
Google™
12

Android WebView

Nicolas Roard

What are we going to talk about?

WebKit timeline

KDE
KHTML
1998

WebCore
+
Safari
2003

WebKit
open sourced
2005

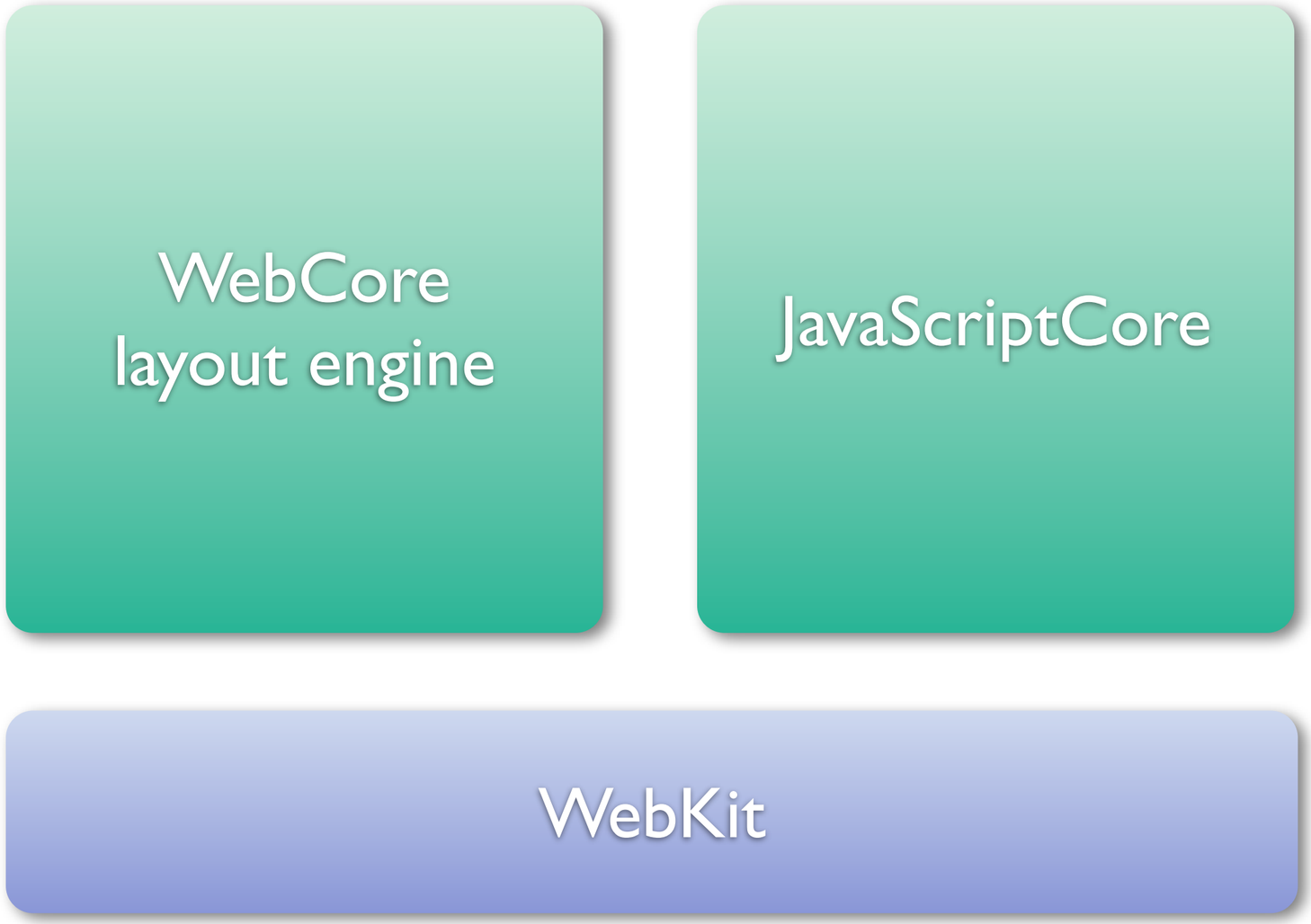
Chrome
+
Android
2008

Architecture

WebKit

Overview

Overview



Overview

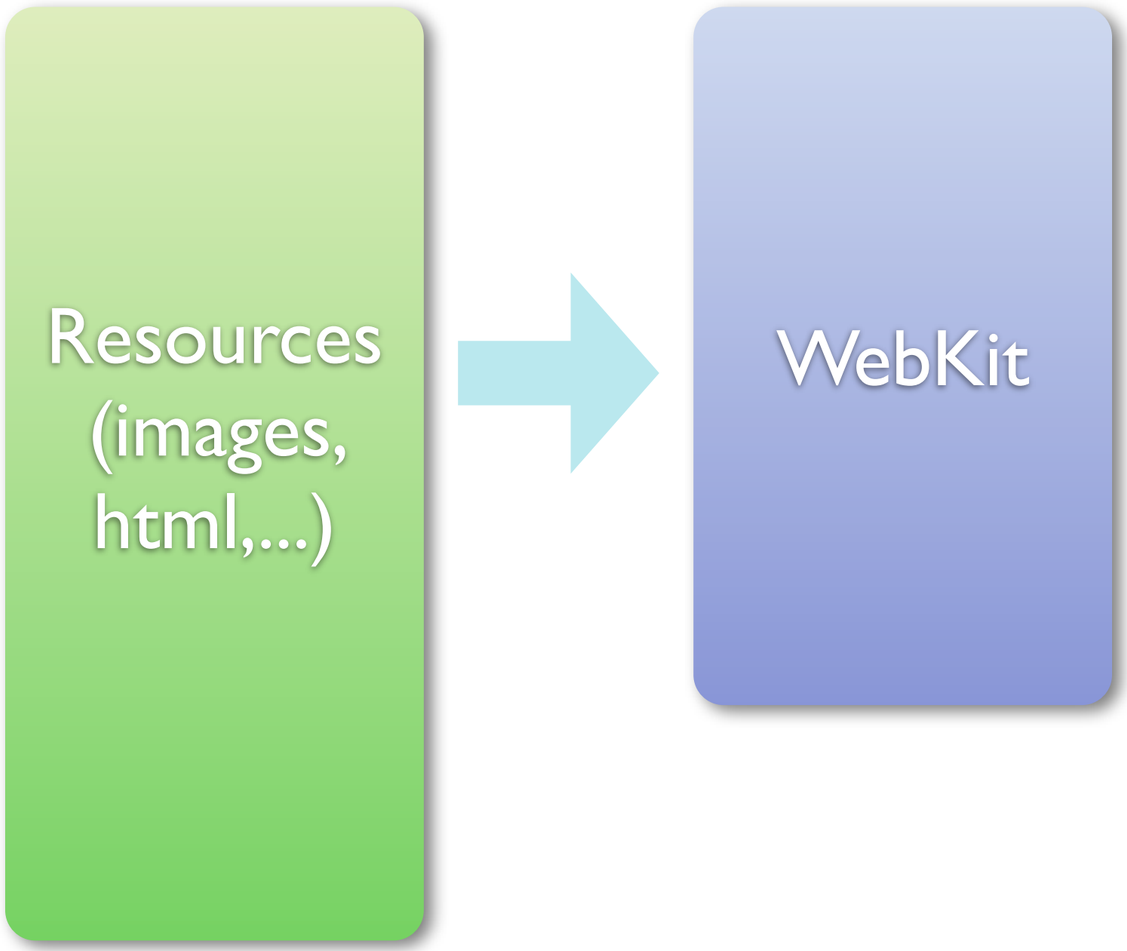
Overview

Overview

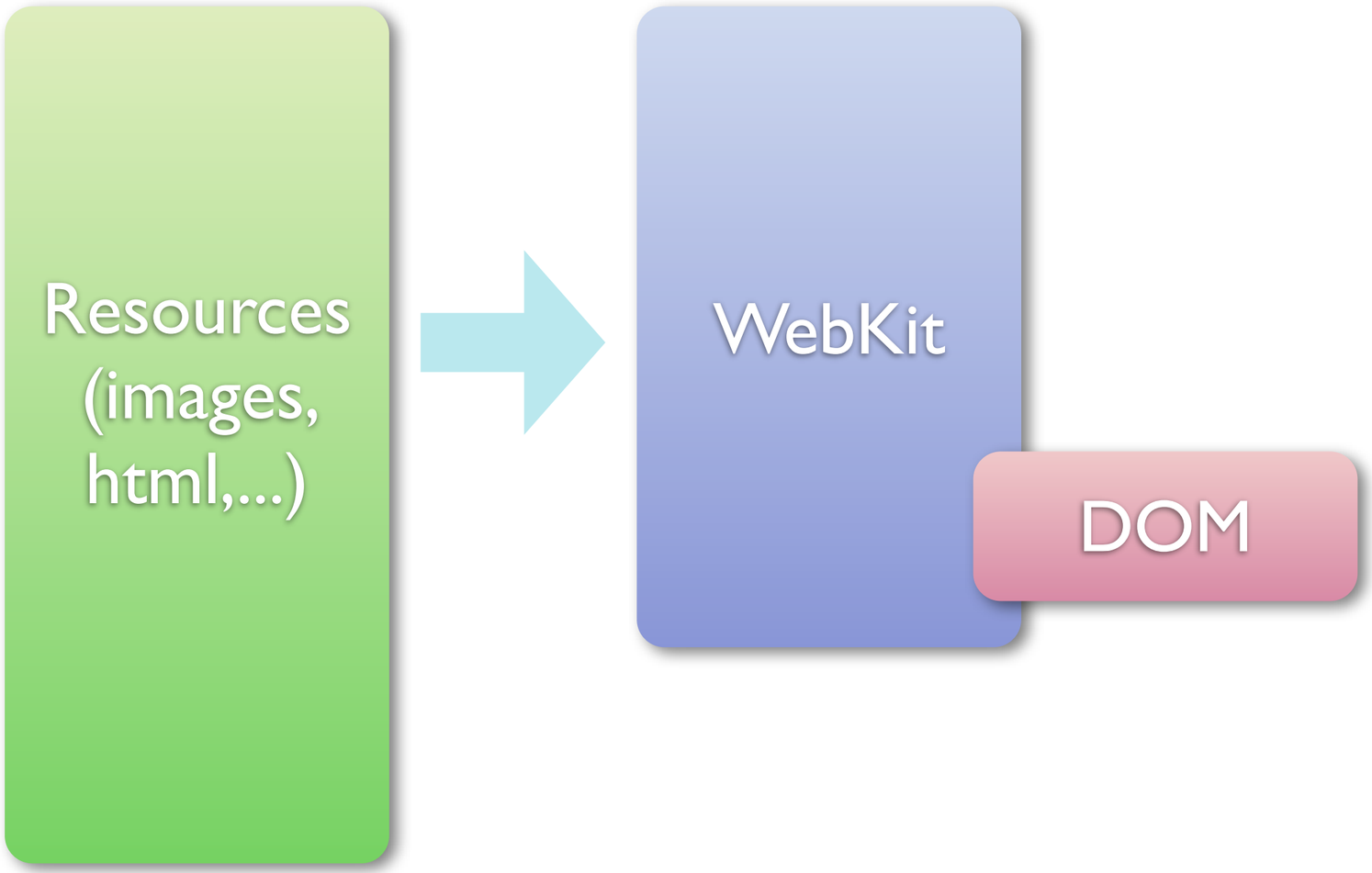
Resources (images, html,...)

WebKit

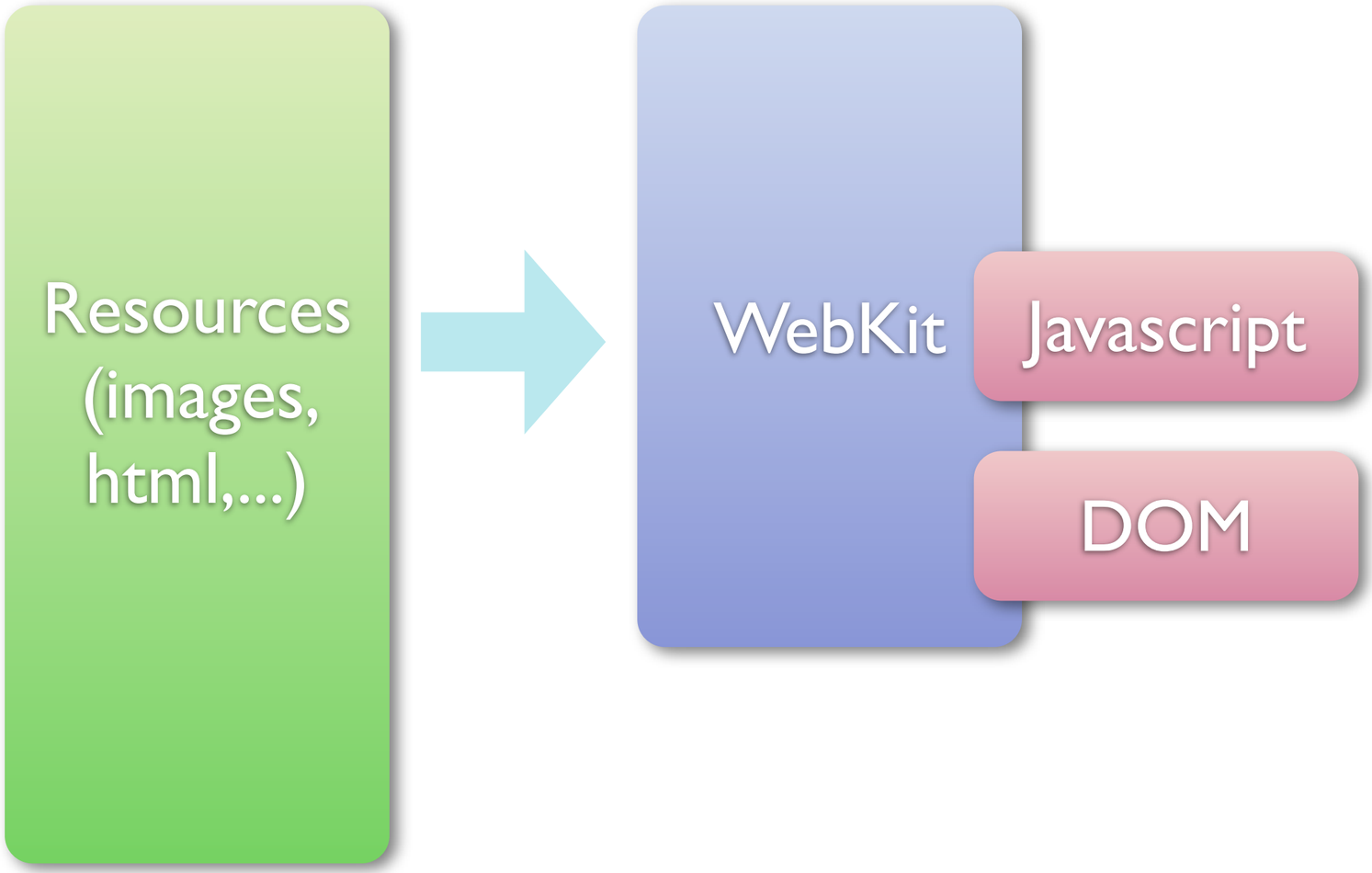
Overview



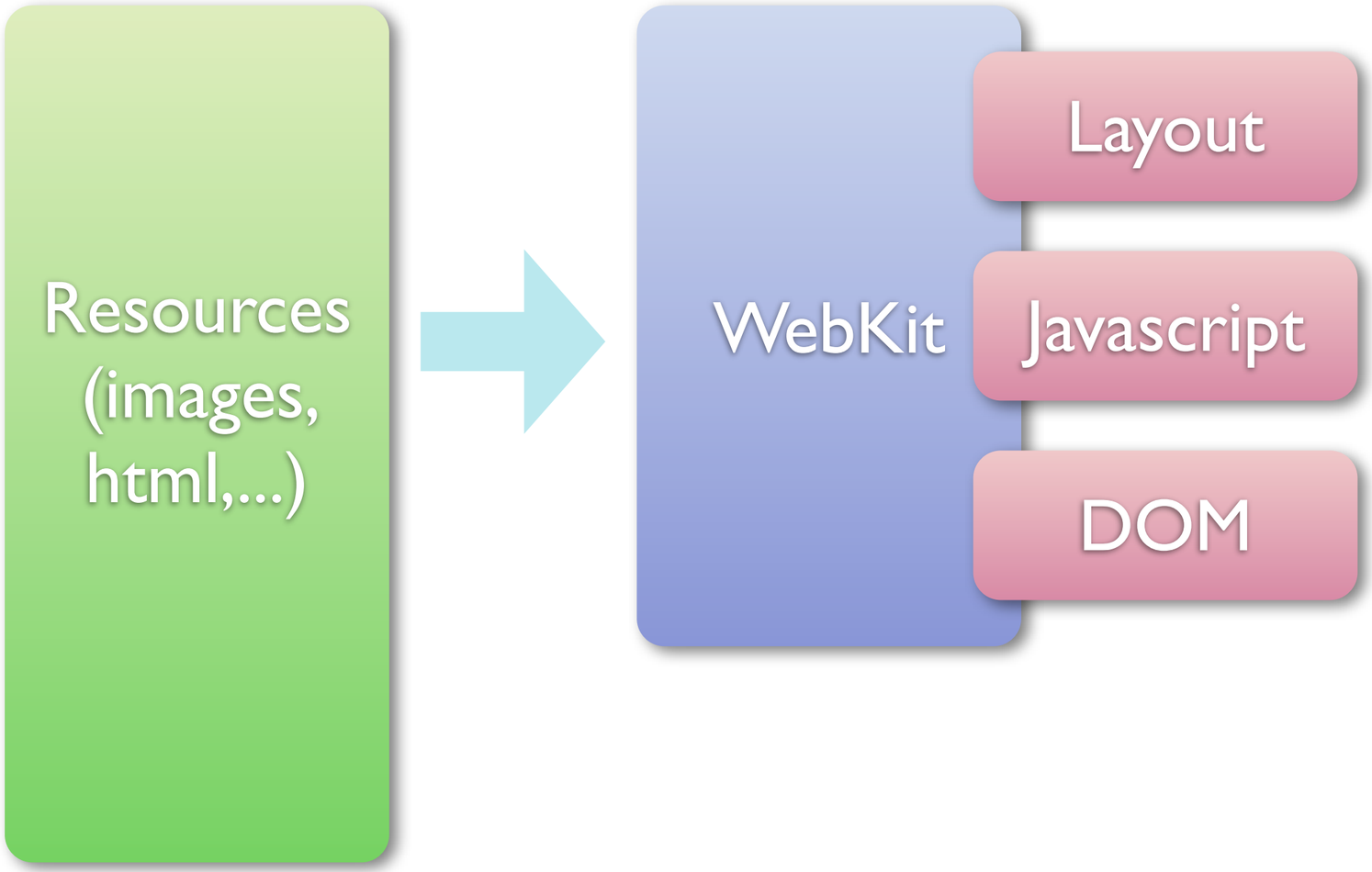
Overview



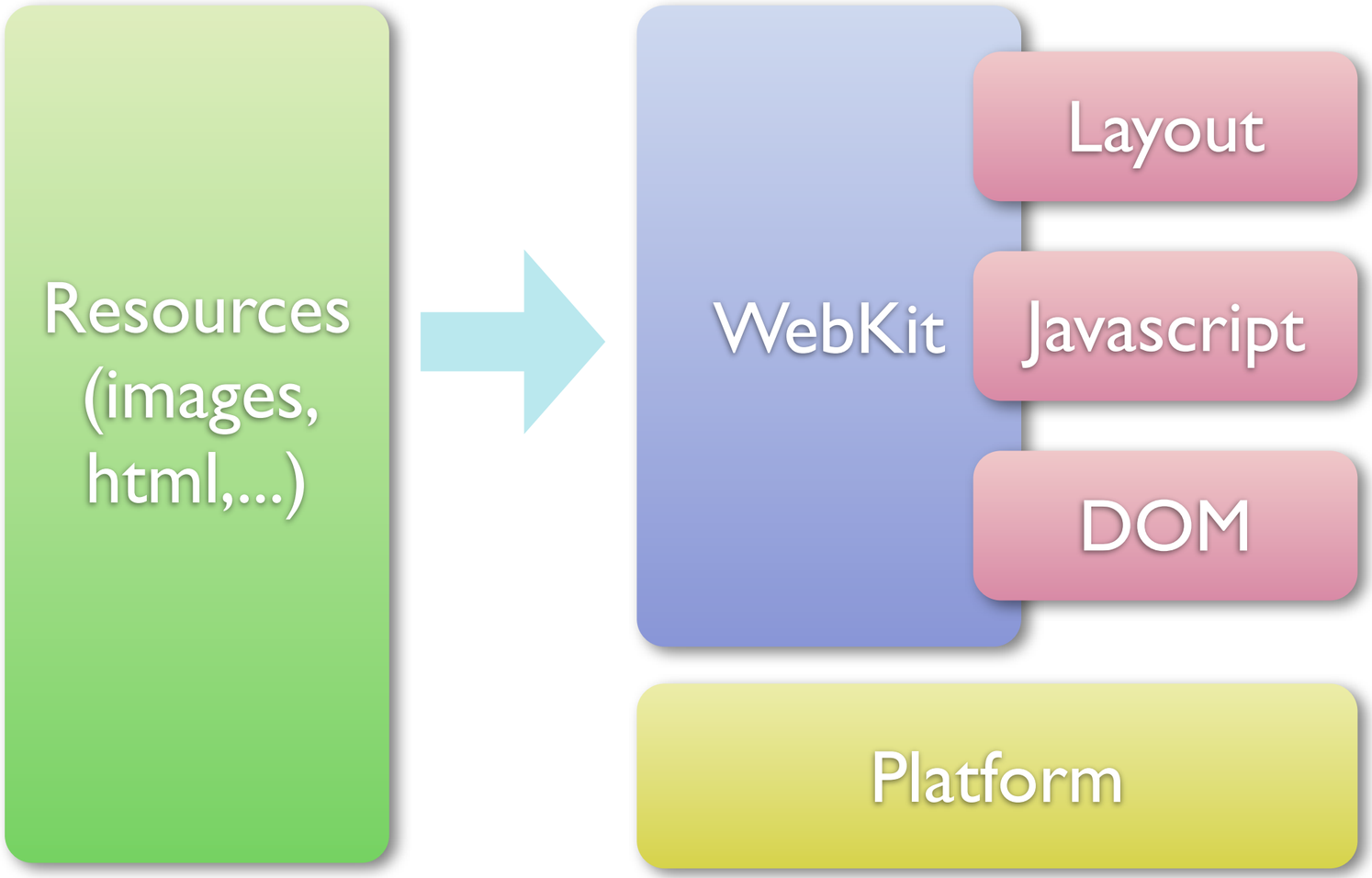
Overview



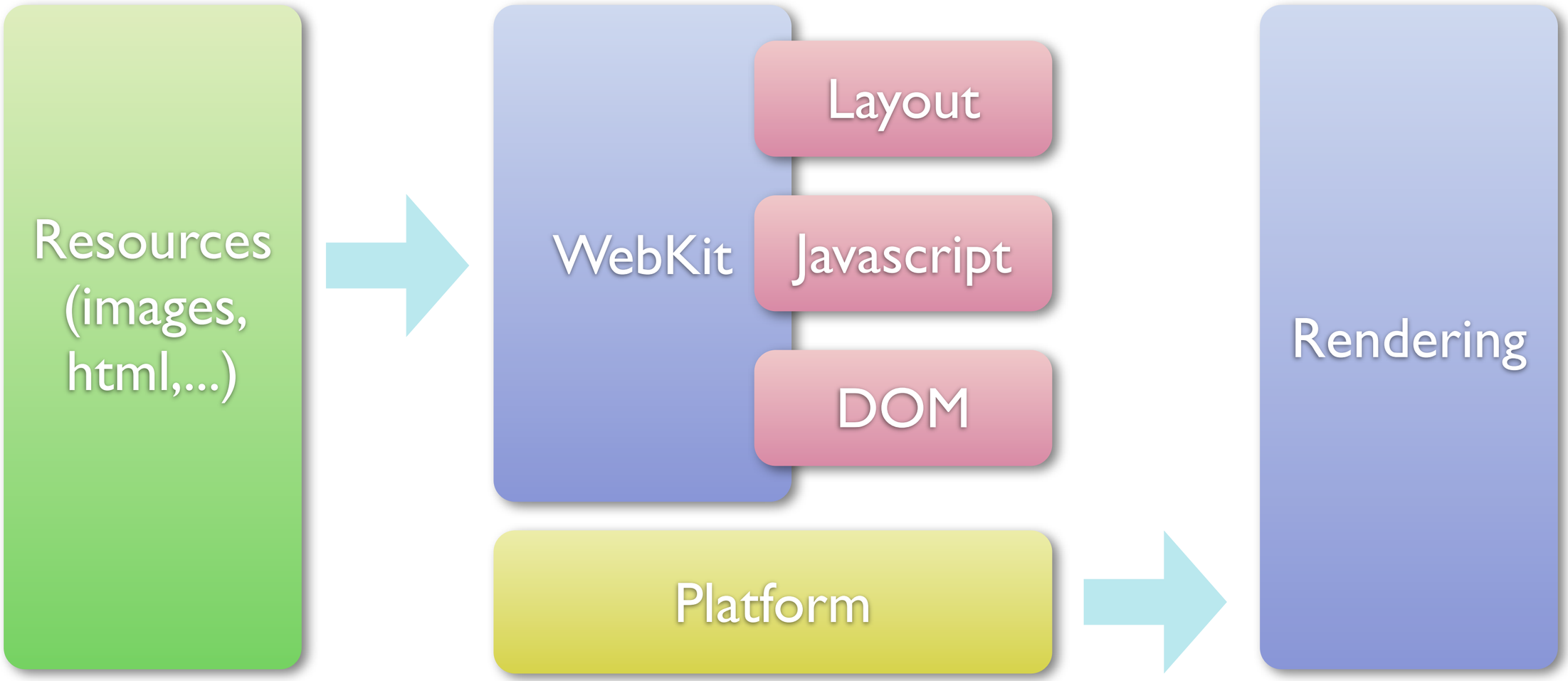
Overview



Overview


Resources
(images,
html,...)
WebKit
Layout
Javascript
DOM
Platform

Overview


Resources
(images,
html,...)
WebKit
Layout
Javascript
DOM
Platform
Rendering

Platform

Platform implementation

Platform implementation

Network / Disk access

Platform implementation

Network / Disk access

System integration

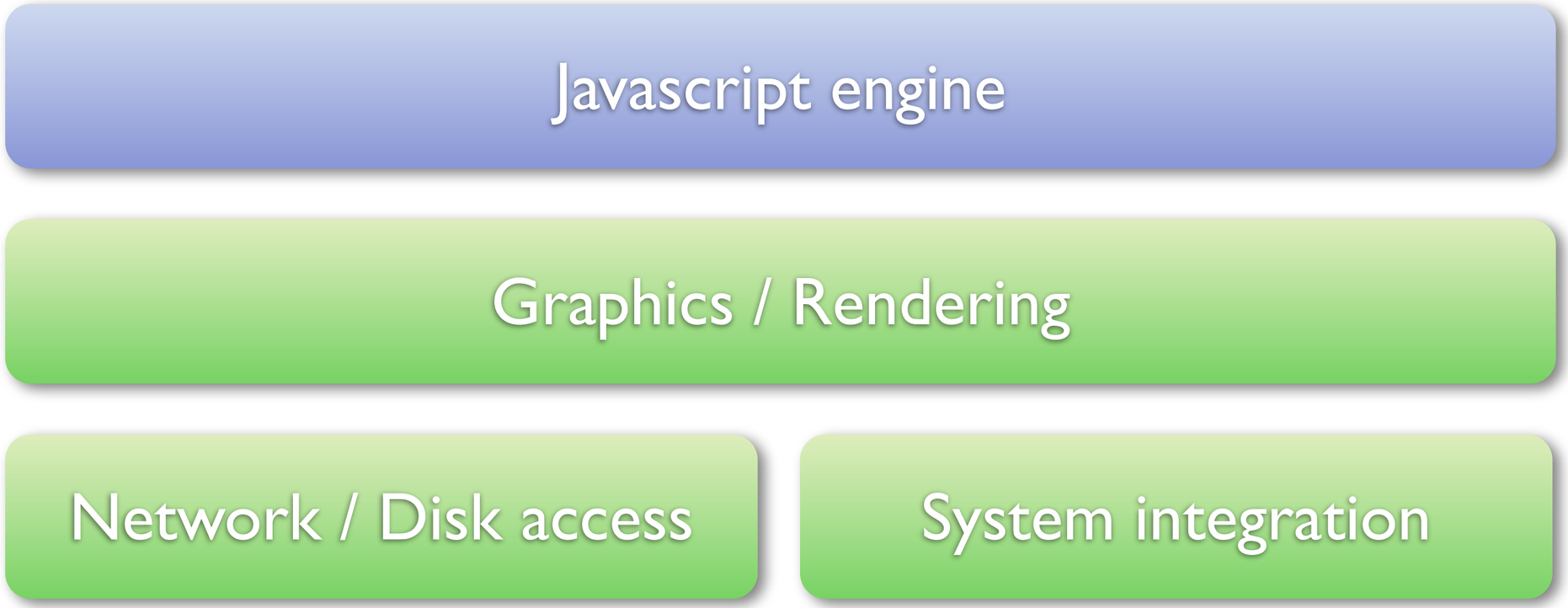
Platform implementation

Graphics / Rendering

Network / Disk access

System integration

Platform implementation

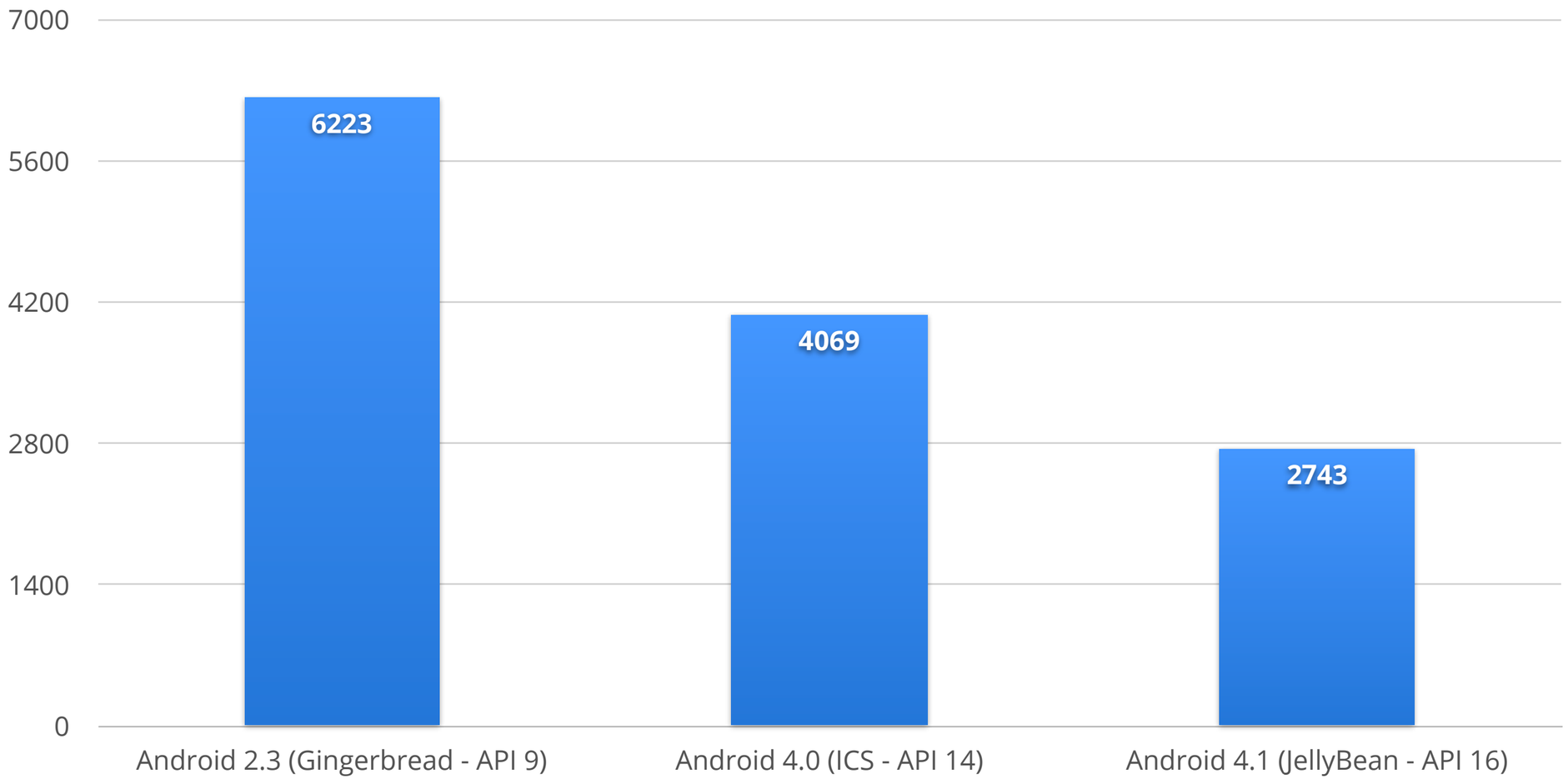


System integration

- **Eclair**
  - Database API support
  - Application cache support (offline webapps)
  - Geolocation API
  - HTML5 video (fullscreen)
- **Honeycomb**
  - Media capture (camera)
  - Device Orientation

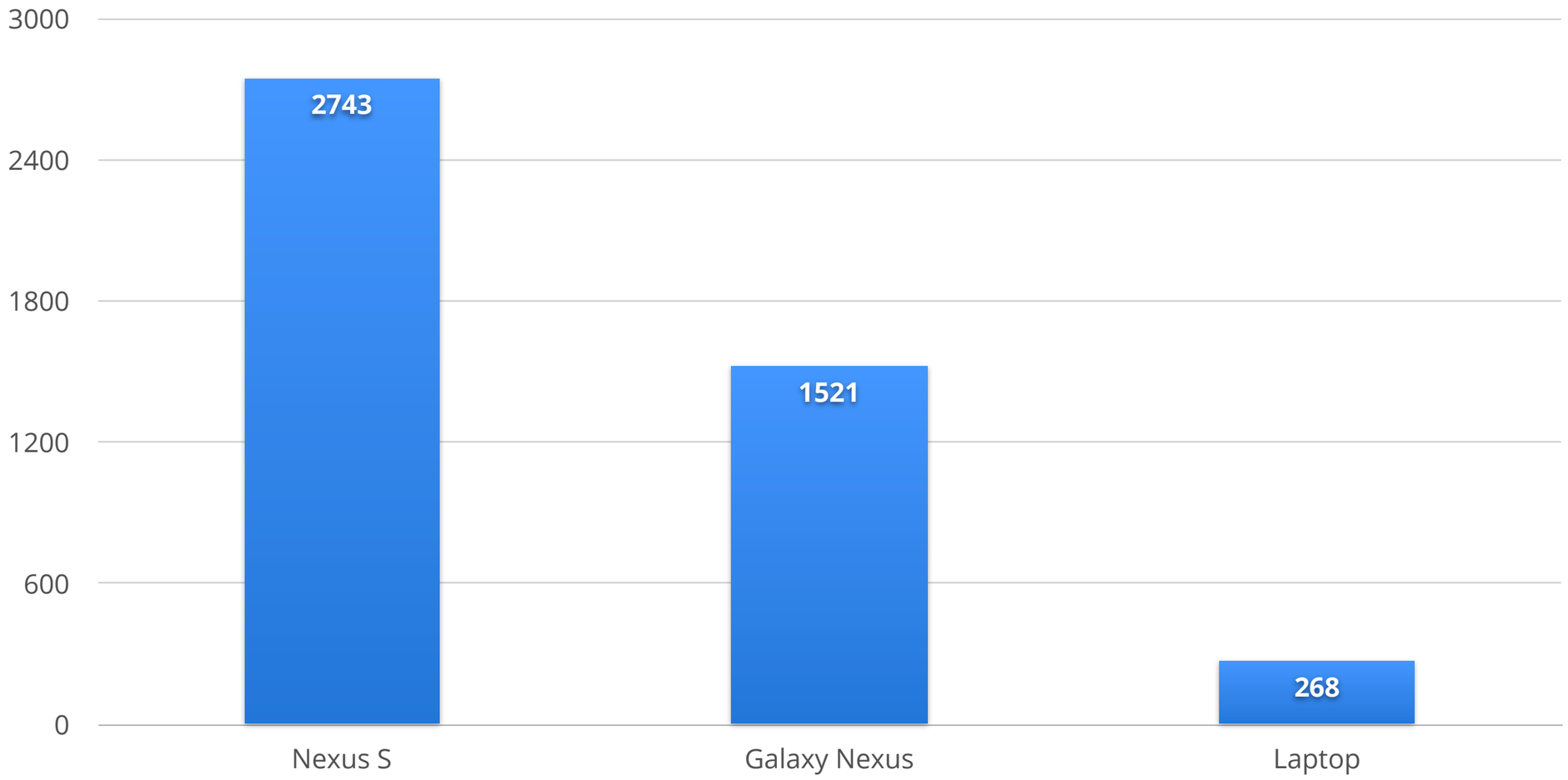
Sunspider benchmark, Nexus S

(lower is better)



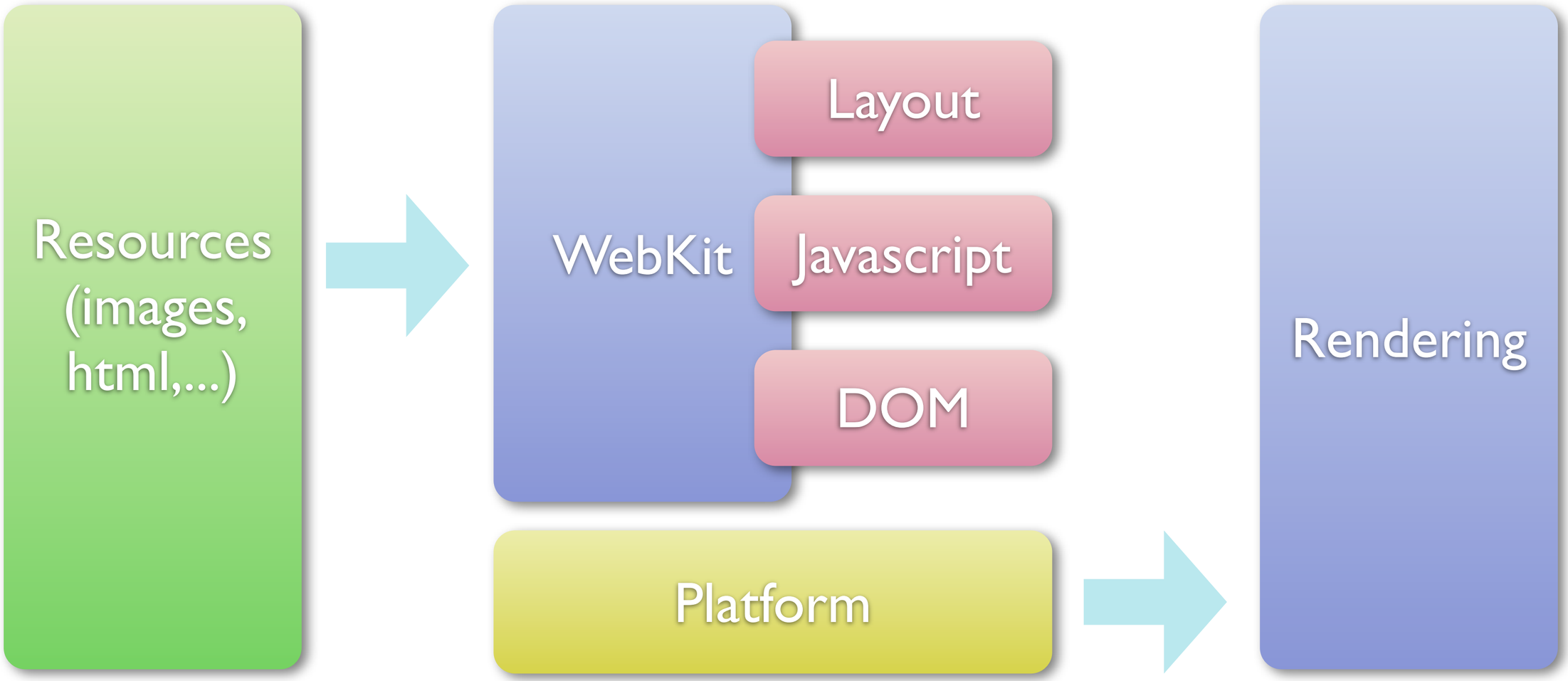
Sunspider benchmark

(lower is better)



Rendering

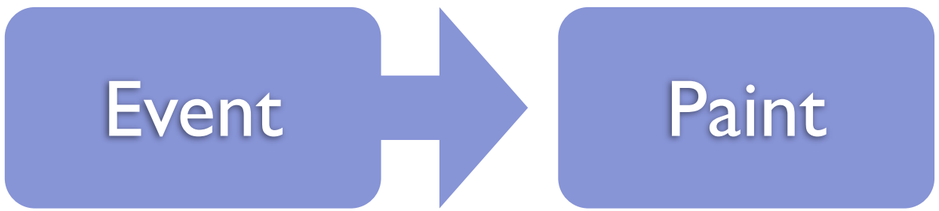
Overview


Resources
(images,
html,...)
WebKit
Layout
Javascript
DOM
Platform
Rendering

Rendering Loop

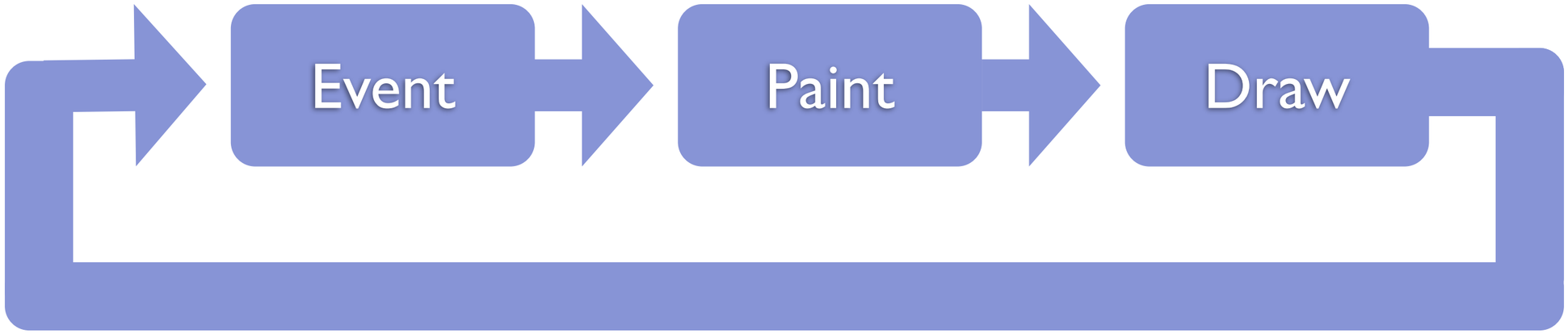
Rendering Loop

Rendering Loop



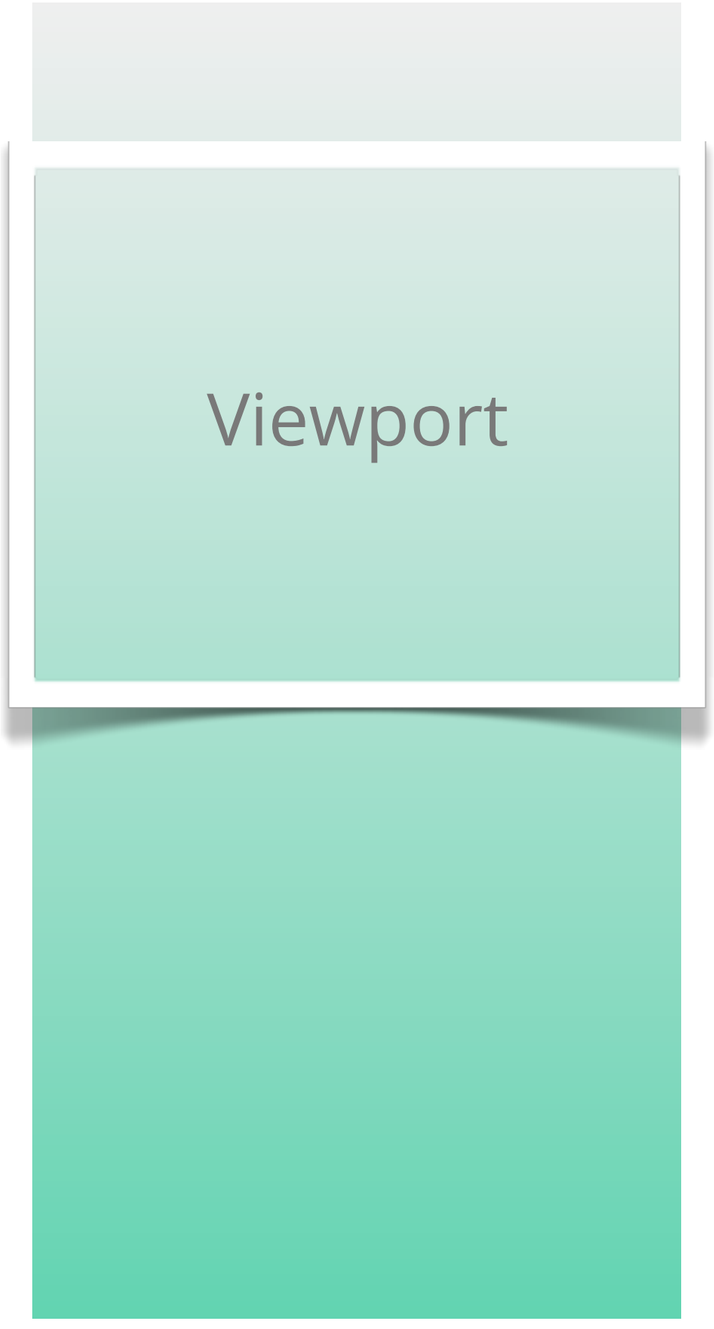
Rendering Loop

Rendering Loop

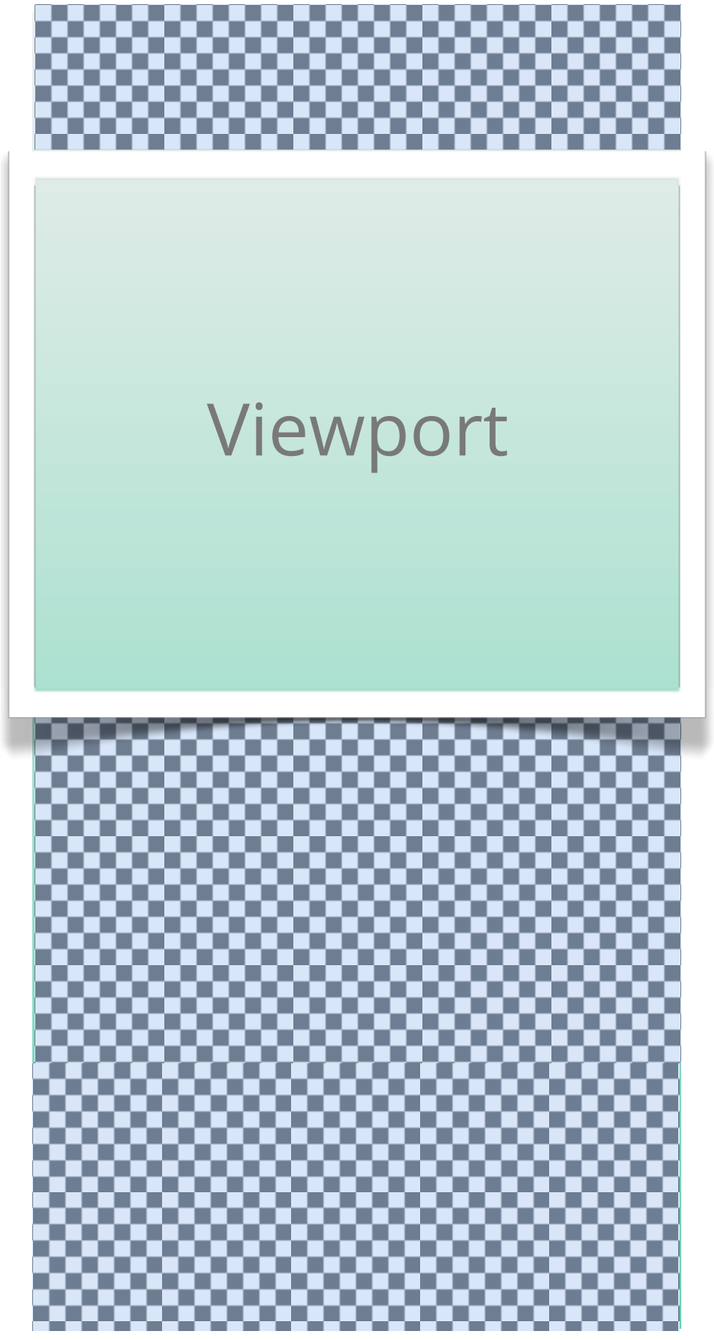


Rendering

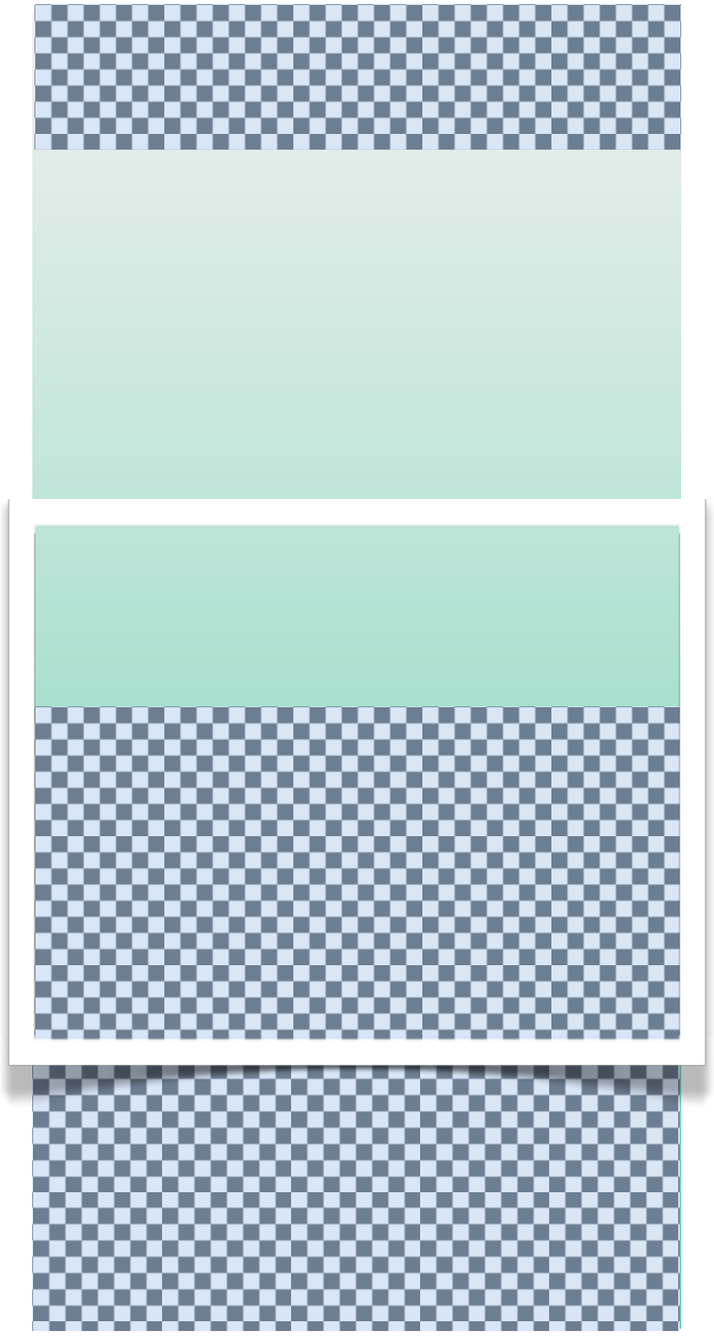
Rendering



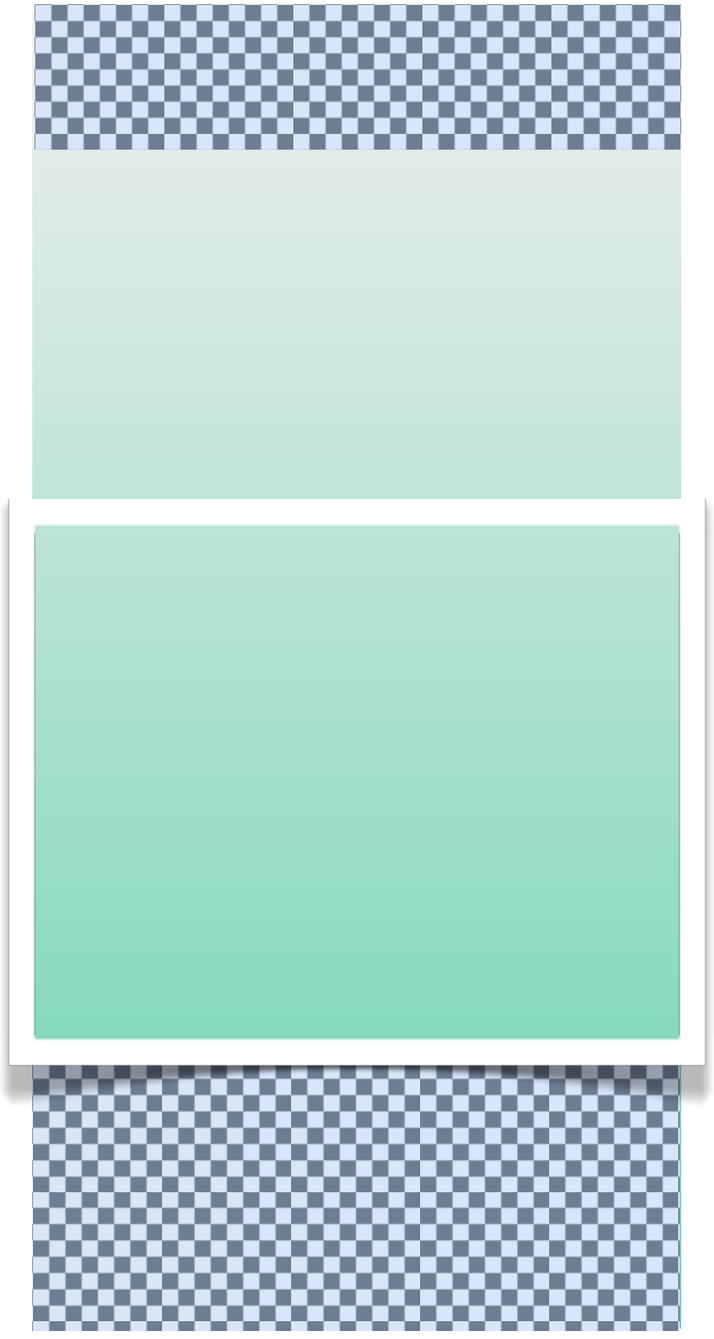
Rendering



Rendering

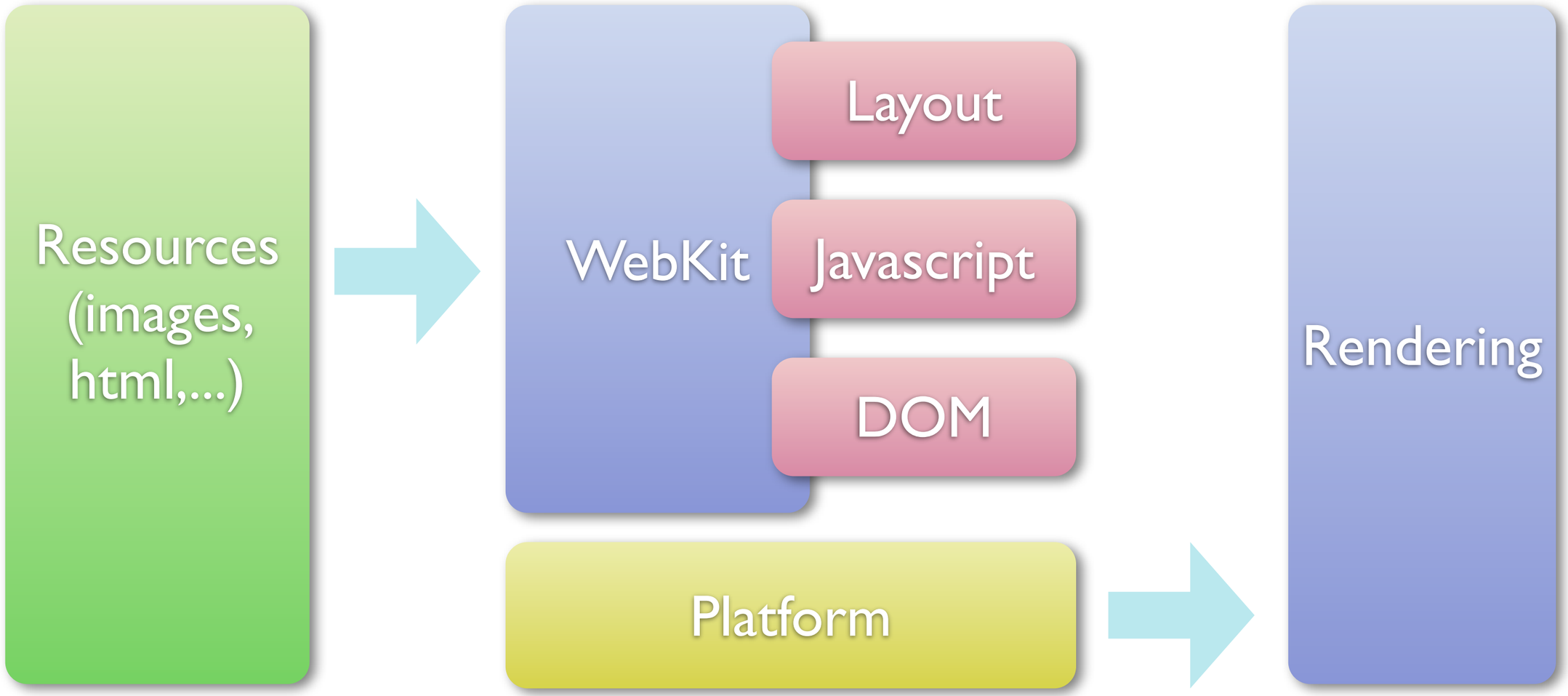


Rendering

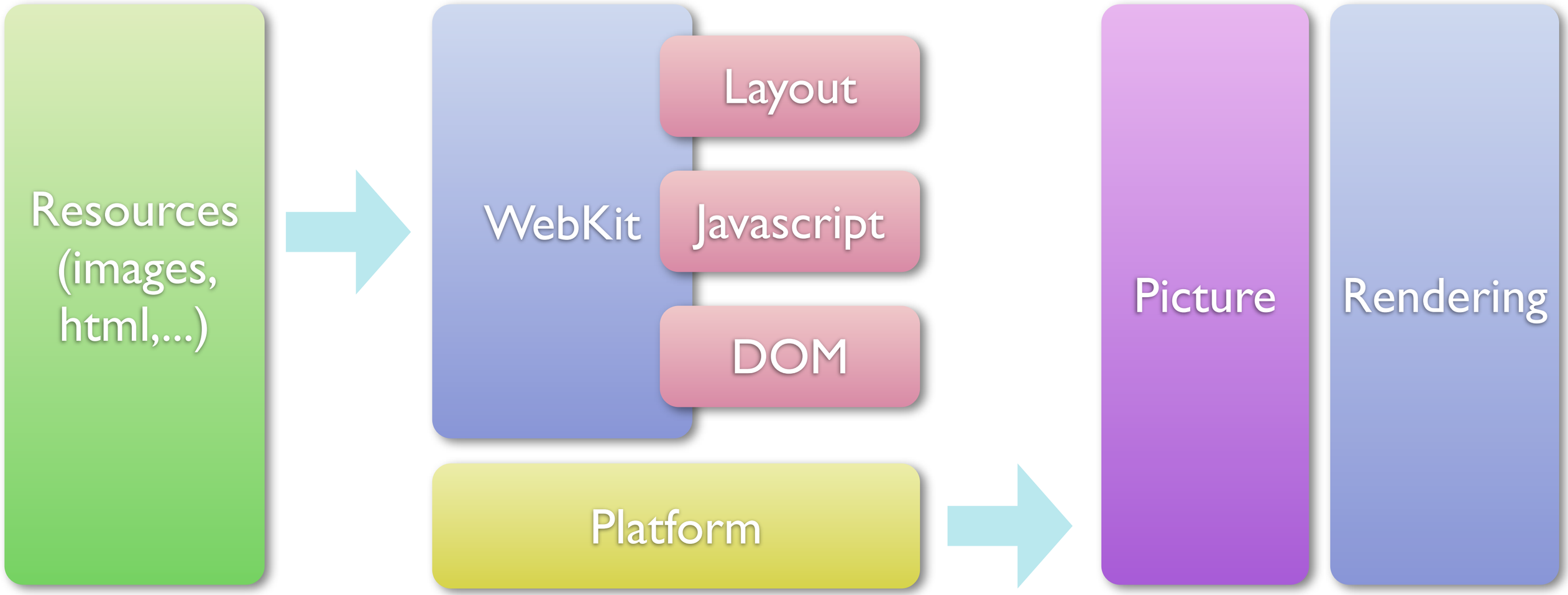


Before Honeycomb

Android WebView Software Rendering



Android WebView Software Rendering

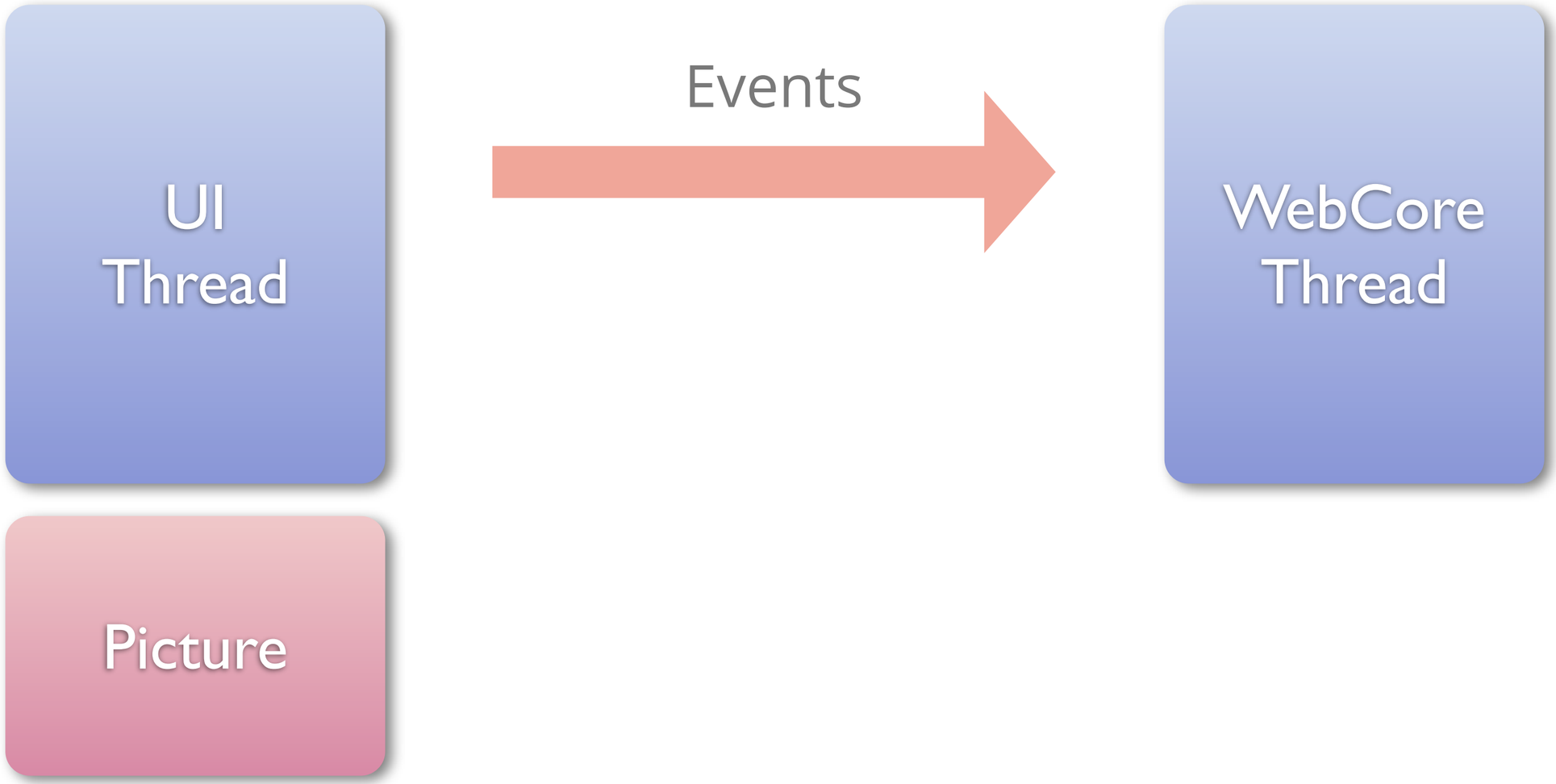


Picture

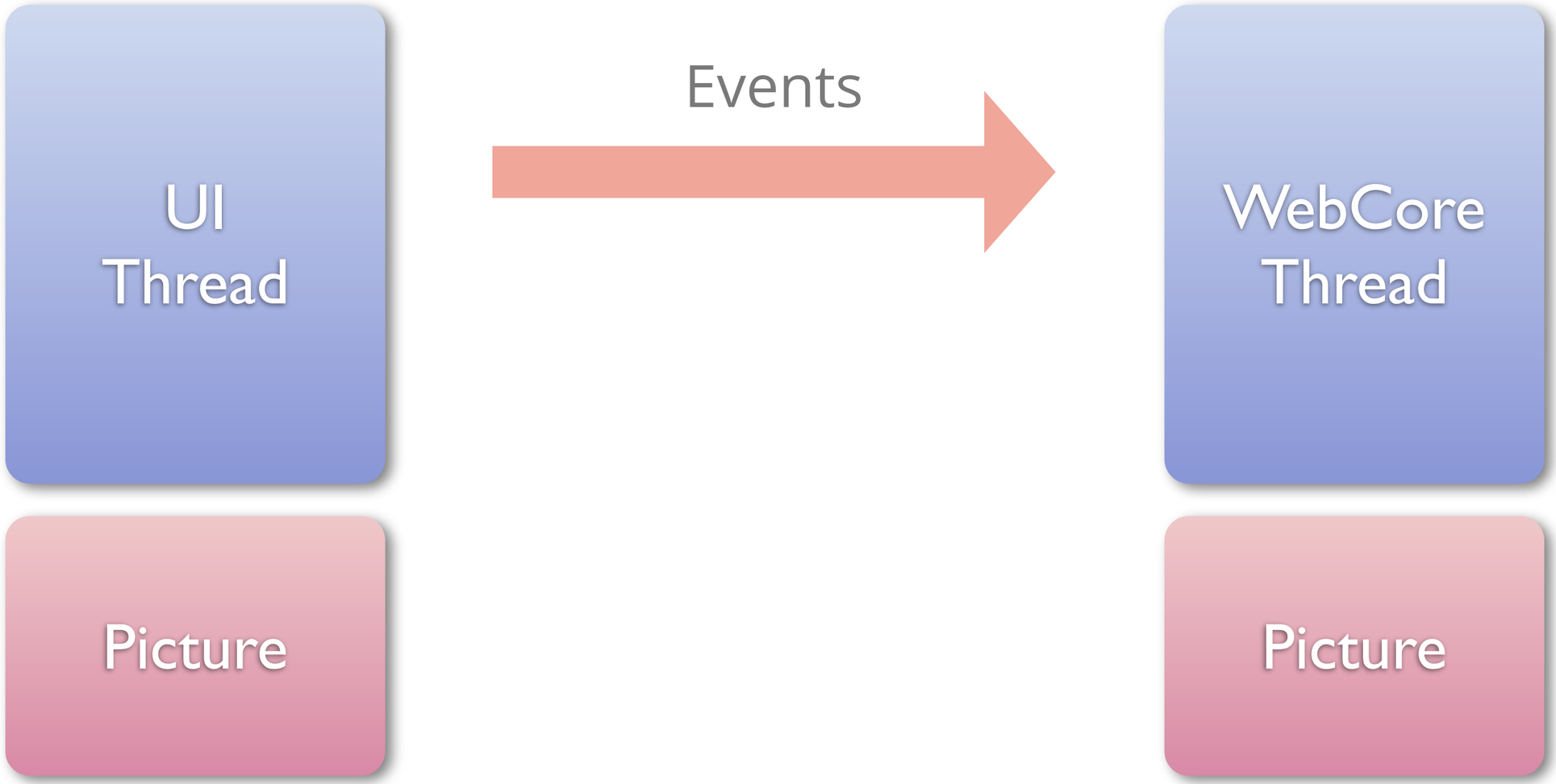
- Vector representation of the entire page (not just the visible area)
- No need to go back to webkit when:
    - Scrolling
    - Zooming

Multithreading

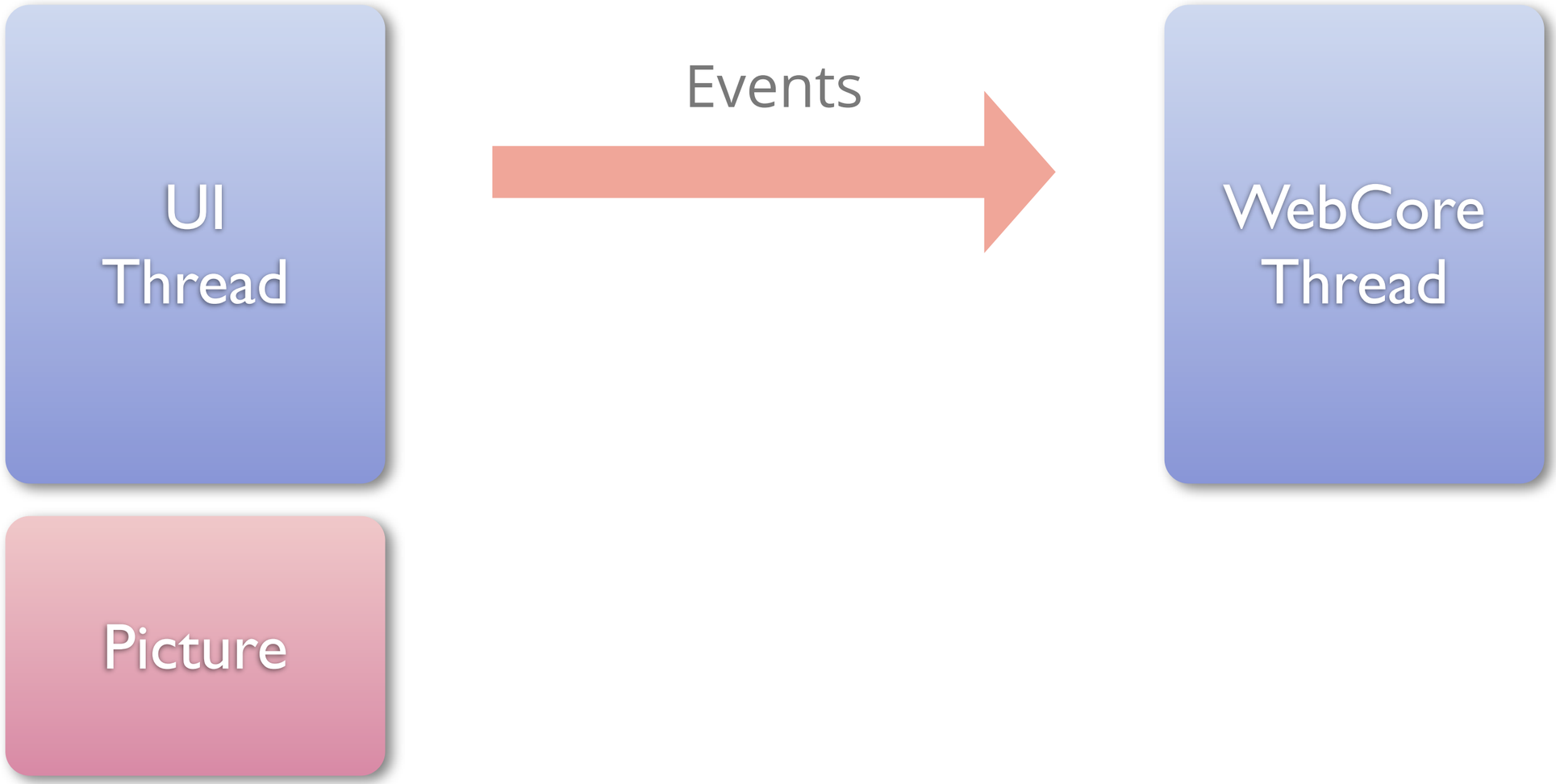
Multithreading



Multithreading



Multithreading



Problems with software rendering

Problems with software rendering

- Speed is highly dependent on the time spent traversing and rasterizing the PictureSet
  - Scroll and Zoom smoothness can vary depending on the website

Problems with software rendering

- Speed is highly dependent on the time spent traversing and rasterizing the PictureSet
  - Scroll and Zoom smoothness can vary depending on the website
- Doesn’t support 3D operations (i.e. for CSS3D)

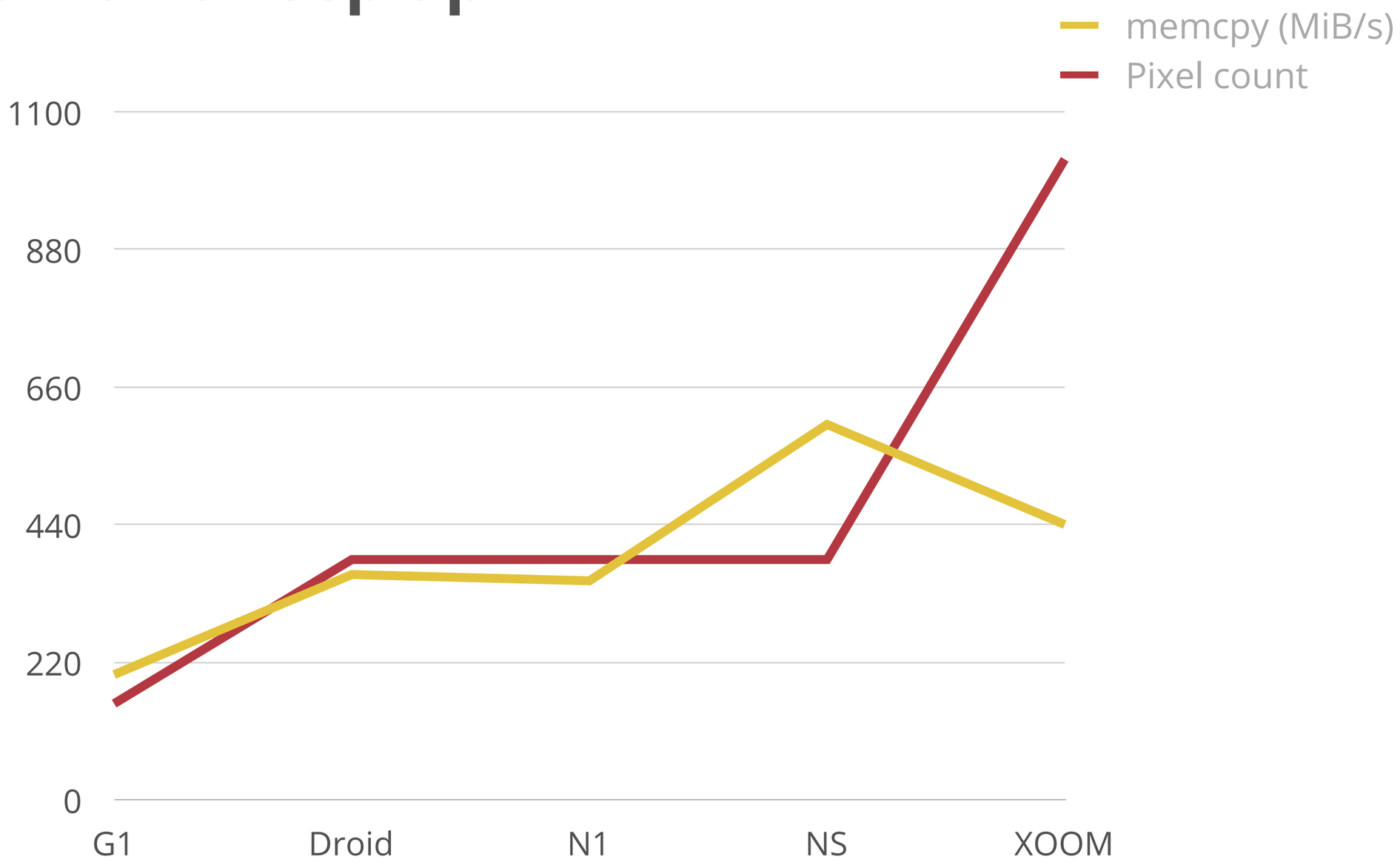
Problems with software rendering

- Speed is highly dependent on the time spent traversing and rasterizing the PictureSet
  - Scroll and Zoom smoothness can vary depending on the website
- Doesn’t support 3D operations (i.e. for CSS3D)
- Poor support of inline plugins or video (cannot have HTML content above a plugin/video; position not in sync when scrolling/zooming)

Honeycomb and beyond

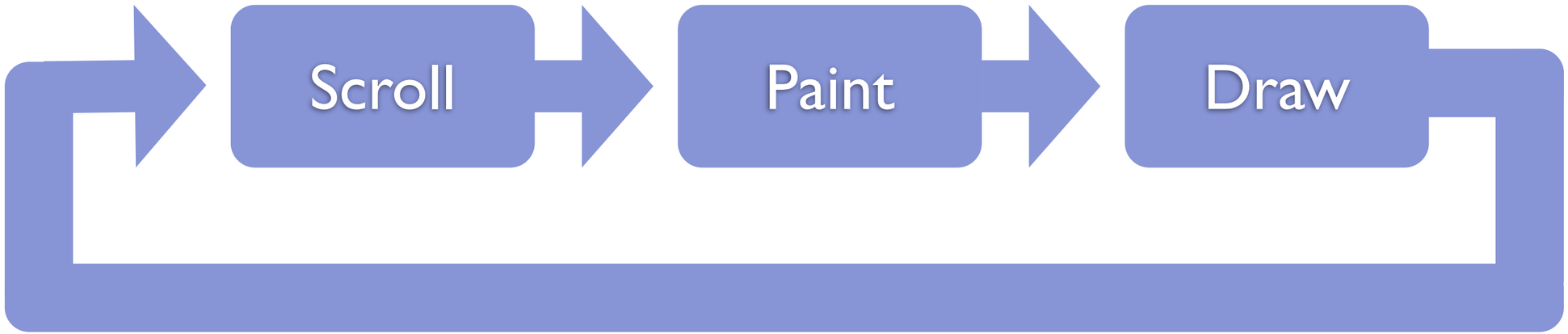
Adding Hardware Acceleration

Software won’t keep up

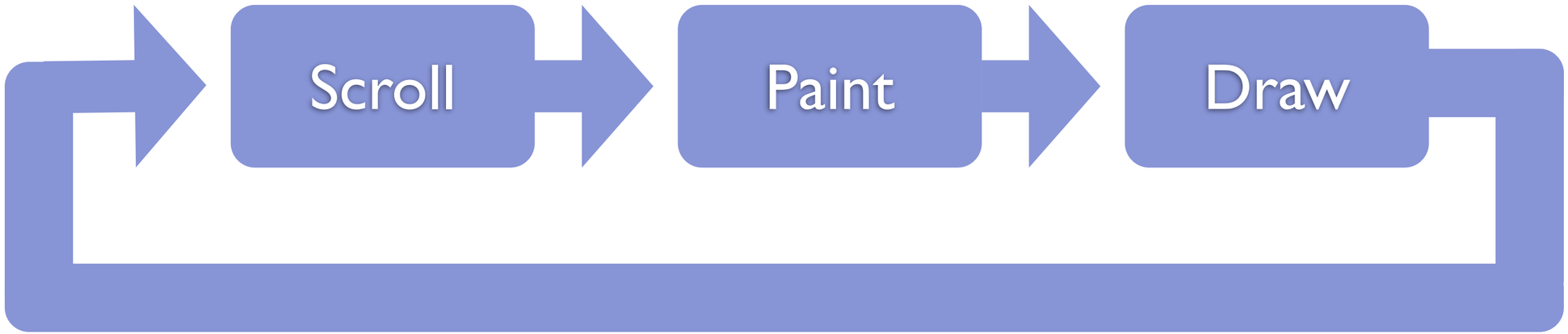


Tiling Content

Software Rendering

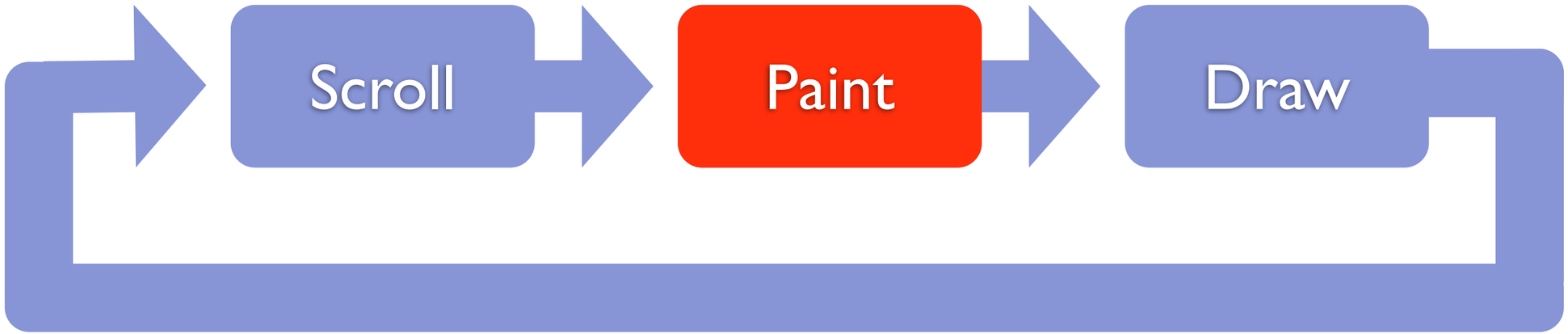


Software Rendering



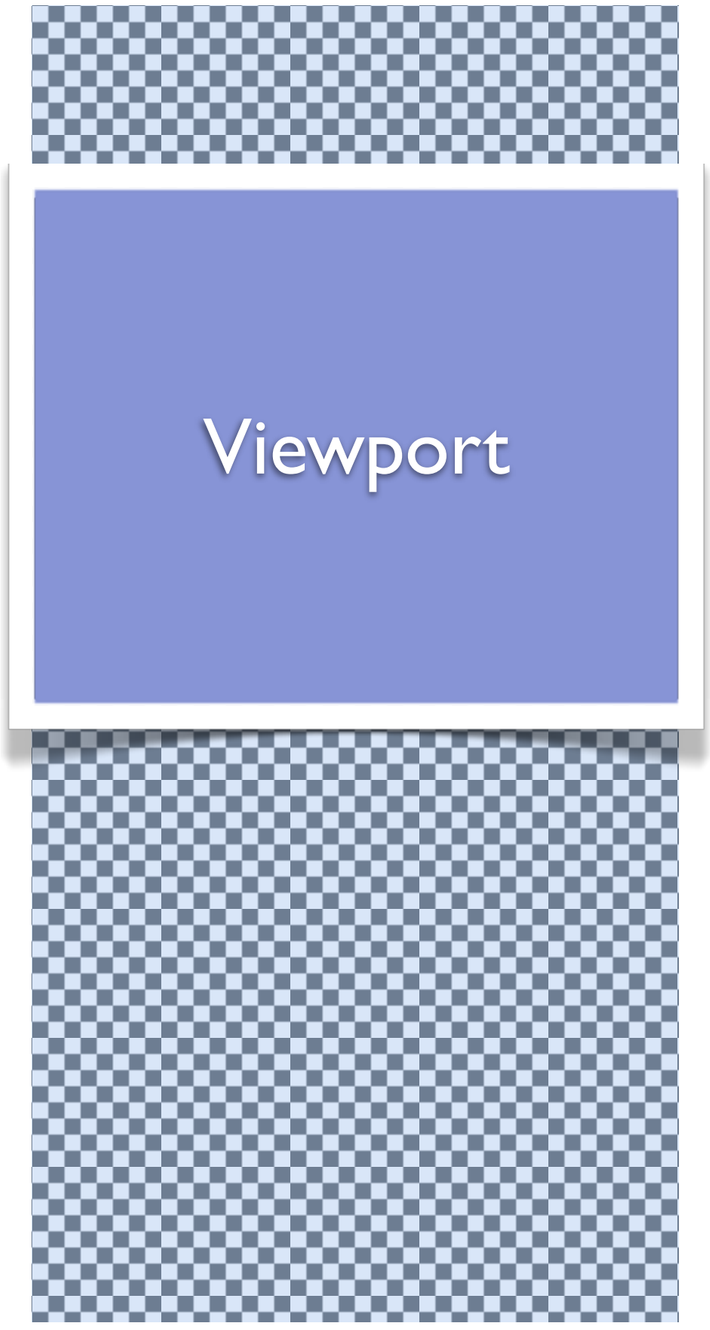
16.66ms

Software Rendering

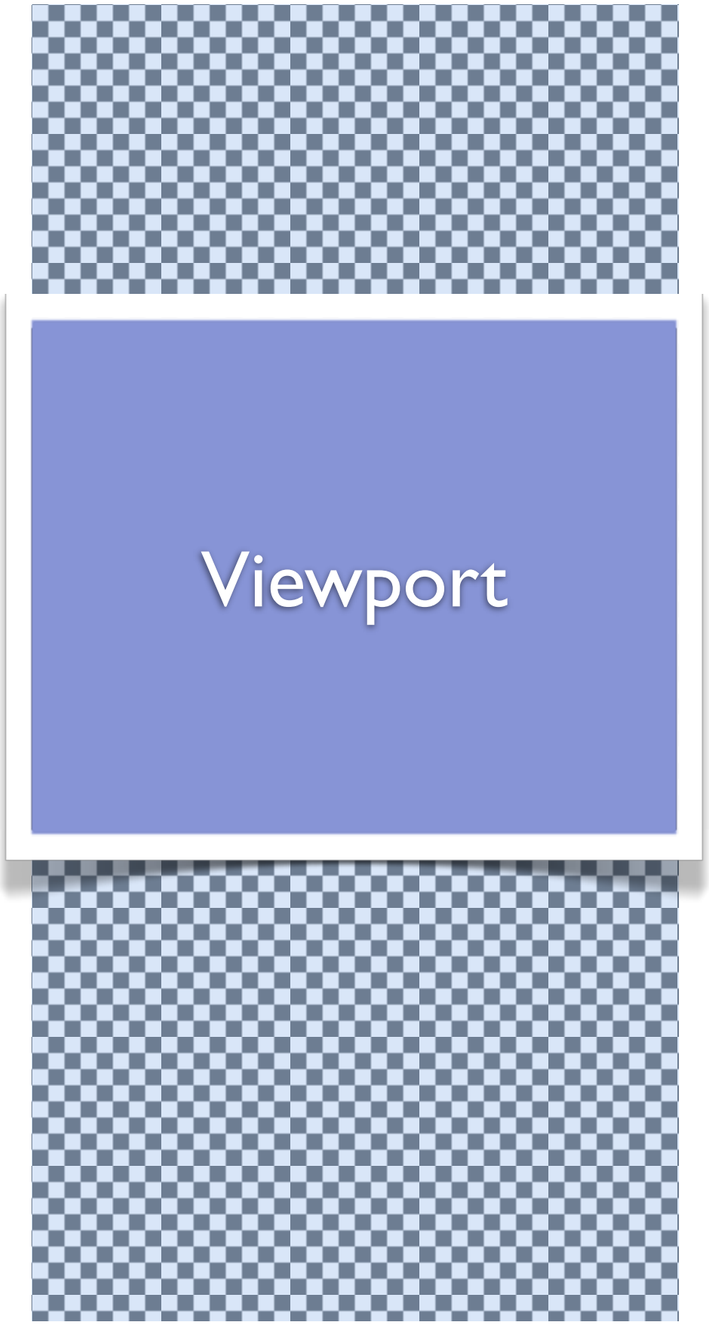


16.66ms

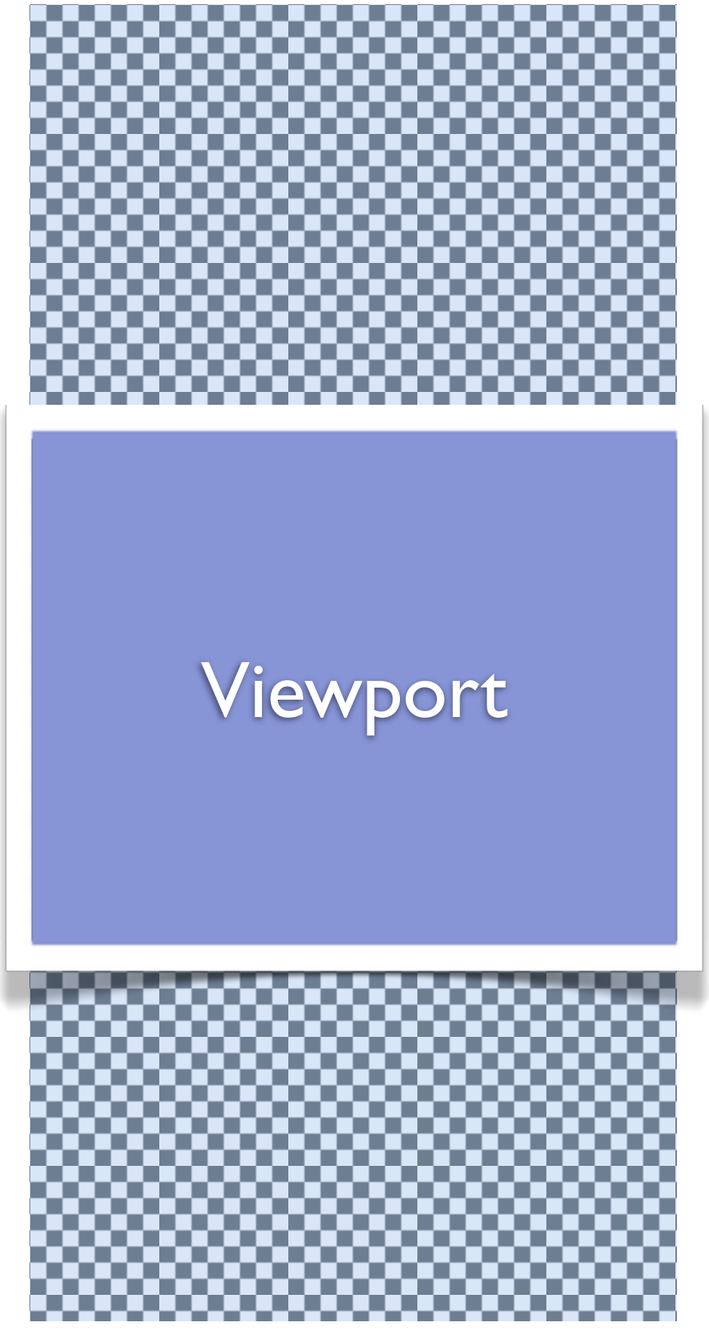
Software Rendering



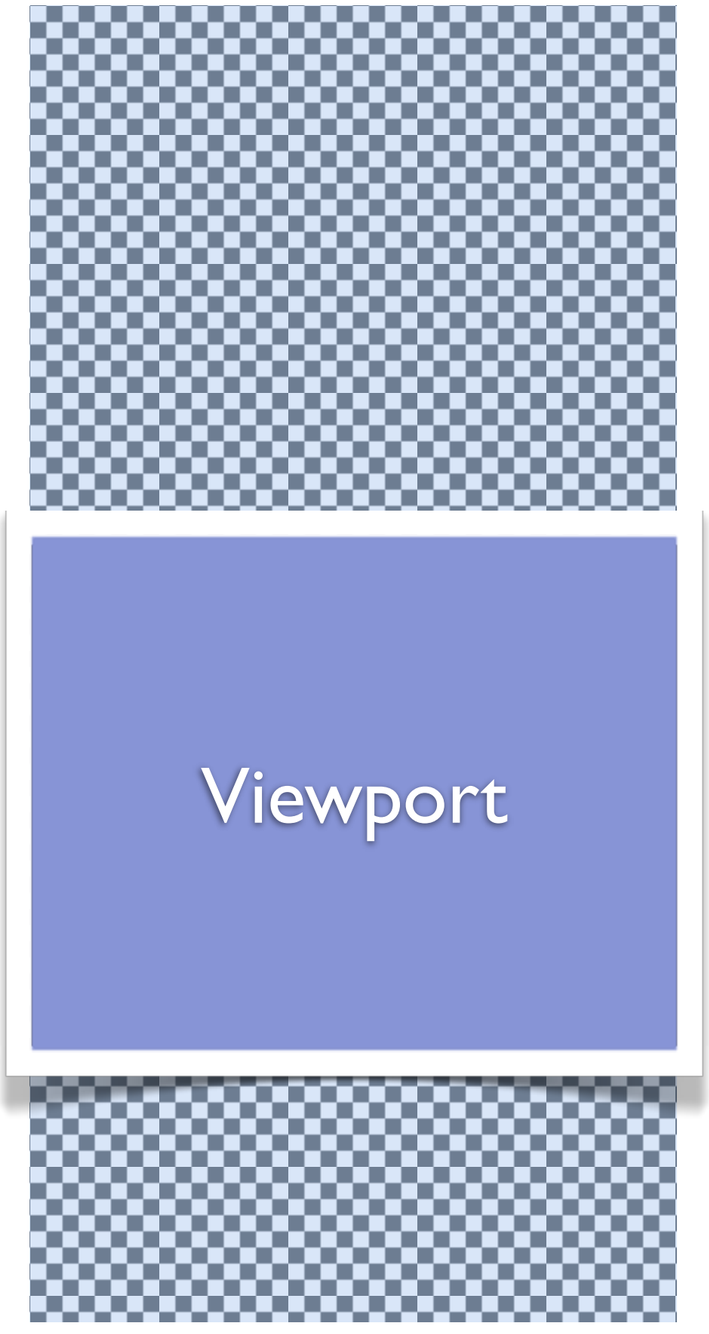
Software Rendering



Software Rendering



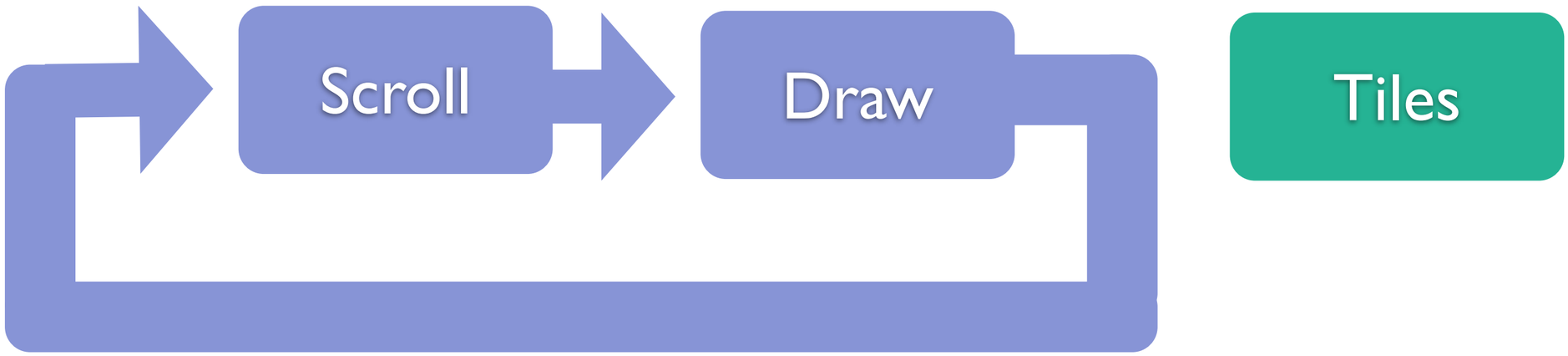
Software Rendering



Ideally...

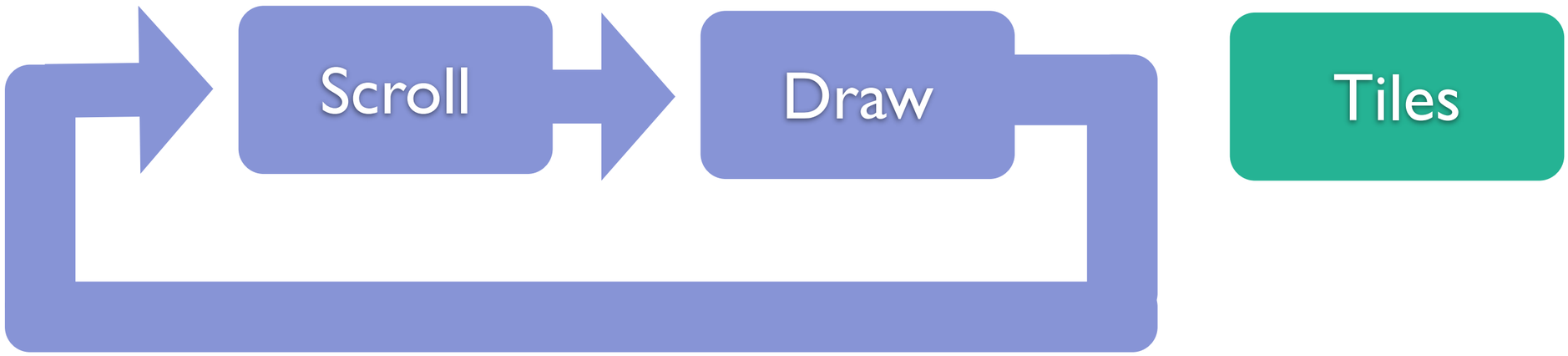
<16.66ms

Hardware Rendering



Hardware Rendering

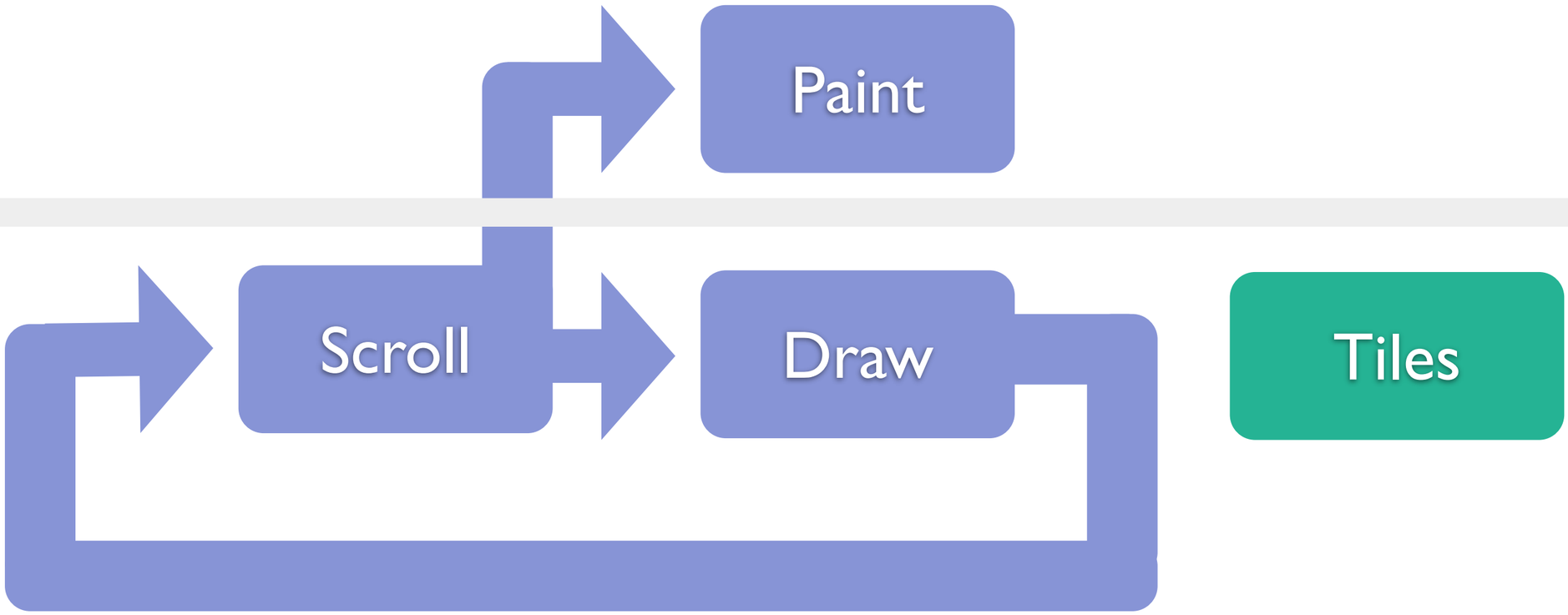
Thread 2



Thread 1

Hardware Rendering

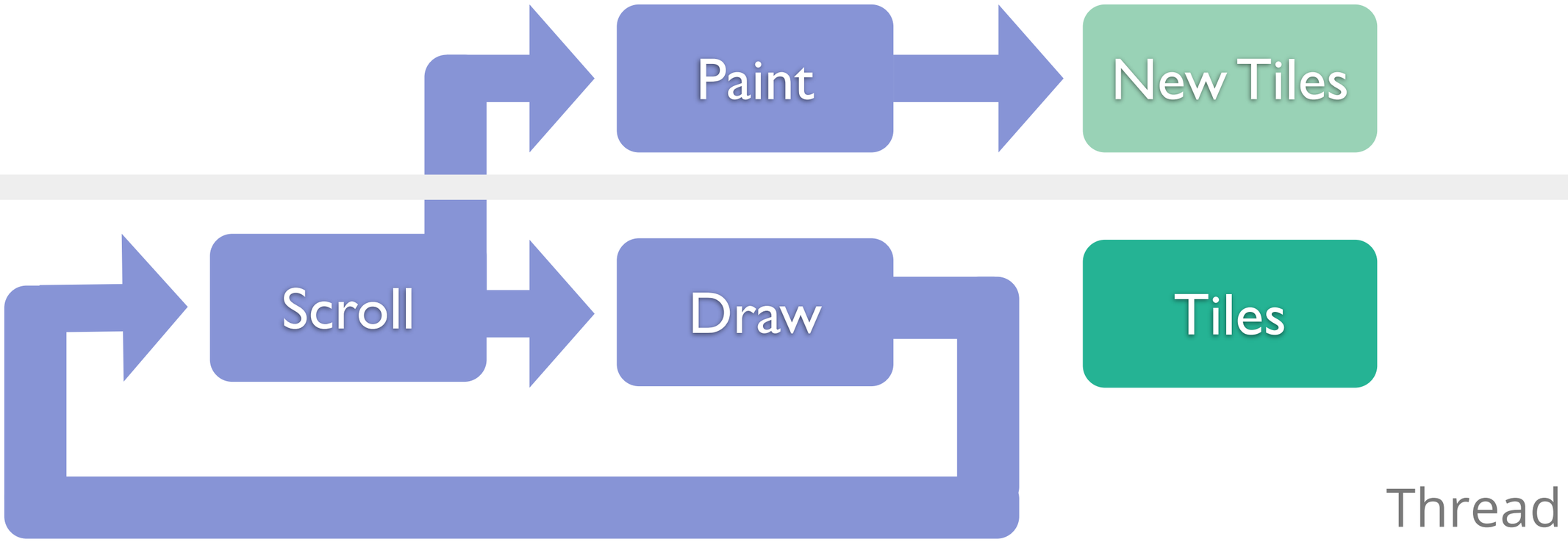
Thread 2



Thread 1

Hardware Rendering

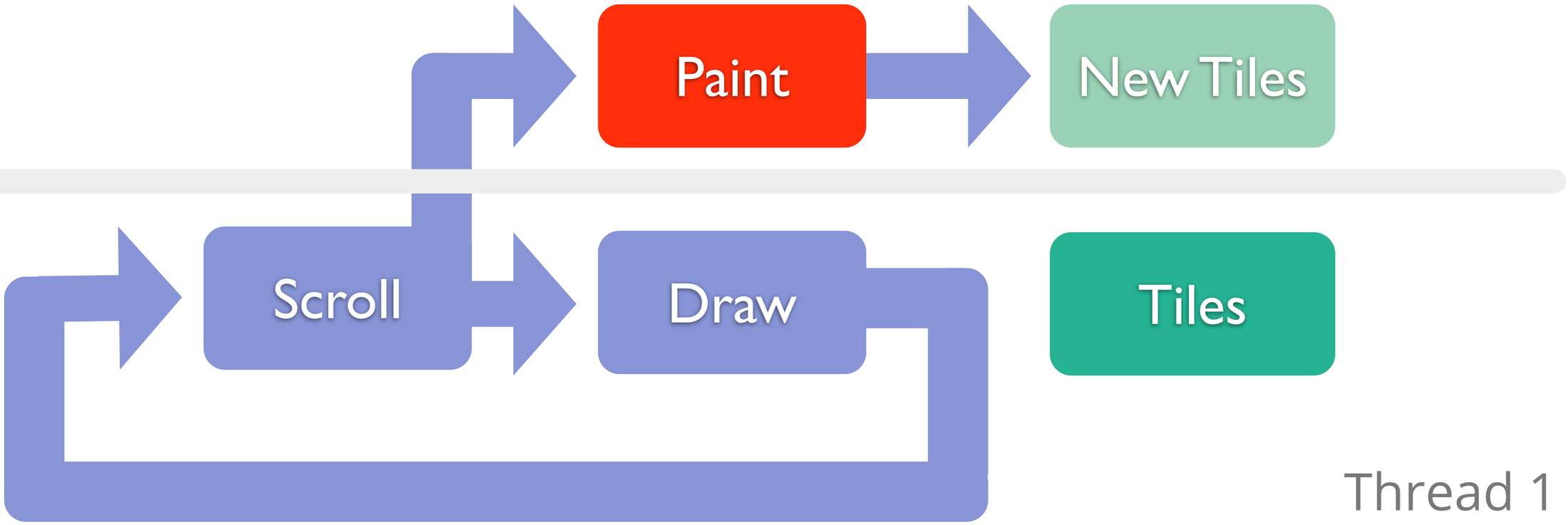
Thread 2



Thread 1

Hardware Rendering

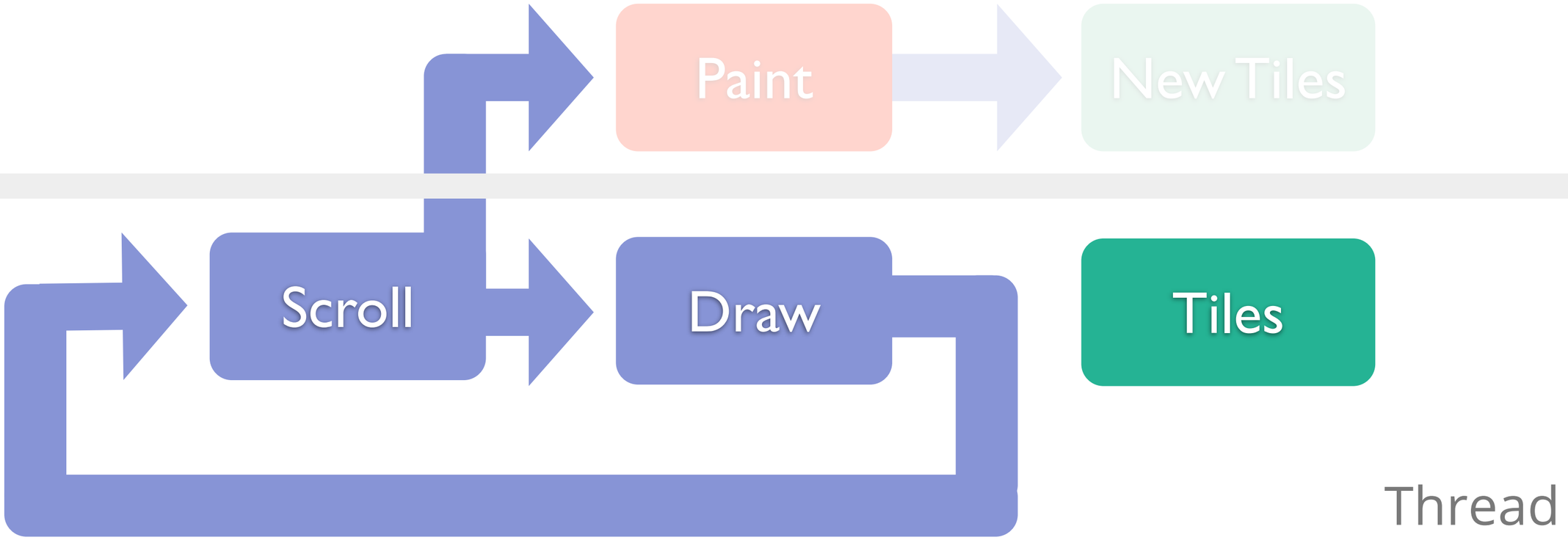
Thread 2



Thread 1

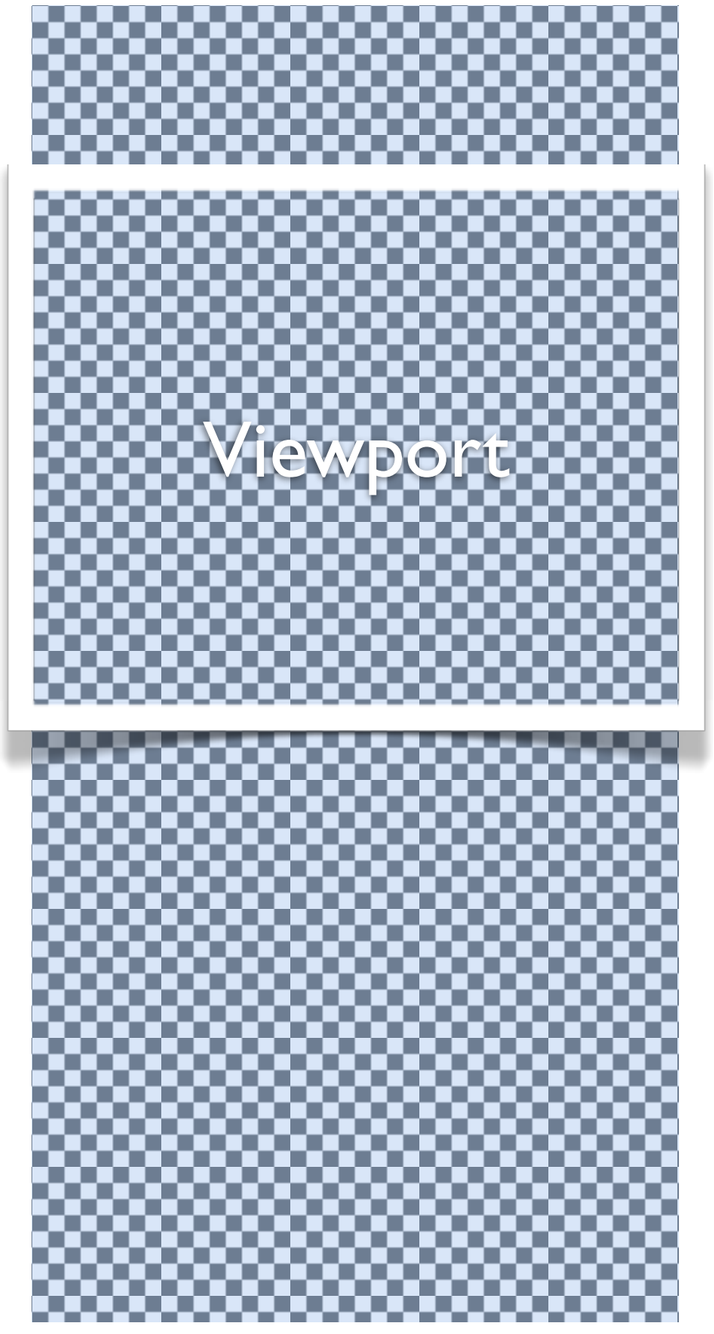
Hardware Rendering

Thread 2

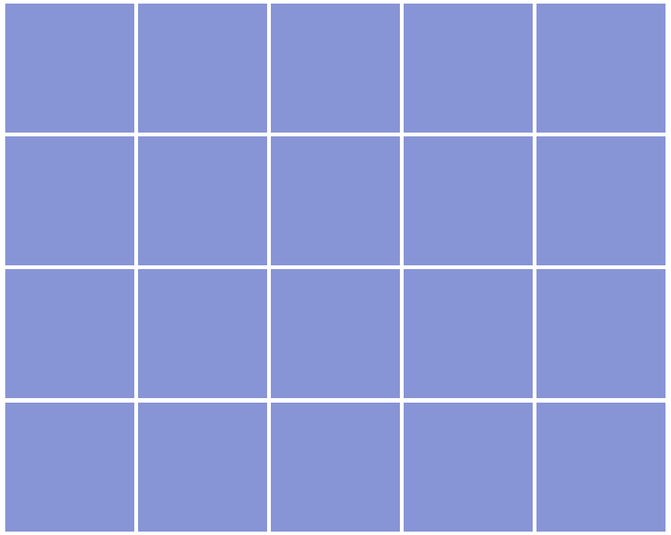


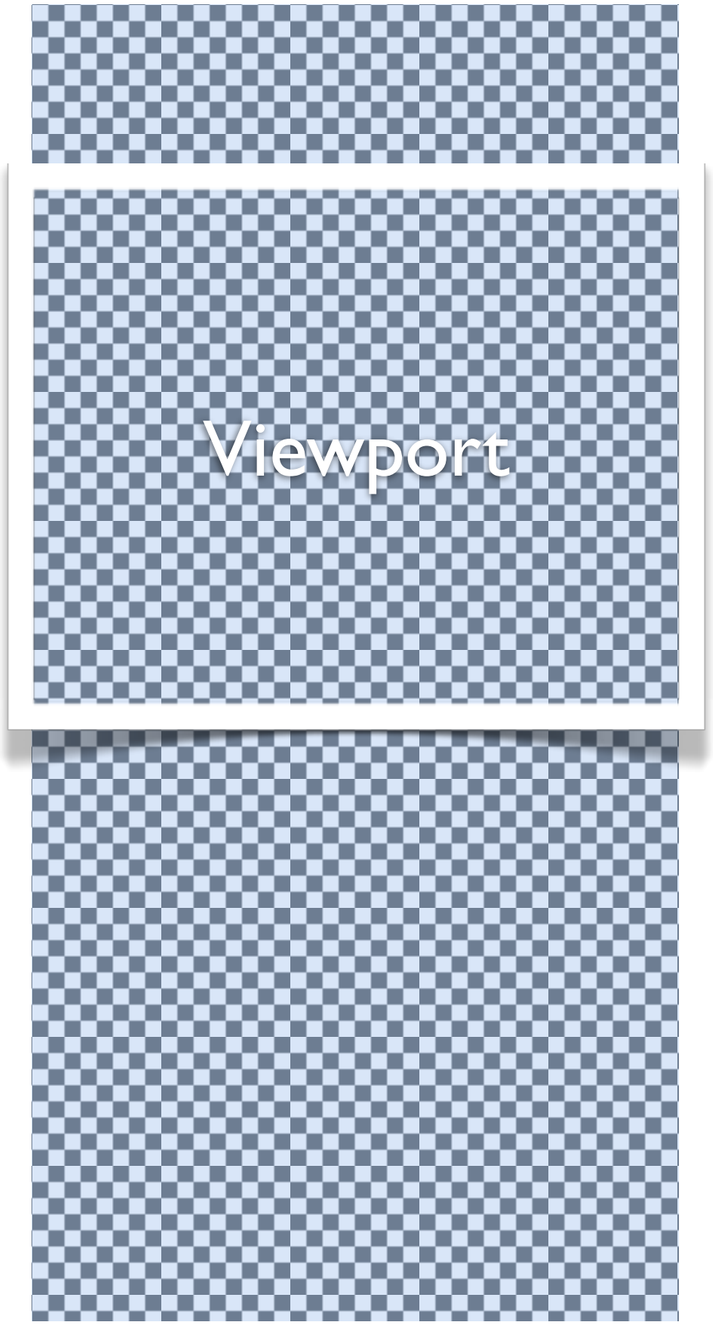
Thread 1

Tiled Rendering

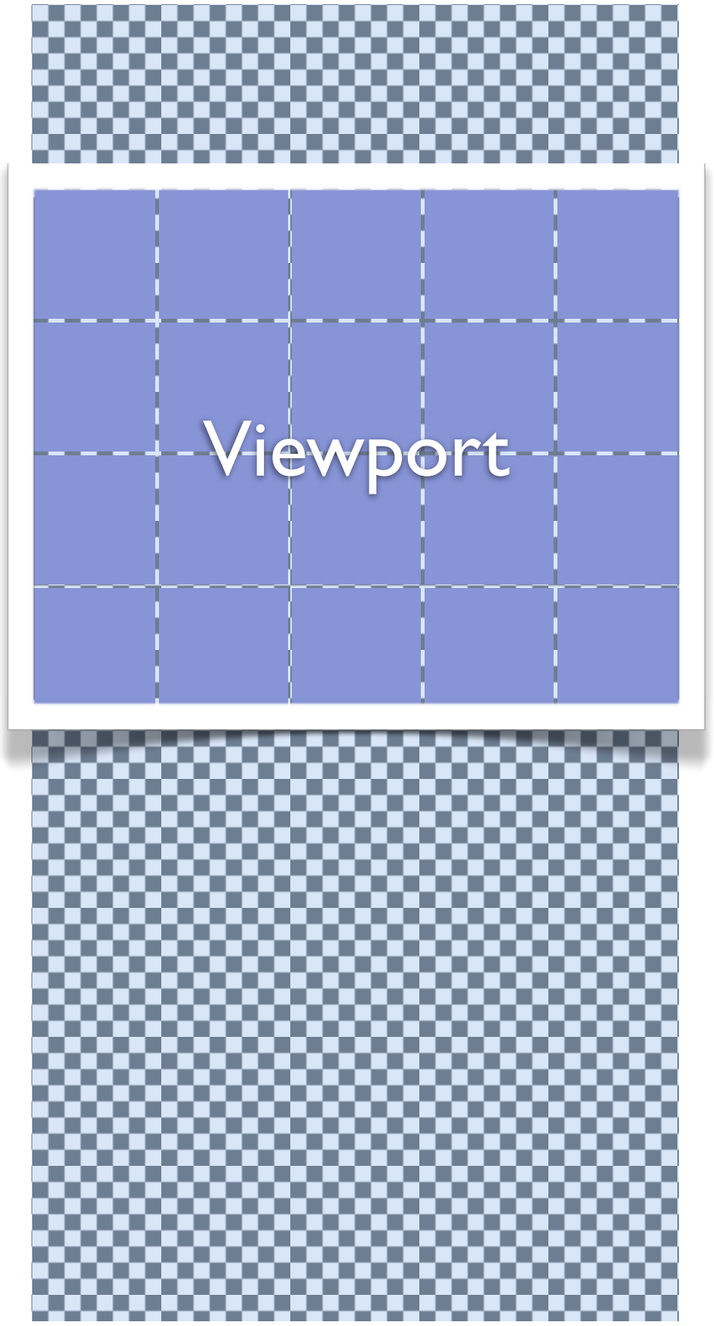


Tiled Rendering

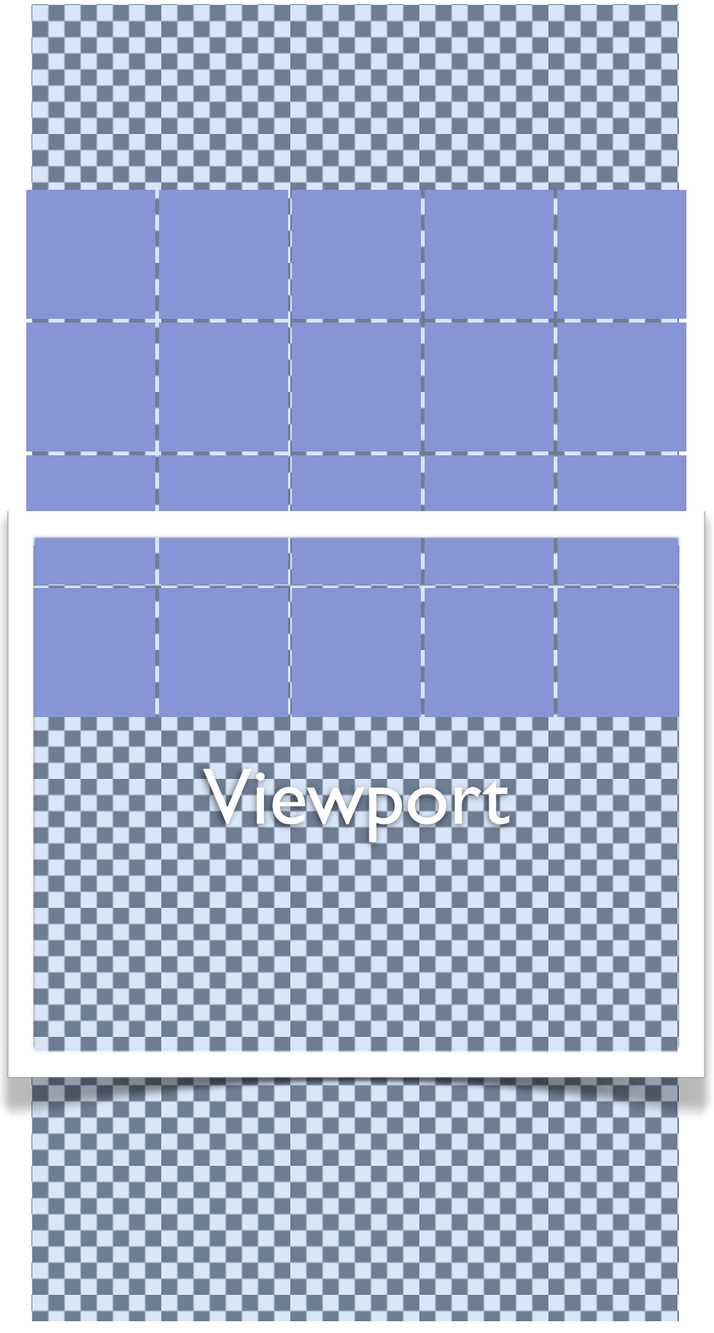




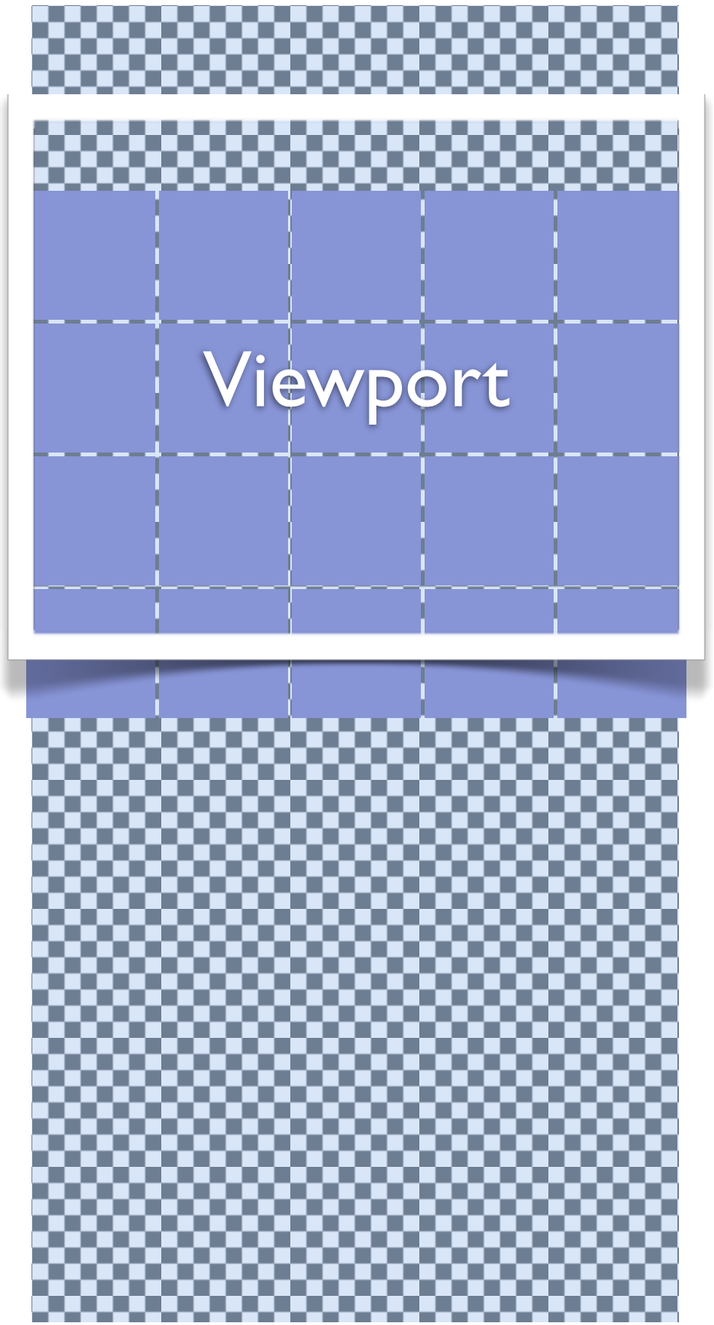
Tiled Rendering



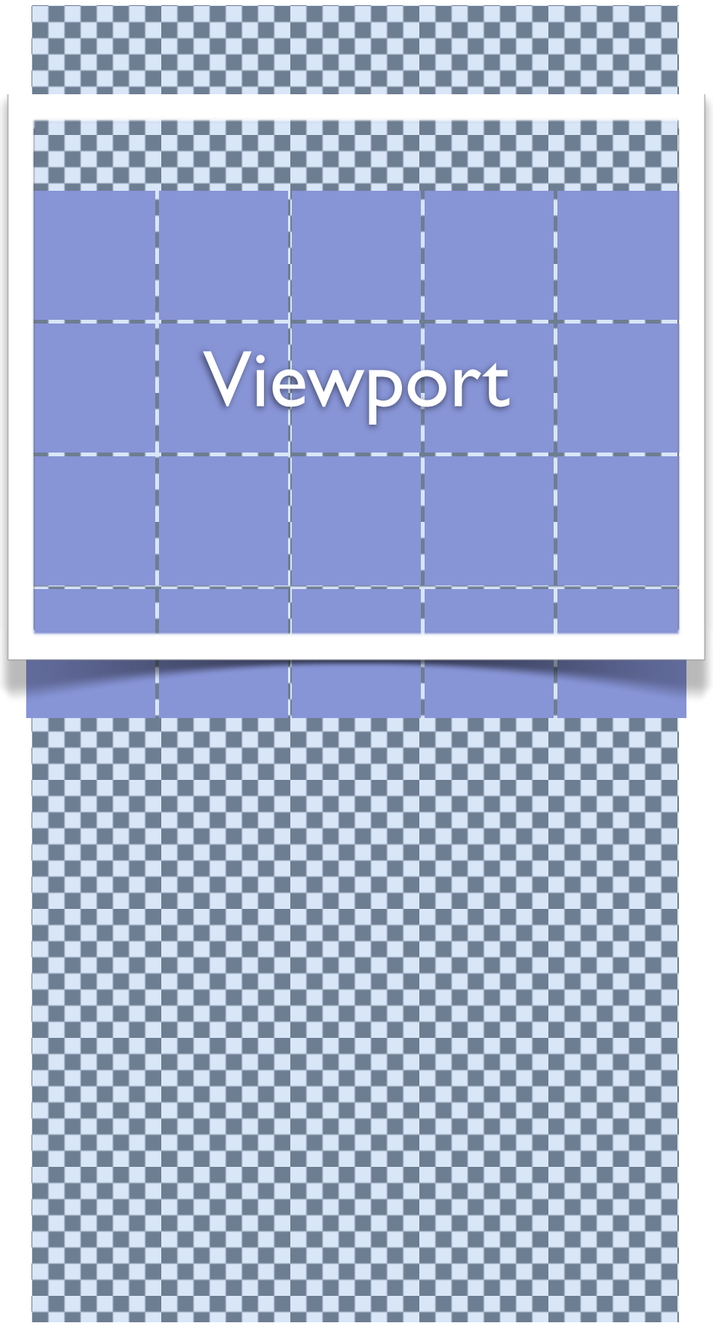
Tiled Rendering



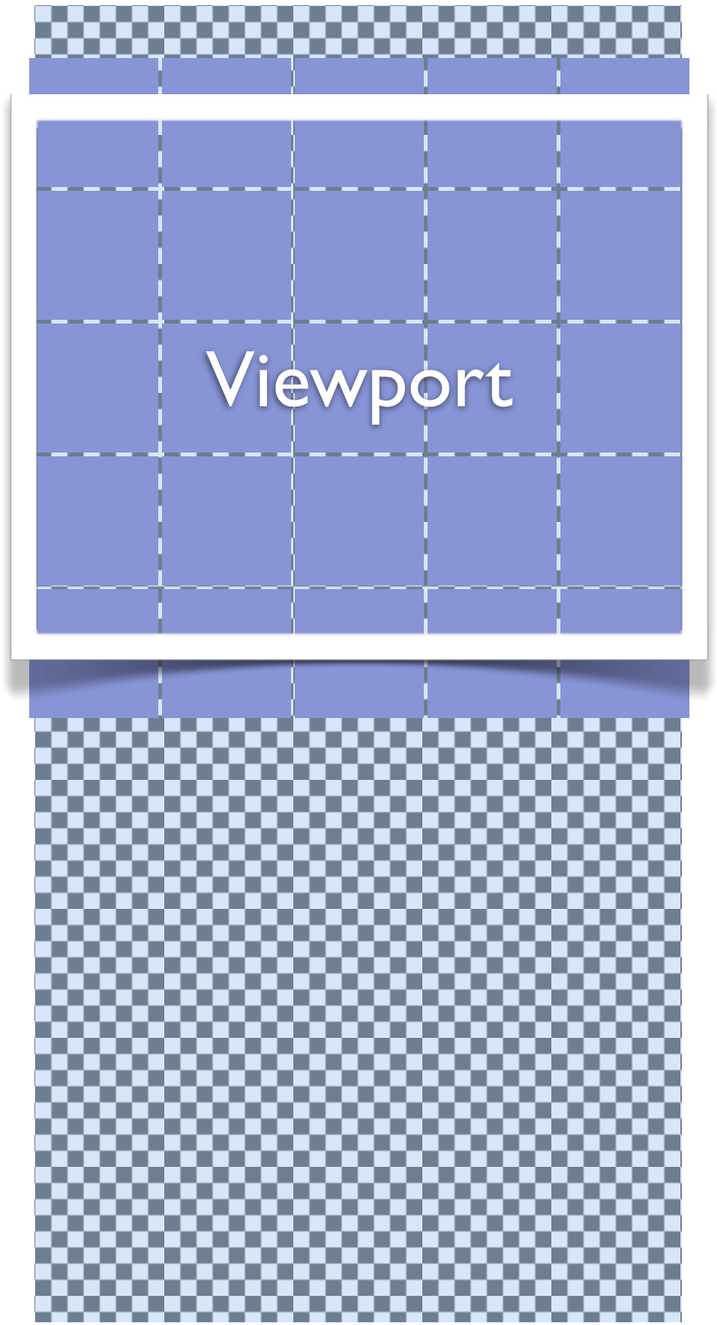
Tiled Rendering



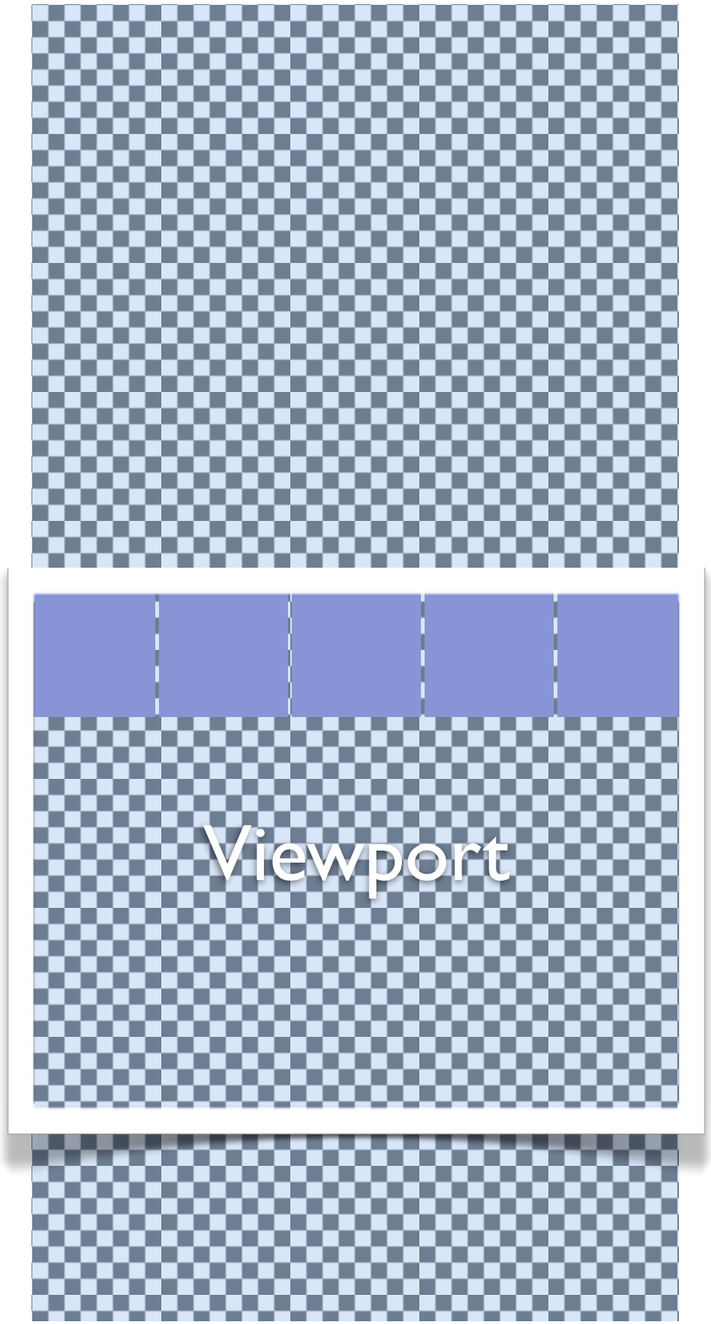
Tiled Rendering



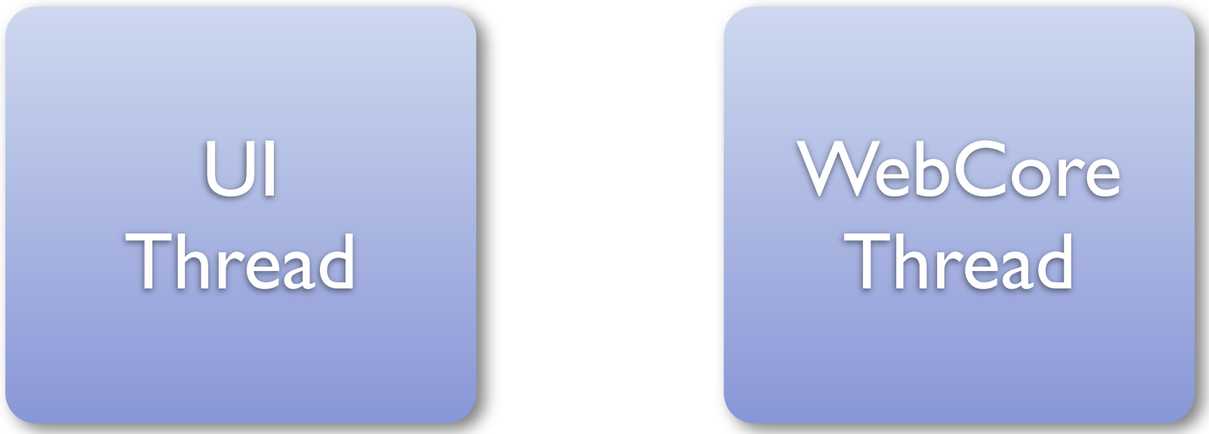
Tiled Rendering



Tiled Rendering



Tiles Generation



Tiles Generation

Texture Generation Thread

UI Thread

WebCore Thread

Tiles Generation

Texture Generation Thread

UI Thread

WebCore Thread

Picture

Tiles Generation

Texture Generation Thread

UI Thread

WebCore Thread

Picture

Tiles Generation

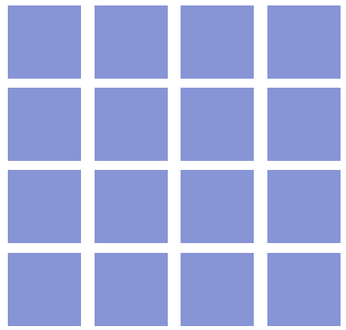
Texture Generation Thread

UI Thread

WebCore Thread

Picture

Tiles Generation



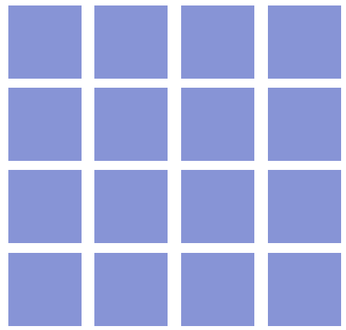
Texture Generation Thread

UI Thread

WebCore Thread

Picture

Tiles Generation



Texture Generation Thread

UI Thread

WebCore Thread

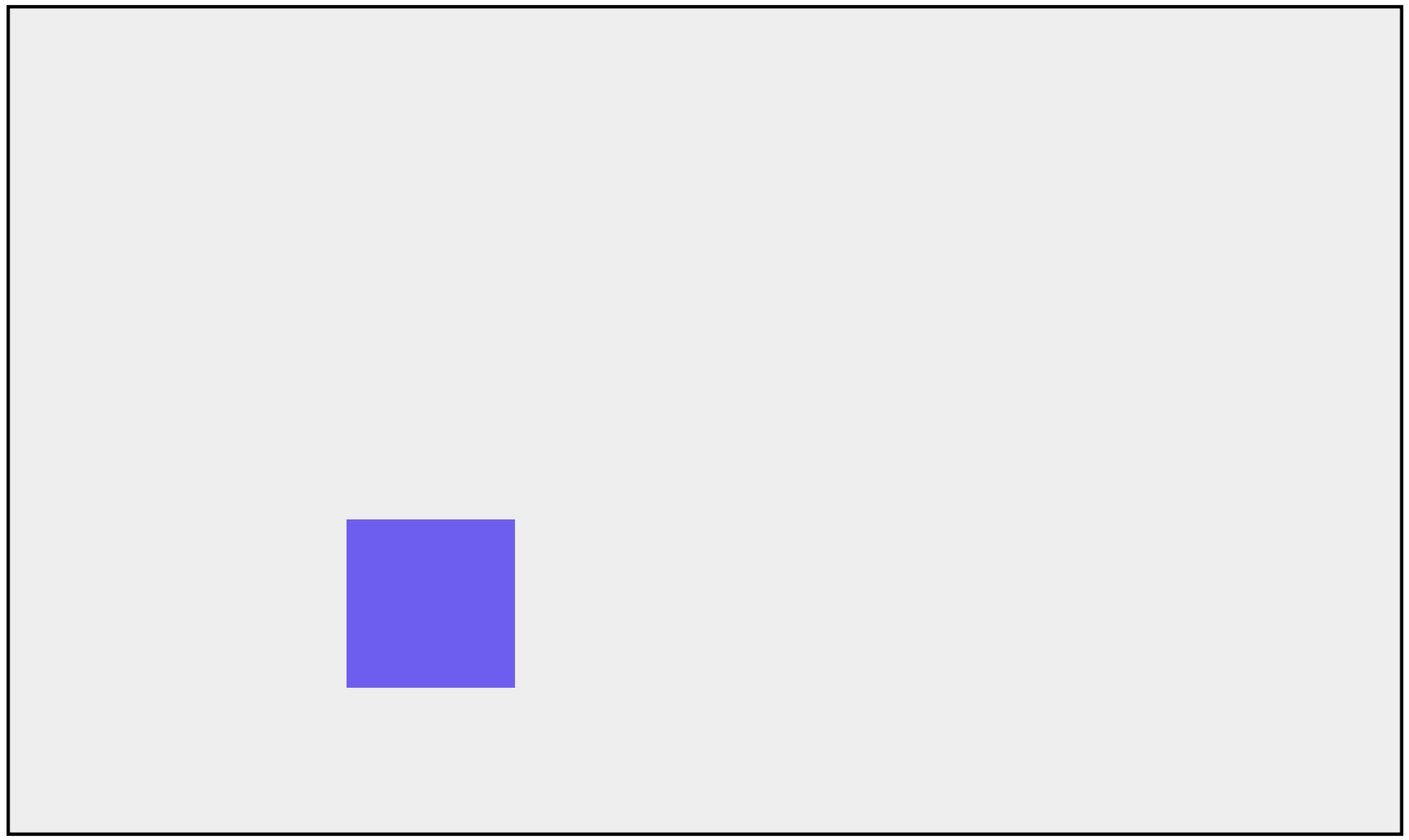
Picture

A few additional things...

- Slower at Painting, Faster at Drawing
  - Software Rendering still useful
- Precaching of tiles
  - Prefetch of surrounding tiles
  - Direction dependent
- Content visibility
  - Schedule low-res tiles (ICS -- API 14)
- Memory usage
  - Use a limited number of tiles (device dependent)
  - Tiles are 256x256 (...so far!)
  - Check if plain colors (JB -- API 16)

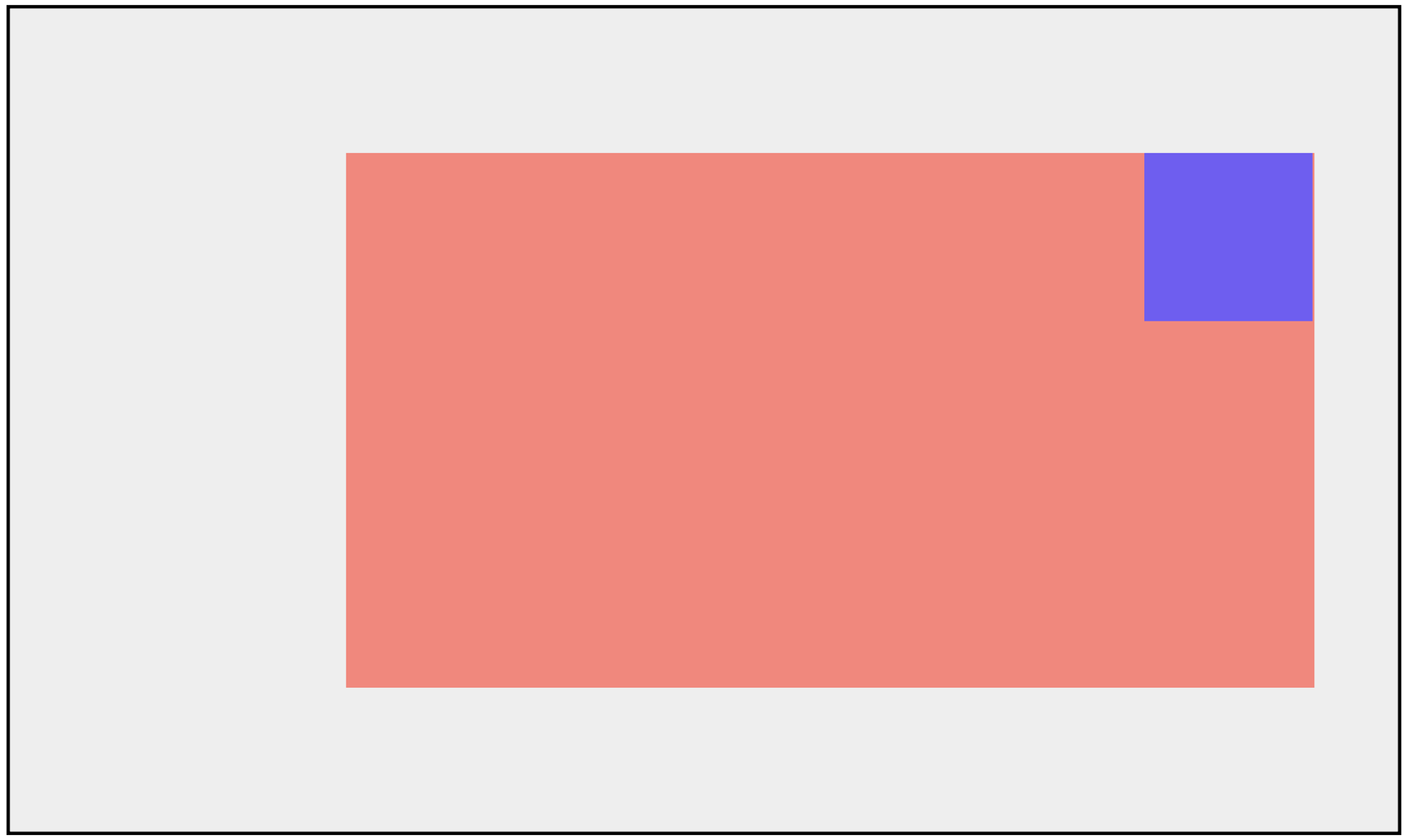
WebKit Layers

Moving content

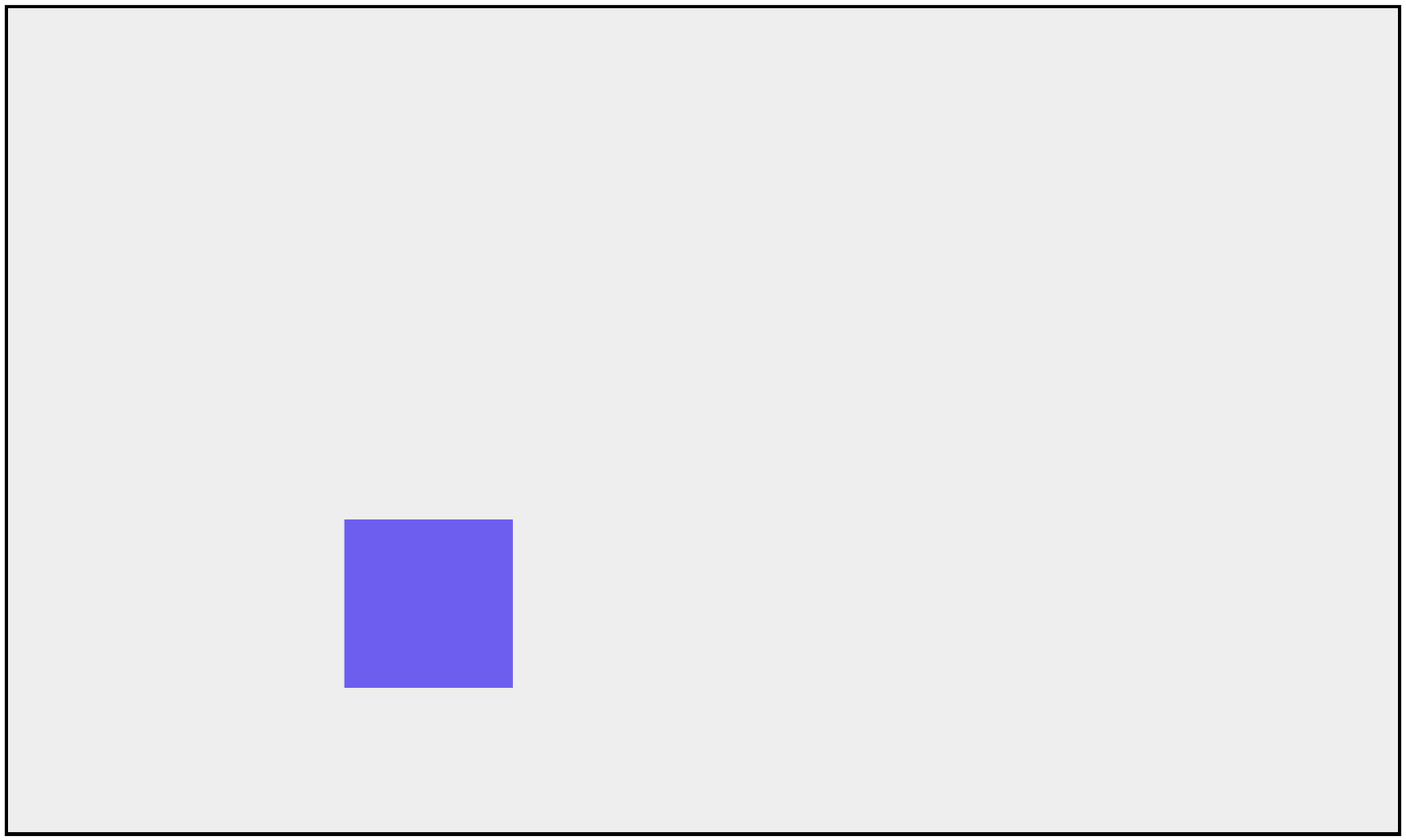


Moving content

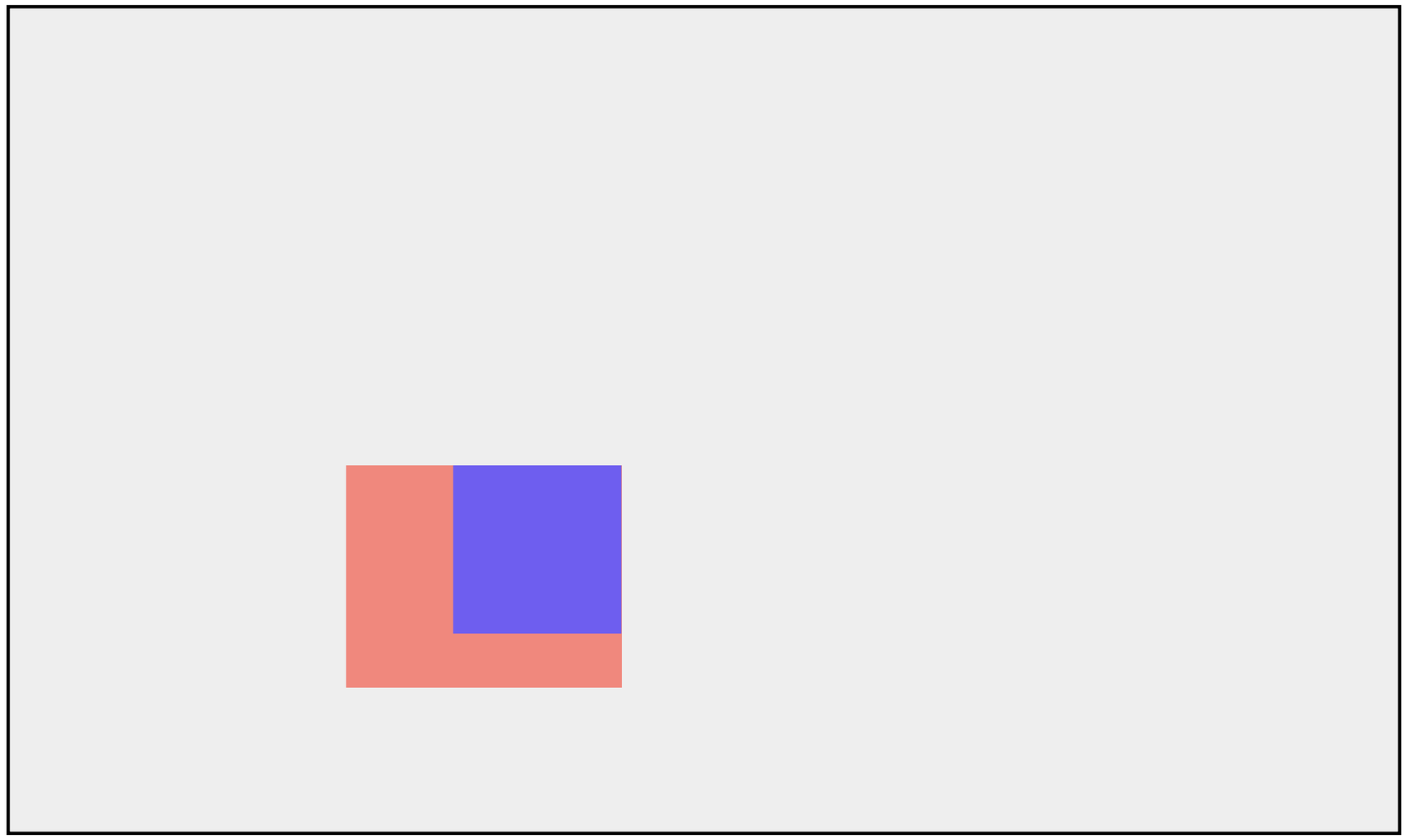
Moving content



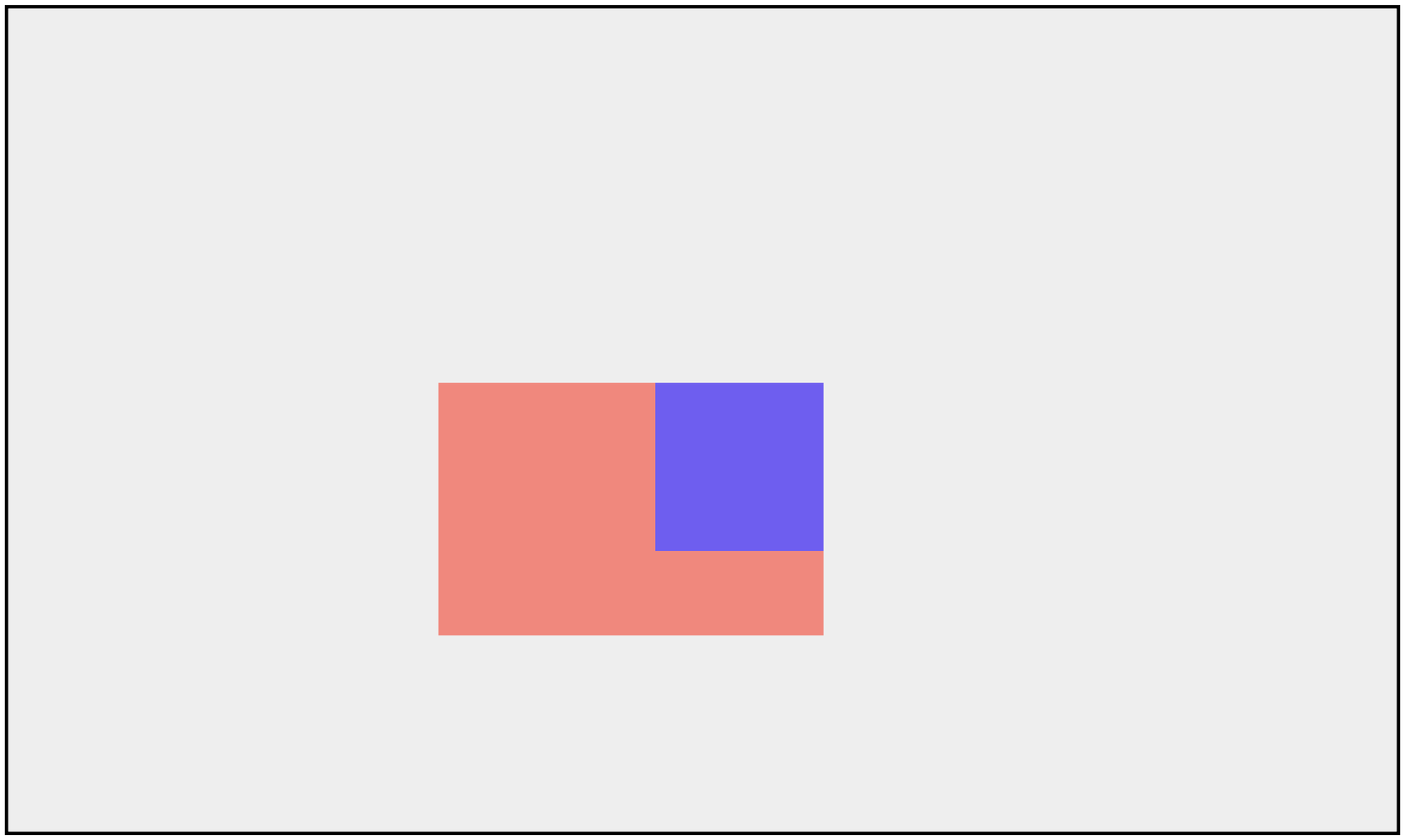
Moving content



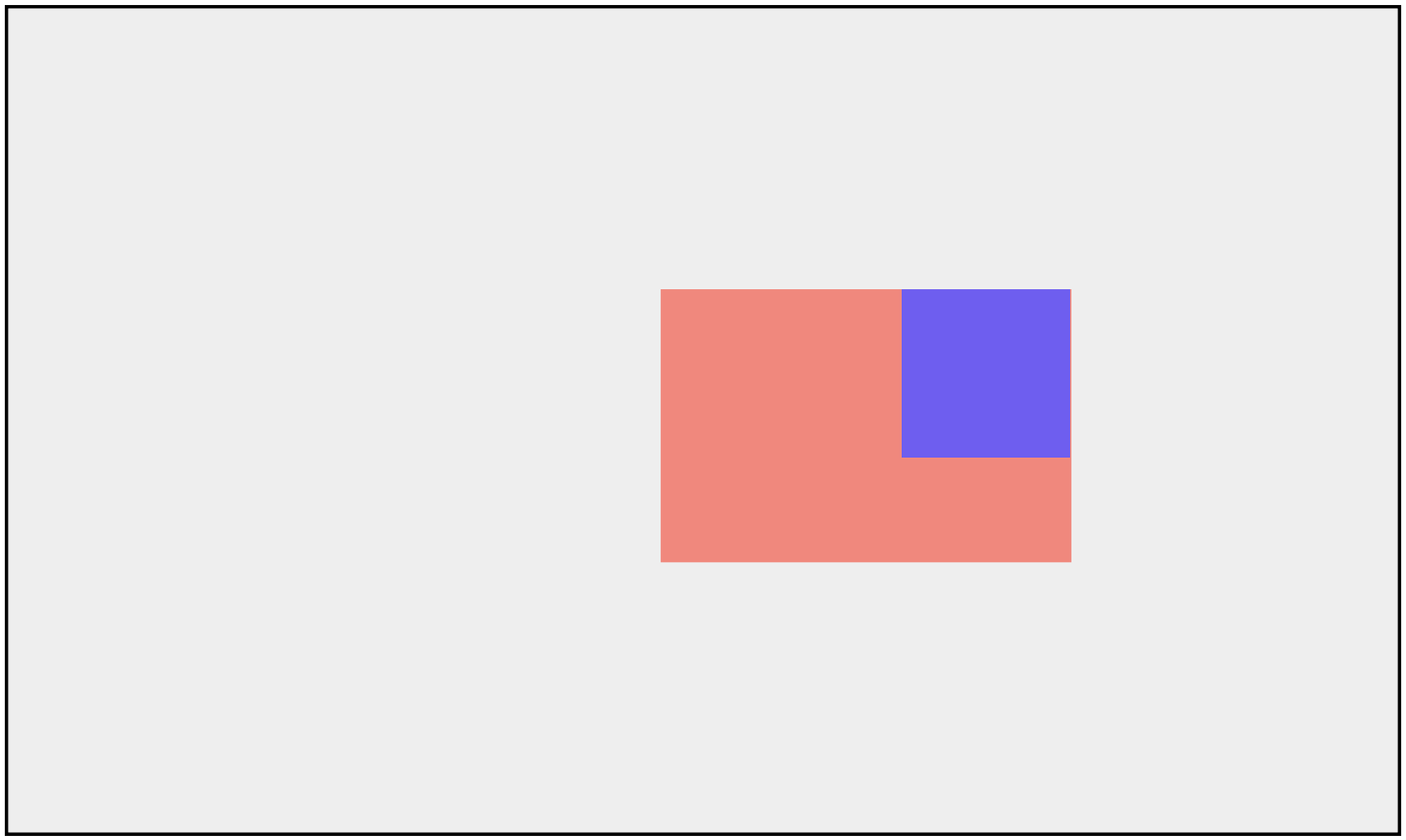
Moving content



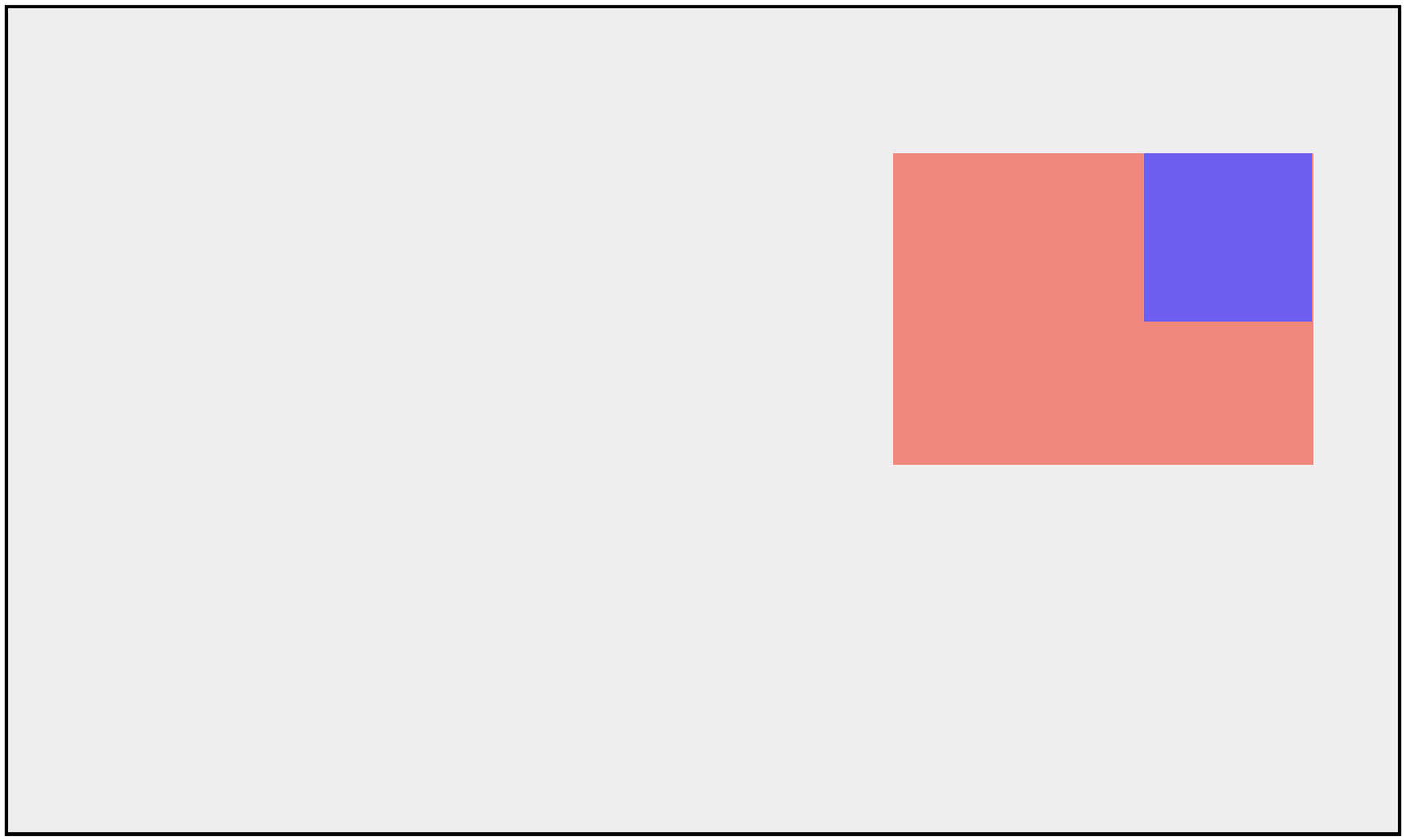
Moving content



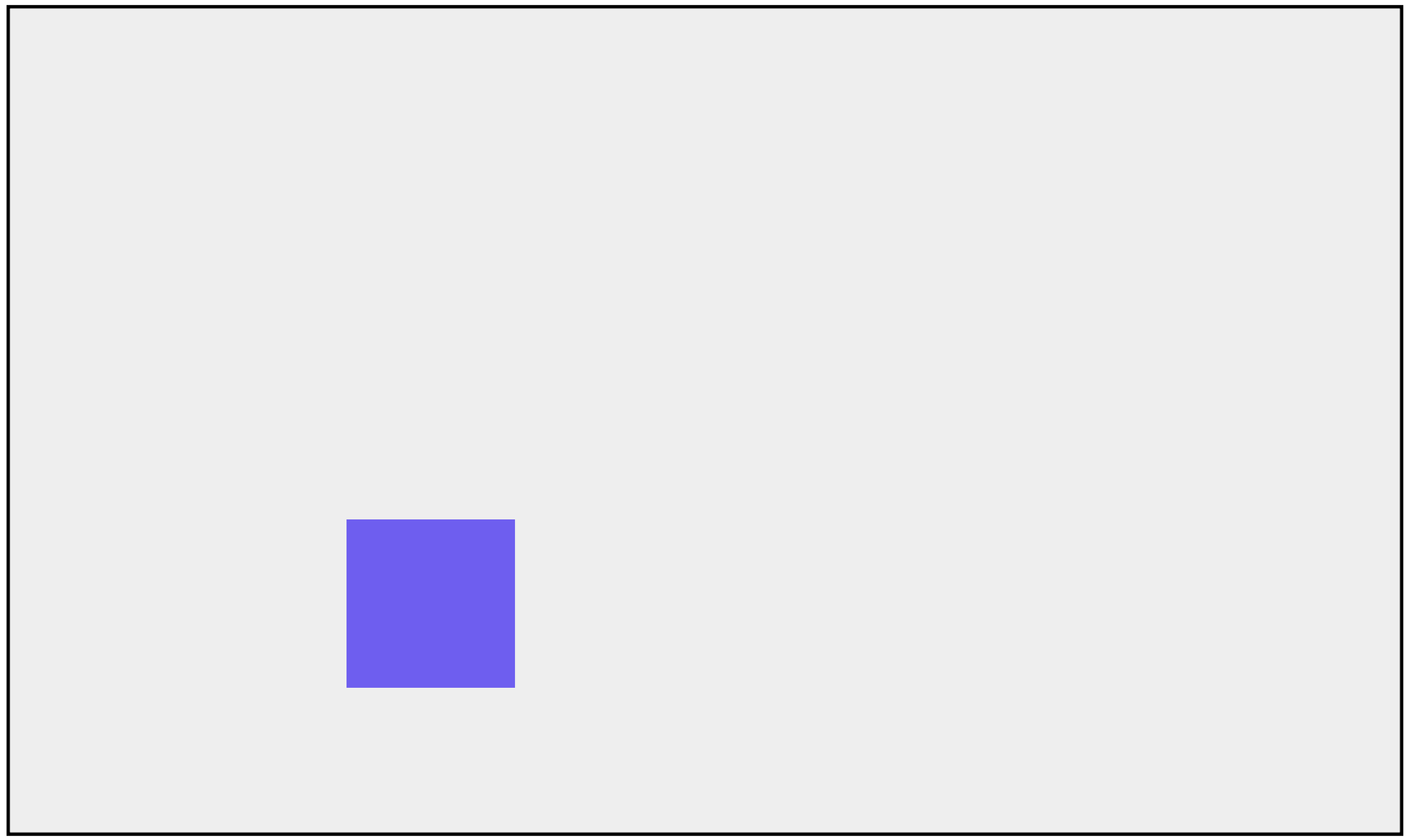
Moving content



Moving content



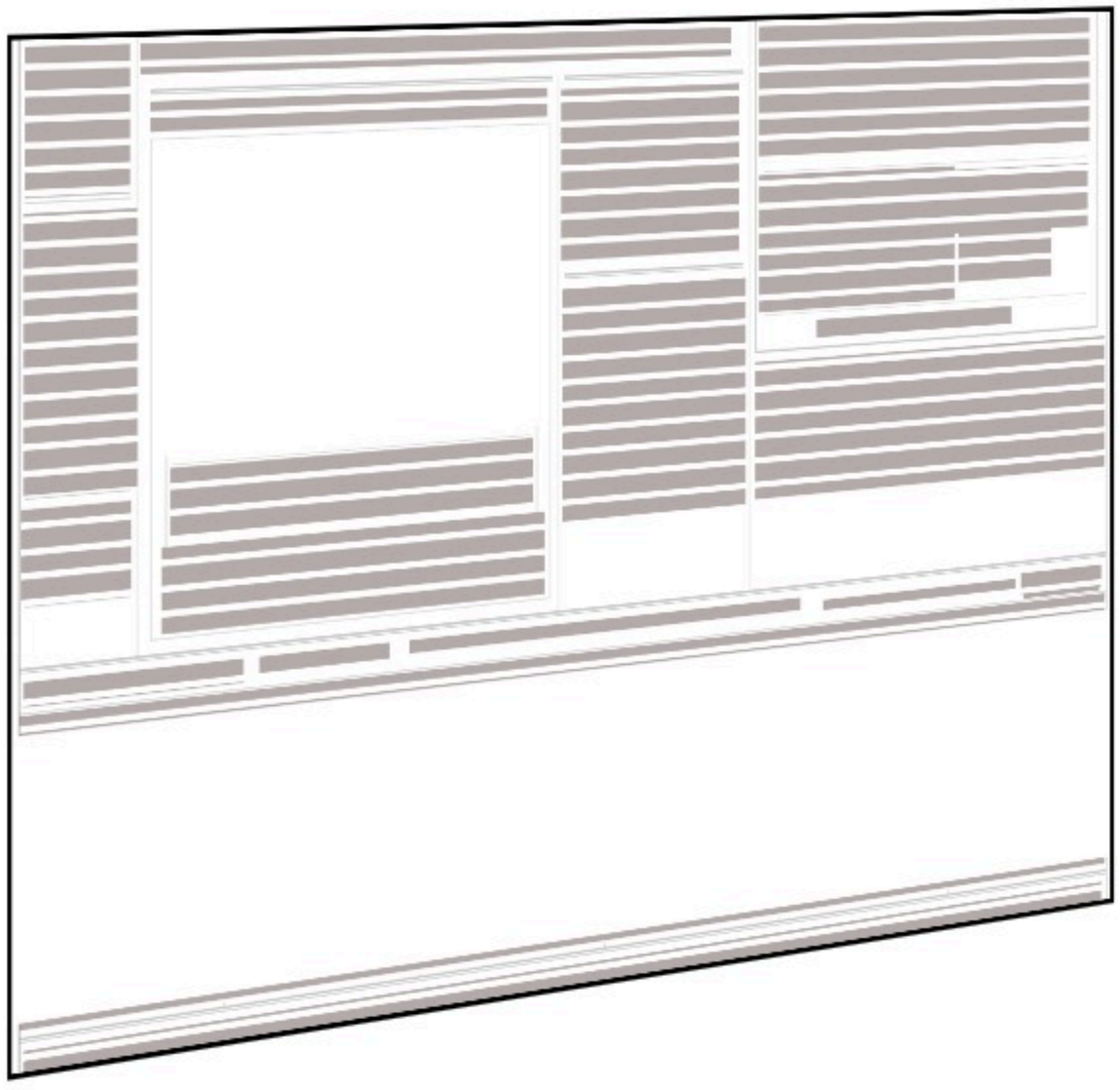
Moving content - with a layer



Moving content - with a layer

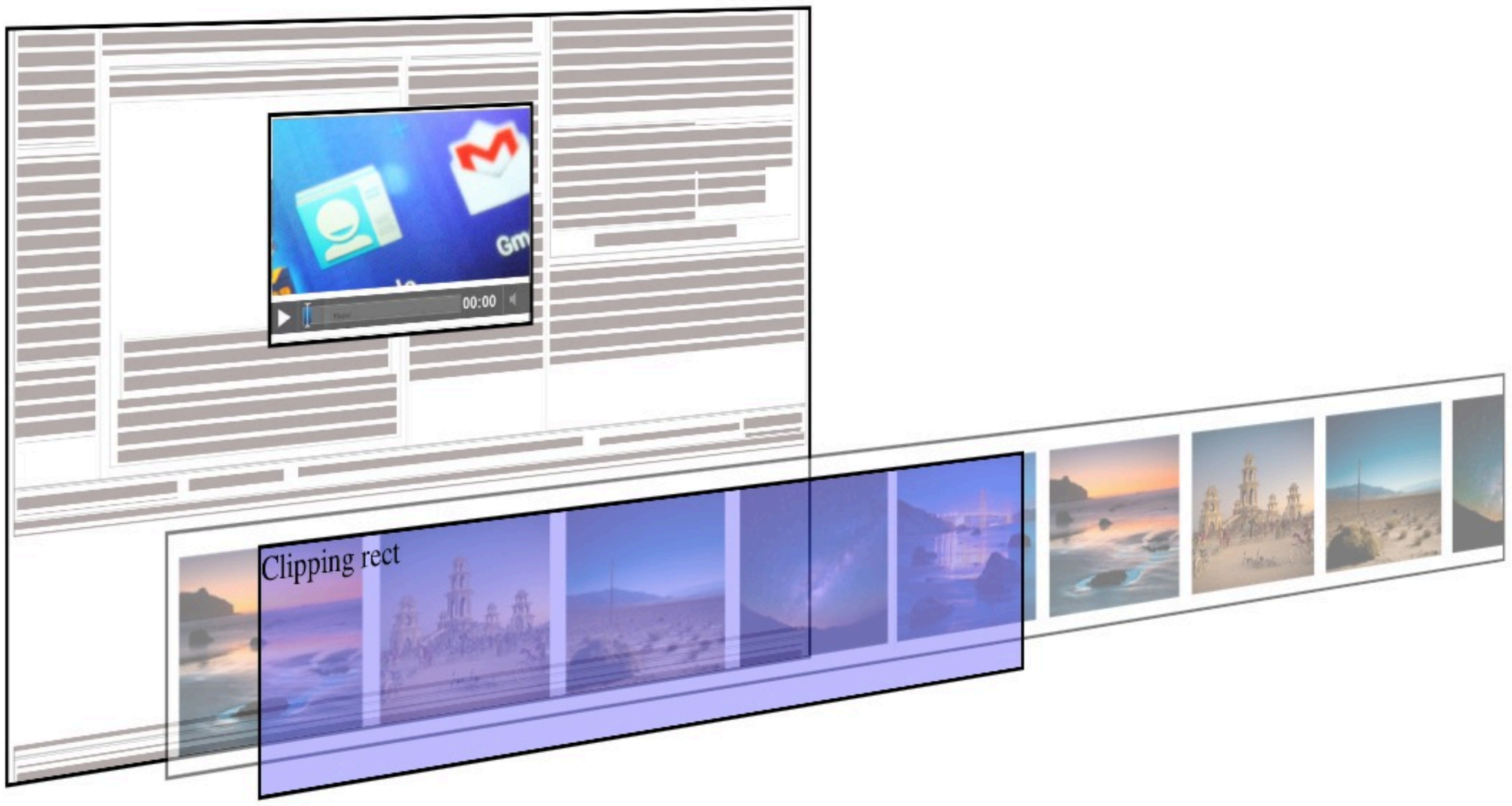

00:00


00:00

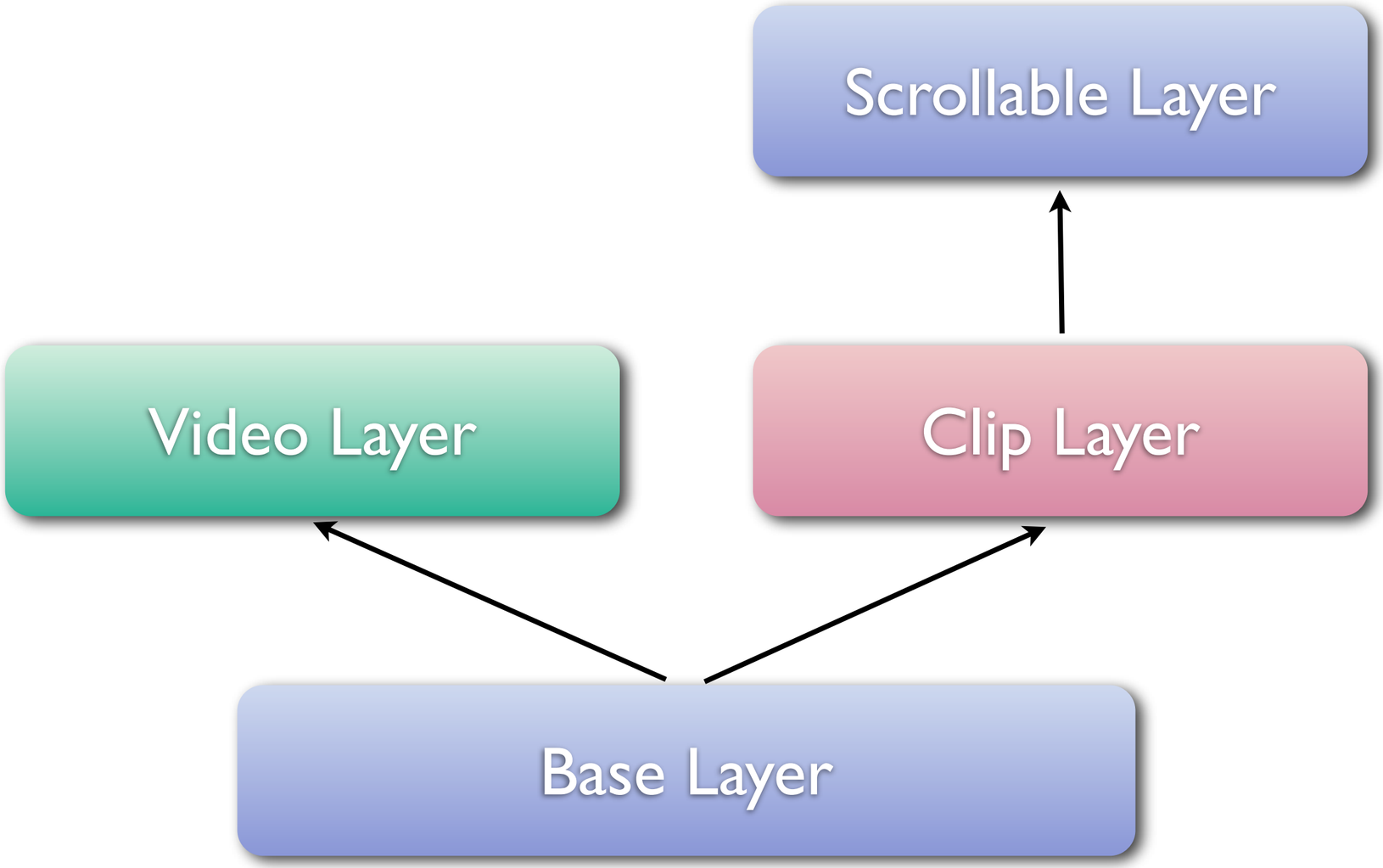



00:00


00:00


00:00
Clipping rect


00:00
Clipping rect


Scrollable Layer
Video Layer
Clip Layer
Base Layer

Layers

- Allow to composite elements of the page on their own surfaces
- Faster to move things around (no need to invalidate/repaint)
- Allow support for scrollable areas
- Nice way of integrating “foreign” content (e.g. video, plugins) into a render tree

48

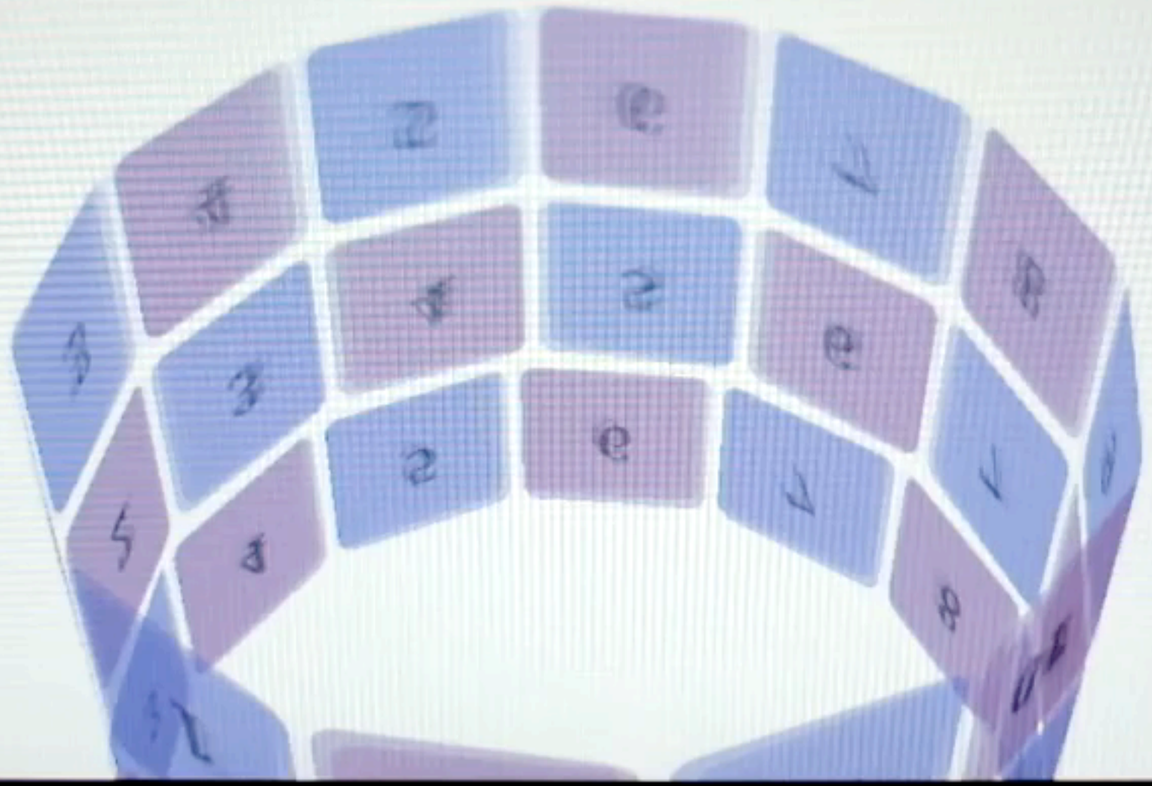
WebKit CSS 3D

Pure CSS 3D

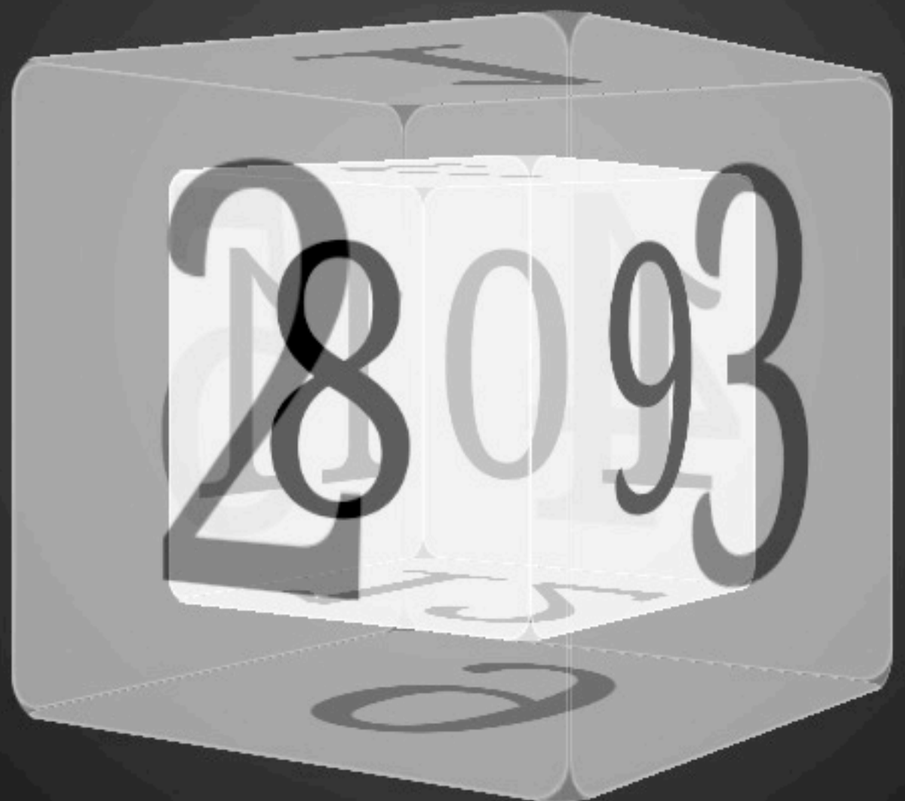
This is a simple example of how to use CSS transformation and animations to get interesting-looking behavior.

The three rings are constructed using a simple JavaScript function that creates elements and assigns them a transform that describes their position in the ring. CSS animations are then used to rotate each ring, and to spin the containing element around too.

Note that you can still select the numbers on the ring; everything remains clickable.



Backfaces visible

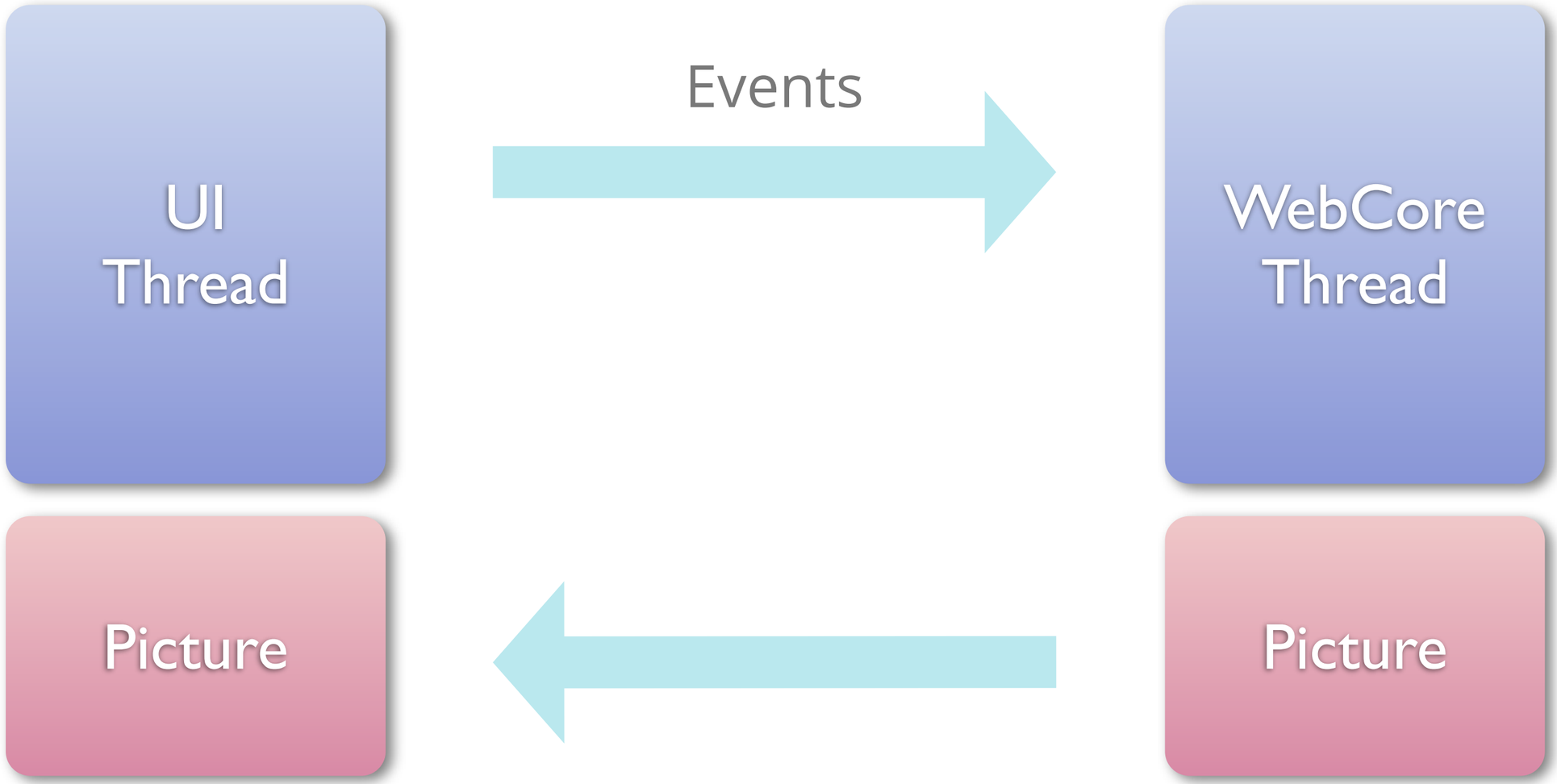


Toggle Shape

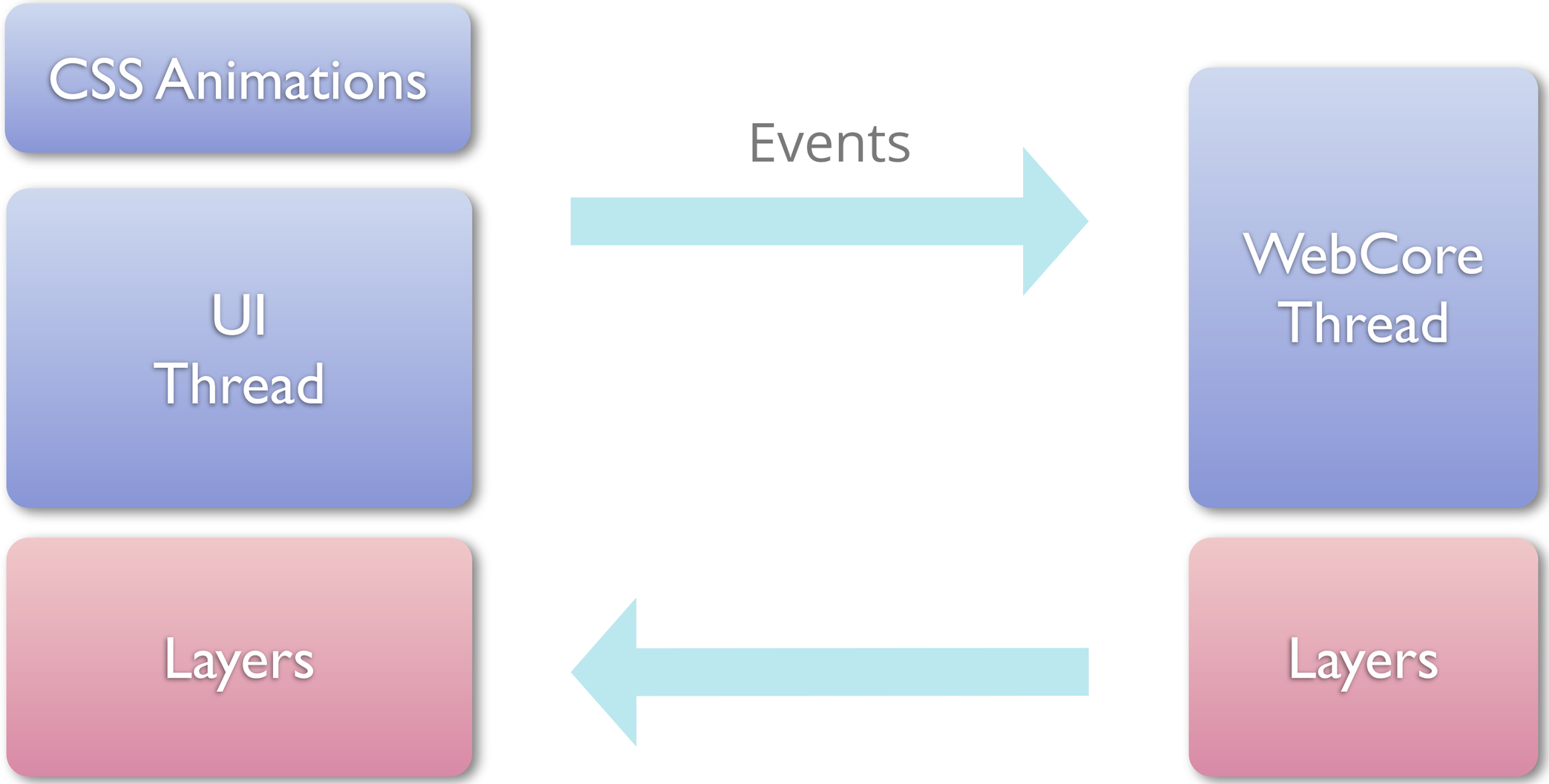
Backfaces visible

WebKit Layers ...on Android

Architecture



Architecture



CSS Animations

CSS Animations

- CSS property to animate HTML elements
- Supported on WebKit-based browsers, Firefox
- High-level specification of an animation
- Allow complex animations

CSS Animations

- CSS property to animate HTML elements
- Supported on WebKit-based browsers, Firefox
- High-level specification of an animation
- Allow complex animations

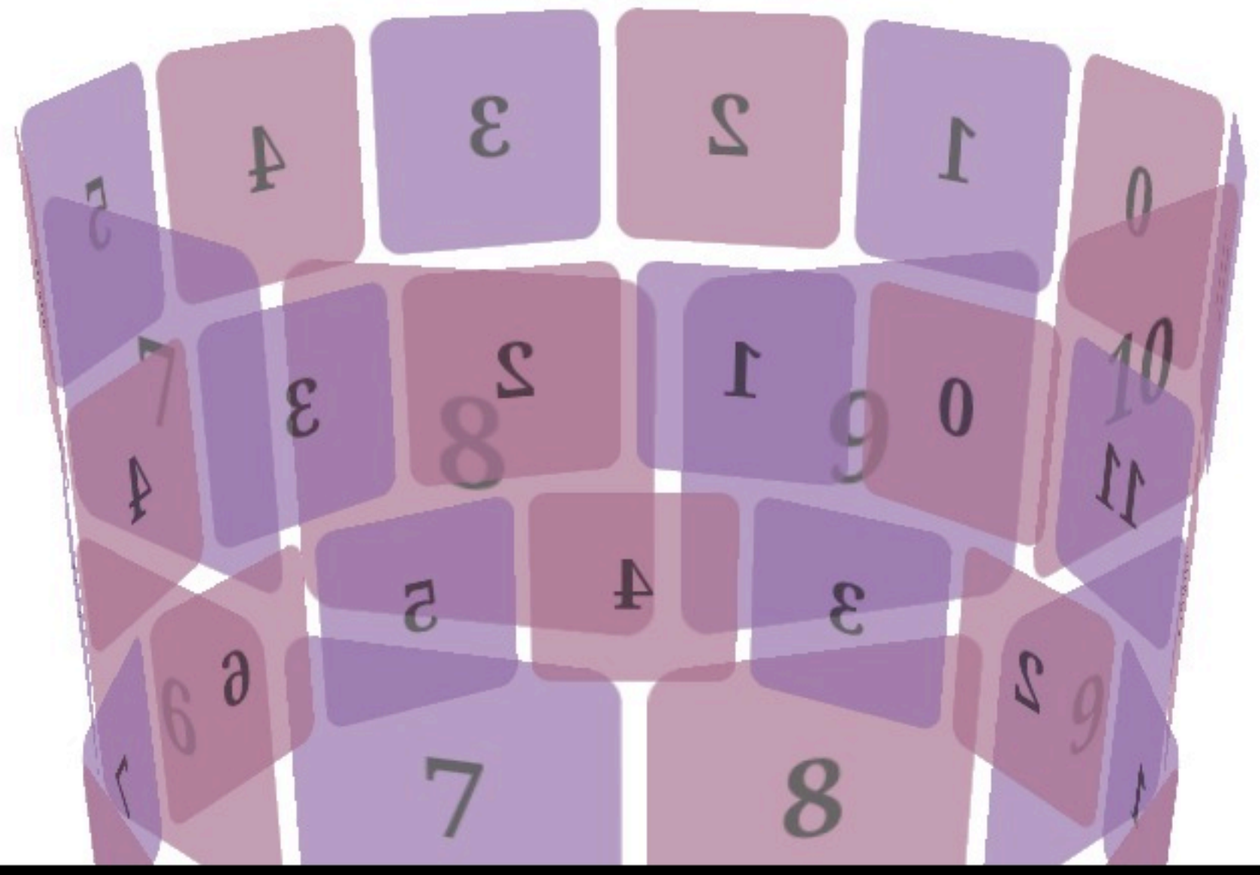
on Android

- Hardware accelerated
  - runs in parallel to webkit, on the UI thread
  - V-Synced
- Much faster than JavaScript...
  - No need to execute webkit or javascript code
  - No repaint -- elements are put onto layers

This is a simple example of how to use CSS transformation and animations to get interesting-looking behavior.

The three rings are constructed using a simple JavaScript function that creates elements and assigns them a transform that describes their position in the ring. CSS animations are then used to rotate each ring, and to spin the containing element around too.

Note that you can still select the numbers on the ring; everything remains clickable.



Keyframes animations / transform

```
@-webkit-keyframes x-spin {
  0% { -webkit-transform: rotateX(0deg); }
  50% { -webkit-transform: rotateX(180deg); }
  100% { -webkit-transform: rotateX(360deg); }
}
```

Taking advantages of Layers

- On mobile, our main problem is to go talk to WebKit and do the repaint/inval cycle
- Moving elements to layers allow us to do things without having to go to WebKit
  - Fixed Positioned elements
- Asynchronously update webkit, no need to wait for WebKit to paint things
  - Smooth interaction and immediate reactivity

Brought to you by Layers

- Seamless integration of inline HTML5 video
- Fixed positioned elements
- IFrames support
- Overflow-scroll elements
- CSS 3D
- Fixed background elements

Sessions | Select a Track | Day 1 | Day 2 | Day 3

Day 2	11:45 - 12:45	13:30 - 14:30	14:45 - 15:45	16:00 - 17:00	17:15 - 18:15	18:30 - 19:30
Chrome	How To Build Apps That Love Each Other With Web Intents	Turning The Web Up To 11	Chrome Developer Tools Evolution	Jank Busters: Building Performant Web Apps	Building High Performance Mobile Web Applications	
		HTML5 And App Engine: The Epic Tag Team Take On Modern Web Apps At Scale			New Web Tools And Advanced CSS/ HTML5 Features From Adobe & Google	
Cloud Platform	Building Mobile App Engine Backends For Android, IOS And The Web	Introducing Google Compute Engine	Big Data: Turning Your Data Problem Into A Competitive Advantage	Getting The Most Out Of Python 2.7 On App Engine	Google Compute Engine -- Technical Details	

complexspiral distorted

The page you are viewing right now exists to show off what can be accomplished with pure CSS1, and that's all. This variant on complexspiral doesn't even use any CSS2 to accomplish its magic. Remember: as you look this demo over, there is *no* Javascript here, nor are *any* PNGs being used, nor do I employ *any* proprietary extensions to CSS or any other language. It's all done using straight W3C-recommended markup and styling, all validated, plus a total of four (4) images.

Unfortunately, not every browser supports all of CSS1, and only those browsers which fully and completely support CSS1 will get this right. Despite some claims to the contrary, IE6/Win's rendering of this page is **not** correct, as it (as well as some other browsers) doesn't correctly support `background-attachment: fixed` for any element other than the `body`. That makes it impossible to pull off the intended effect. Other browsers may or may not get the effect right.

Hands-on: Things to Examine

Before you start, make sure you're viewing this page in one of the browsers mentioned above. Otherwise the descriptions to follow won't match what you see.

The first, easiest thing to do is scroll the page vertically. Make sure you scroll all the way to the very end of the page and back. Notice how the various areas with colored backgrounds also appear to distort the

complexspiral distorted

The page you are viewing right now exists to show off what can be accomplished with pure CSS1, and that's all. This variant on complexspiral doesn't even use any CSS2 to accomplish its magic. Remember: as you look this demo over, there is *no* Javascript here, nor are *any* PNGs being used, nor do I employ *any* proprietary extensions to CSS or any other language. It's all done using straight W3C-recommended markup and styling, all validated, plus a total of four (4) images.

Unfortunately, not every browser supports all of CSS1, and only those browsers which fully and completely support CSS1 will get this right. Despite some claims to the contrary, IE6/Win's rendering of this page is **not** correct, as it (as well as some other browsers) doesn't correctly support `background-attachment: fixed` for any element other than the `body`. That makes it impossible to pull off the intended effect. Other browsers may or may not get the effect right.

Hands-on: Things to Examine

Before you start, make sure you're viewing this page in one of the browsers mentioned above. Otherwise the descriptions to follow won't match what you see.

The first, easiest thing to do is scroll the page vertically. Make sure you scroll all the way to the very end

Some additional things...

- Same code path as base surface
- Memory usage
  - Limited number of tiles
- Drawing
  - We merge layers
  - Content check
- Painting
  - Fast invals

Using WebView

What can it do

- Desktop-class HTML support
- Viewport metatag support
- IFrames, Scrollable elements
- Fixed positioned elements
- CSS Fixed background
- CSS3D / CSS animations / Layers
- HTML5 Video / Audio
- Embedded fonts
- Complex text: Japanese / Arabic text / etc

When...

Eclair

- Database API support, for client-side databases using SQL.
- Application cache support, for offline applications.
- Geolocation API support, to provide location information about the device.
- <video> tag support in fullscreen mode.
  viewport tag support

Froyo

- Layers
- Fixed position elements
- Accelerated CSS animations (2 frames)

When...

Honeycomb

- Media capture (camera)
- Device Orientation
- CSS3D transforms
- IFrames
- Overflow-scroll elements
- Inline video
- Fixed position elements (full support)
- Accelerated CSS animations (full support)

When...

ICS

- Accessibility
- Support for Indic fonts (Devanagari, Bengali, and Tamil)
- Support for Ethiopic, Georgian, and Armenian fonts
- Support for WebDriver

JellyBean

- Layers performance improvement
- Fixed background element
- Vertical text support

Setup

Use a webview

```
<?xml version="1.0" encoding="utf-8"?>
<WebView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/webview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
/>
```

71

Internet permission

```
<manifest ... >
  <uses-permission
          android:name="android.permission.INTERNET" />
  ...
</manifest>
```

WebViewClient -- URL loading

```
private class myWebViewClient extends WebViewClient {
  @Override
   public boolean shouldOverrideUrlLoading(WebView view, String url) {
     return false;
   }
}

(...)

myWebView.setWebViewClient(new myWebViewClient());
```

Load content

```
WebView myWebView = (WebView) findViewById(R.id.webview);
myWebView.loadUrl("http://www.example.com");

(...)
```

```
String summary = "<html><body>Hello World</body></html>";
myWebView.loadData(summary, "text/html", null);
```

Using javascript

```
WebView myWebView = (WebView) findViewById(R.id.webview);
WebSettings webSettings = myWebView.getSettings();
webSettings.setJavaScriptEnabled(true);
```

75

Advantage and Disadvantages

Software Rendering mode

- (may) use less memory
- faster repaints
- slower draw
- UI responsiveness content-dependent
- content always there
- no CSS3D support
- no inline video support

Hardware Rendering mode

- use more memory
- slower repaints
- faster draw
- UI responsiveness constant
- can miss content temporarily
- CSS3D support
- inline video support

Are you Accelerated?

- If Activity or Application accelerated
  - android:hardwareAccelerated=”true”
  - API level 11 (Android 3.0 Honeycomb)

- Forcing 2D rendering
  - setLayerType with LAYER_TYPE_SOFTWARE

WebView automatically accelerated

Accessibility

Accessibility in WebView

- Same solution as Chrome / ChromeOS
- Enable JavaScript
  - Explore by Touch (ICS)
  - Gesture Navigation (JB)
- Settings => Accessibility => Enhance web accessibility

Inline / Fullscreen Video

WebChromeClient

```
public void onShowCustomView(View view,
            WebChromeClient.CustomViewCallback callback) {
    mOriginalOrientation = getRequestedOrientation();
    FrameLayout decor = (FrameLayout) getWindow().getDecorView();
    mFullscreenContainer = new FrameLayout(getBaseContext());
    mFullscreenContainer.addView(view, ViewGroup.LayoutParams.MATCH_PARENT);
    decor.addView(mFullscreenContainer, ViewGroup.LayoutParams.MATCH_PARENT);
    mCustomView = view;
    mCustomViewCallback = callback;
    setRequestedOrientation(getRequestedOrientation());
}
```

WebChromeClient

```
@Override
public void onHideCustomView() {
    FrameLayout decor = (FrameLayout) getWindow().getDecorView();
    decor.removeView(mFullscreenContainer);
    mFullscreenContainer = null;
    mCustomView = null;
    mCustomViewCallback.onCustomViewHidden();
    // Show the content view.
    setRequestedOrientation(mOriginalOrientation);
}
```

Binding javascript code to dalvik

Binding javascript code to java

```
public class JavaScriptInterface {
    Context mContext;

    /** Instantiate the interface and set the context */
    JavaScriptInterface(Context c) {
        mContext = c;
    }

    /** Show a toast from the web page */
    public void showToast(String toast) {
        Toast.makeText(mContext, toast,
                       Toast.LENGTH_SHORT).show();
    }
}
```

Binding javascript code to java

```
WebView webView = (WebView) findViewById(R.id.webview);
webView.addJavascriptInterface(new JavaScriptInterface(this), "Android");
```

HTML side

```
<input type="button" value="Say hello"
  onClick="showAndroidToast('Hello Android!')" />

<script type="text/javascript">
    function showAndroidToast(toast) {
        Android.showToast(toast);
    }
</script>
```

Viewport Tag

Viewport tag

```
<head>
    <title>Example</title>
    <meta name="viewport" content="width=device-width, user-scalable=no" />
</head>
```

Viewport tag

```
<meta name="viewport"
      content="
          height = [pixel_value | device-height] ,
          width = [pixel_value | device-width ] ,
          initial-scale = float_value ,
          minimum-scale = float_value ,
          maximum-scale = float_value ,
          user-scalable = [yes | no] ,
          target-densitydpi = [dpi_value | device-dpi |
                               high-dpi | medium-dpi | low-dpi]
      " />
```

Layers / Animations

Use layers!

Use layers!

```
var elem = document.getElementById("myElement");
```

Use layers!

```
var elem = document.getElementById("myElement");

elem.style.left = x;
elem.style.top = y;
```

Use layers!

```
var elem = document.getElementById("myElement");

elem.style.left = x;
elem.style.top = y;

#myElement {
  -webkit-transform: transform3d(0, 0, 0);
}
```

Use layers!

```
var elem = document.getElementById("myElement");

elem.style.left = x;
elem.style.[redacted]

#myElement {
  -webkit-transform: transform3d(0, 0, 0);
}
```

Use layers!

```
var elem = document.getElementById("myElement");

elem.style.left = x;
elem.style.[redacted]

#myElement {
  -webkit-transform: transform3d(0, 0, 0);
}

elem.style.webkitTransform = ”translate3d(x, y, 0)”;
```

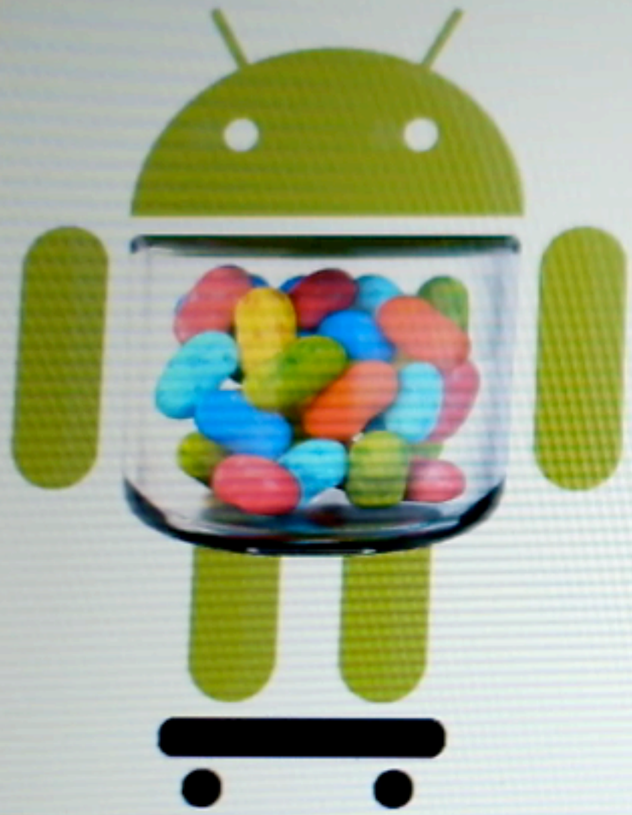
Animation

Animation

Animation with JavaScript

```
var animTime = 2500; // ms
var animDist = 730; // px
var dx = (animDist / animTime);
var x = 0;
var startTime = Date.now();

function run() {
    var delta = Date.now() - startTime;
    startTime = Date.now();
    x += dx * delta;
    if (x > animDist) {
        x = 0;
    }
    var elem = document.getElementById("droid");
    elem.style.left = x + "px";
    setTimeout("run()", 0);
}
```



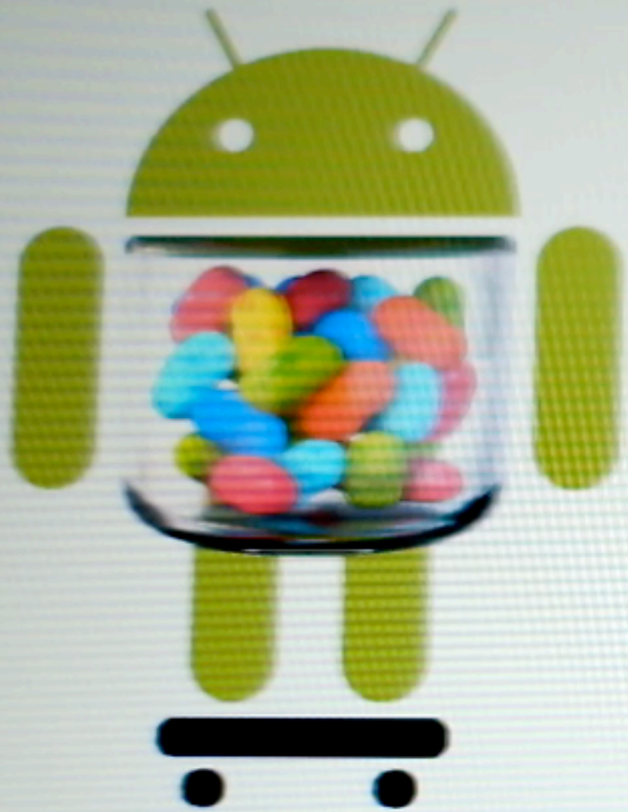
Google 12

2:36

Animation with CSS

```
@-webkit-keyframes droid-anim-horiz {
  0% { -webkit-transform: translate3d(0, 0, 0); }
  100% { -webkit-transform: translate3d(750px, 0, 0); }
}
```

```
#droid {
  position: absolute;
  -webkit-animation: droid-anim-horiz 2.5s;
}
```



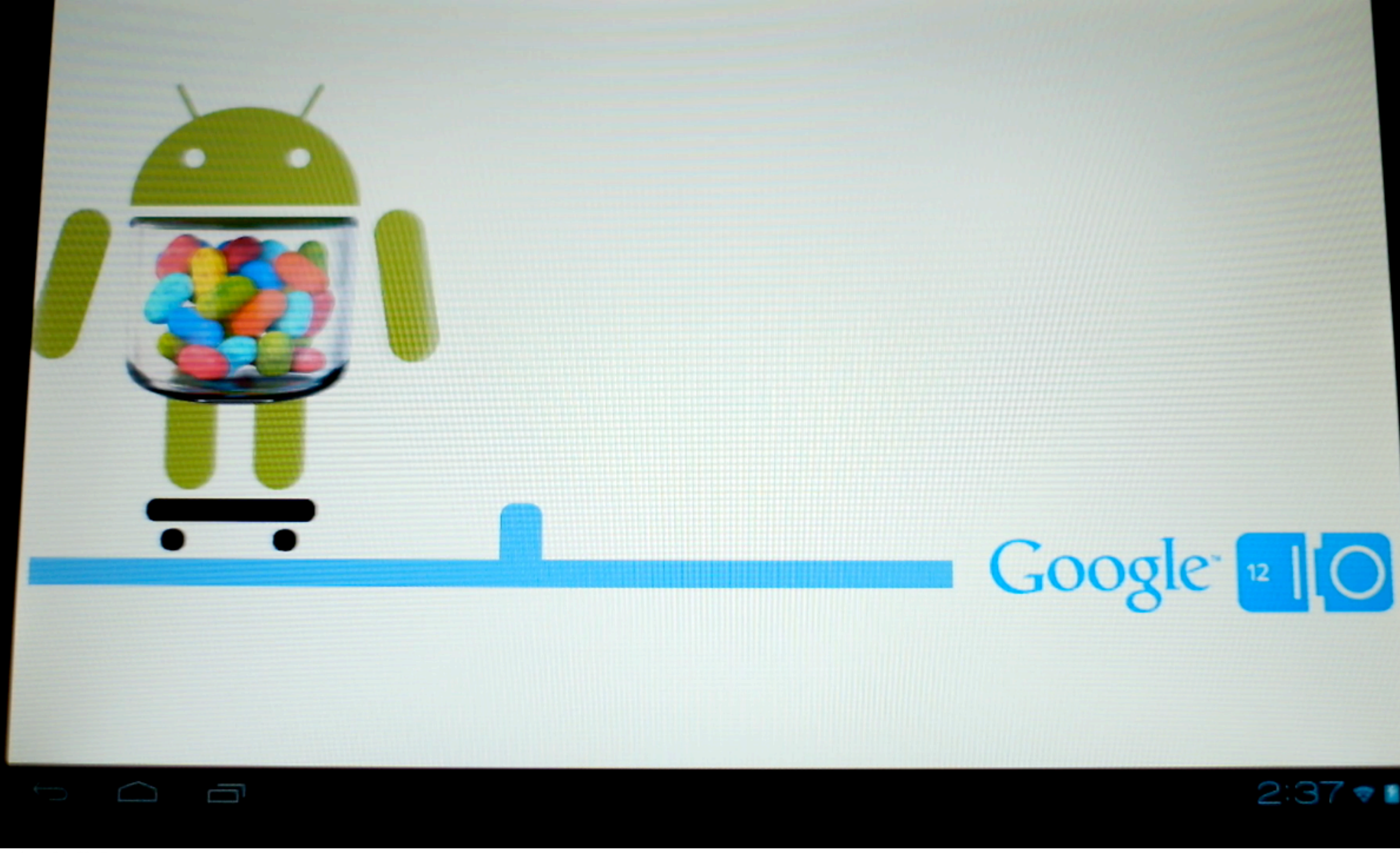
Google 12

2:37

Use CSS animations

- Much Faster
- V-Synced
- Higher level
- Allow for complex animations
- Trigger the creation of layers

97


Google
12
2:37

Layer compositing rules

- Default in WebKit
  - transform3d
  - video
  - canvas (if accelerated)
  - CSS animations

Layer compositing rules

- Default in WebKit
  - transform3d
  - video
  - canvas (if accelerated)
  - CSS animations

- Additional rules on Android
  - fixed positioned
  - overflow-scroll
  - iframes
  - canvas
  - fixed background

Layer compositing rules

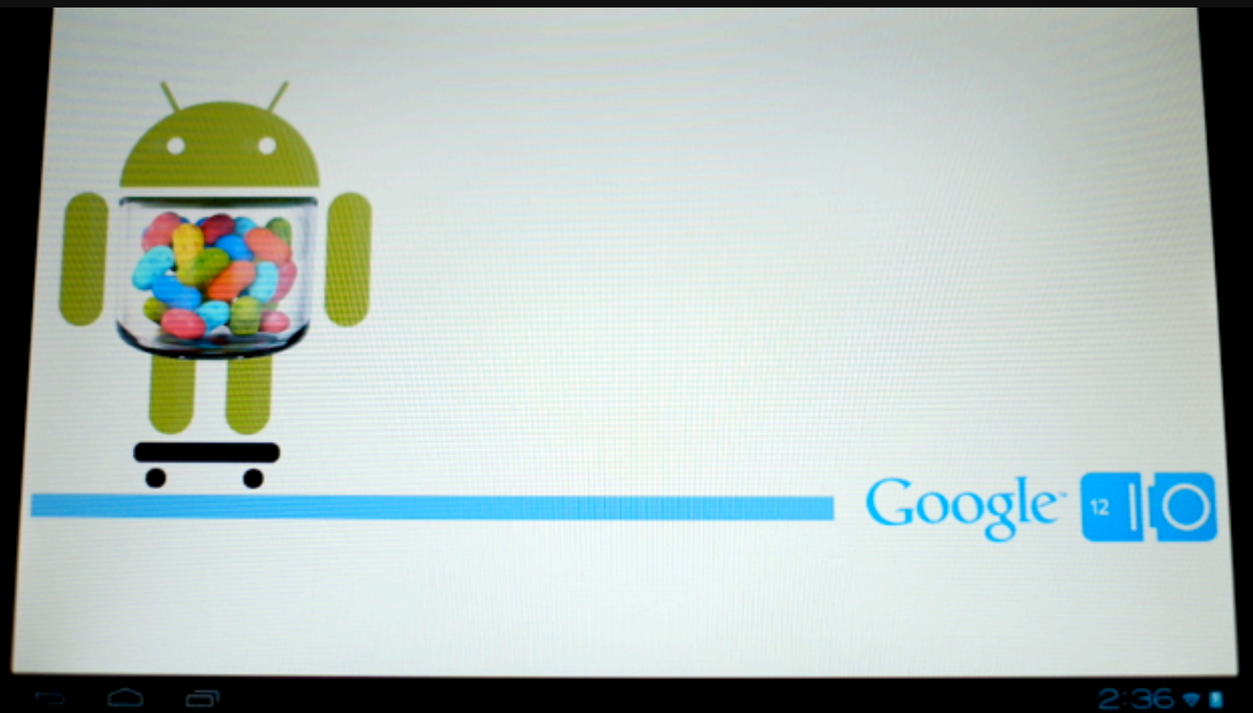
- Default in WebKit
  - transform3d
  - video
  - canvas (if accelerated)
  - CSS animations

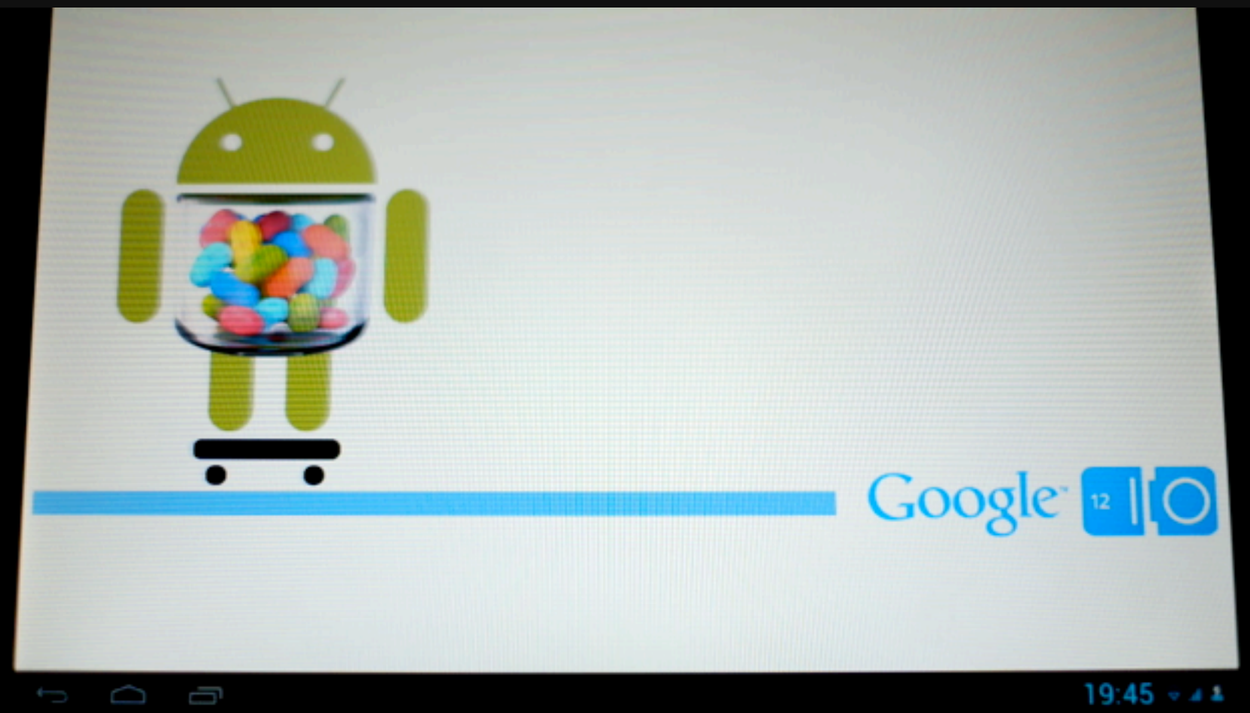
- Additional rules on Android
  - fixed positioned
  - overflow-scroll
  - iframes
  - canvas
  - fixed background

Layers are an implementation detail: the final rendering is the same

Tips

- Put animated / appearing content on layers
- Creating a layer is **expensive**
- If something is on layer, leave it if you can
- Pay attention to the size (256x256 tiles)
- Prepare offscreen content (via translate3d(100%, 0, 0))
- If animating images, create a layer for them
    - e.g. apply translate3d(0, 0, 0) on the <img> element directly
- Try software rendering!



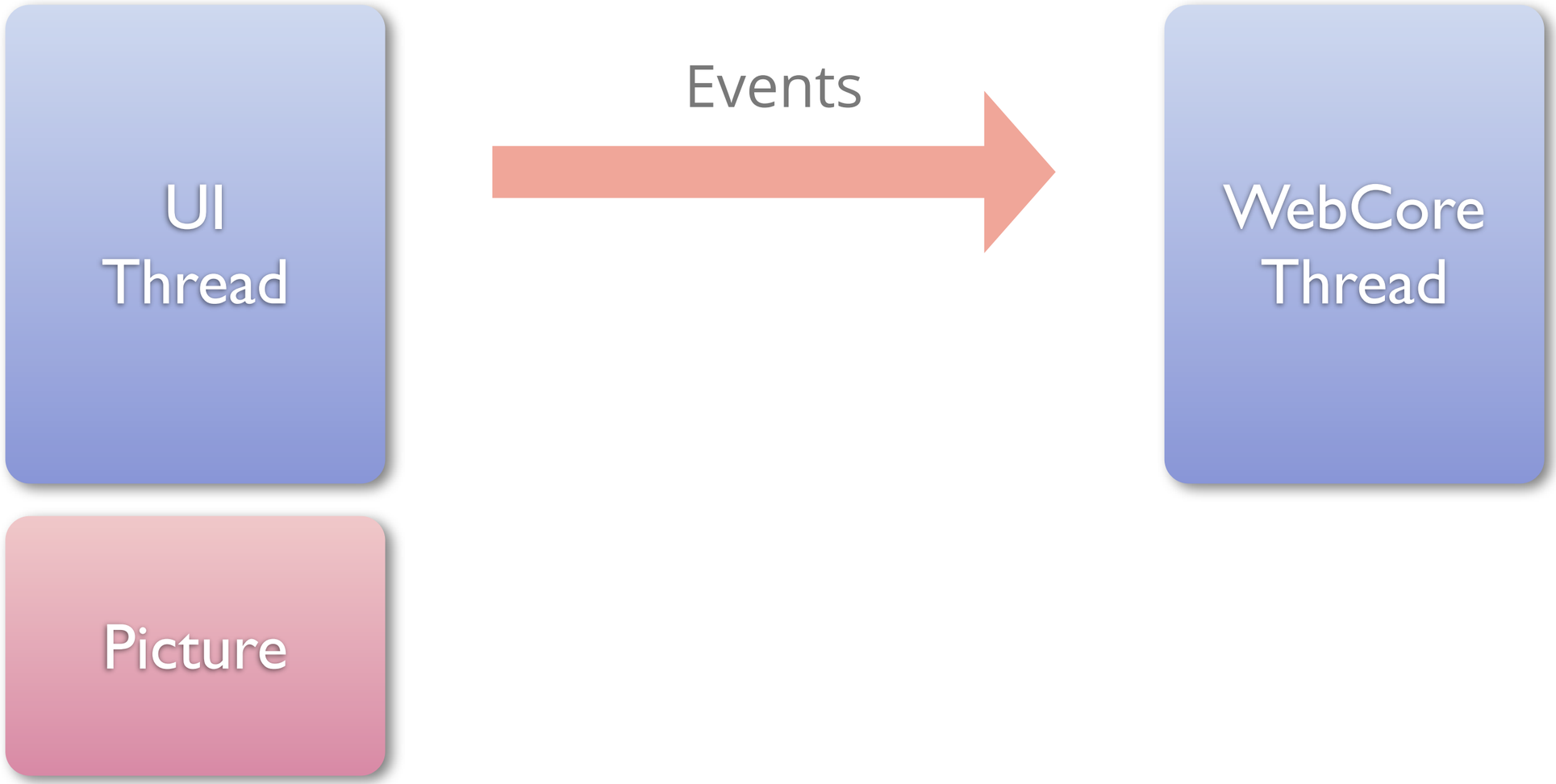


ICS

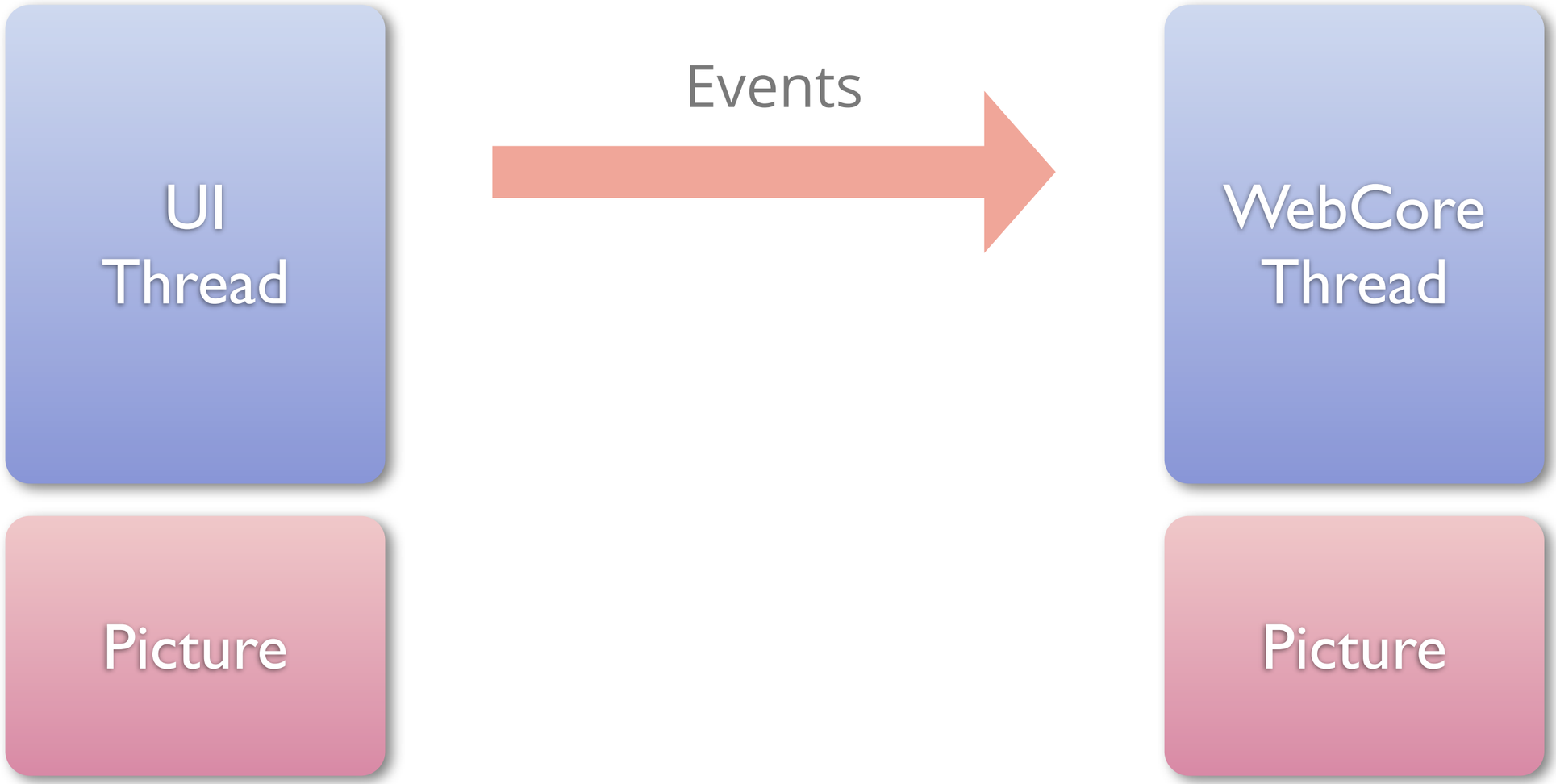
JB

Synchronization

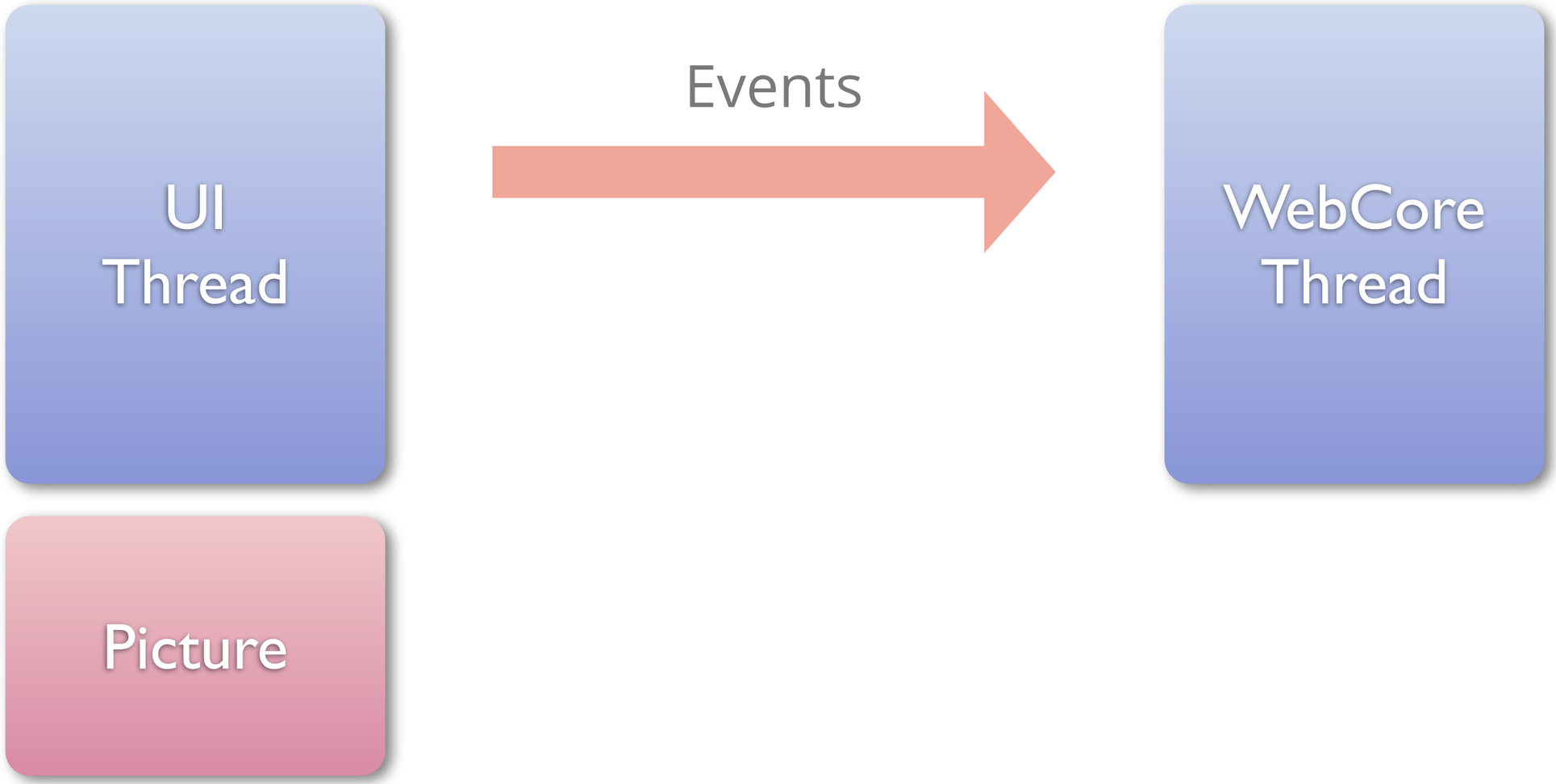
Multithreading



Multithreading

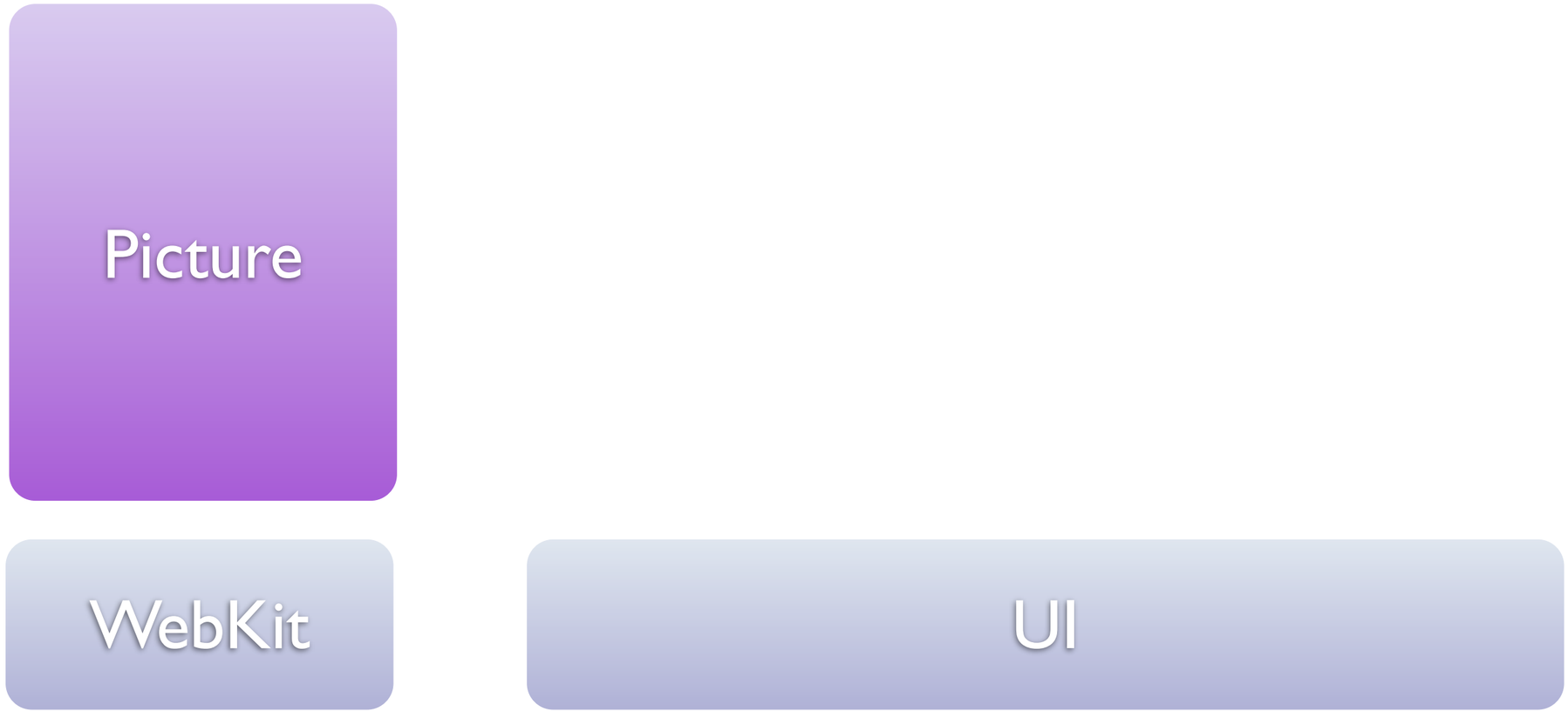


Multithreading

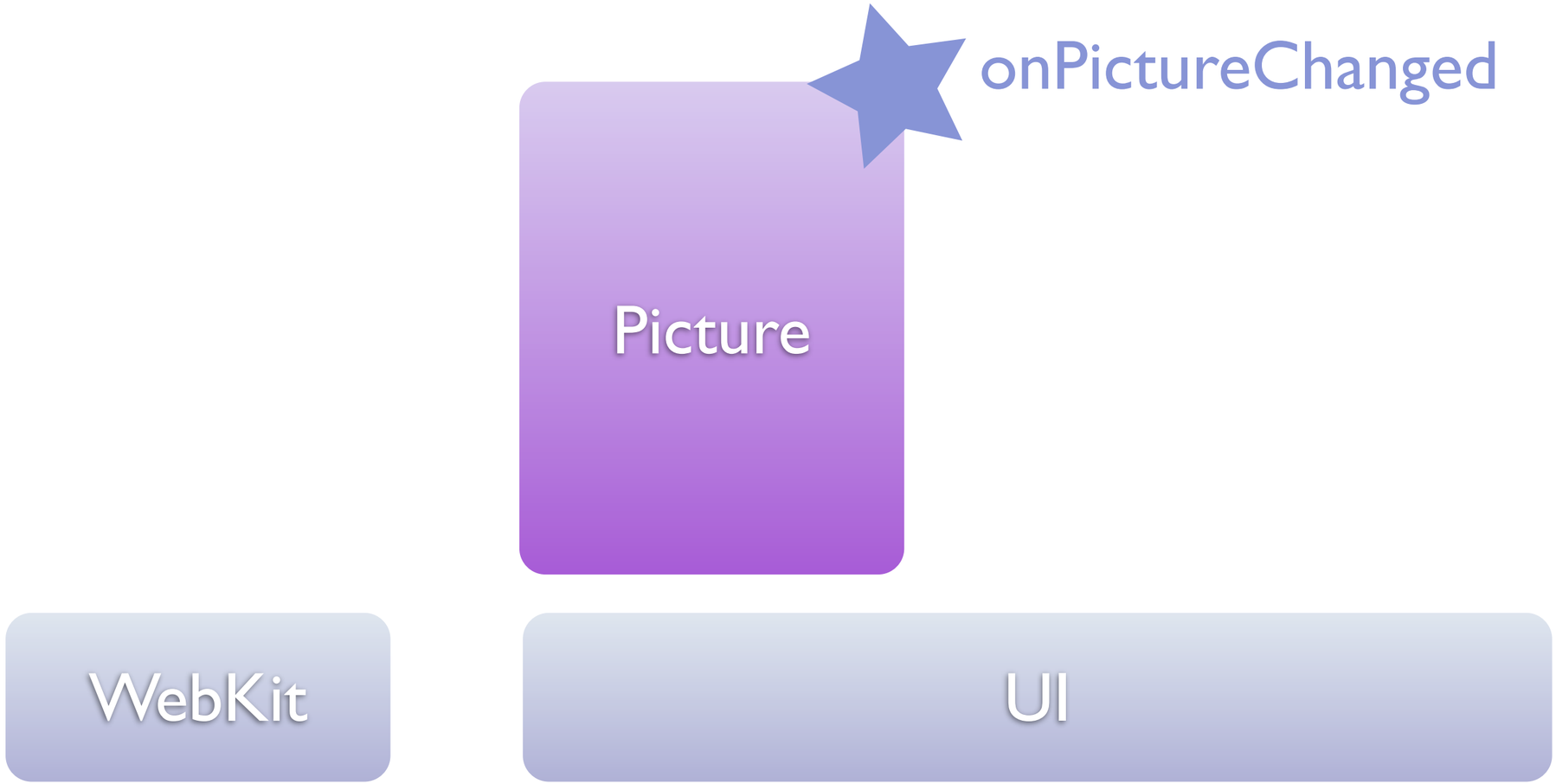


Asynchronous drawing

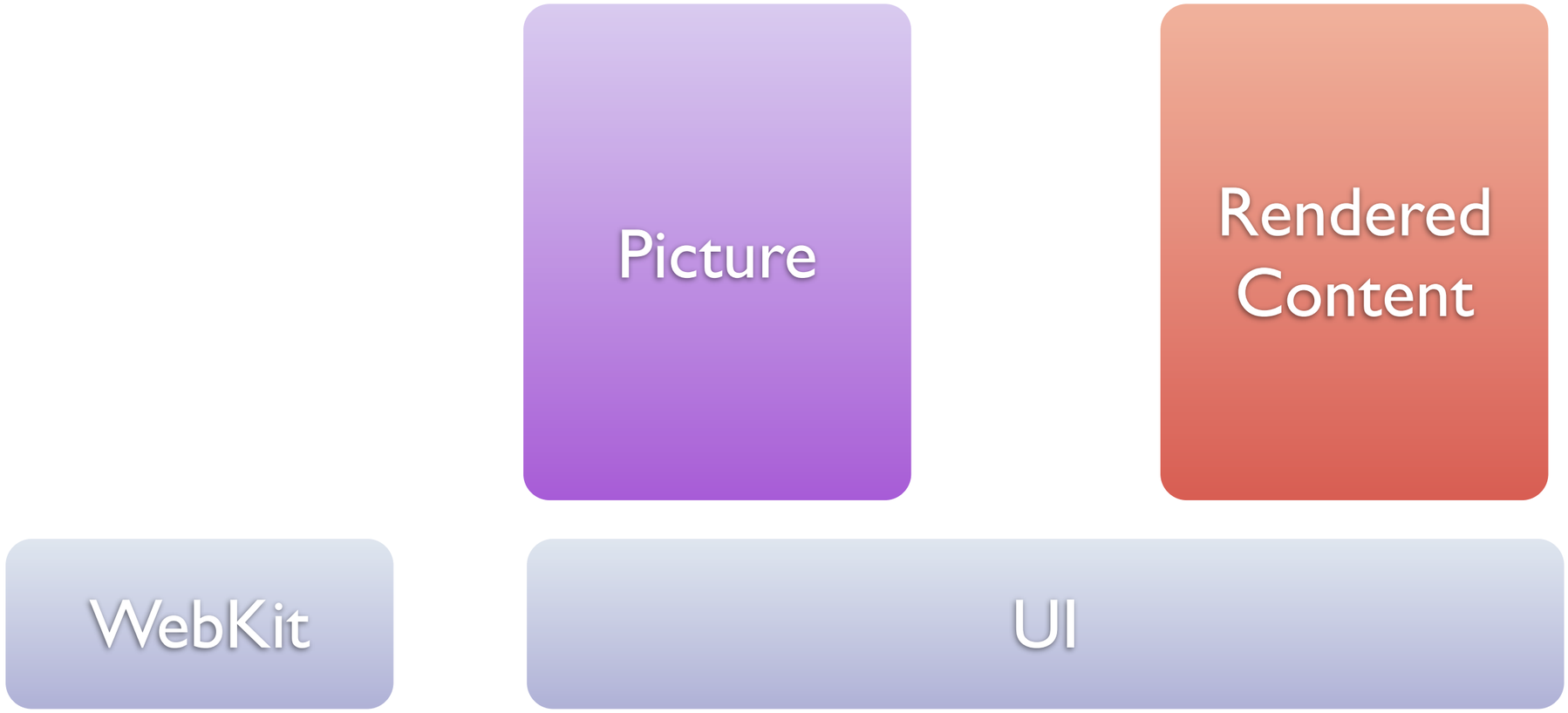
Asynchronous drawing



Asynchronous drawing



Asynchronous drawing



Synchronizing rendering, example

- We want to know when some content has finished rendering
  - show some text message in the meantime
- Use the fact that CSS animations are synchronized
- Solution in 4 parts:
  - bridge object
  - CSS
  - HTML
  - JavaScript

Bridge

```
public class JavaScriptInterface {
    View mView;

    JavaScriptInterface(View v) {
        mView = v;
    }

    public void finishedRendering() {
        mView.setVisibility(View.INVISIBLE);
    }
}

...

mWebView.addJavascriptInterface(
            new JavaScriptInterface(mTextView), "android");
```

CSS Magic

```
#transitionElement {
    -webkit-transform: translate3d(0, 0, 0);
}
```

```
#transitionElement.finish {
    -webkit-transform: translate3d(0, 0, 1px);
    -webkit-transition-duration: 1ms;
}
```

HTML

```
<body onload="load()">
...

<div id="transitionElement">
</body>
```

JavaScript

```
function load() {
  document.getElementById('transitionElement').className = 'finish';
  document.getElementById('transitionElement').addEventListener(
                      'webkitTransitionEnd',
                          function(e) {
                              android.finishedRendering()
                          }
                      );
}
```

Debug

Using the Console

Using the Console

```
console.log("Hello World"); // Javascript
```

Using the Console

```
console.log("Hello World"); // Javascript
```

```
adb logcat
```

111

Using the Console

```
console.log("Hello World"); // Javascript

adb logcat
```

Console: Hello World http://www.example.com/hello.html :82

Using the Console

```
console.log(String)
console.info(String)
console.warn(String)
console.error(String)
```

112

Using Console in WebView

```
WebView myWebView = (WebView) findViewById(R.id.webview);
myWebView.setWebChromeClient(new WebChromeClient() {
    public boolean onConsoleMessage(ConsoleMessage cm) {
        Log.d("MyApplication", cm.message() + " -- From line "
                             + cm.lineNumber() + " of "
                             + cm.sourceId() );
        return true;
    }
});
```

Selenium WebDriver

Selenium WebDriver

- Supports touch, gestures, screenshot captures
- Can be started from Eclipse

- http://seleniumhq.wordpress.com/
- http://code.google.com/p/selenium/wiki/AndroidDriver
- http://selenium.googlecode.com/svn/trunk/docs/api/java/index.html

Example

```
public void testGoogleShouldWork() {

    // Create a WebDriver instance with the activity in
    // which we want the test to run
    WebDriver driver = new AndroidDriver(getActivity());

    // Let's open a web page
    driver.get("http://www.google.com");

    // Lookup for the search box by its name
    WebElement searchBox = driver.findElement(By.name("q"));

    // Enter a search query and submit
    searchBox.sendKeys("weather in san francisco");
    searchBox.submit();
```

Example

```
    // Making sure that Google shows 11 results
    WebElement resultSection = driver.findElement(By.id("ires"));
    List<WebElement> searchResults =
                            resultSection.findElements(By.tagName("li"));
    assertEquals(11, searchResults.size());

    // Let's ensure that the first result shown is the weather widget
    WebElement weatherWidget = searchResults.get(0);
    assertTrue(weatherWidget.getText().contains(
                                        "Weather for San Francisco, CA"));
}
```

Android Browser debug mode

Google I/O 2012

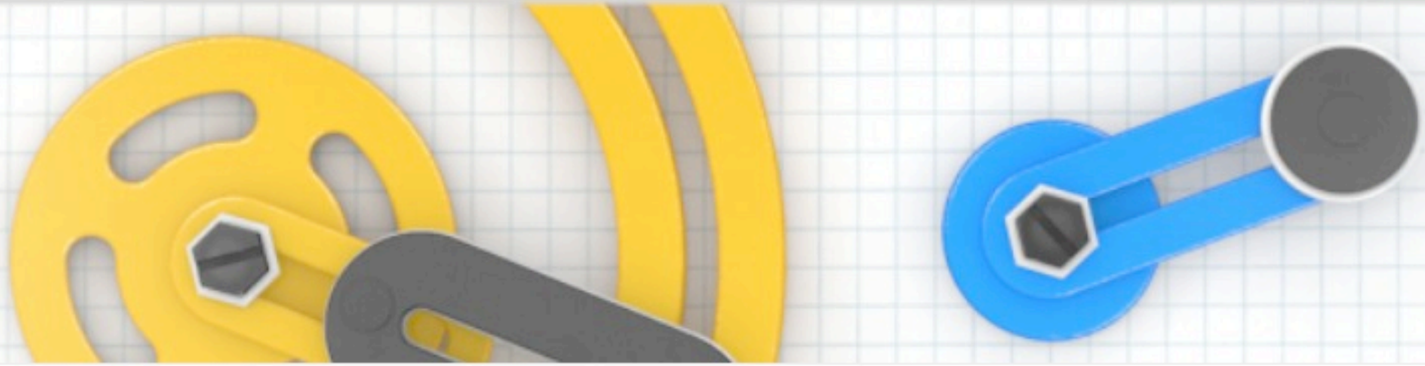
https://developers.google.com/events/io/sessions

Developers Home +1 36k

Google 12 I/O

June 27-29, 2012 Moscone Center, San Francisco

Schedule: Sessions Sandbox Events Info



input/output

Create your own machine
in our Chrome Experiment.

Start Building

Sessions | Select a Track | Day 1 Day 2 Day 3

Google I/O 2012

about:debug

Developers Home

+1 36k

Google 12 I/O

June 27-29, 2012 Moscone Center, San Francisco

Schedule: Sessions Sandbox Events Info

Google I/O 2012

about:debug

Developers Home

+1 36k

Google 12 I/O

June 27-29, 2012 Moscone Center, San Francisco

Schedule: Sessions Sandbox Events Info

input/output

5:53

Settings

General

Privacy & security

Accessibility

Advanced

Bandwidth management

Labs

Debug

5:53

Settings

General

Privacy & security

Accessibility

Advanced

Bandwidth management

Labs

Debug

5:53

Settings

Enable OpenGL Rendering

Enable HW Accelerated Skia

UAString

Enable Visual Indicator

Enable Cpu Upload Path

Show JavaScript Console

Single column rendering

5:53

Settings

Enable OpenGL Rendering

Enable HW Accelerated Skia

UAString

Enable Visual Indicator

Enable Cpu Upload Path

Show JavaScript Console

Single column rendering

Enable OpenGL Re

Enable HW Accele

UAString

Enable Visual Indic

Enable Cpu Upload

Show JavaScript C

Single column rendering

UAString

Android

Desktop

iPhone

iPad

Froyo-N1

Honeycomb-Xoom

Cancel

Settings

Enable OpenGL Rendering ☑

Enable HW Accelerated Skia ☐

UAString

Enable Visual Indicator ☐

Enable Cpu Upload Path ☐

Show JavaScript Console ☑

Single column rendering ☐

Settings

Enable OpenGL Rendering ☑

Enable HW Accelerated Skia ☐

UAString

Enable Visual Indicator ☐

Enable Cpu Upload Path ☐

Show JavaScript Console ☑

Single column rendering ☐

5:54

Settings

Enable OpenGL Rendering

Enable HW Accelerated Skia

UAString

Enable Visual Indicator

Enable Cpu Upload Path

Show JavaScript Console

Single column rendering

(0,0) 1.28x 56 11.8ms (1,0) 1.28x 56 11.2ms (2,0) 1.28x 56 8.5ms (3,0) 1.28x 56 10.4ms (4,0) 1.28x 56 13.0ms

Developers Home

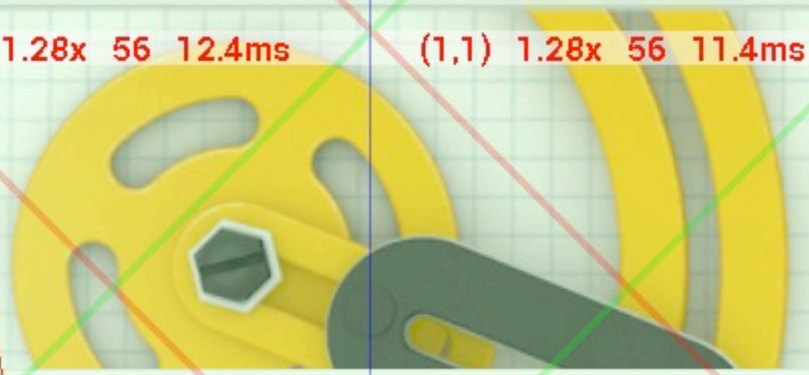
+1 36k

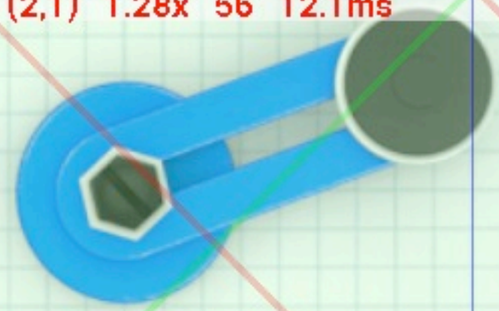
Google 12 I/O

June 27-29, 2012 Moscone Center, San Francisco

Schedule: Sessions Sandbox Events Info

(0,1) 1.28x 56 12.4ms (1,1) 1.28x 56 11.4ms (2,1) 1.28x 56 12.1ms (3,1) 1.28x 56 13.7ms (4,1) 1.28x 56 15.7ms





input/output

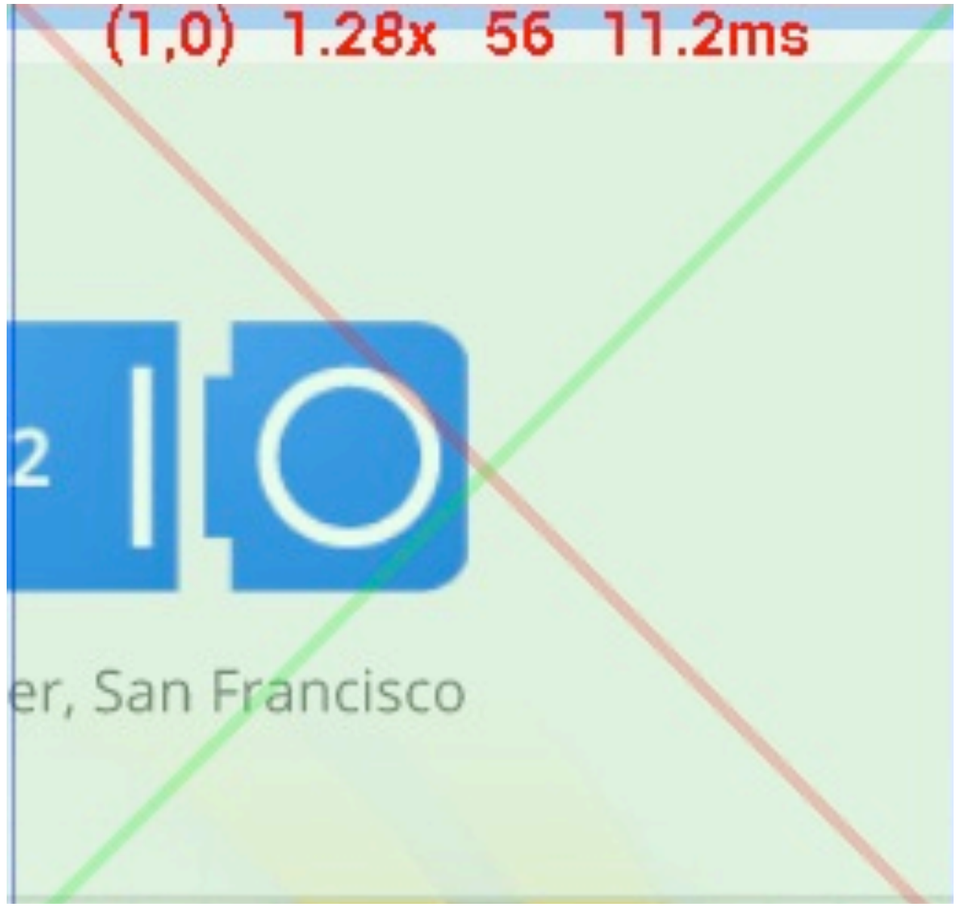
Create your own machine
in our Chrome Experiment.

Start Building

Sessions Select a Track Day 1 Day 2 Day 3

(0,2) 1.28x 56 23.6ms (1,2) 1.28x 56 13.6ms (2,2) 1.28x 56 13.4ms (3,2) 1.28x 56 15.1ms (4,2) 1.28x 56 10.5ms

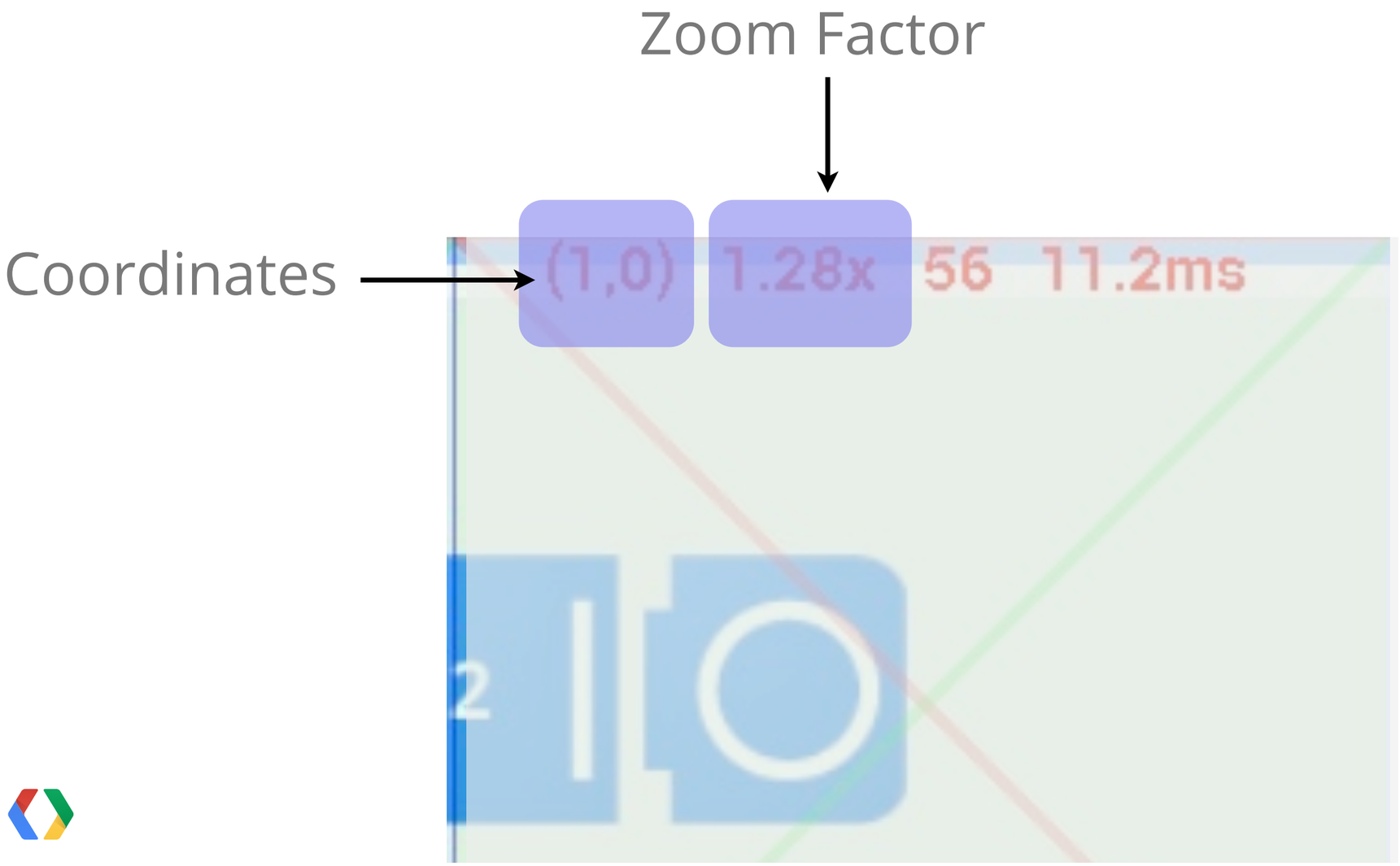
Anatomy of a Tile



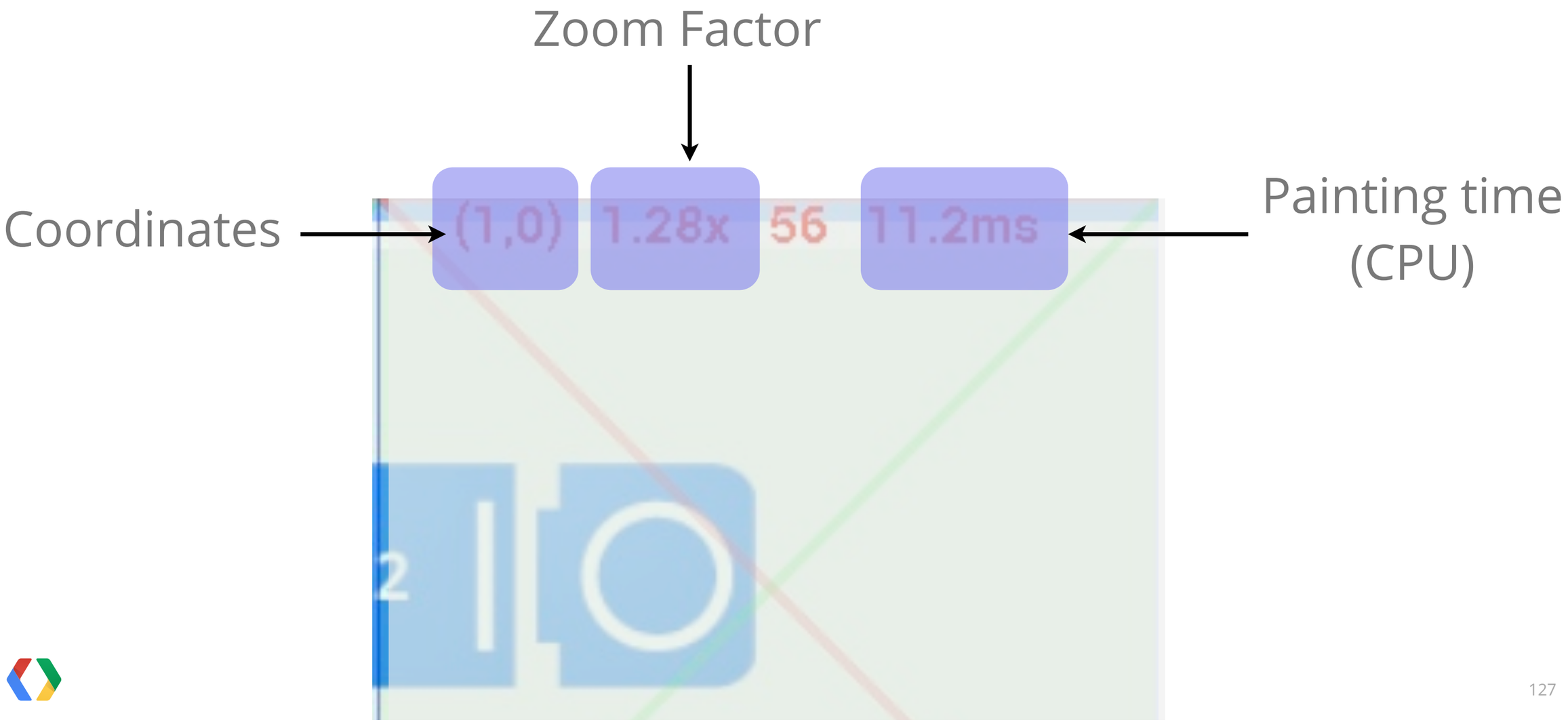
Anatomy of a Tile

Coordinates

Anatomy of a Tile



Anatomy of a Tile




Google I/O 2012
(0,0) 1.00x 58 1.6ms
(1,0) 1.00x 58 1.3ms
(2,0) 1.00x 57 0.5ms
(3,0) 1.00x 57 0.6ms
Sessions
Select a Track
Day 1
Day 2
Day 3
370
(0,0) 1.00x 71 1.6ms
(1,0) 1.00x 71 1.4ms
(2,0) 1.00x 71 3.5ms
(3,0) 1.00x 70 1.9ms
391
Android
Ten Things Game Developers Should Know
Android WebView
Measuring The End-To-End Value Of Your App
Security And Privacy In Android Apps
(3,0) 1.28x 71 0.6ms
(0,16) 1.00x 71 12.0ms
(1,16) 1.00x 71 10.8ms
(2,16) 1.00x 71 13.3ms
(3,16) 1.00x 71 6.5ms
Chrome
Chrome/OS Fireside Chat
High Performance HTML5
The History And Future Of Google Web Toolkit
Writing Secure Web Apps And Chrome Extensions
Beyond Paper: Google Cloud Print And The Future Of Printing
(3,0) 1.28x 71 0.5ms
Crunching Big Data With BigQuery
Managing Google Compute Engine Virtual Machines
Optimizing Your Google App Engine App
Gaming In The Cloud
Powering Your Application's Data Using Google Cloud

(0,0) 1.31x 79 0.4ms (1,0) 1.31x 79 0.3ms (2,0) 1.31x 79 0.2ms (3,0) 1.31x 79 0.5ms (4,0) 1.31x 79 0.3ms

(0,1) 1.31x 79 0.2ms (1,1) 1 2,1) 1.31x 79 0.3ms (3,1) 1.31x 79 0.2ms (4,1) 1.31x 79 0.2ms

(0,2) 1.31x 79 0.7ms (1,2) 1.31x 79 0.3ms (2,2) 1.31x 79 0.3ms (3,2) 1.31x 79 1.3ms (4,2) 1.31x 79 2.5ms

Google™ 12


(0,0) 1.00x 101 2.8ms
(1,0) 1.
(2,1) 2.70x 79 0.2ms
(3,1) 2.70x 79 0.4ms
(6,1) 2.70x 79 0.6ms
(0,0) 1.00x 32
(0,0) 1.00x 57 2.0ms
(1,0) 1
(3,2) 2
ms
0.4ms
2.70x 79 0.2ms

about:debug commands

about:debug.render
about:debug.render.file
about:debug.dom
about:debug.dom.file
about:debug.display

RenderTree

RenderTree

about:debug.render.file

RenderTree

about:debug.render.file /sdcard/renderTree.txt

RenderTree

about:debug.render.file /sdcard/renderTree.txt

adb pull /sdcard/renderTree.txt

RenderTree

```
layer at (0,0) size 962x520
  RenderView at (0,0) size 962x50
 layer at (0,0) size 962x520
  RenderBlock {HTML} at (0,0) size 962x520
    RenderBody {BODY} at (0,0) size 962x520 [color=#222222] [bgcolor=#FFFFFF]
      RenderBlock {DIV} at (0,0) size 962x30
        RenderBlock {DIV} at (0,0) size 962x0
RenderBlock {DIV} at (0,30) size 962x0
RenderBlock {DIV} at (0,69) size 962x451
  RenderBlock {SPAN} at (0,0) size 962x402
    RenderBlock {CENTER} at (0,0) size 962x402
      RenderBlock {DIV} at (0,0) size 962x231
        RenderImage {IMG} at (299,0) size 364x207
      RenderBlock {DIV} at (0,231) size 962x102
      RenderBlock {DIV} at (0,333) size 962x69
```

LayersTree

about:debug.display /sdcard/layersTree.plist

```
adb pull /sdcard/layersTree.plist
```

about:debug.display

layer = 616fe7e8;
layerId = 370;
haveClip = 0;
isFixed = 1;
opacity = 1.000;
size = { w = 1000.000; h = 57.000; };
position = { x = 0.000; y = 3991.406; };
anchor = { x = 0.500; y = 0.500; };
drawMatrix = { (1.00,0.00,0.00,0.00),(0.00,1.00,0.00,0.00),
(0.00,0.00,1.00,0.00),(0.00,3991.41,0.00,1.00) };
transformMatrix = { (1.00,0.00,0.00,0.00),(0.00,1.00,0.00,0.00),
(0.00,0.00,1.00,0.00),(0.00,0.00,0.00,1.00) };
clippingRect = { x = 0.000; y = 332.000; w = 1000.000; h = 5347.000; };
m_content.width = 1000;
m_content.height = 57;
fixedLeft = { type = 4; value = 0.00; };
fixedTop = { type = 4; value = 0.00; };
fixedRight = { type = 4; value = 0.00; };
fixedMarginLeft = { type = 4; value = 0.00; };

about:debug.display

layer = 616fe7e8;
layerId = 370;
haveClip = 0;
isFixed = 1;
opacity = 1.000;
size = { w = 1000.000; h = 57.000; };
position = { x = 0.000; y = 3991.406; };
anchor = { x = 0.500; y = 0.500; };
drawMatrix = { (1.00,0.00,0.00,0.00),(0.00,1.00,0.00,0.00),
(0.00,0.00,1.00,0.00),(0.00,3991.41,0.00,1.00) };
transformMatrix = { (1.00,0.00,0.00,0.00),(0.00,1.00,0.00,0.00),
(0.00,0.00,1.00,0.00),(0.00,0.00,0.00,1.00) };
clippingRect = { x = 0.000; y = 332.000; w = 1000.000; h = 5347.000; };
m_content.width = 1000;
m_content.height = 57;
fixedLeft = { type = 4; value = 0.00; };
fixedTop = { type = 4; value = 0.00; };
fixedRight = { type = 4; value = 0.00; };
fixedMarginLeft = { type = 4; value = 0.00; };

(0,0) 1.00x 58 1.6ms (1,0) 1.00x 58 1.3ms (2,0) 1.00x 57 0.5ms (3,0) 1.00x 57 0.6ms

Sessions Select a Track Day 1 Day 2 Day 3

370

(0,0) 1.00x 71 1.6ms (1,0) 1.00x 71 1.4ms (2,0) 1.00x 71 3.5ms (3,0) 1.00x 70 1.9ms

391

Android

Ten Things Game Developers Should Know

Android WebView

Measuring The End-To-End Value Of Your App

Security And Privacy In Android Apps

(3,16) 1.28x 71 0.6ms

(0,16) 1.00x 71 12.0ms (1,16) 1.00x 71 10.8ms (2,16) 1.00x 71 13.3ms (3,16) 1.00x 71 6.5ms

Chrome

Chrome/OS Fireside Chat

High Performance HTML5

The History And Future Of Google Web Toolkit

Writing Secure Web Apps And Chrome Extensions

Beyond Paper: Google Cloud Print And The Future Of Printing

(3,17) 1.28x 71 0.5ms

Crunching Big Data With BigQuery

Managing Google Compute Engine Virtual Machines

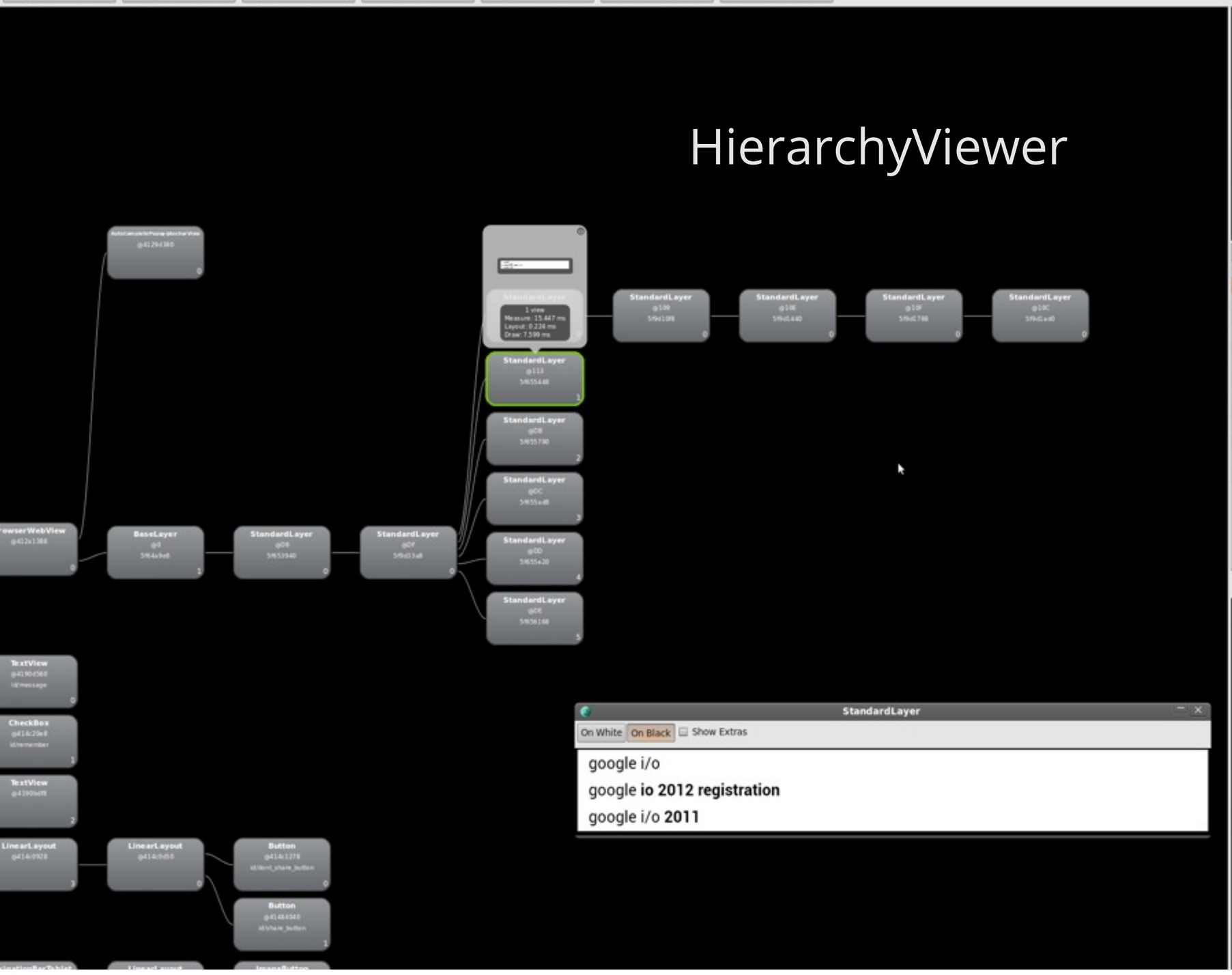
Optimizing Your Google App Engine App

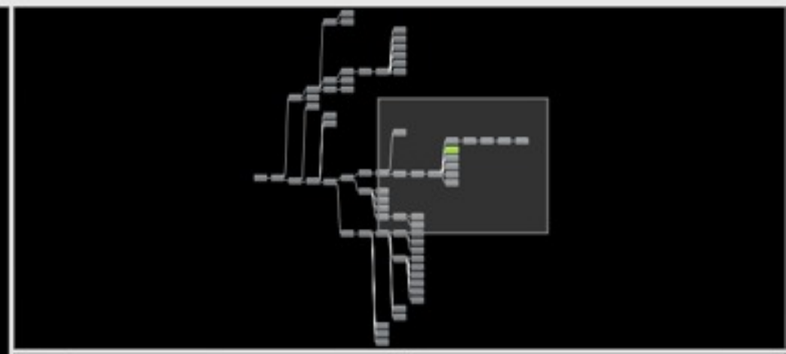
Gaming In The Cloud

Powering Your Application's Data Using Google Cloud

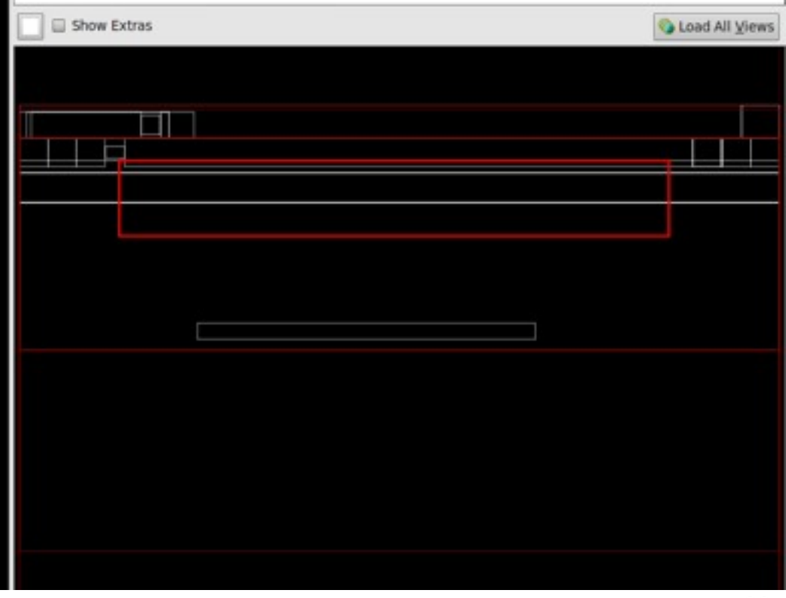
What’s coming up

HierarchyViewer





Property	Value
▷ Layout	
▽ Miscellaneous	
anchor	{ x = 0.500; y = 0.500; }
clippingRect	{ x = 0.000; y = 0.000; w = 1280.000; h = 2461.000; }
drawMatrix	{ (1.00,0.00,0.00,0.00),(0.00,1.00,0.00,0.00),(0.00,0.00,1.00
haveClip	0
isFixed	0
layerId	275
mID	5f655448
m_content.height	127
m_content.width	926
opacity	1.000
position	{ x = 15168.000; y = 26854.000; }
size	{ w = 926.000; h = 127.000; }
transformMatrix	{ (1.00,0.00,0.00,0.00),(0.00,1.00,0.00,0.00),(0.00,0.00,1.00



Questions?