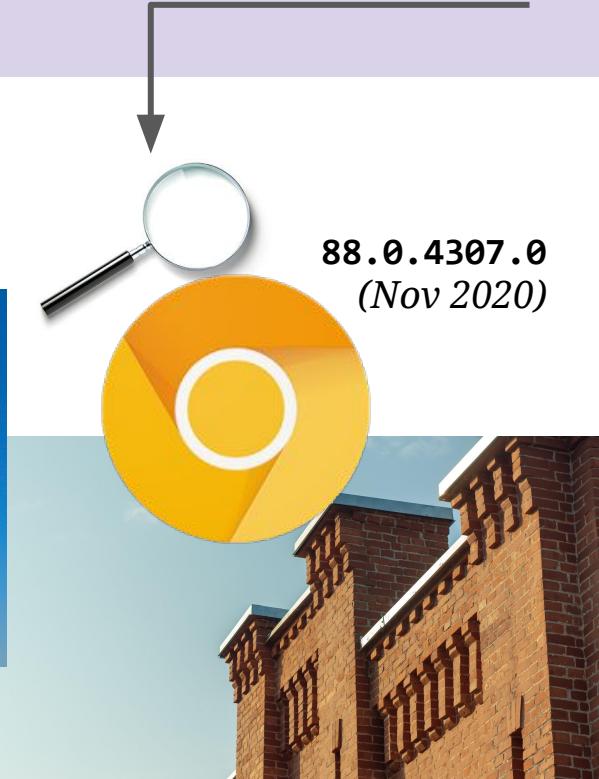
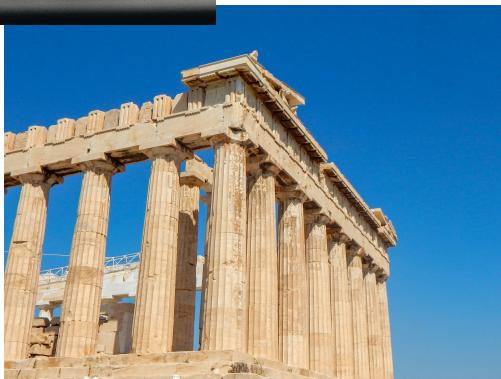
Nov 2020

slides: bit.ly/lifeofapixel

with special thanks to past presenters
dknox@, chrishter@, pdr@

"The unexamined pixel is not worth rendering."
— Socrates

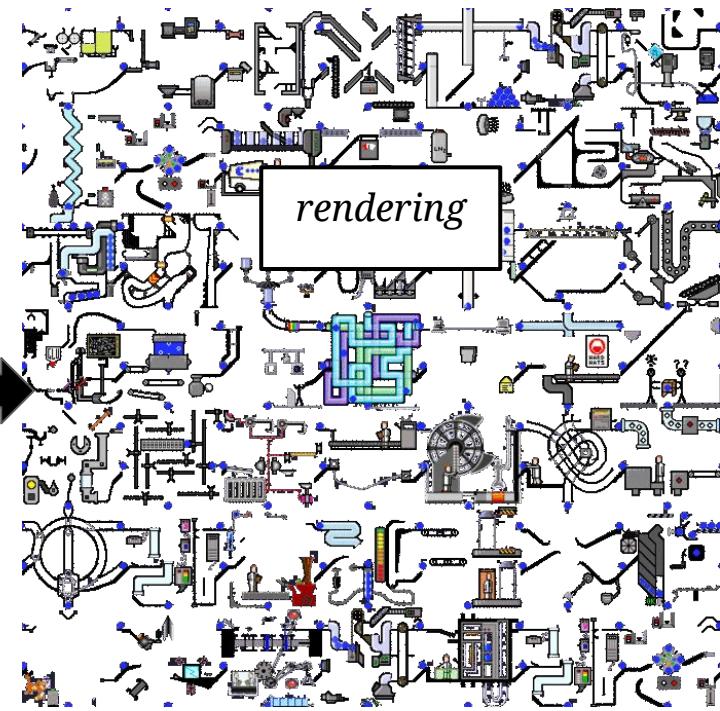
LIFE OF A pixel



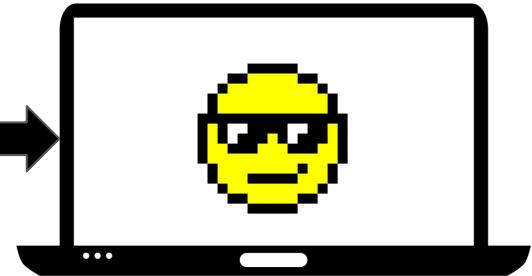
LIFE OF A pixel

content

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="u
    <title>Page name<
    <meta name="descri
    <link href="assets/
    <link rel="shortcut
  </head>
```



pixels



content

A screenshot of a web browser displaying the Wikipedia article for 'Cat'. The page includes the Wikipedia logo, a sidebar with links like 'Main page', 'Contents', and 'Recent changes', and the main content area with text about the domestic cat species. A red box highlights the entire page content.

NOT "content"

content::WebContents

sandboxed renderer process

Blink

CC

content

- **HTML**

(Hyper-Text Markup Language)

`<p> hello </p>`



- **CSS**

(Cascading Style Sheets)

`p { color: red }`

- **JS**

(JavaScript)

`p.innerHTML = "goodbye";`

- **images**

``

(other kinds of rendered content: <video>, <canvas>, WebAssembly, WebGL, WebVR, PDF, ...)

content

<https://www.nytimes.com>

Sections Search

ENGLISH 中文 (CHINESE) ESPAÑOL

Covering stories from up close. Firsthand reporting that brings you the facts.

The New York Times

Sunday, October 8, 2017 Today's Paper Video 77°F Hang Seng -0.21%
From \$15.00 \$9.99 a month.

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Style Food Travel Magazine T Magazine Real Estate ALL

Take a Fit Test
Get your scores in games that challenge your memory, attention, and more.

Memory Matrix 55% Speed Match 75% Train of Thought 30%

lumosity Take Fit Test →

White House to Tie Hard-Line Conditions Into Dreamers' Deal
By MICHAEL D. SHEAR 8:27 PM ET
In exchange for letting the young undocumented immigrants known as Dreamers stay in the U.S., the White House will push for terms that include a crackdown on children fleeing Central America. The deal will also insist on the construction of a wall across the southern border.
256 Comments

Suspected ISIS Members Surrender in Iraq
By ROD NORDLAND and JOHN BICKEL, Photos by Ivor Prickett for The New York Times
Video by CAMILLA SCHUM, ROD NORDLAND and JOHN BICKEL
A Family Battle With Opioids
BY JUDITH CHERKEL
A friend, reflecting on his son's opioid journey, once remarked, "I wish I'd held him closer." Is it ever close enough?

2:35
Fake Assumptions About the Iran Nuclear Deal
Who Invented 'Zero'?
Confessions of a Sensible Gun Owner
N.R.A. and G.O.P. Together
EDITORIAL: Mr. Trump's Attack on Birth Control
Bruni: Be Afraid of Bannon
Douthat: Pigs of Liberalism

Reviews for America
By JONATHAN LIEB
I used to give this place five stars! But since the change of management, it's going downhill.

More Than 1,000 ISIS Fighters Surrender
The militants turned themselves in after the latest in a string of humiliating defeats in Iraq and Syria. Many claim not to have been soldiers, but only cooks or clerks.

256 Comments

Trump Risks Inciting World War III', G.O.P. Senator Says
By JONATHAN MAZER 8:27 AM ET
In a remarkable rebuke of his party's president, Bob Corker, the influential head of the Senate Foreign Relations

Listen to 'The New Washington'
Senator Marco Rubio spoke with Carl Hulse about President Trump, the recovery efforts in Puerto Rico, U.S.-Cuban relations and more.
Follow The New Washington: Apple Podcasts Radio Public Stitcher Once Racing to Flee the Senate, Rubio Now Digs In

TIMES INSIDER » What's a Science Reporter to Do When Sound Evidence Isn't Sound?
THE CROSSWORD » Play Today's Puzzle

```
</div>
<div class="column">
  <div class="collection crosswords-collection">
    <article class="story">
      <h3 class="kicker">
        <a href="http://wordplay.blogs.nytimes.com">Wordplay
      </h3>
    </article>
  </div><!-- close column -->
</div><!-- close layout -->
</div><!-- close crossword-subscription -->
</div><!-- close collection -->
</div>
</div><!-- close user-subscriptions -->
<div id="HMPmiddle" class="ad hpmiddle-ad nocontent robots-nocontent"></div>
<div class="region c-column-middle-span-region">
  <div class="collection">
    <link rel="stylesheet" type="text/css" href="https://int.nyt.com/app
<style type="text/css">
/*HIDE WATCHING HEADER*/
.portal-container>header { display: none }
</style>
<div id="nytint-hp-watching">
  <div class="portal-container">
    <header class="portal-header">
      <h4>Watching</h4>
    </header>
    <div class="portal-posts-frame expanded"></div>
    <footer class="portal-footer"></footer>
  </div>
</div>
<script type="text/javascript">
require(['foundation/main'], function() {
  require('homepage/main', 'https://int.nyt.com/apps/portals/assets/portal/app
}, function() {
  require(['portal/app'], function(Portal) {
    var opts = {
      env: 'production_published',
      matchHeight: {
        match: ':span-ab-layout.layout > .ab-column',
        container: '.c-column.column',
        maxHeight: 2000
      }
    }
    if (window.location.search.indexOf('portal_variant=watchingNoScroll') !=-
      opts.variation = 'simple';
      opts.poll = false;
      opts.limit = 20;
    }

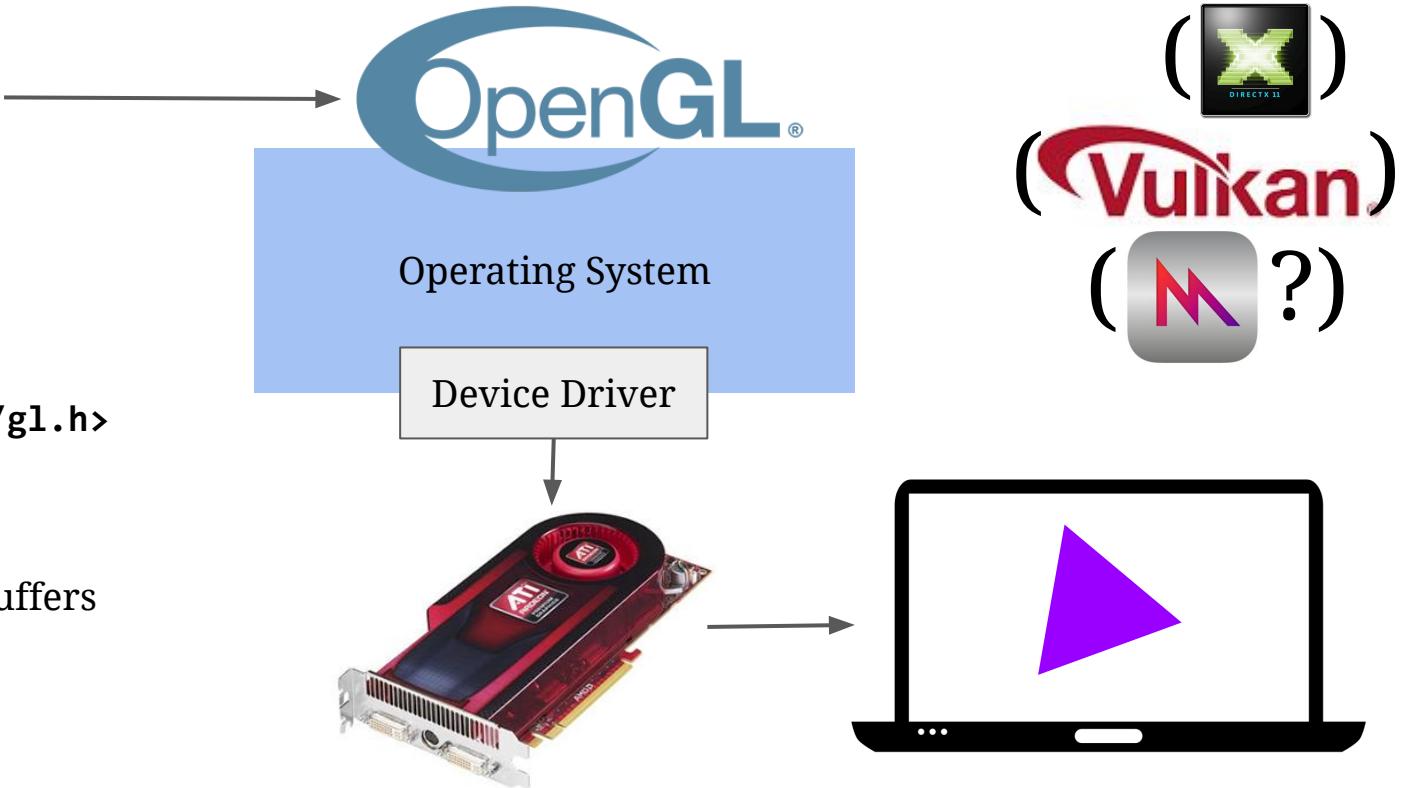
    var watching = Portal.create('#nytint-hp-watching', opts);
  });
});
```

pixels



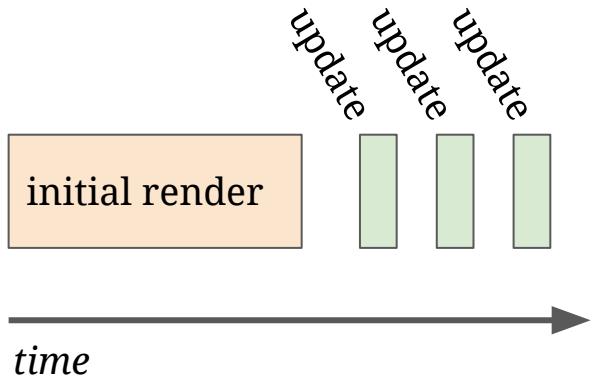
```
#include <GL/gl.h>
```

- textures
- shaders
- vertex buffers
- ...



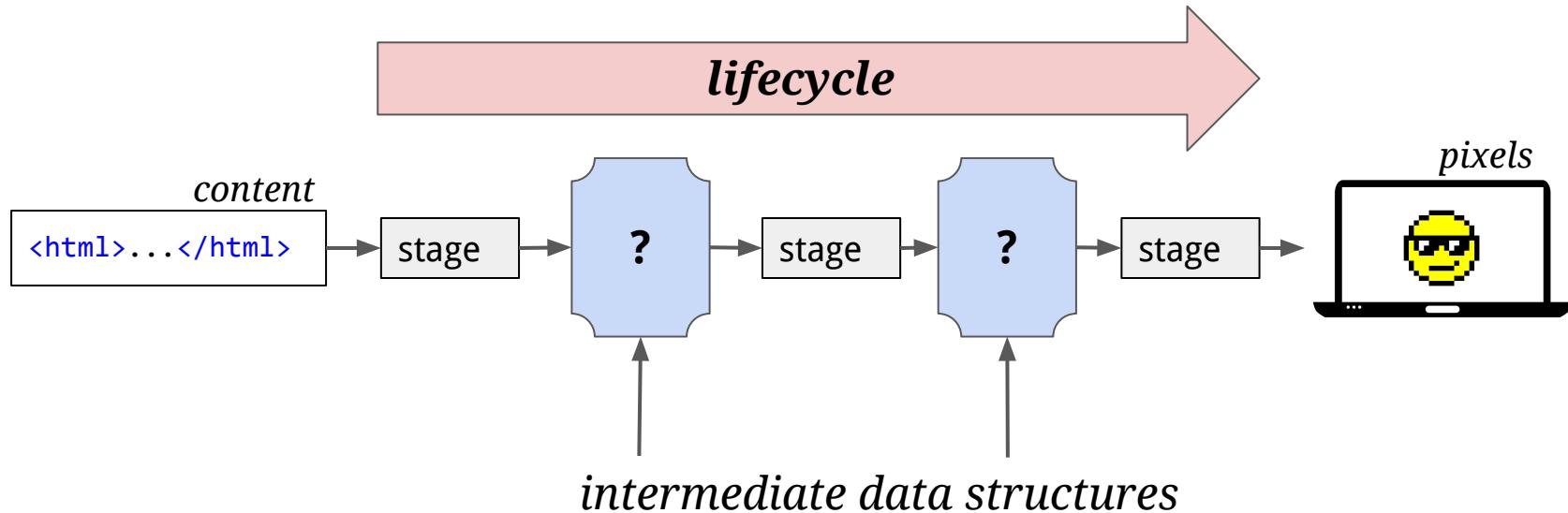
goals

1. *render content into pixels*
2. *build data structures to enable **updating** the rendering efficiently*

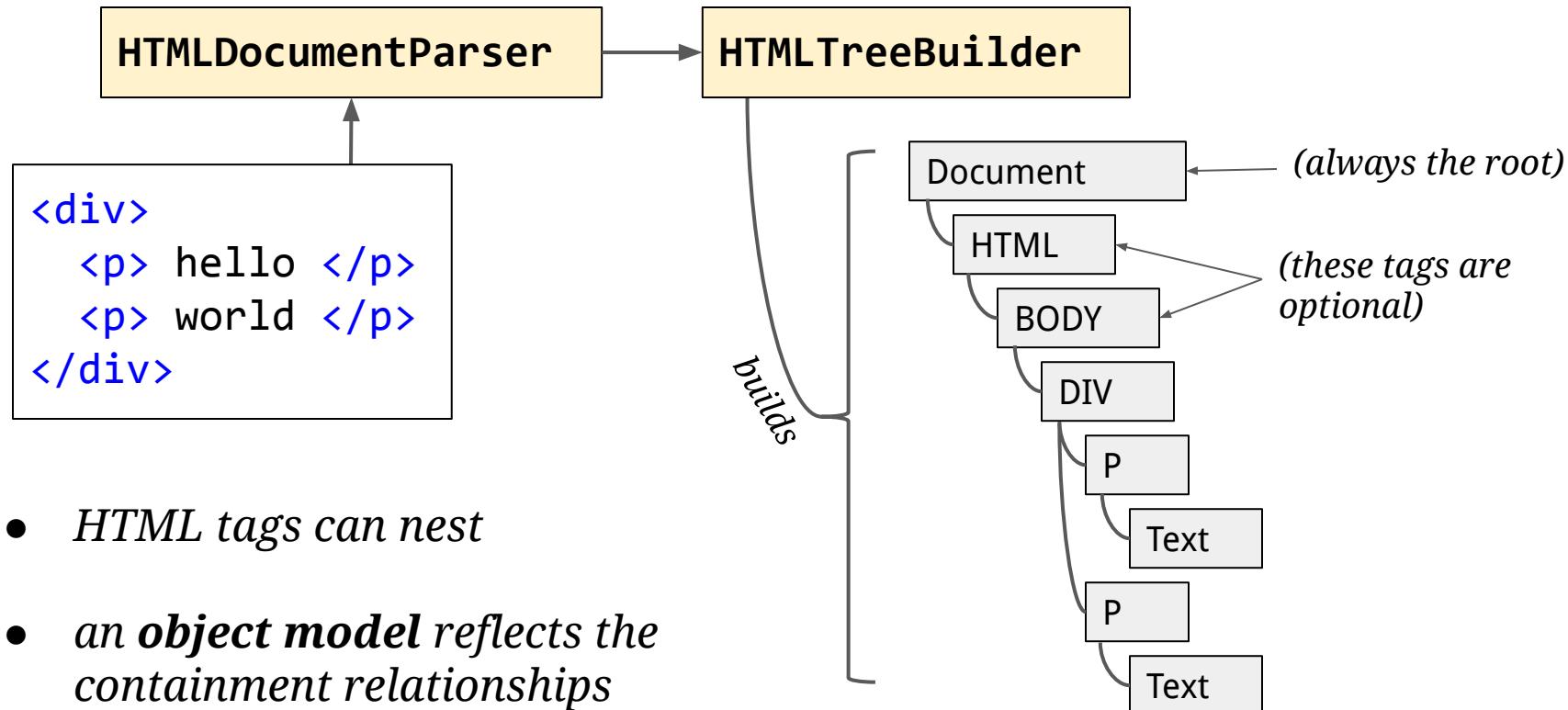


- JavaScript
- user input
- asynchronous loading
- animations
- scrolling
- zooming

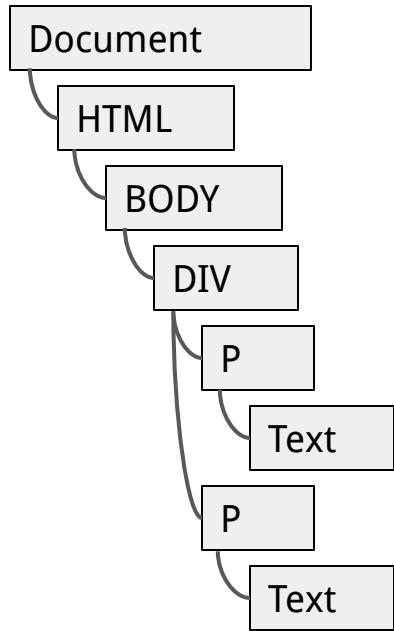
stages



parsing



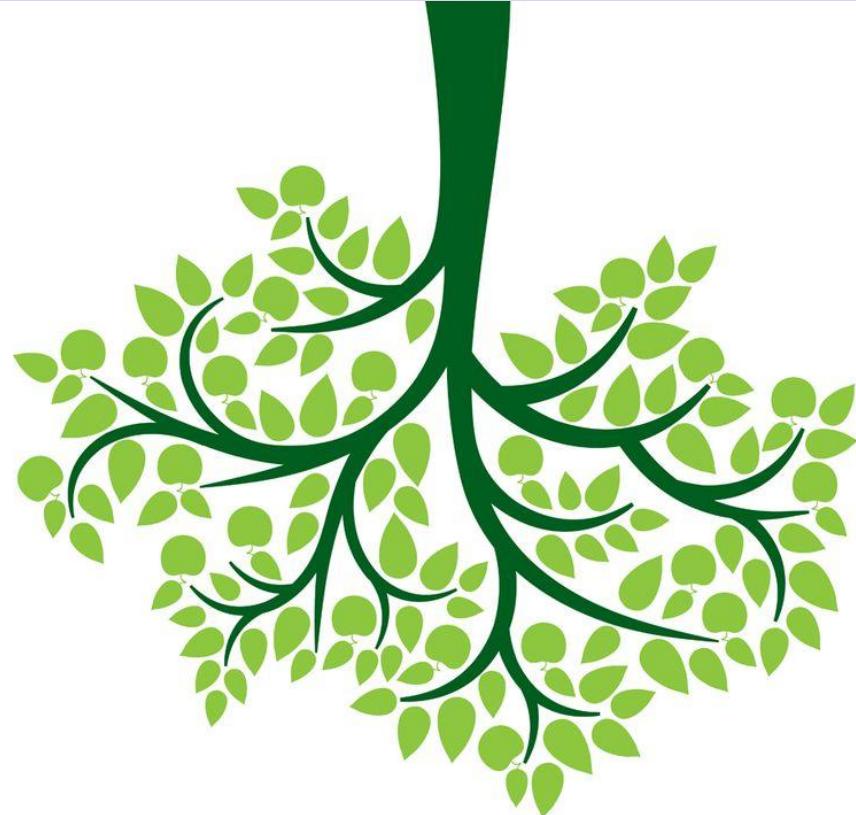
DOM



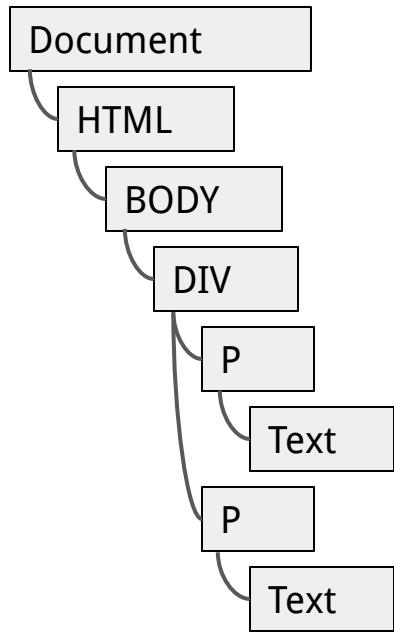
*This is the
Document
Object
Model.*

*The DOM
is a tree!*

- *parents*
- *children*
- *siblings*



DOM

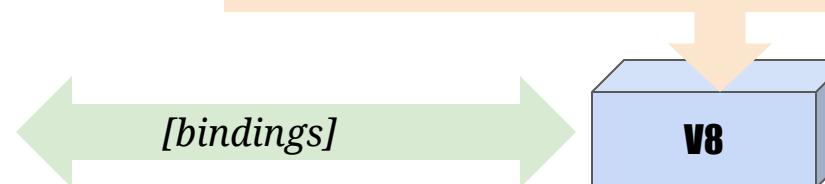


The DOM is both

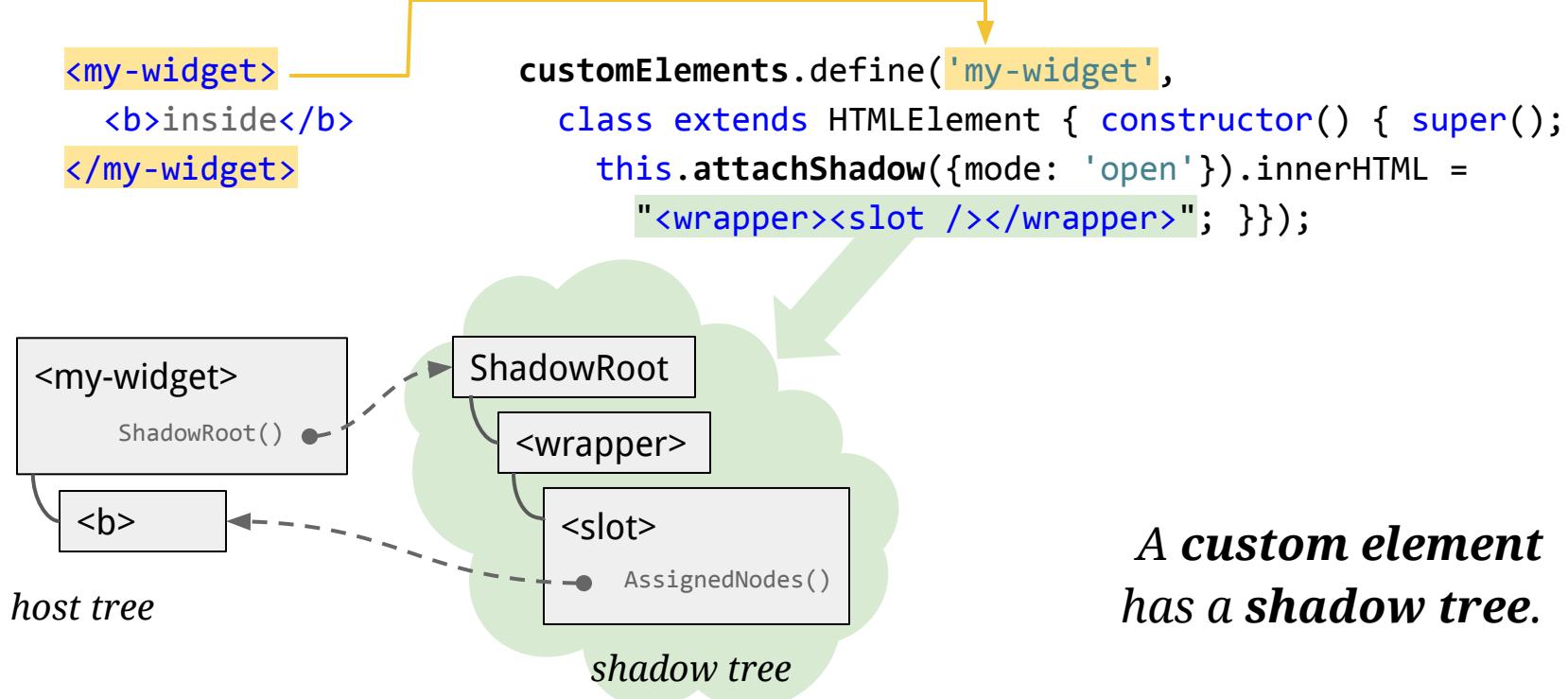
- *Chrome's internal representation*
AND
- *the API exposed to JavaScript*

```
var div = document.body.firstChild;
var p2 = div.childNodes[1];
p2.appendChild(document.createElement("span"));
p2.foo = {arbitrary: "state"};
```

[JS]

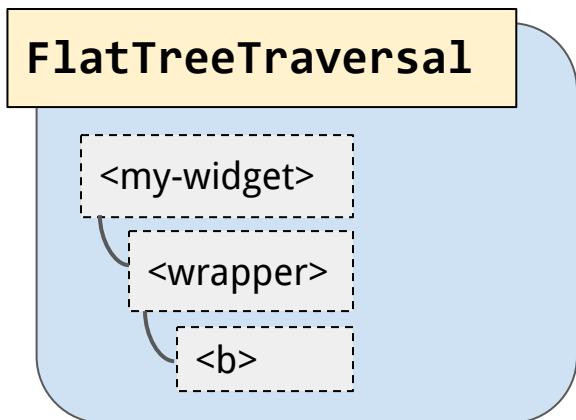
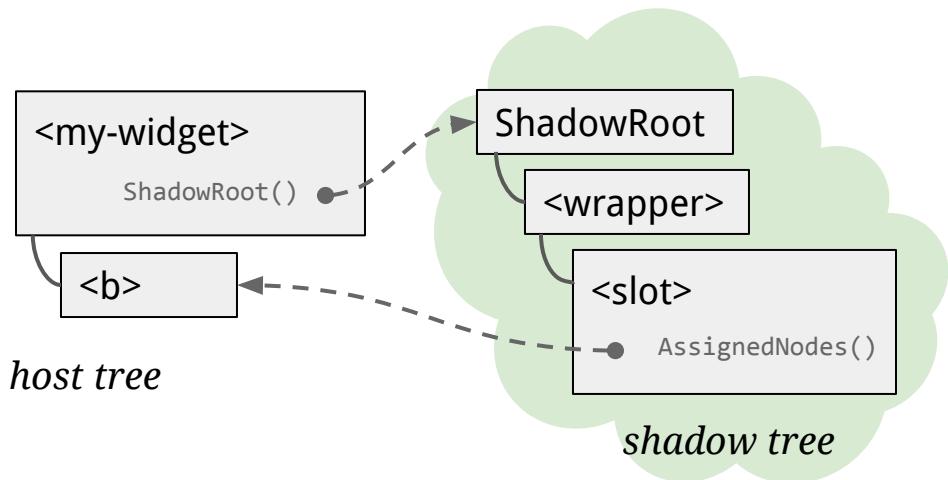


DOM

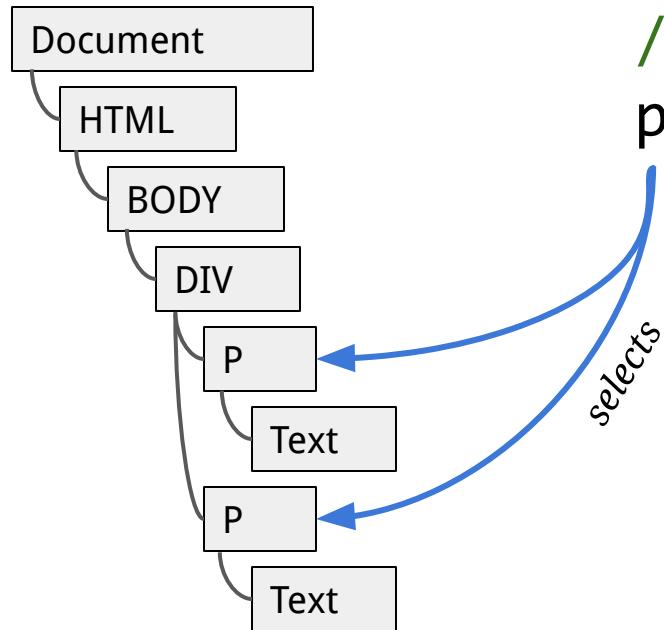


DOM

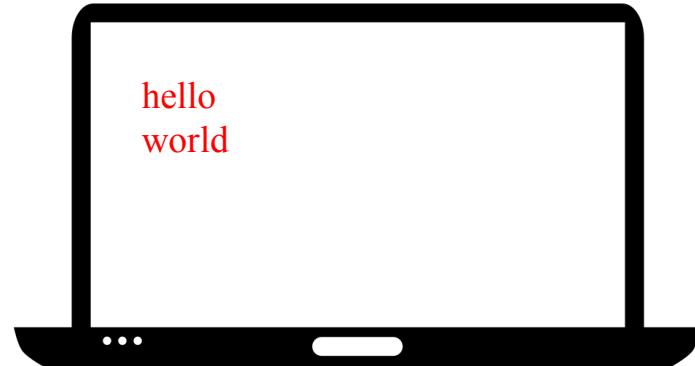
A *flat tree traversal* presents a composed view.



style



```
/* every <p> has red text */  
p { color: red }
```



style

font-weight: bold;

hello

margin-left: 2cm;

hello

outline: dashed blue;

hello

transform: rotate(20deg);

hello

background: url(kitten.jpg);



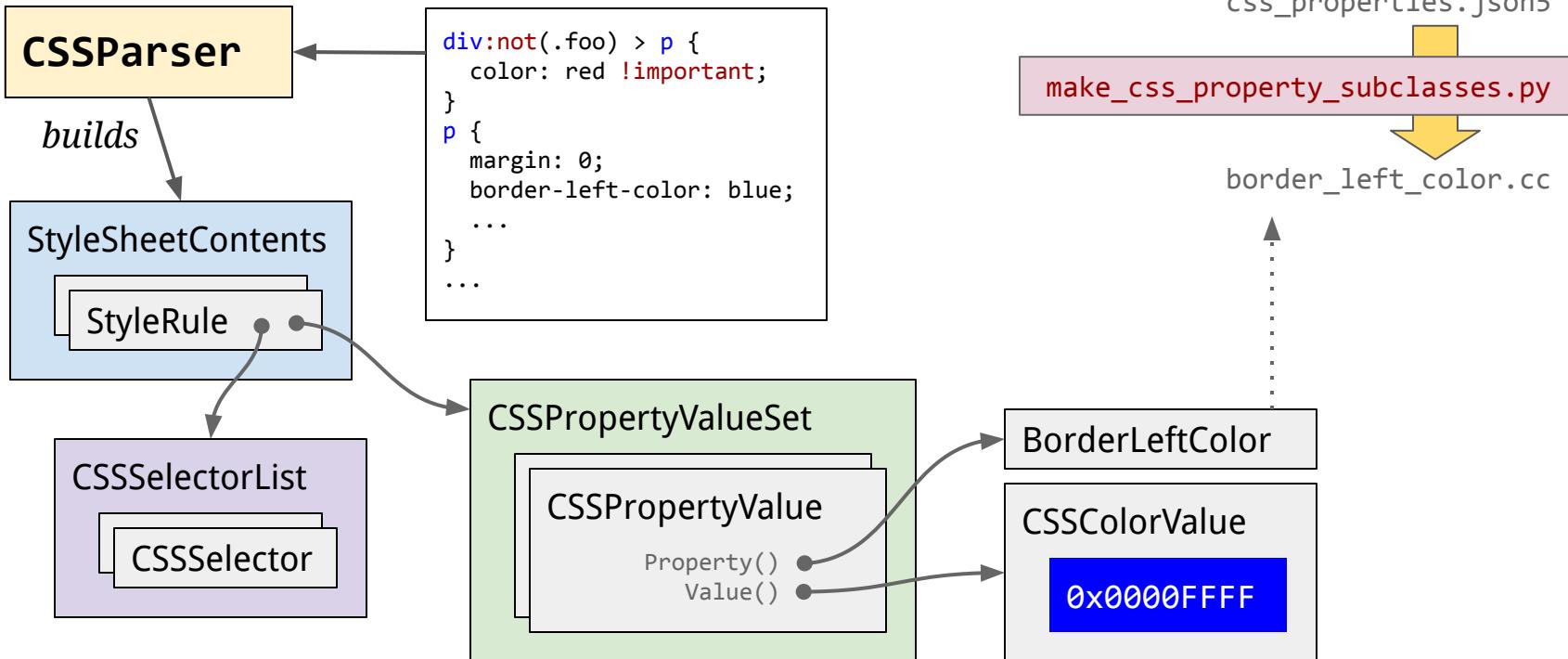
style

```
/* every other <p> in any <div> without class="foo" */
div:not(.foo) > p:nth-of-type(2n) {
    color: red !important;
}

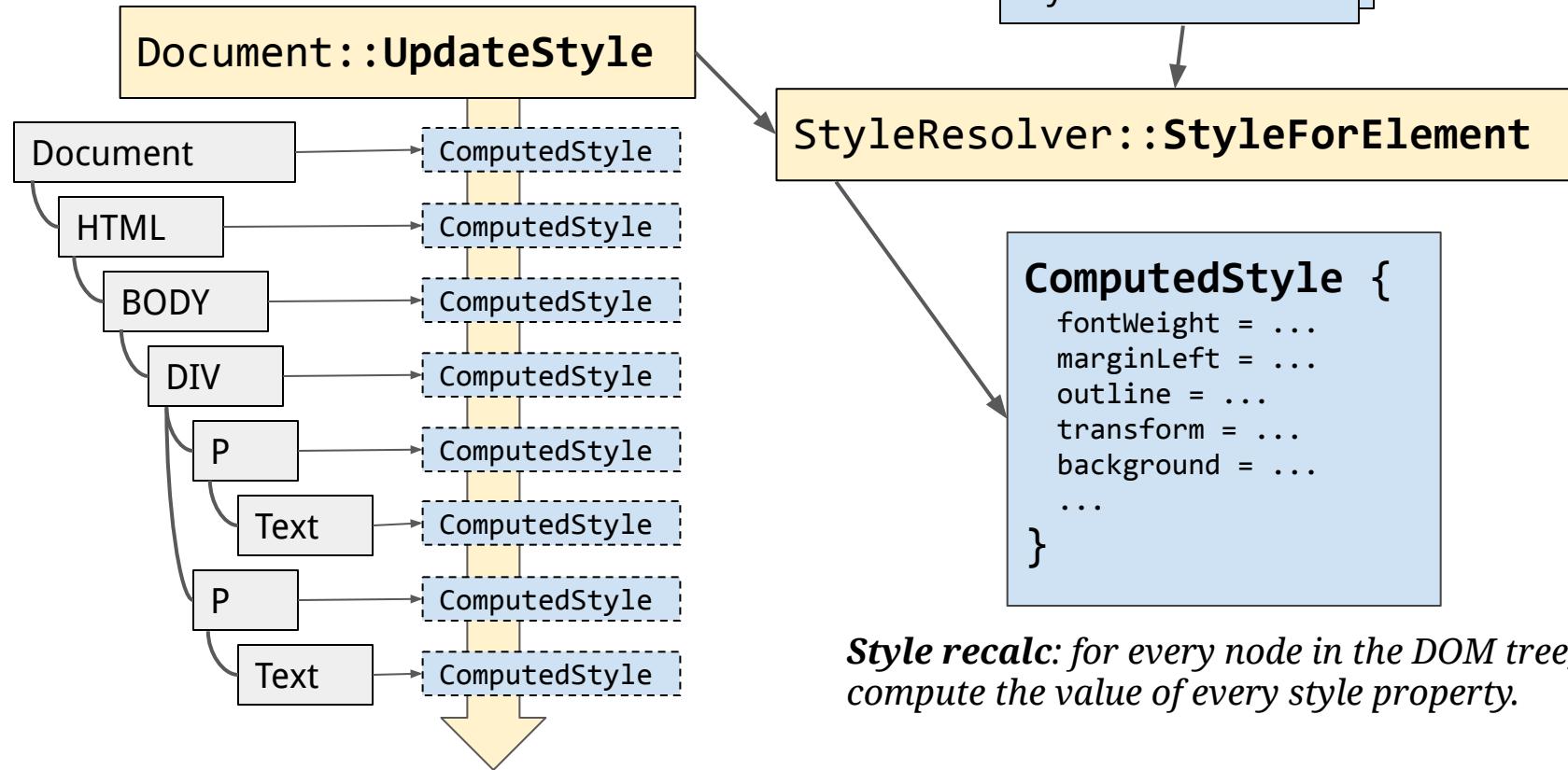
p {
    color: blue;
}
```

- *selectors can be complex!*
- *declarations may conflict!*

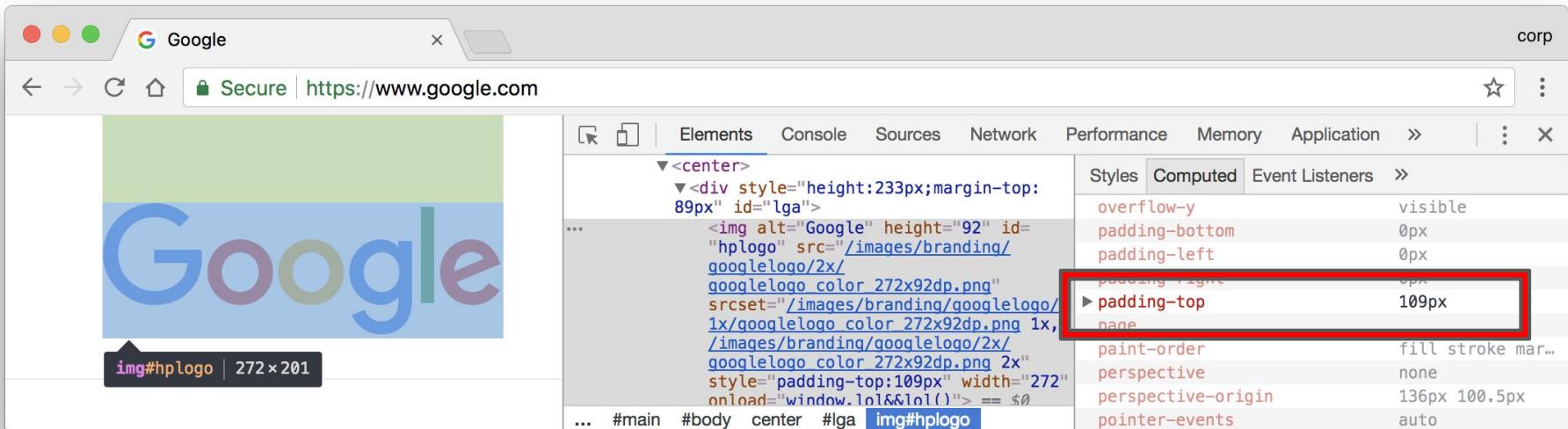
style



style



style



The screenshot shows a Mac OS X desktop with a web browser window for Google. The browser title bar says "Google". The address bar shows "Secure | https://www.google.com". The main content area displays the Google logo. In the bottom-left corner of the logo, there is a tooltip-like box with the text "img#hplogo | 272 x 201". To the right of the logo, the developer tools are open. The "Elements" tab is selected. A specific "img#hplogo" element is highlighted. In the "Computed" tab of the styles panel, the "padding-top" property is listed with a value of "109px", which is highlighted with a red box.

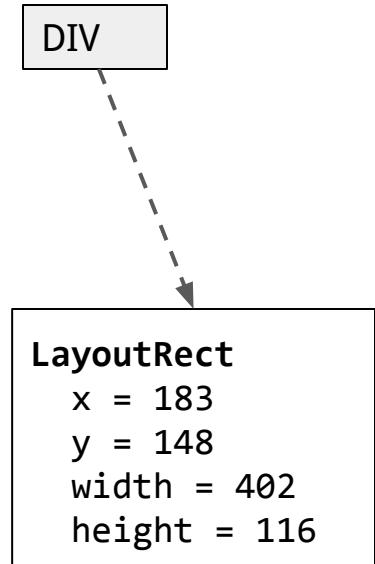
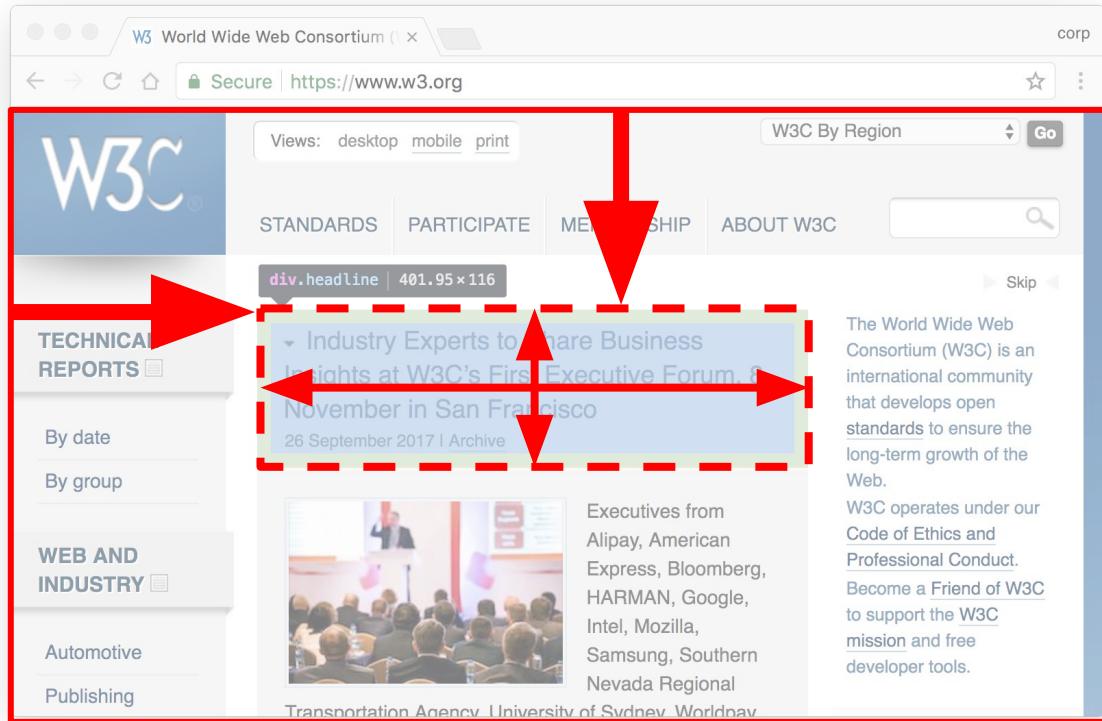
Style	Value
padding-top	109px

```
getComputedStyle(element)["padding-top"]
```

[JS]

(But: `getComputedStyle` returns some layout data also.)

layout



layout

A screenshot of a web browser window displaying a page from Gutenberg.org. The page contains several paragraphs of text, each enclosed in a separate rectangular box, illustrating a block layout structure. The browser interface includes standard controls like back, forward, and search, along with a URL bar showing the secure connection.

The last burst carried the mail to the summit of the hill. The horses stopped to breathe again, and the guard got down to skid the wheel for the descent, and open the coach-door to let the passengers in.

"Tst! Joe!" cried the coachman in a warning voice, looking down from his box.

"What do you say, Tom?"

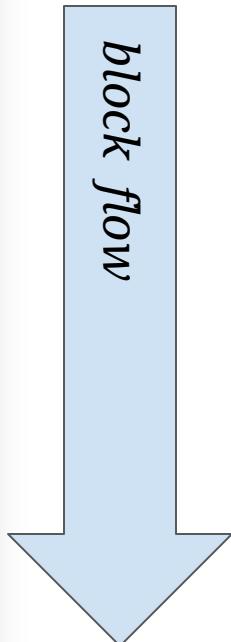
They both listened.

"I say a horse at a canter coming up, Joe."

"I say a horse at a gallop, Tom," returned the guard, leaving his hold of the door, and mounting nimbly to his place. "Gentlemen! In the king's name, all of you!"

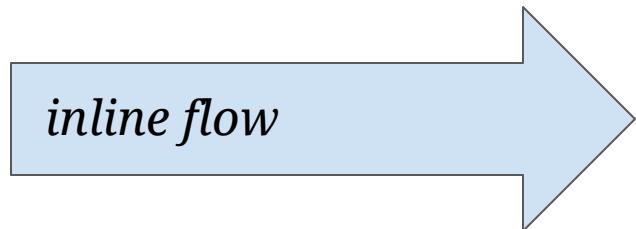
With this hurried adjuration, he cocked his blunderbuss, and stood on the offensive.

The passenger booked by this history, was on the coach-step, getting in; the two other passengers were close behind him, and about to follow. He remained on the step, half in the coach and half out of; they remained in the road below him. They all looked from the coachman to the guard, and from the guard to the coachman, and listened. The coachman looked back and the guard looked back, and even the emphatic leader pricked up his ears and looked back, without contradicting.



Simple "block" layout objects are placed one after another, flowing down the page.

layout



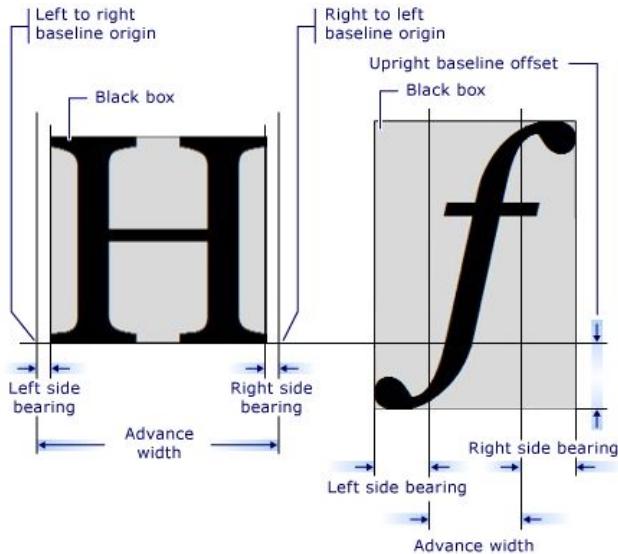
Text and "inline" elements flow left-to-right, and are broken into lines.



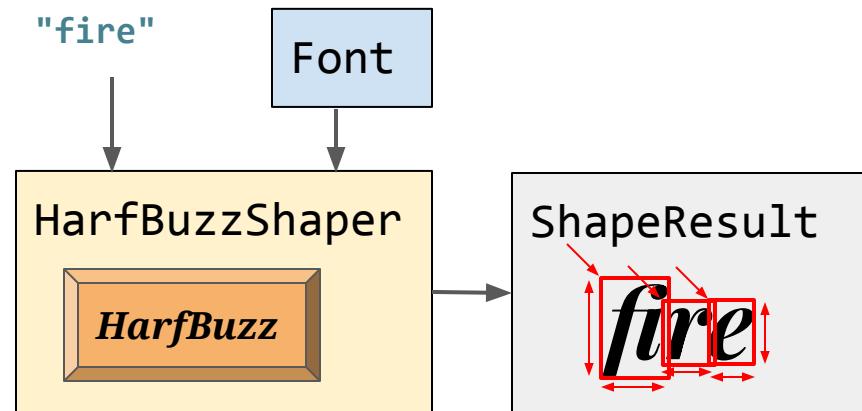
العربية ←
עברית ←

(Some languages flow right-to-left.)

layout

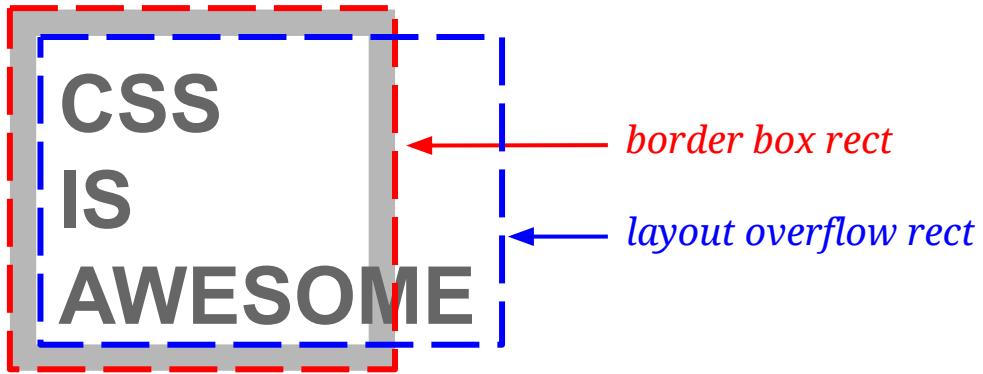


Layout measures runs of text in fonts.



Shaping selects the glyphs and computes their placement.

layout



*Overflow can be visible,
hidden, or scrollable.*

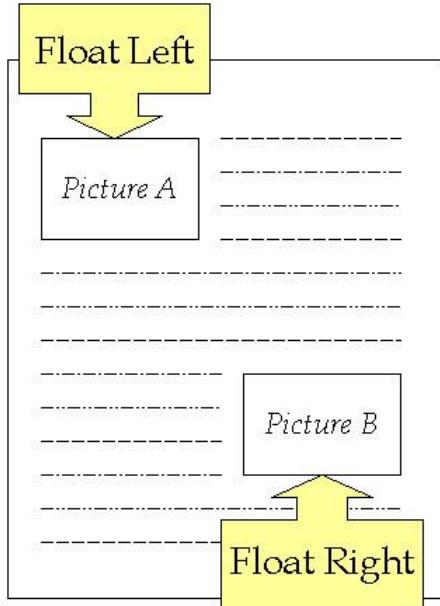
*The contents of a layout object
can **overflow** its border box.*

A screenshot of a web browser window showing a yellow-highlighted text block. The text is a long Latin placeholder (Lorem ipsum). A horizontal scrollbar is visible at the bottom of the block, indicating that the content exceeds the available vertical space and is therefore scrollable. The text block has a thin black border.

*...
Lorem ipsum dolor sit amet, Praesent tristique feugiat ex neque nec gravida molestie, maximus blandit lectus turpis posuere arcu id risus voluptate purus eu rhoncus maximus, i aliquet dui orci id erat. Phas tincidunt libero ultrices ac. 1 malesuada eleifend a eu nunc et vehicula. Sed a iaculis justus
...
purus facilisis risus, in laoreet ante est a quam. Aenean odio eros, pulvinar eu tristique id, suscipit et est. Proin interdum vel lorem ac pretium. Quisque pulvinar eleifend tellus, vitae fringilla leo varius vitae. Nullam a ligula viverra, egestas purus id, molestie nulla. Sed pulvinar aliquet orci vitae molestie.*

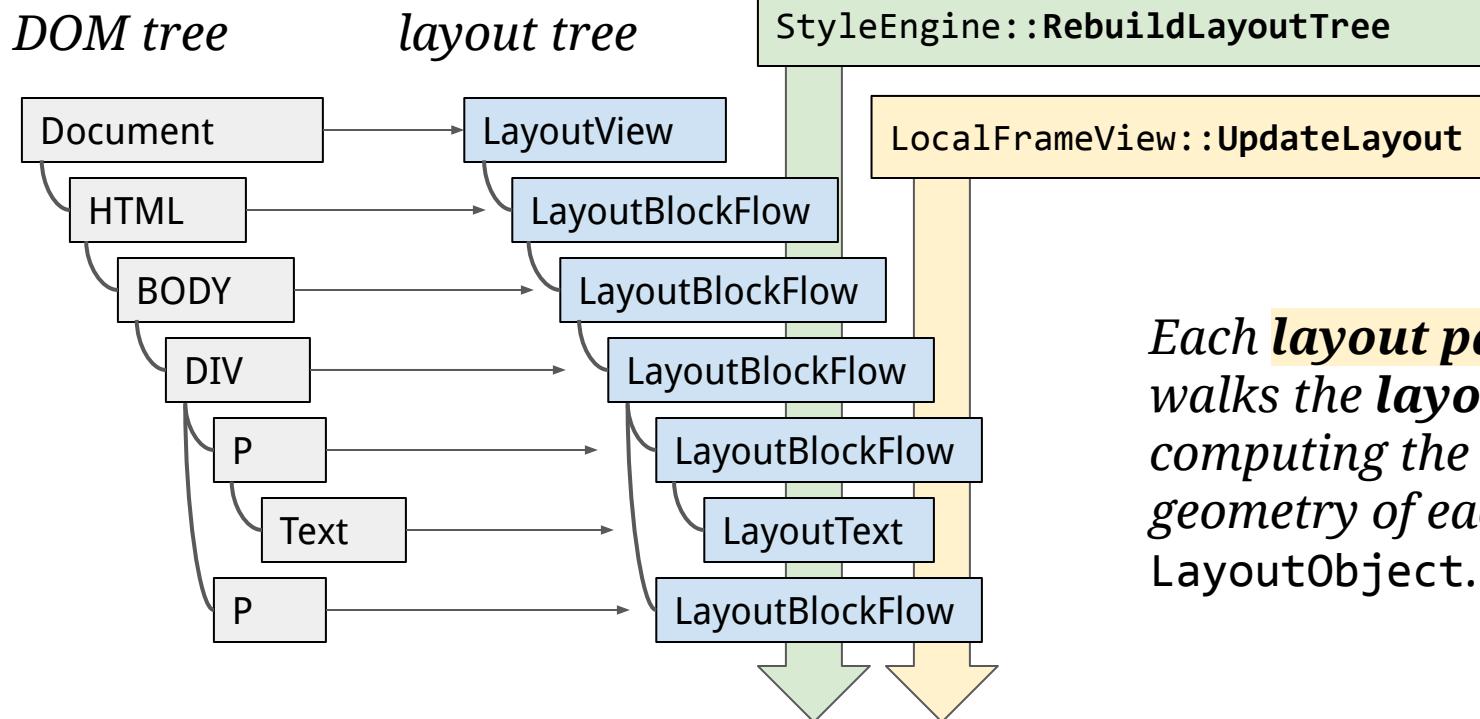
layout

Other kinds of layout are even more complex.



- `<table>`
- `float: left`
- `column-count: 3`
- `display: flex`
- `writing-mode: vertical-lr`
- ...

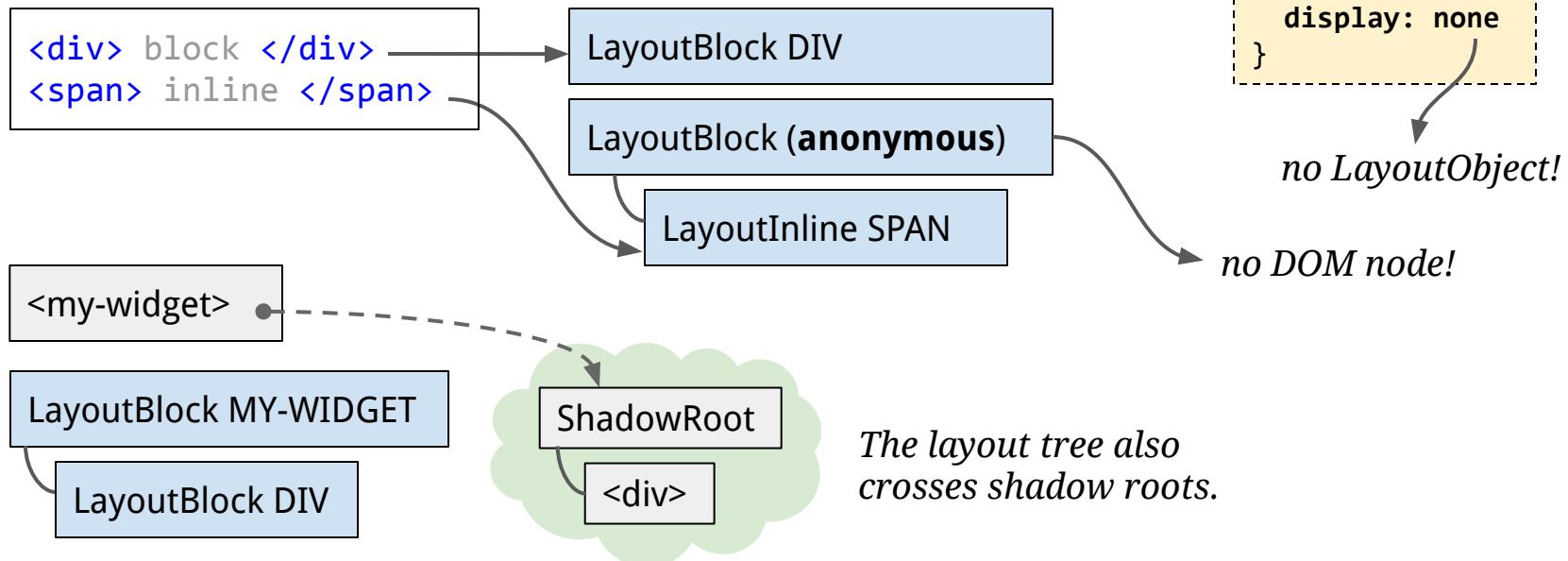
layout



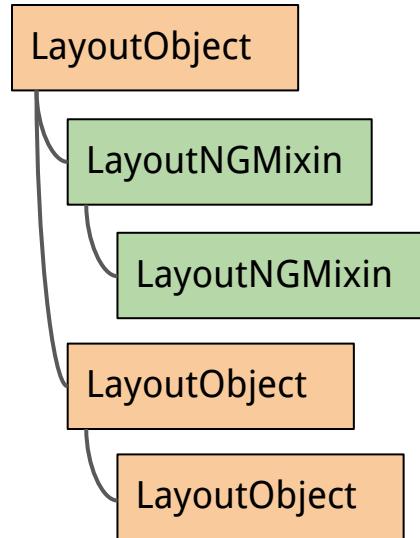
Each **layout pass** walks the **layout tree**, computing the visual geometry of each **LayoutObject**.

layout

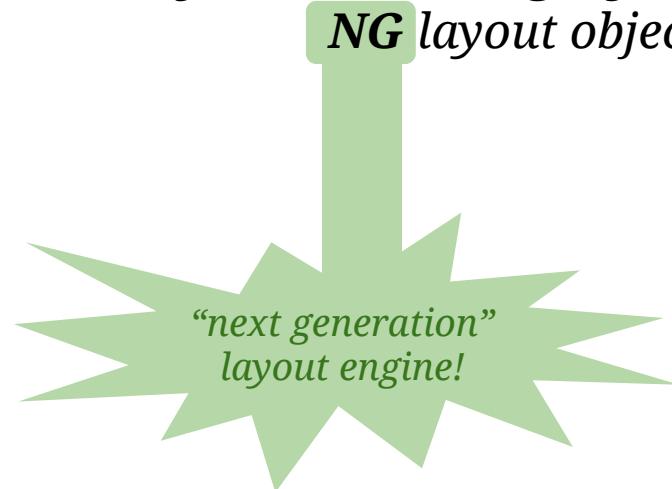
DOM nodes are not 1:1 with layout objects.



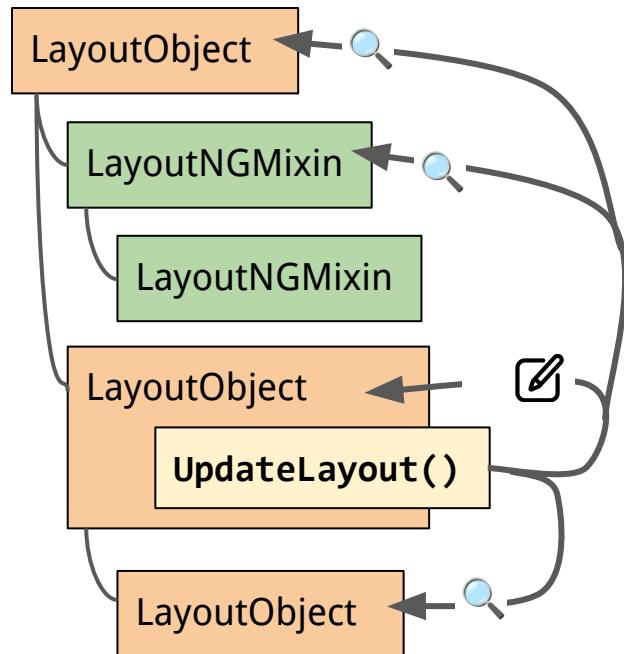
layout



*The layout tree has **legacy** and **NG** layout objects.*



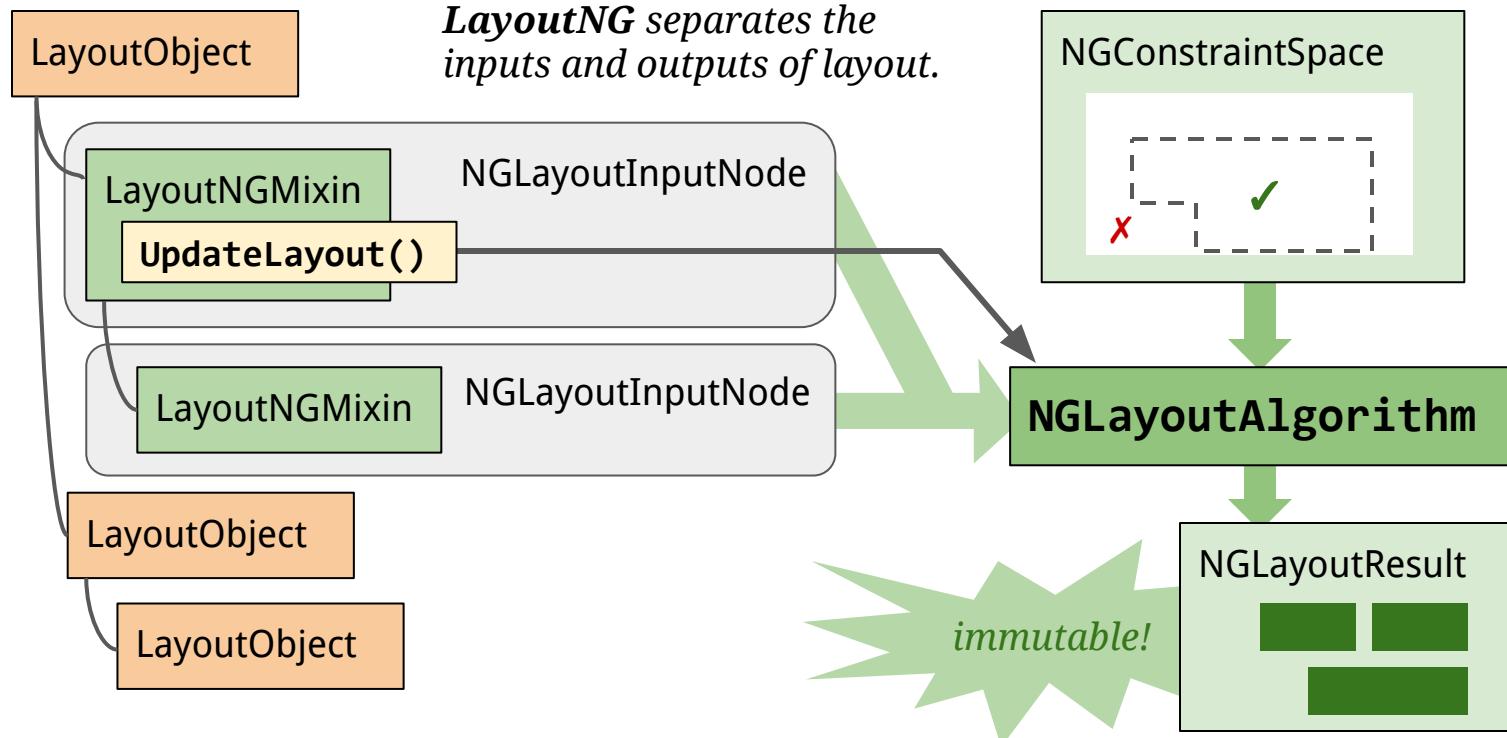
layout



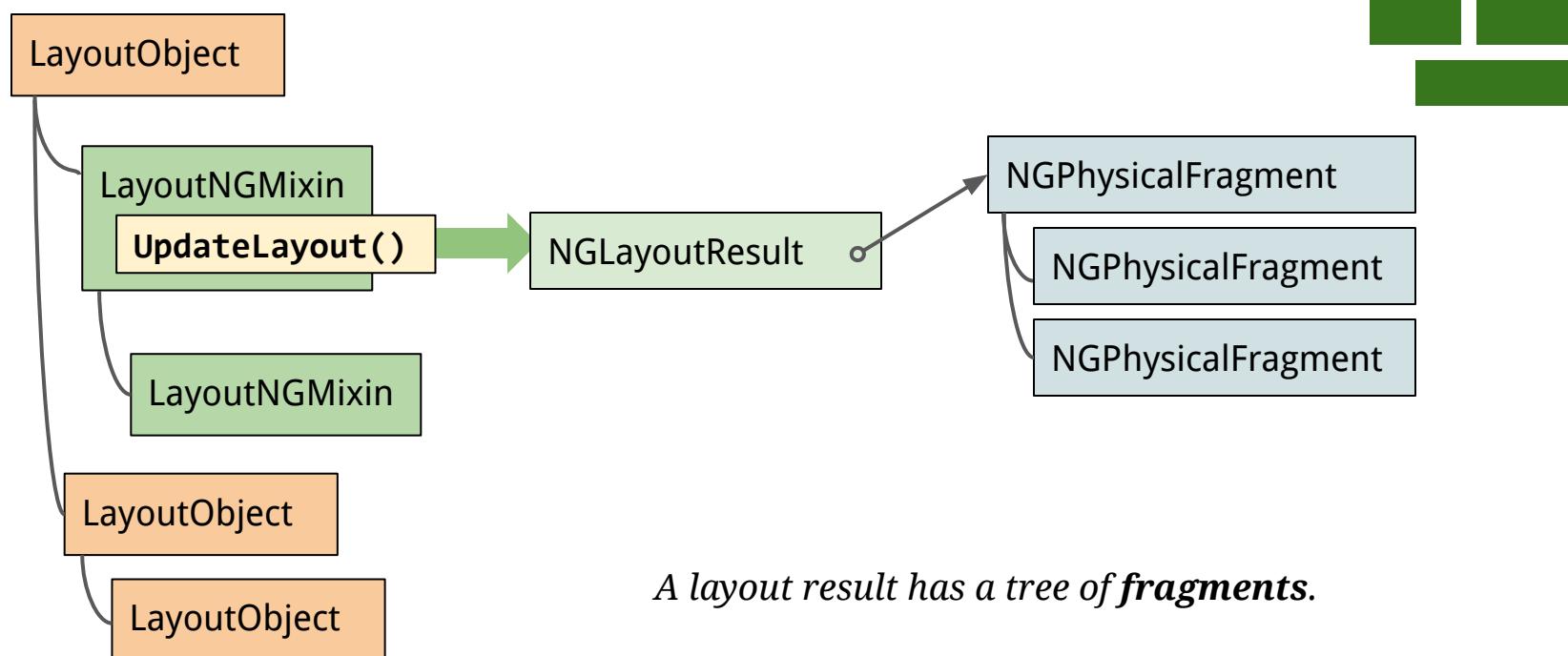
A **legacy layout object** holds inputs, outputs, and algorithms of layout.

It can see the state of the whole tree.

layout



layout



layout — example

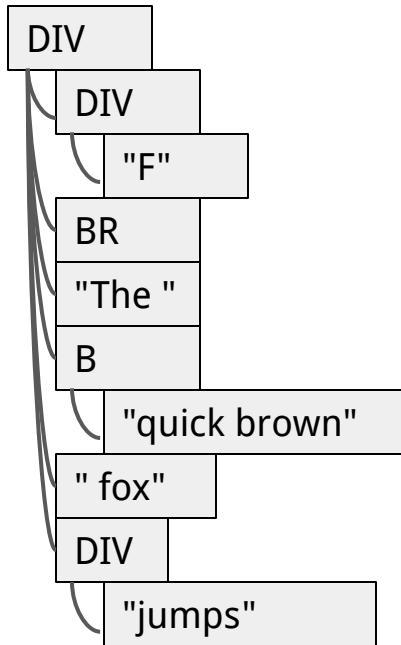
```
<div style="max-width: 100px">
    <div style="float: left; padding: 1ex">F</div>
    <br>The <b>quick brown</b> fox
    <div style="margin: -60px 0 0 80px">jumps</div>
</div>
```

layout — example

```
<div style="max-width: 100px">
    <div style="float: left; padding: 1ex">F</div>
    <br>The <b>quick brown</b> fox
    <div style="margin: -60px 0 0 80px">jumps</div>
</div>
```

F jumps
The **quick**
brown fox

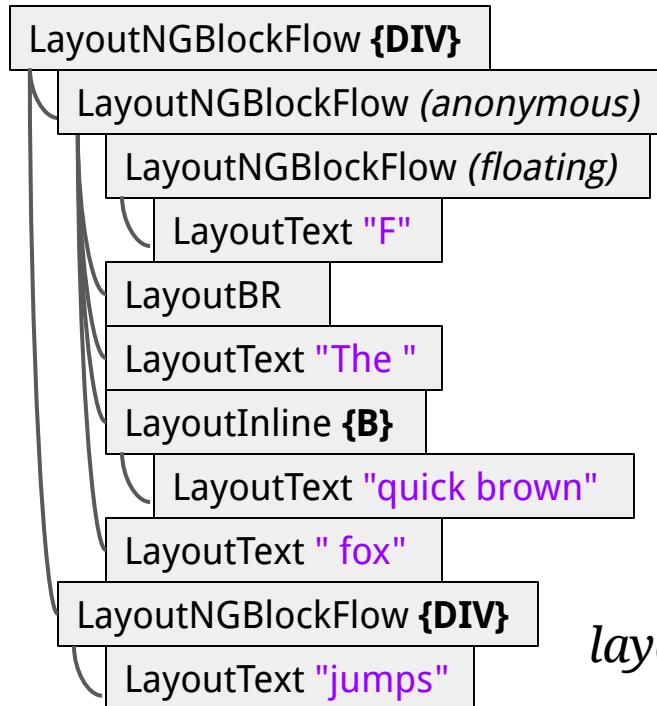
layout — example



```
<div style="max-width: 100px">
  <div style="float: left; padding: 1ex">F</div>
  <br>The <b>quick brown</b> fox
  <div style="margin: -60px 0 0 80px">jumps</div>
</div>
```

F jumps
The **quick**
brown fox

layout — example



layout tree

```
<div style="max-width: 100px">
  <div style="float: left; padding: 1ex">F</div>
  <br>The <b>quick brown</b> fox
  <div style="margin: -60px 0 0 80px">jumps</div>
</div>
```

F jumps
The **quick**
brown fox

layout — example

```
Box (block-flow) at 0,0 100x12
  Box (block-flow children-inline) at 0,0 100x54
    LineBox at 24.9,0 0x18
      Box (floating block-flow children-inline) at 0,0 24.9x34
        LineBox at 8,8 8.9x18
          Text 'F' at 8,8 8.9x17
          Text '\n' at 24.9,0 0x17
        LineBox at 24.9,18 67.1x18
          Text 'The ' at 24.9,18 28.9x17
          Text 'quick' at 53.8,18 38.25x17
        LineBox at 0,36 69.5x18
          Text 'brown' at 0,36 44.2x17
          Text ' fox' at 44.2,36 25.3x17
  Box (block-flow children-inline) at 80,-6 20x18
    LineBox at 0,0 39.125x18
      Text 'jumps' at 0,0 39.125x17
```

fragment tree

F jumps
The quick
brown fox

layout — example

```
Box (block-flow) at 0,0 100x12
  Box (block-flow children-inline) at 0,0 100x54
    LineBox at 24.9,0 0x18
      Box (floating block-flow children-inline) at 0,0 24.9x34
        LineBox at 8,8 8.9x18
          Text 'F' at 8,8 8.9x17
          Text '\n' at 24.9,0 0x17
        LineBox at 24.9,18 67.1x18
          Text 'The ' at 24.9,18 28.9x17
            Text 'quick' at 53.8,18 38.25x17
        LineBox at 0,36 69.5x18
          Text 'brown' at 0,36 44.2x17
          Text ' fox' at 44.2,36 25.3x17
  Box (block-flow children-inline) at 80,-6 20x18
    LineBox at 0,0 39.125x18
      Text 'jumps' at 0,0 39.125x17
```

fragment tree

The diagram illustrates the fragment tree for the text "The quick brown fox jumps". A red dot at the top left represents the root node. A red arrow points from this root to the letter "F" in "The". Another red arrow points from the word "The" to the word "quick". The word "quick" is enclosed in a red rectangle. Below "quick", the words "brown fox" are shown together, with "brown" preceding "fox". The word "jumps" is positioned to the right of the word "fox".

jumps
The **quick**
brown fox

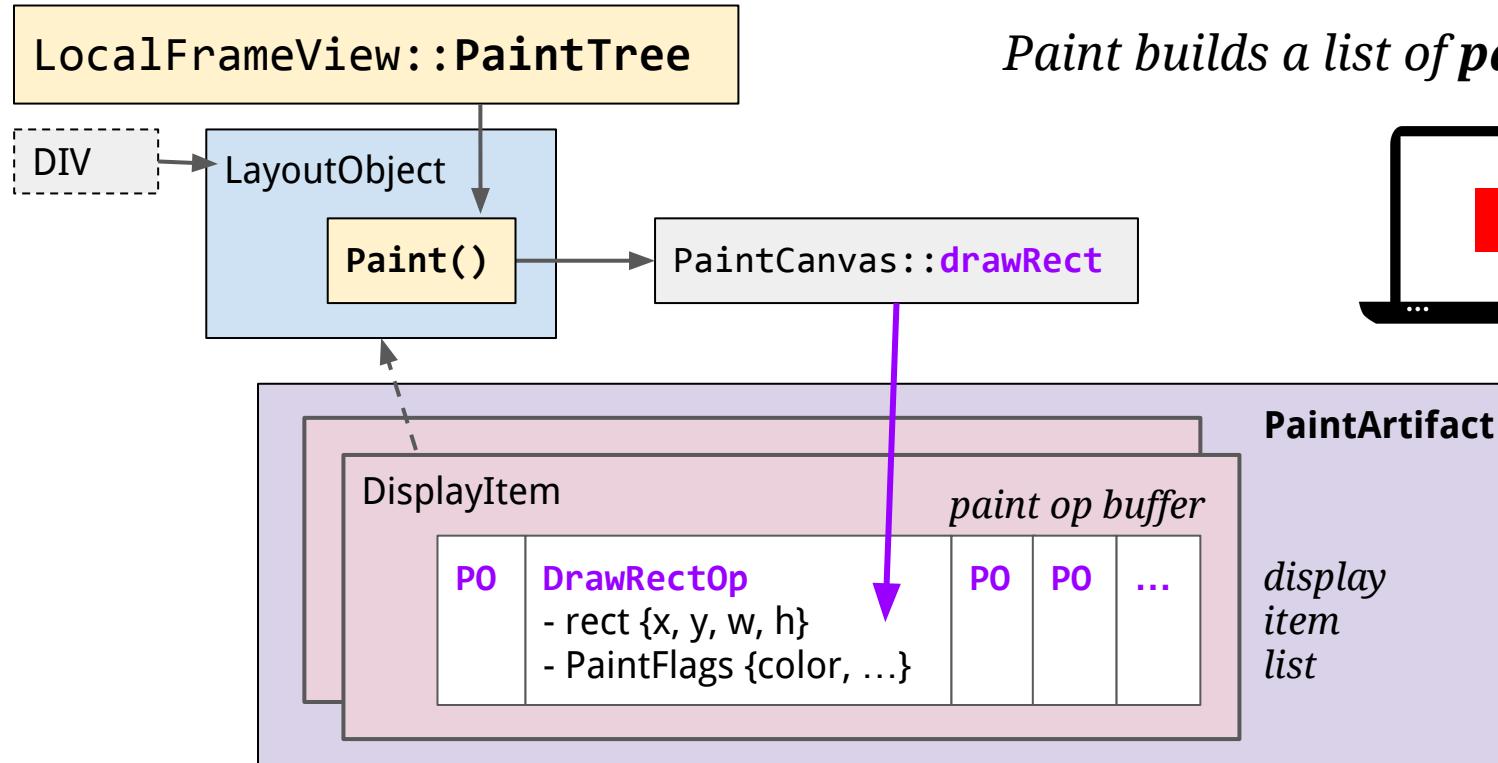
layout — example

```
Box (block-flow) at 0,0 100x12
  Box (block-flow children-inline) at 0,0 100x54
    LineBox at 24.9,0 0x18
      Box (floating block-flow children-inline) at 0,0 24.9x34
        LineBox at 8,8 8.9x18
          Text 'F' at 8,8 8.9x17
          Text '\n' at 24.9,0 0x17
        LineBox at 24.9,18 67.1x18
          Text 'The ' at 24.9,18 28.9x17
          Text 'quick' at 53.8,18 38.25x17
        LineBox at 0,36 69.5x18
          Text 'brown' at 0,36 44.2x17
          Text ' fox' at 44.2,36 25.3x17
  Box (block-flow children-inline) at 80,-6 20x18
    LineBox at 0,0 39.125x18
      Text 'jumps' at 0,0 39.125x17
```

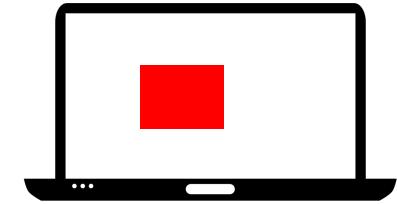
fragment tree

F → jumps
The quick
brown fox

paint



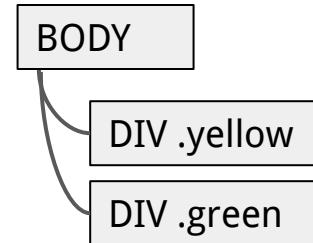
*Paint builds a list of **paint ops**.*



paint

*Paint uses **stacking order**,
not DOM order.*

```
<div class="yellow"></div>
<div class="green"></div>
<style>
  .yellow { z-index: 2; ... }
  .green  { z-index: 1; ... }
</style>
```



DOM tree

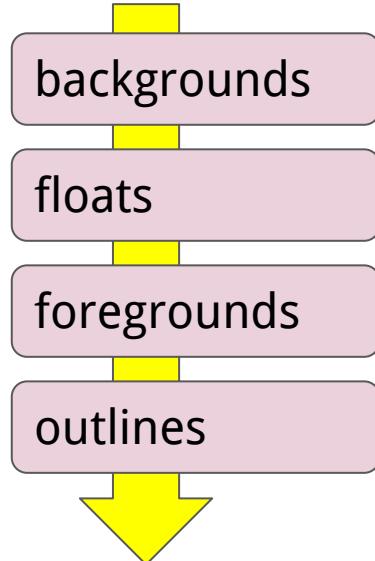


yellow paints last

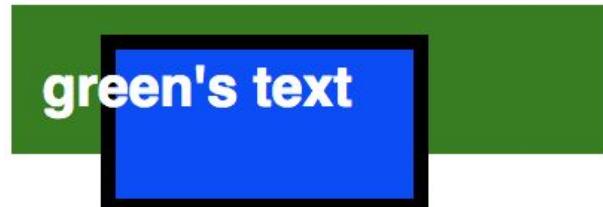
paint

*Each **paint phase** is a separate traversal of a stacking context.*

*paint phases
(simplified)*



```
<div id="green">  
  green's text  
</div>  
<div id="blue"></div>
```



*blue after green, but
foregrounds after backgrounds*

paint — example

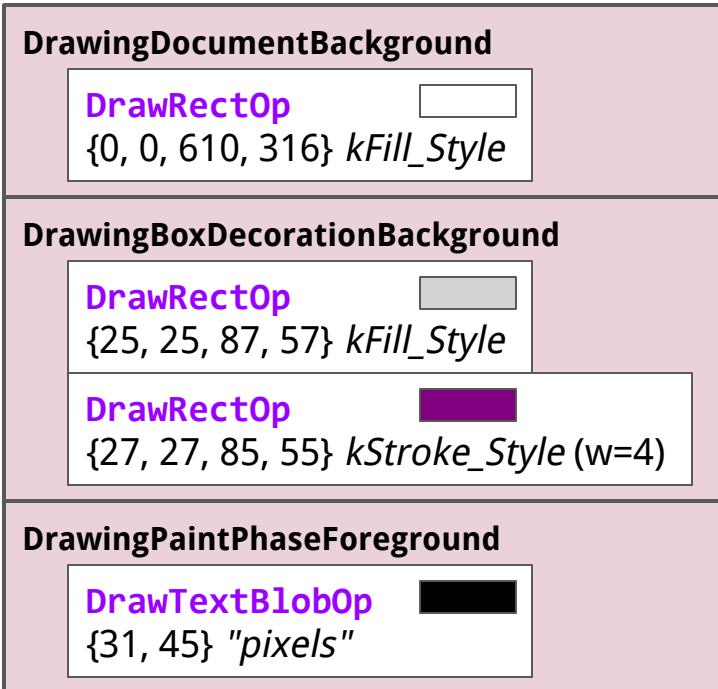
```
<style> #p {  
    position: absolute; padding: 2px;  
    width: 50px; height: 20px;  
    left: 25px; top: 25px;  
    border: 4px solid purple;  
    background-color: lightgrey;  
} </style>  
<div id=p> pixels </div>
```

paint — example

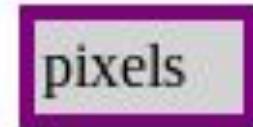
```
<style> #p {  
    position: absolute; padding: 2px;  
    width: 50px; height: 20px;  
    left: 25px; top: 25px;  
    border: 4px solid purple;  
    background-color: lightgrey;  
} </style>  
<div id=p> pixels </div>
```

pixels

paint — example



```
<style> #p {
    position: absolute; padding: 2px;
    width: 50px; height: 20px;
    left: 25px; top: 25px;
    border: 4px solid purple;
    background-color: lightgrey;
}</style>
<div id=p> pixels </div>
```



display items

paint — example

DrawingPaintPhaseForeground

DrawTextBlobOp
{31, 45} 

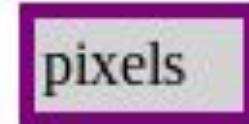
SkTextBlob

53	4C	5B	48	4F	56
0.0	8.0	12.4	20.4	27.5	32.0

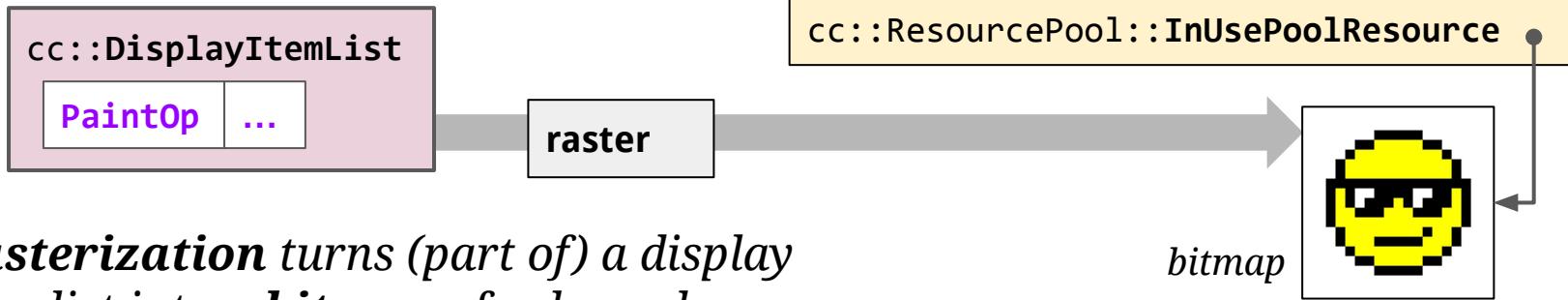


pixels

```
<style> #p {  
    position: absolute; padding: 2px;  
    width: 50px; height: 20px;  
    left: 25px; top: 25px;  
    border: 4px solid purple;  
    background-color: lightgrey;  
}</style>  
<div id=p> pixels </div>
```



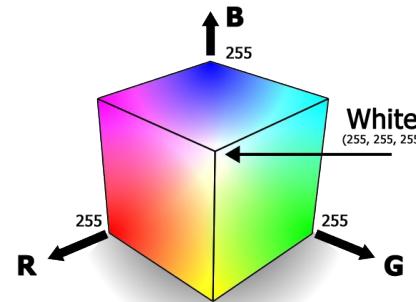
raster



Rasterization turns (part of) a display item list into a **bitmap** of color values.

FFFF	FFFF	0000
FFFF	FFFF	00FF
FFFF	0000	FFFF
FFFF	00FF	00FF
0000	FFFF	FFFF
00FF	00FF	00FF

(red, green,
blue, alpha)

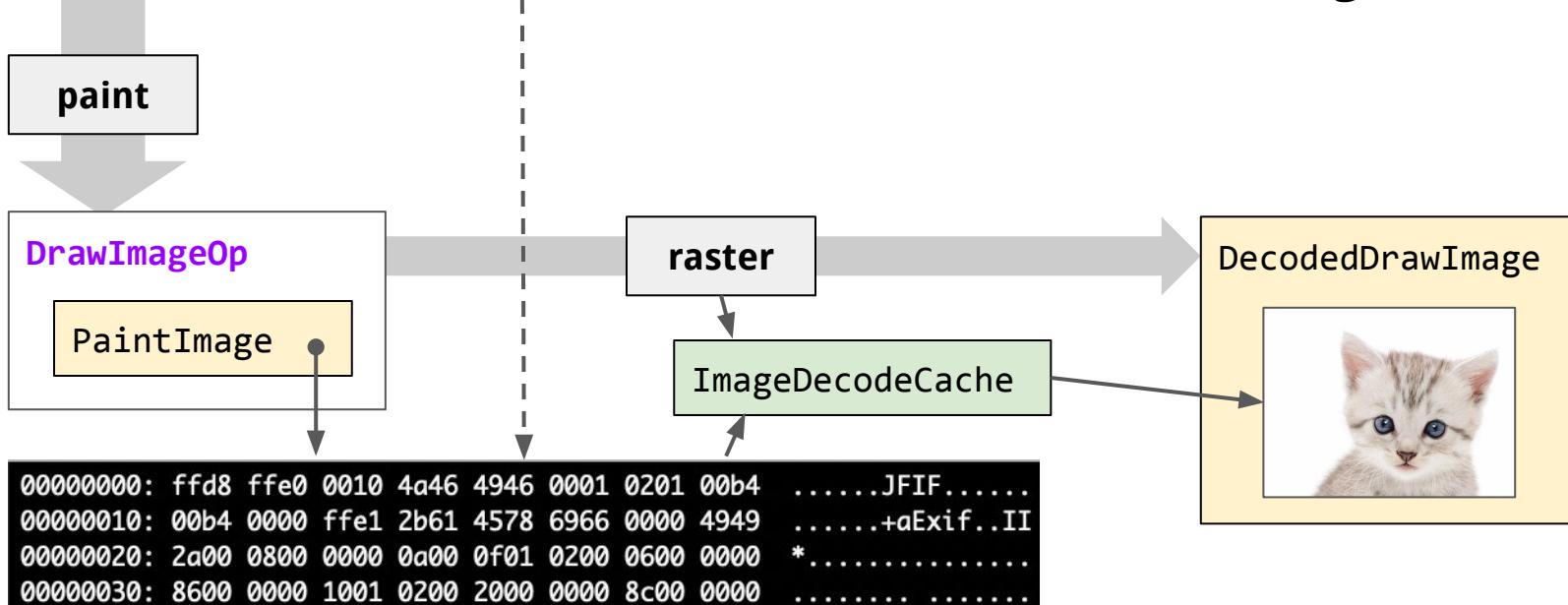


raster

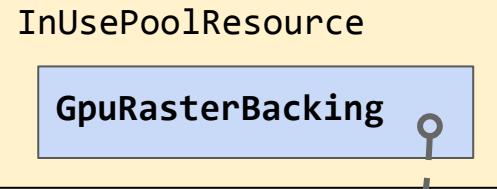
```

```

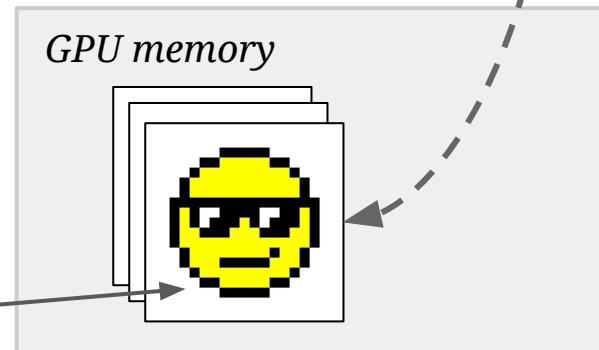
Raster includes image decoding.



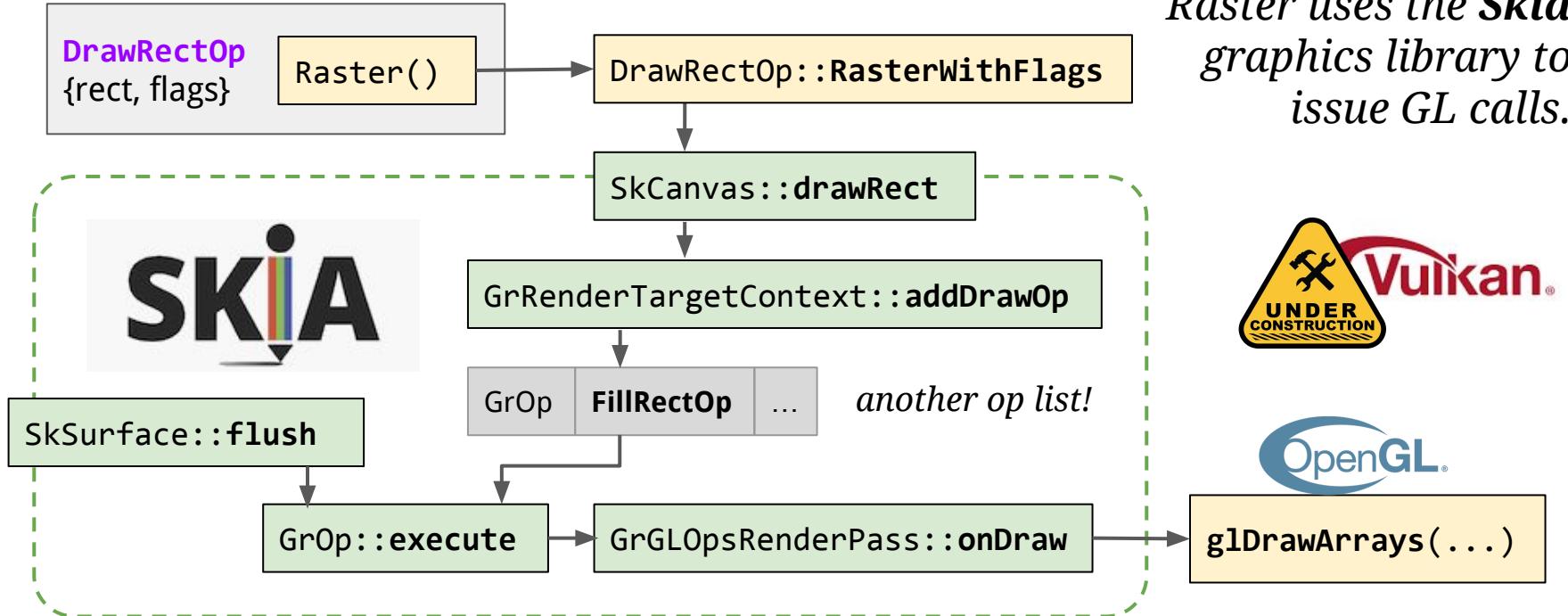
raster



Raster can be accelerated by the GPU.

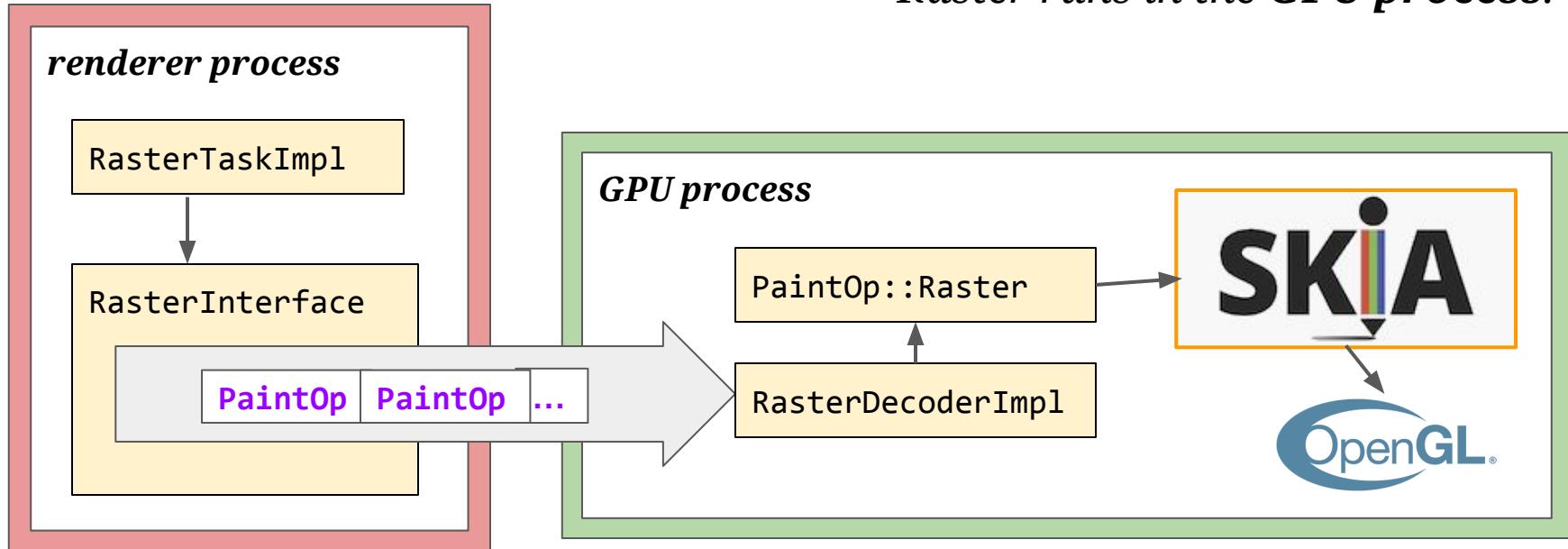


raster



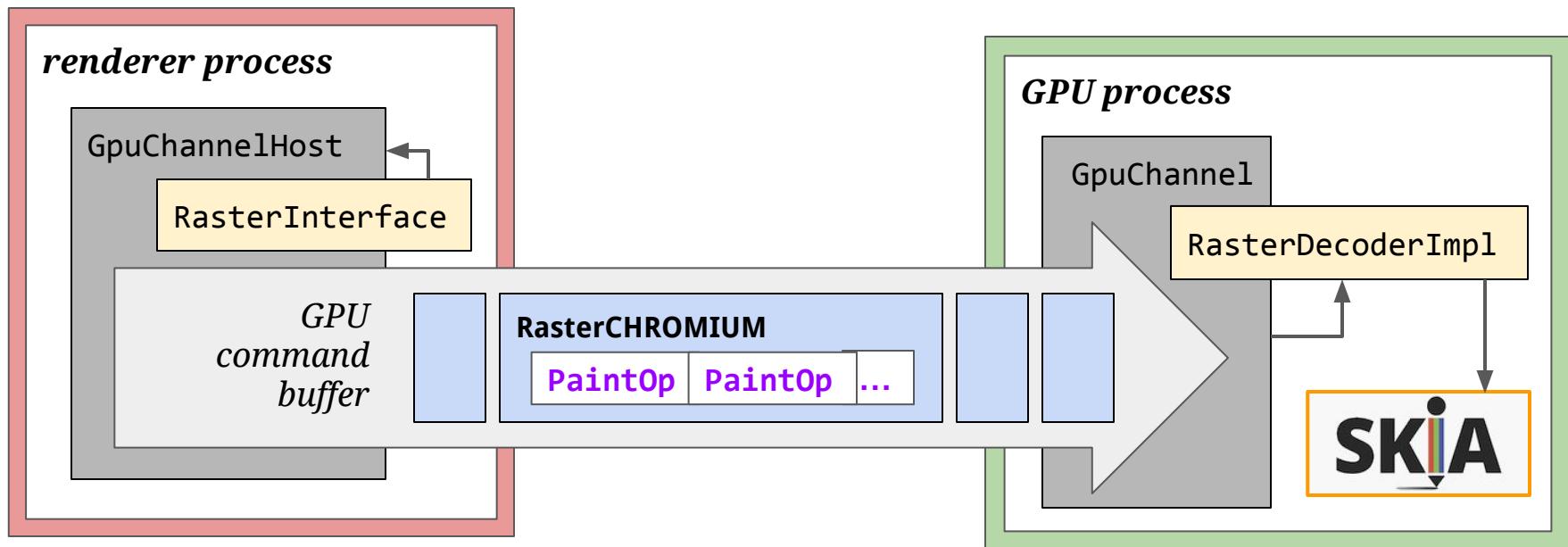
gpu

Raster runs in the GPU process.



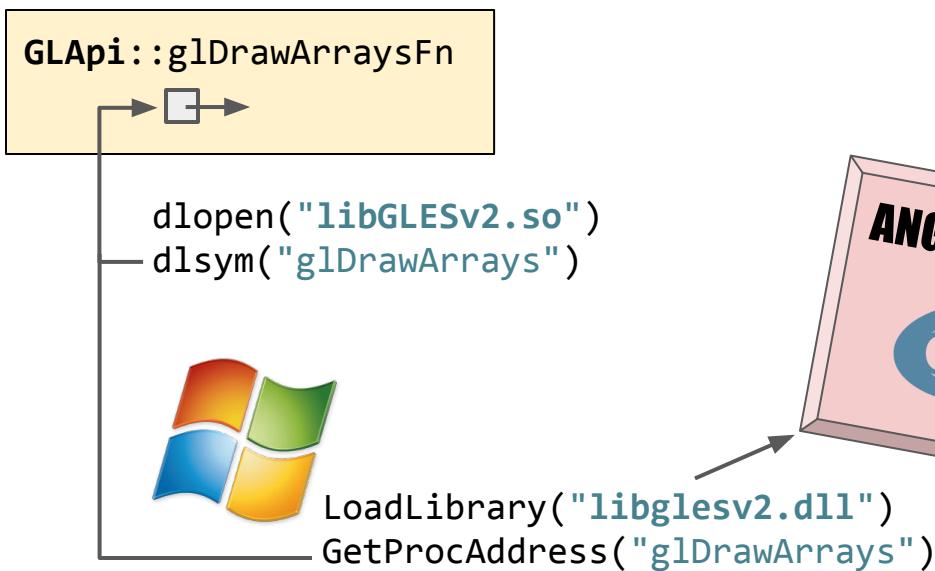
gpu

*The request is sent through a **command buffer**.*



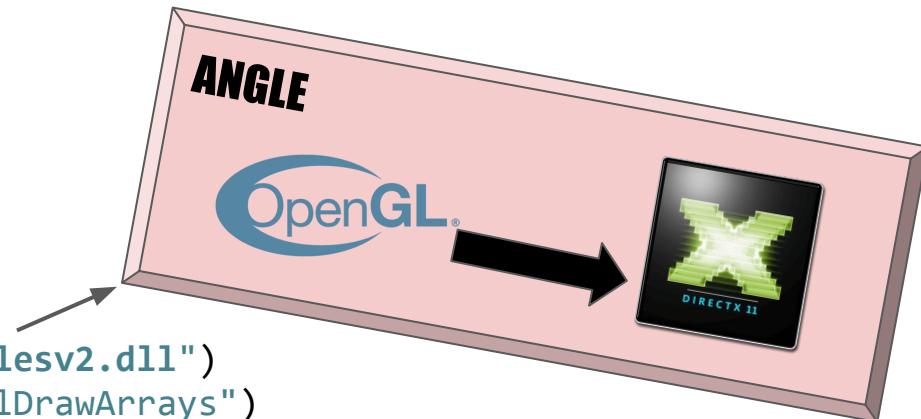
gpu

GPU process

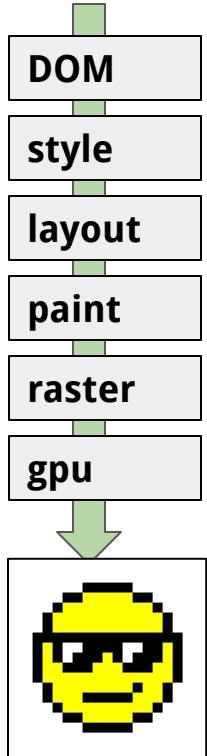


In the GPU process, the GL functions link dynamically to native OpenGL.

On Windows, we translate to DirectX.



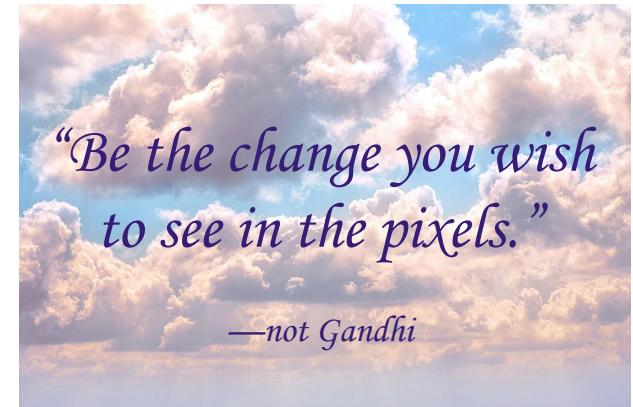
change



*We now have a complete **pipeline**.*

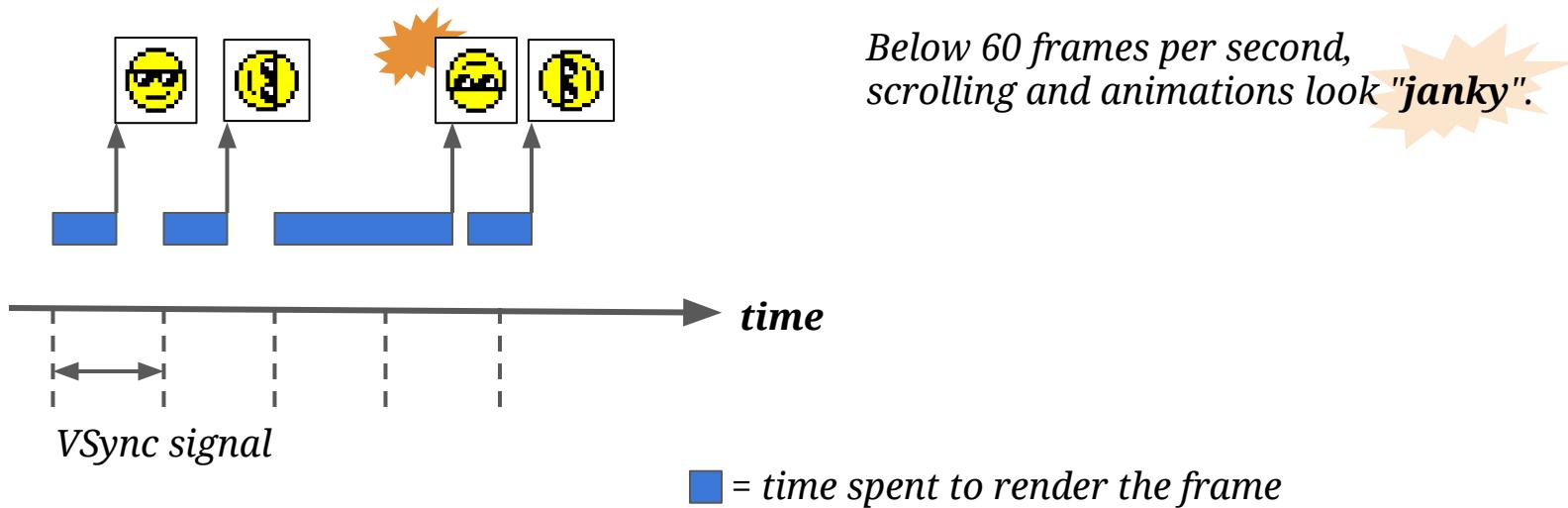
*But what if state can **change**?*

- scrolling
- zooming
- animations
- incremental loading
- JavaScript
- ...



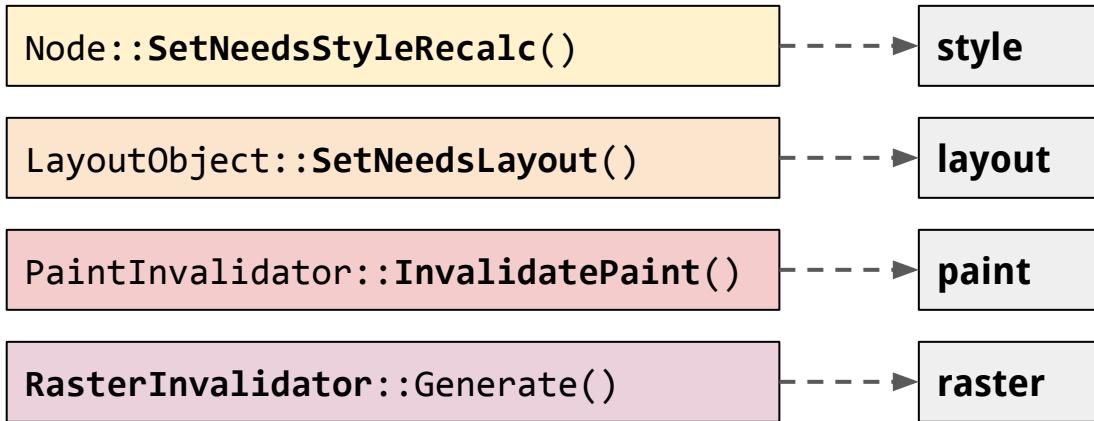
frames

*The renderer produces **animation frames**.*



invalidation

*Each pipeline stage tracks granular asynchronous **invalidations**.*



Outputs are reused from previous frames when possible.

repaint

Paint + raster remain expensive if a large region is transformed...

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam commodo est orci, quis sodales diam tristique et. Fusce fringilla ac libero eget facilisis. Fusce a vulputate ligula. In tortor

scroll

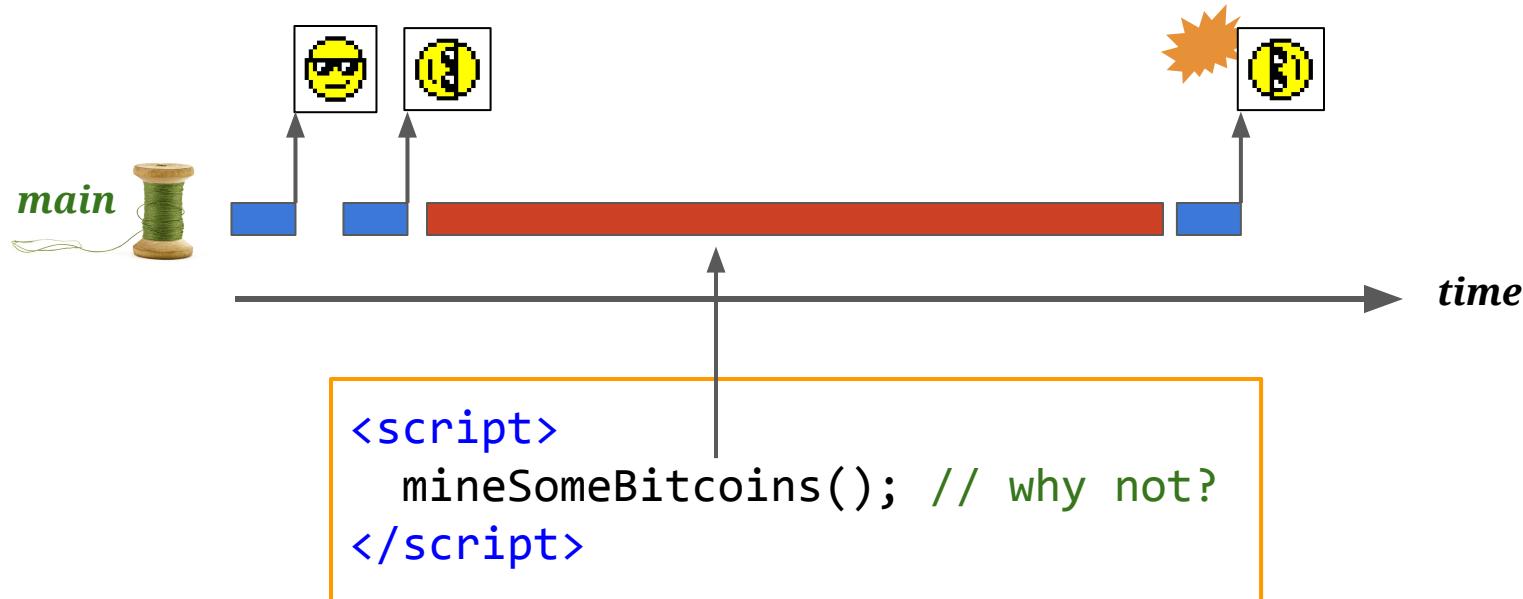


adipiscing elit. Aliquam commodo est orci, quis sodales diam tristique et. Fusce fringilla ac libero eget facilisis. Fusce a vulputate ligula. In tortor ex, porta faucibus fringilla quis, congue a

all the pixels changed!

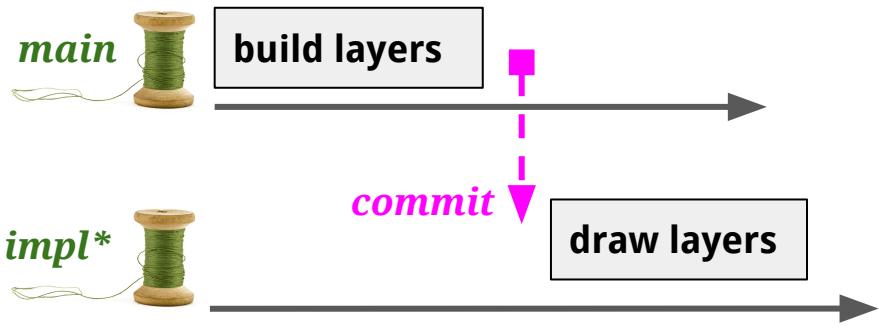
jank

*... and anything on the **main thread** competes with JavaScript.*



enter: compositing

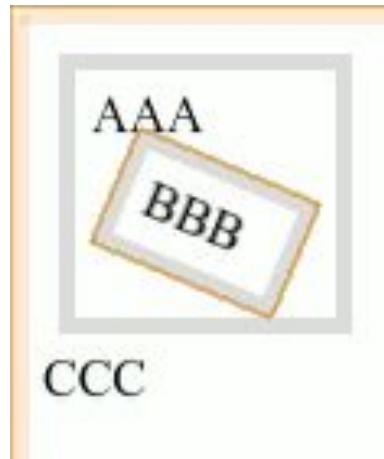
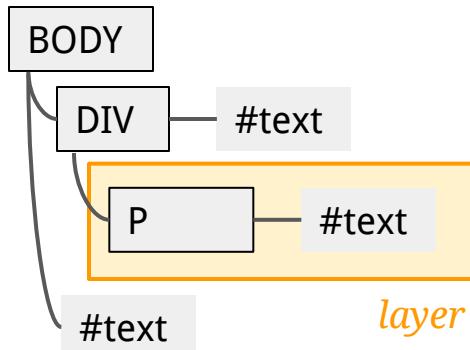
- *Decompose the page into layers which raster independently.*
- *Combine the layers on another thread.*



* ("impl" = compositor thread) ↴(;) ↵

layers

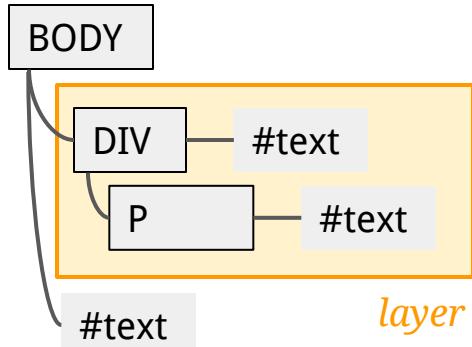
```
<div>  
  AAA  
  <p class="wobble"> BBB </p>  
</div>  
CCC
```



A simple layer captures a subtree of content.

layers

```
<div class="wobble">  
  AAA  
  <p> BBB </p>  
</div>  
CCC
```



A simple layer captures a subtree of content.

compositing



Animation:
a layer moves

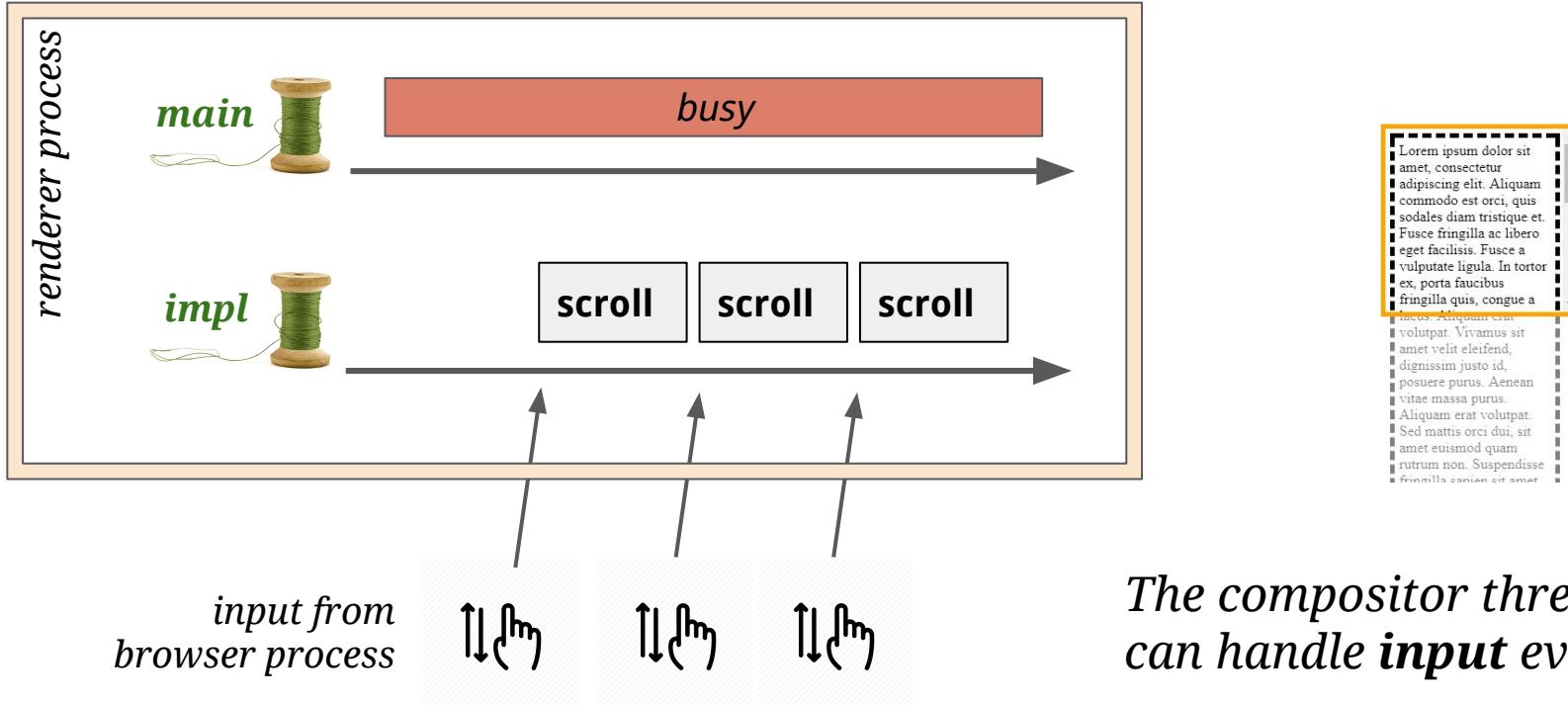
Scrolling:
a layer moves;
another clips

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam commodo est orci, quis sodales diam tristique et. Fusce fringilla ac libero eget facilisis. Fusce a vulputate ligula. In tortor ex, porta faucibus fringilla quis, congue a
fucus. Aliquam erat volutpat. Vivamus sit amet velit eleifend, dignissim justo id, posuere purus. Aenean vitae massa purus. Aliquam erat volutpat. Sed mattis orci dui, sit amet euismod quam rutrum non. Suspendisse fringilla canien sit amet

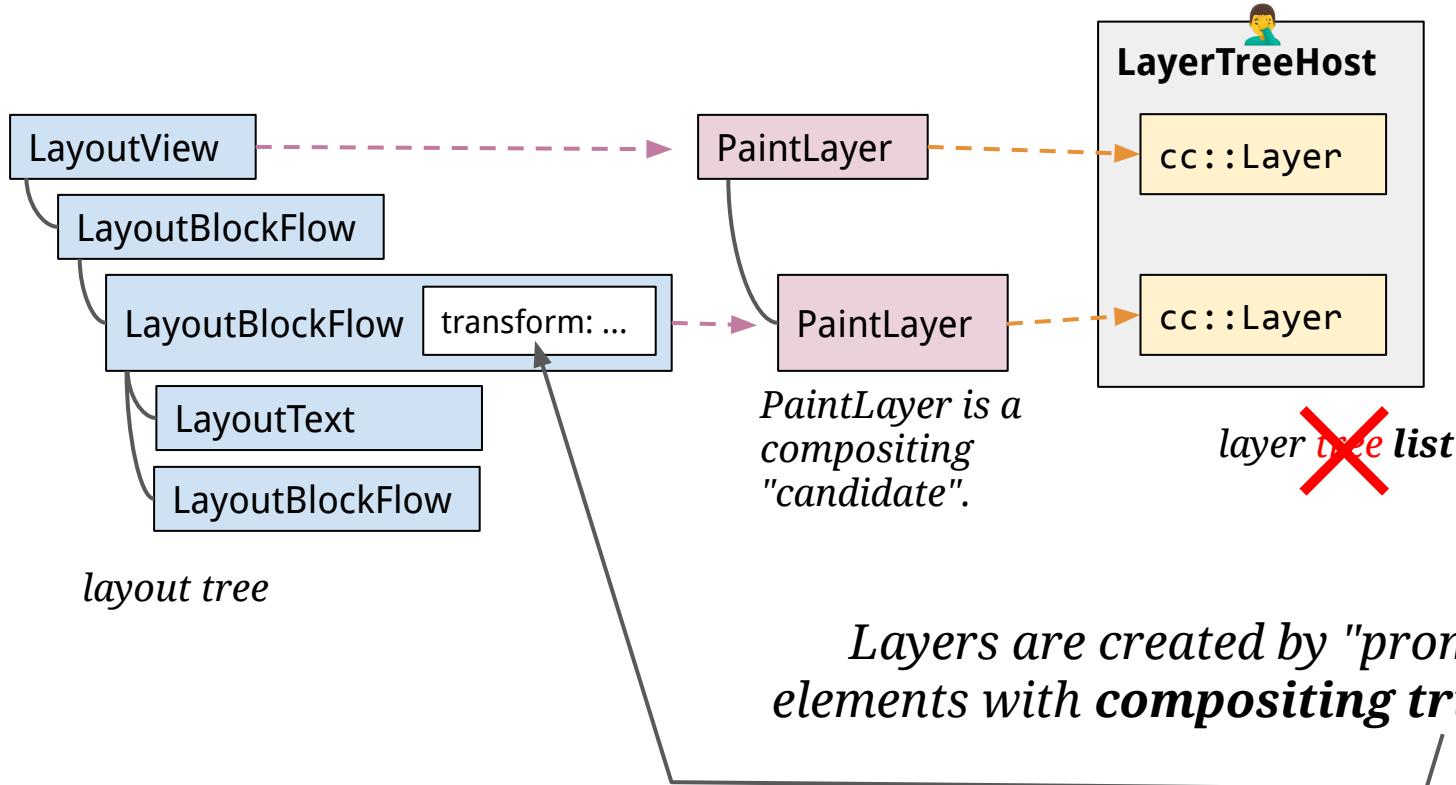
Pinch Zoom:
a layer scales



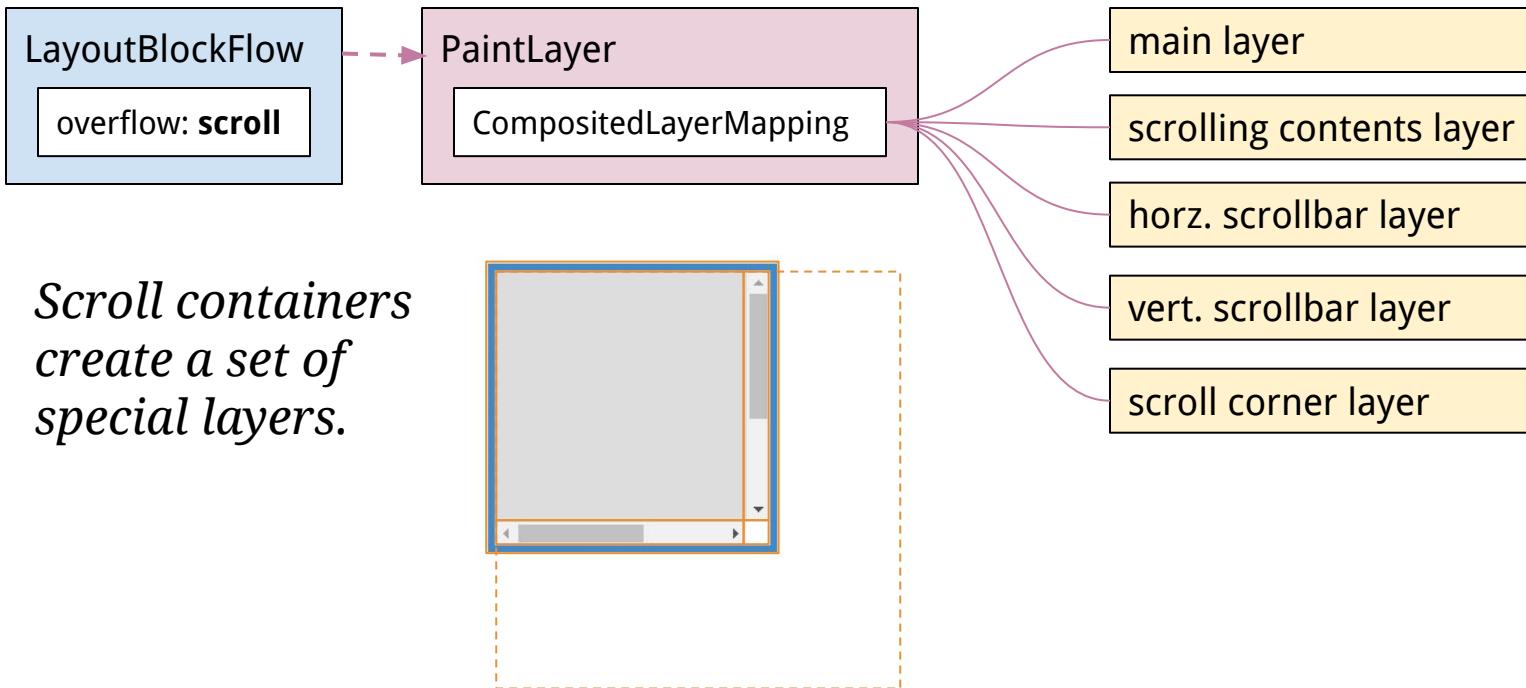
threaded input



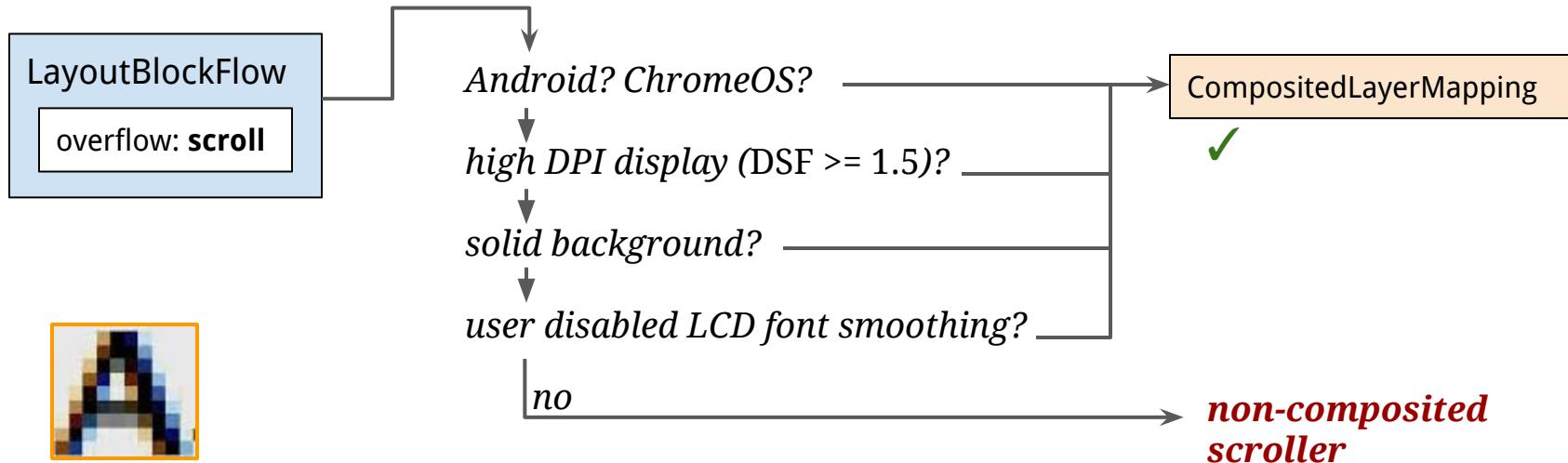
layer promotion



scrolling layers

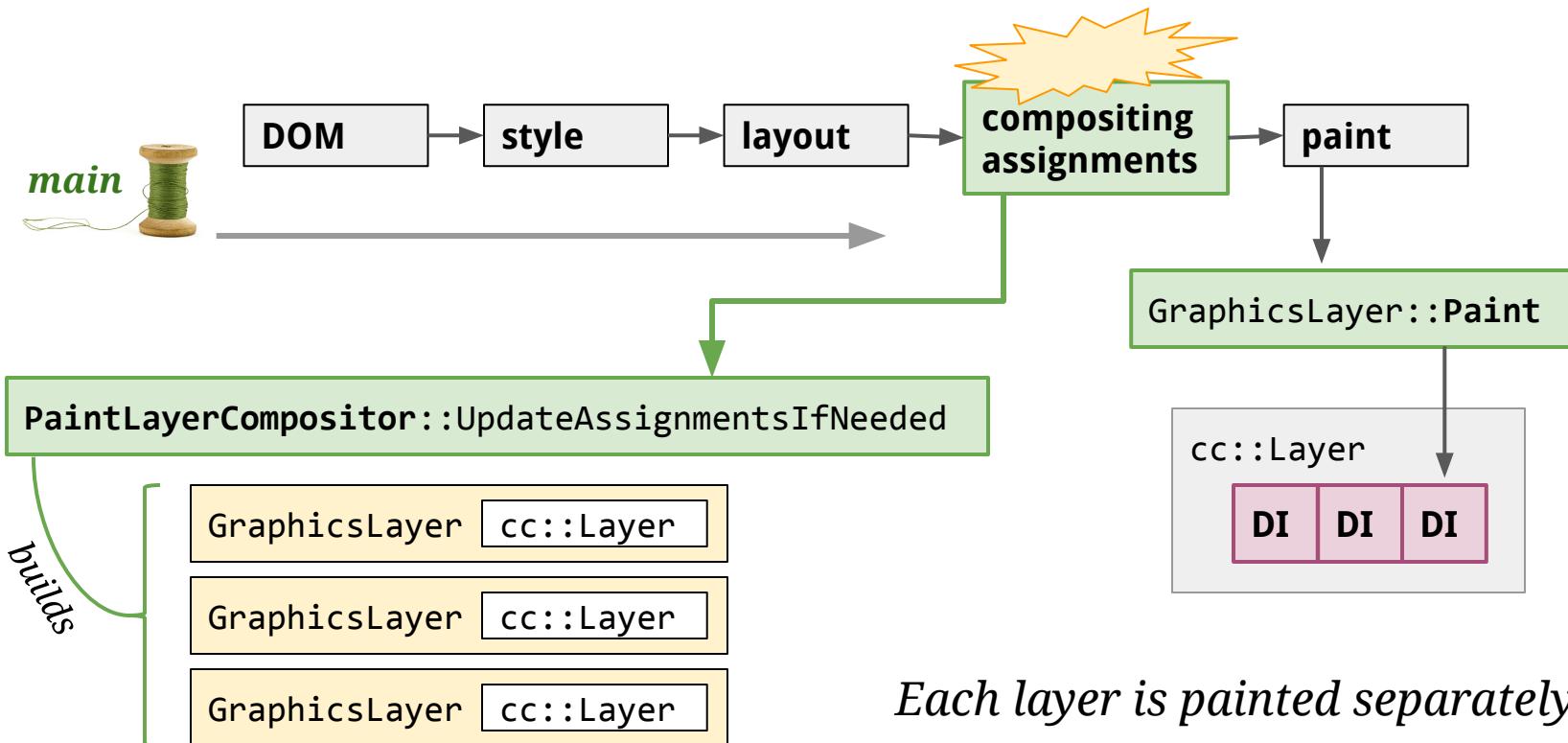


scrolling layers

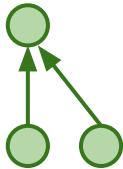


*Compositing a transparent scroller
disables subpixel antialiasing.*

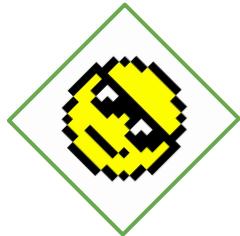
compositing assignments



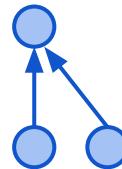
property trees



*transform
tree*



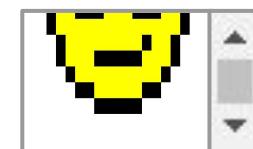
*clip
tree*



*effect
tree*

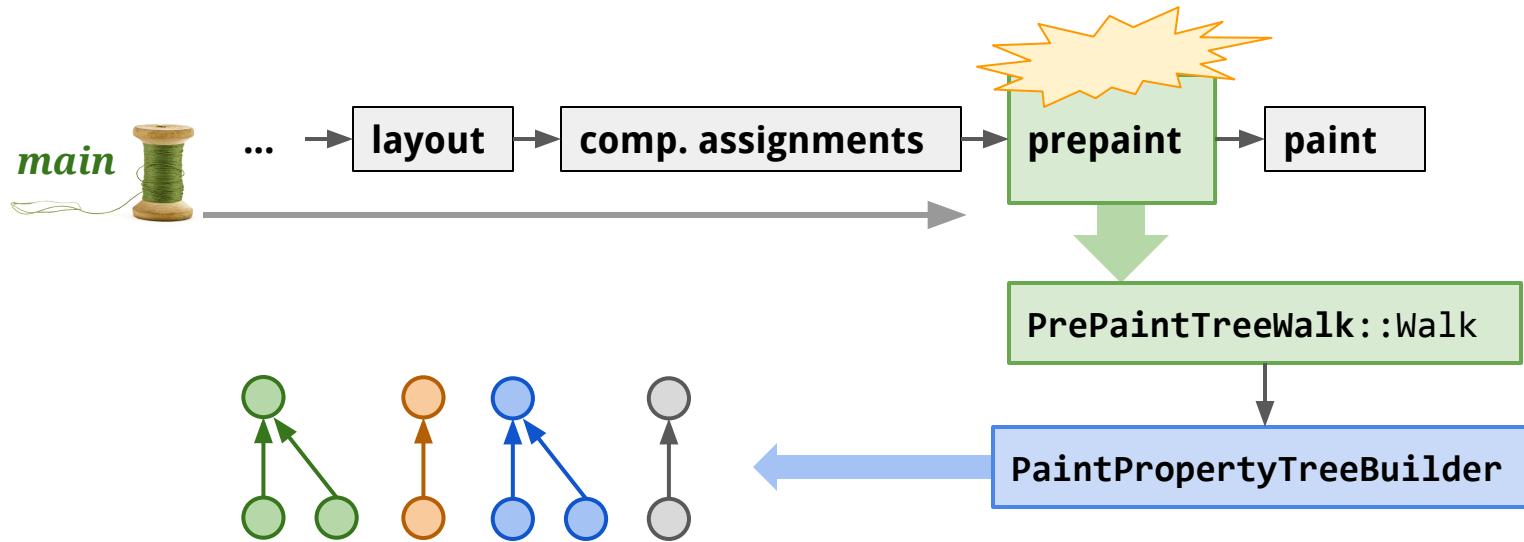


*scroll
tree*



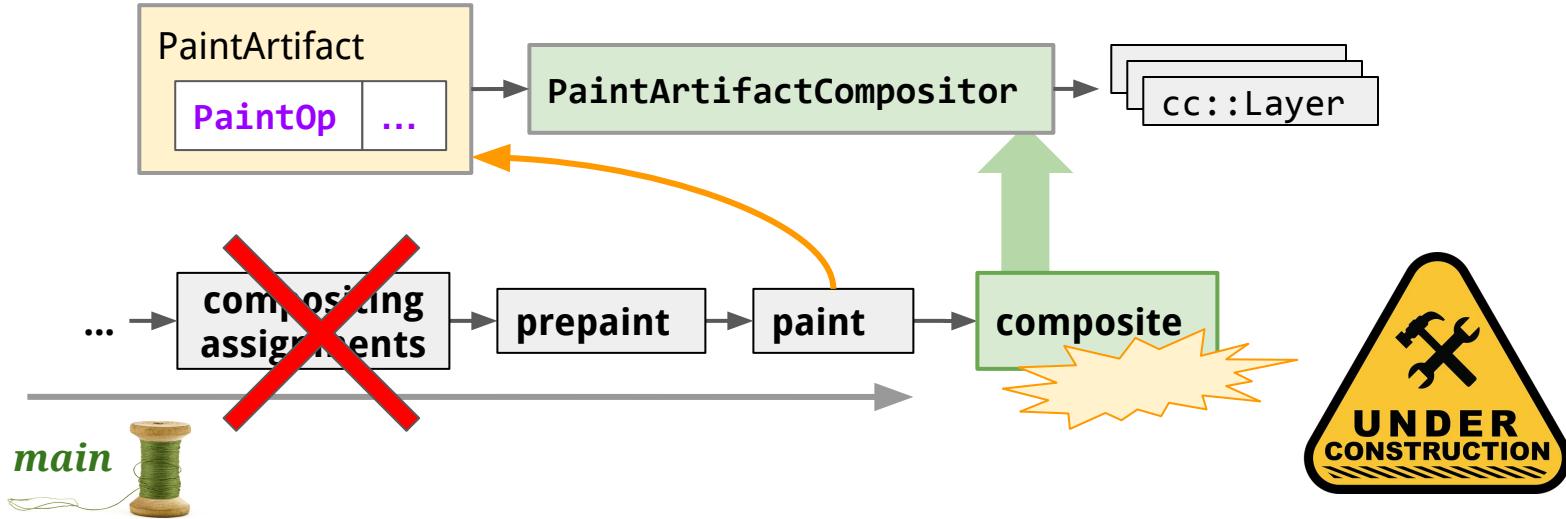
*The compositor can apply some **paint properties** to a layer.*

property trees



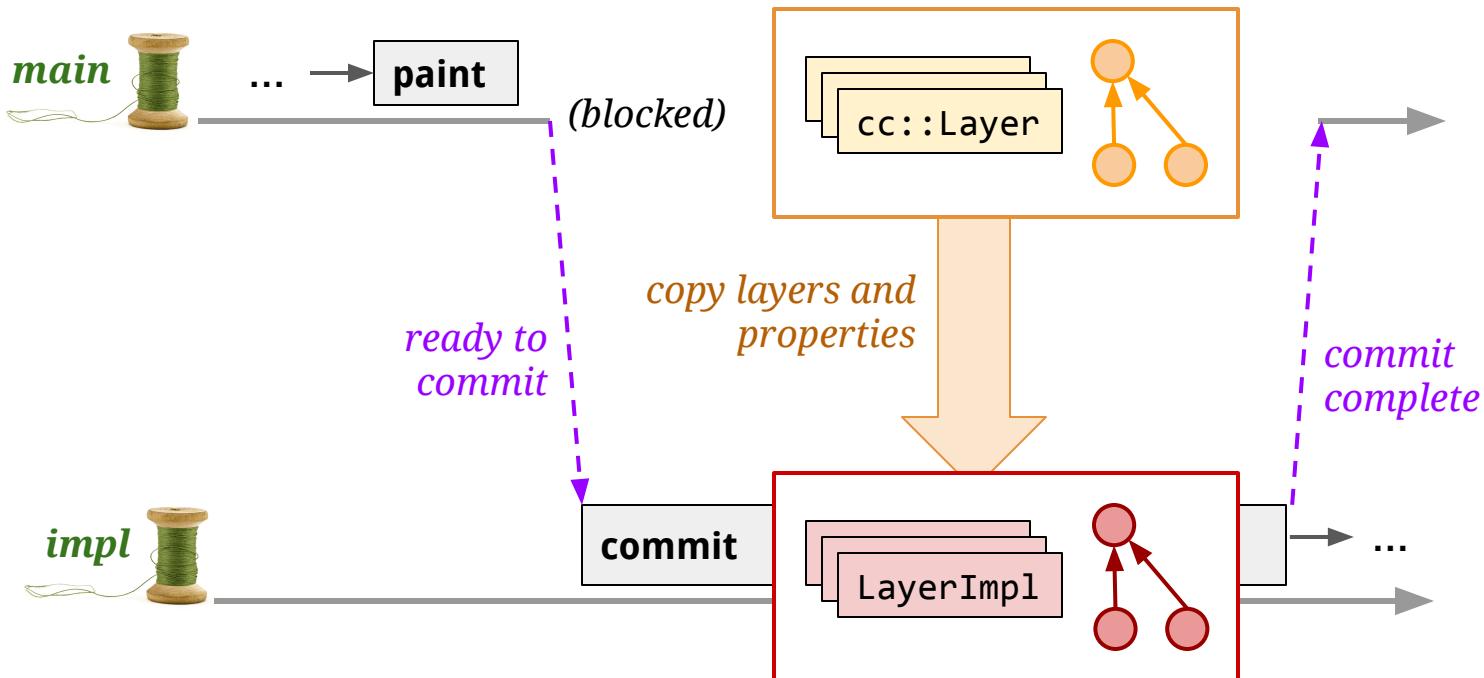
*The **prepaint** stage builds the property trees.*

composite after paint (CAP)



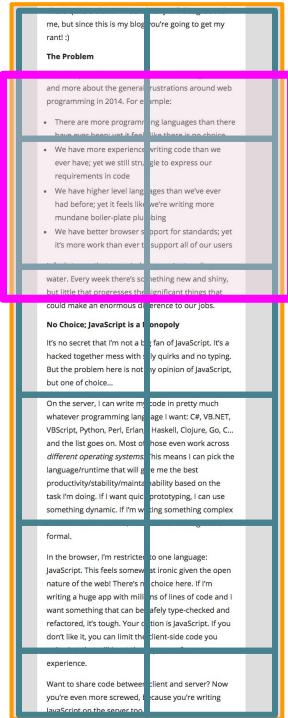
In the future, layers will be created after paint.

commit



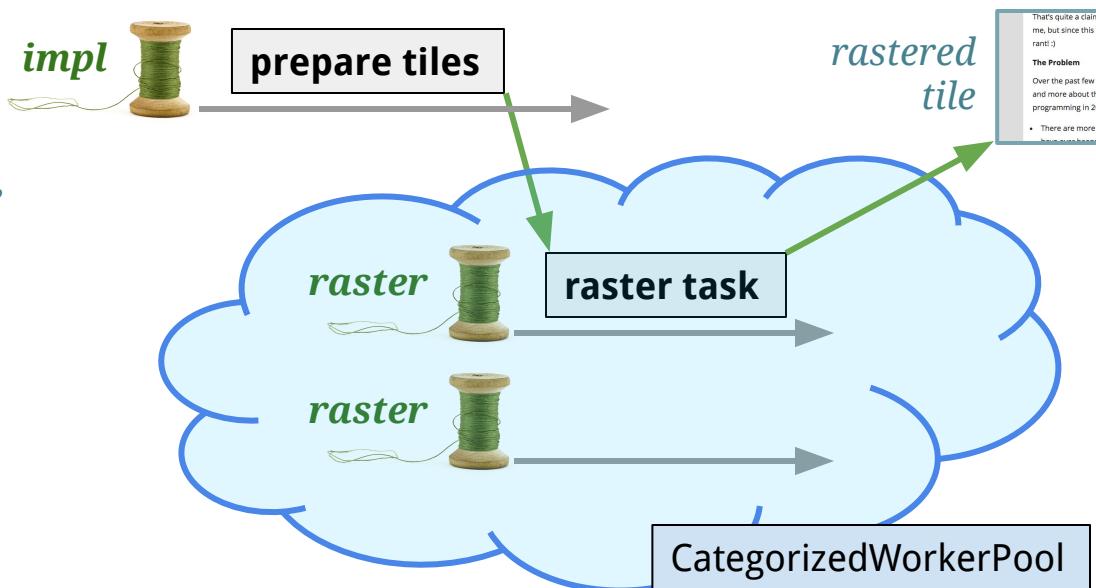
tiling

viewport



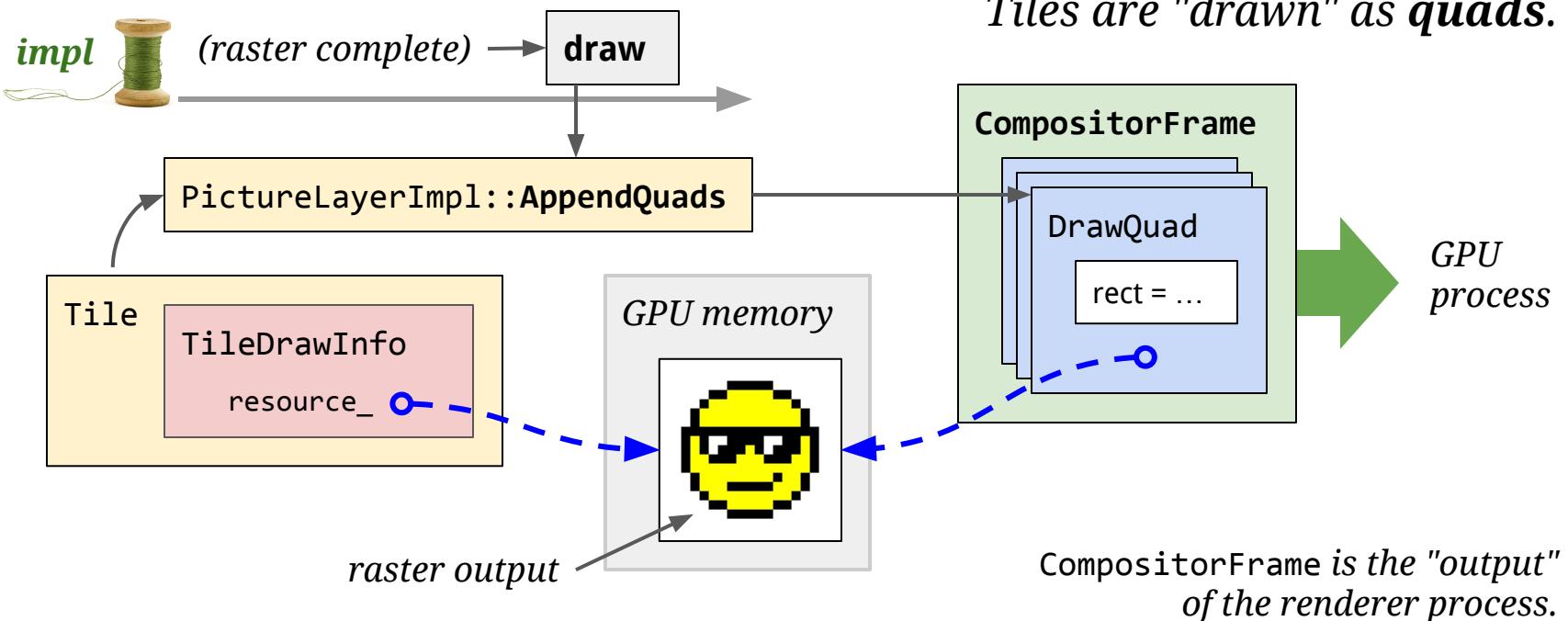
visible tiles

layer

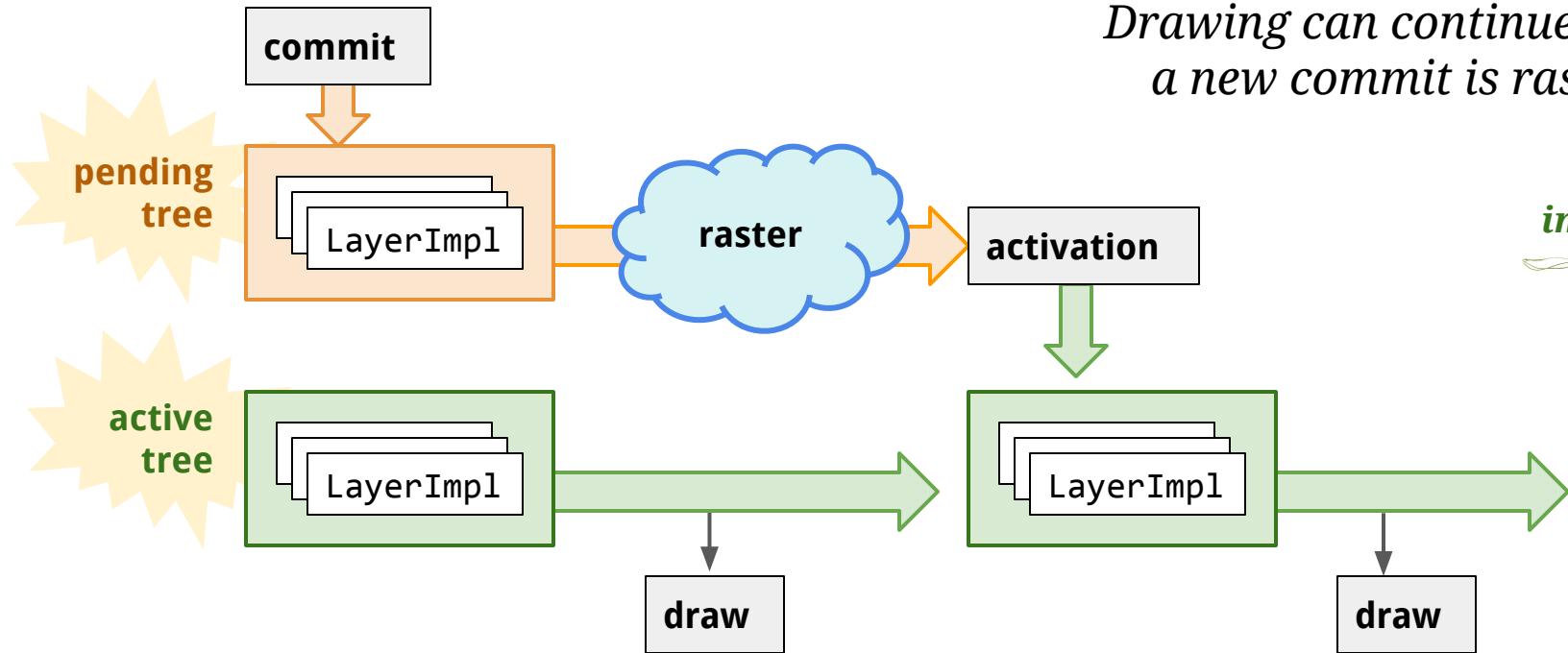


*Layers are broken into **tiles** for raster.*

draw layers

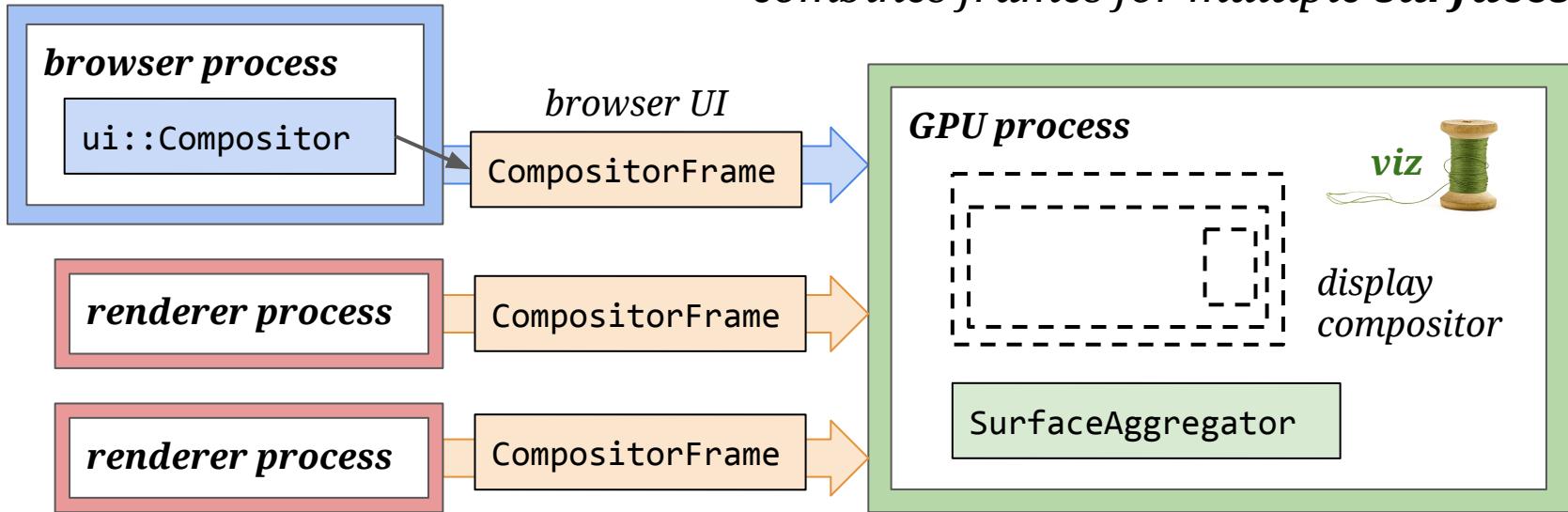


activation

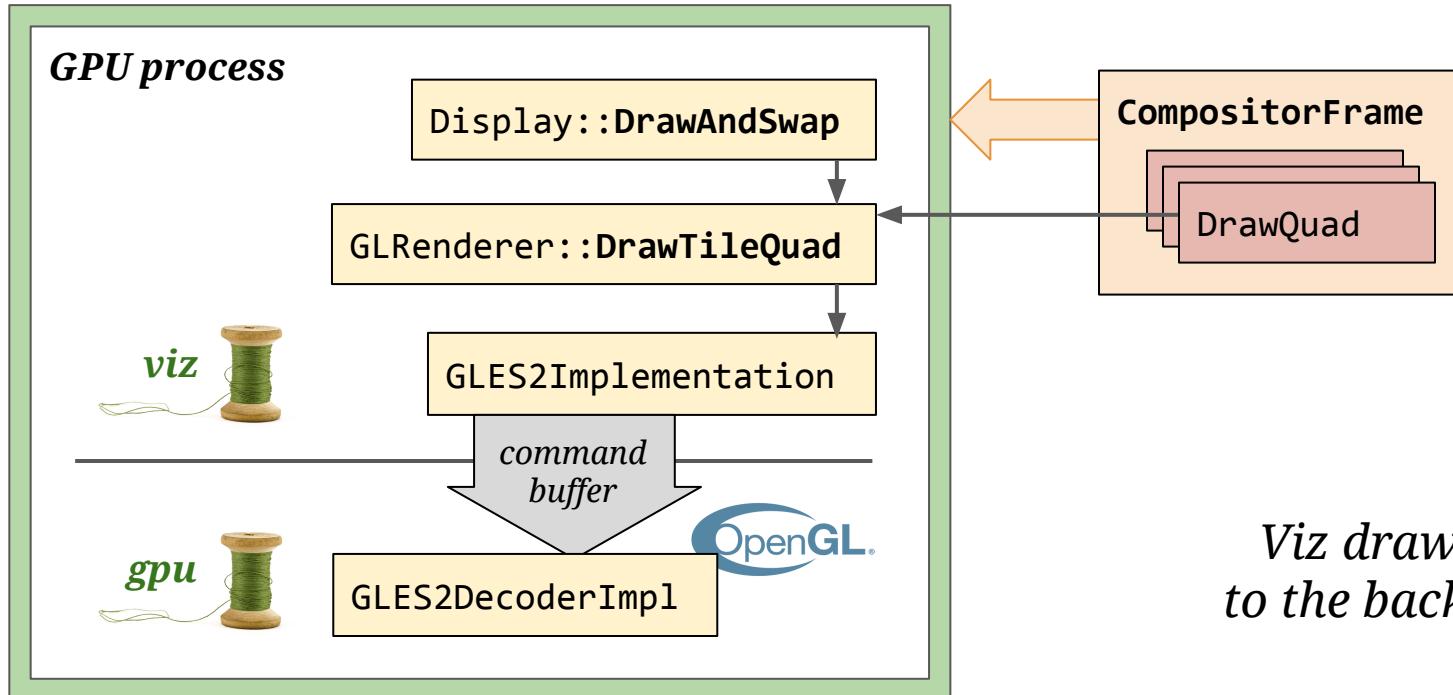


display (viz)

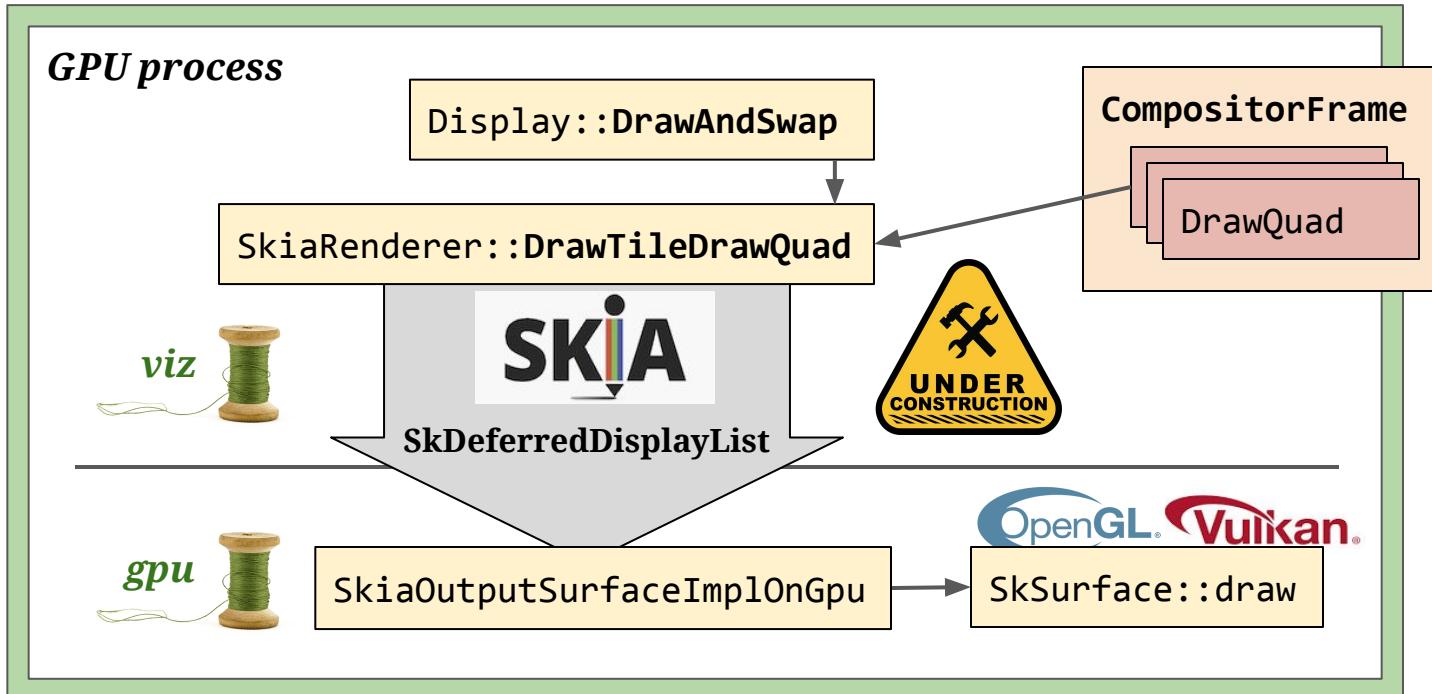
The display compositor combines frames for multiple surfaces.



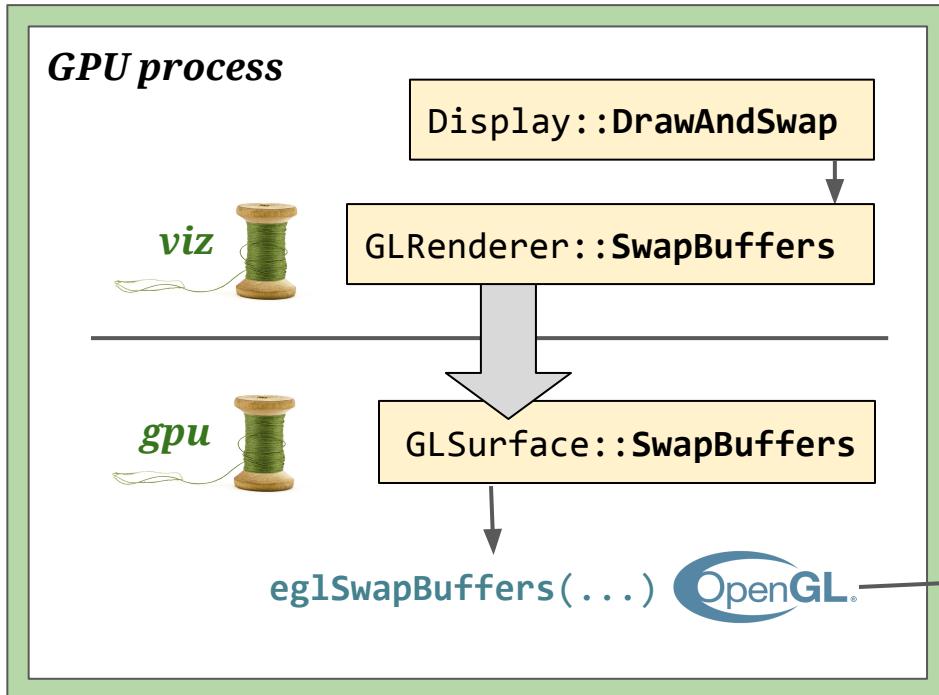
display (viz)



display (viz)

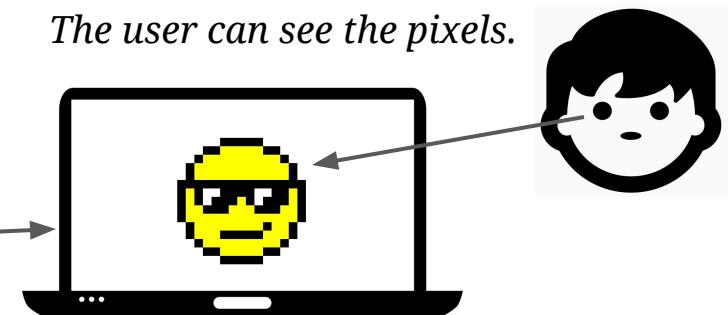


display (viz)

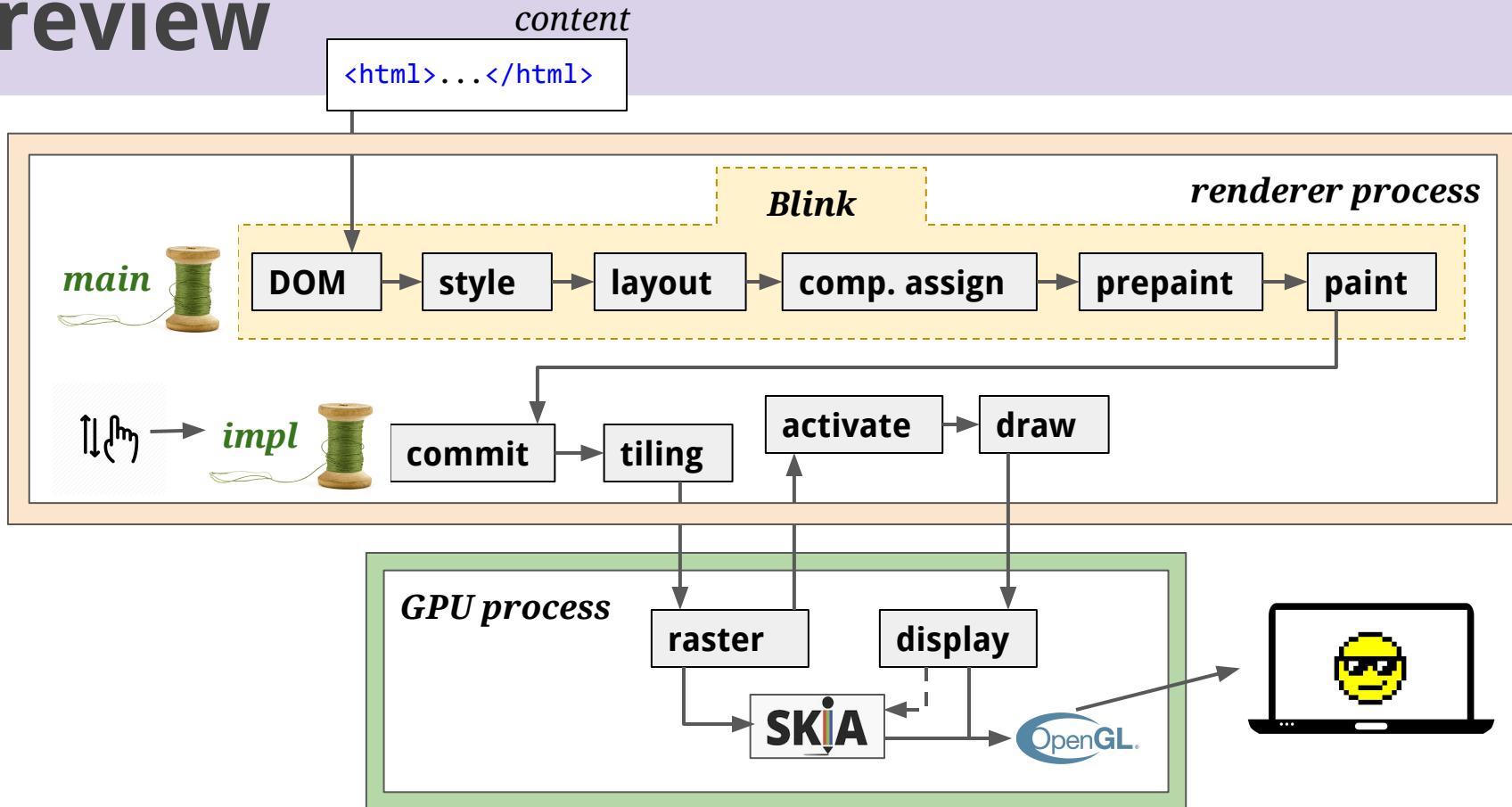


Finally, we swap the buffers.

The user can see the pixels.



review



end

LIFE OF A pixel

slides: bit.ly/lifeofapixel

feedback: skobes@chromium.org