



Shading Language Update

Bill Licea-Kane

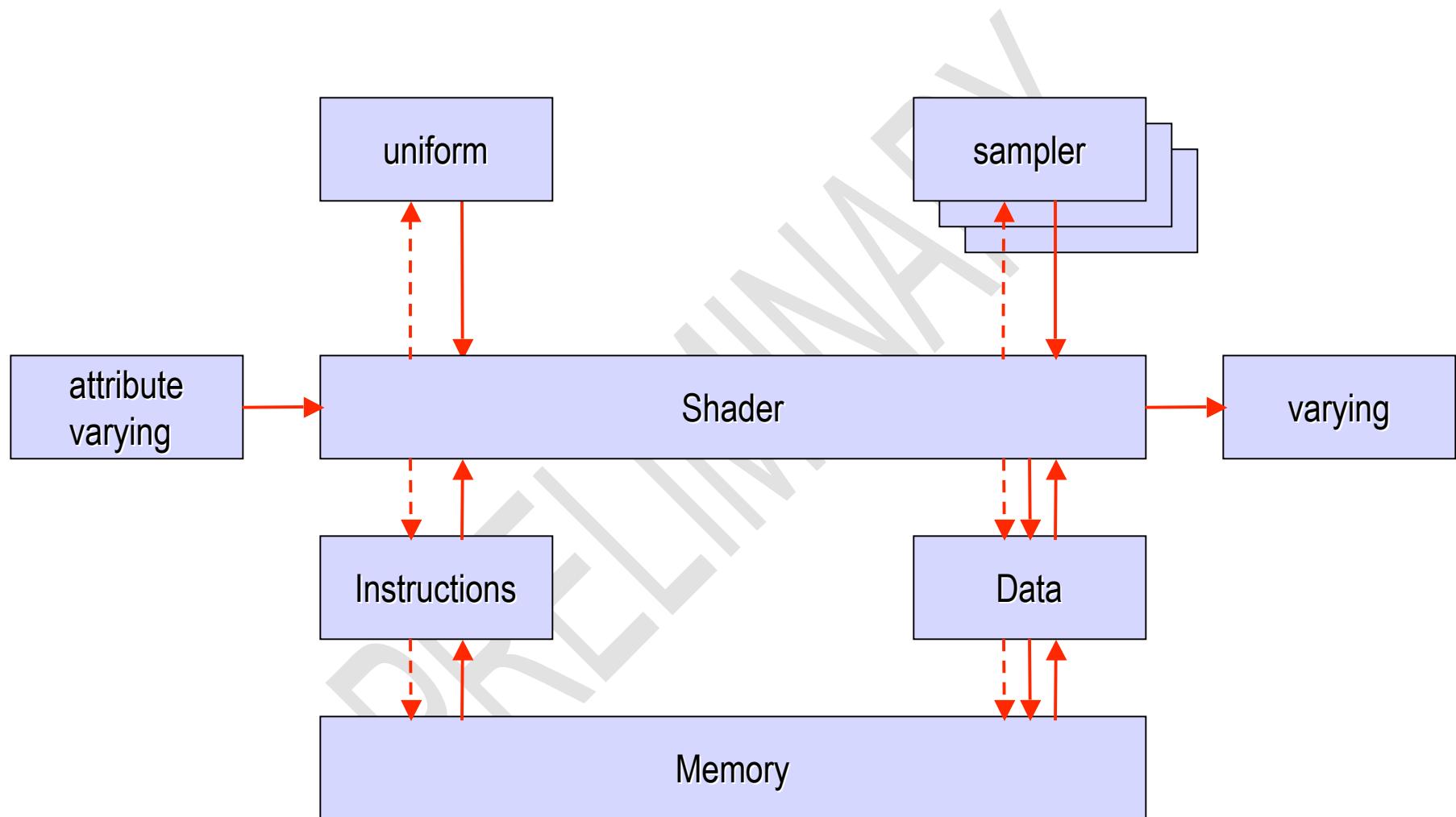


About this talk - Caveat BOFor

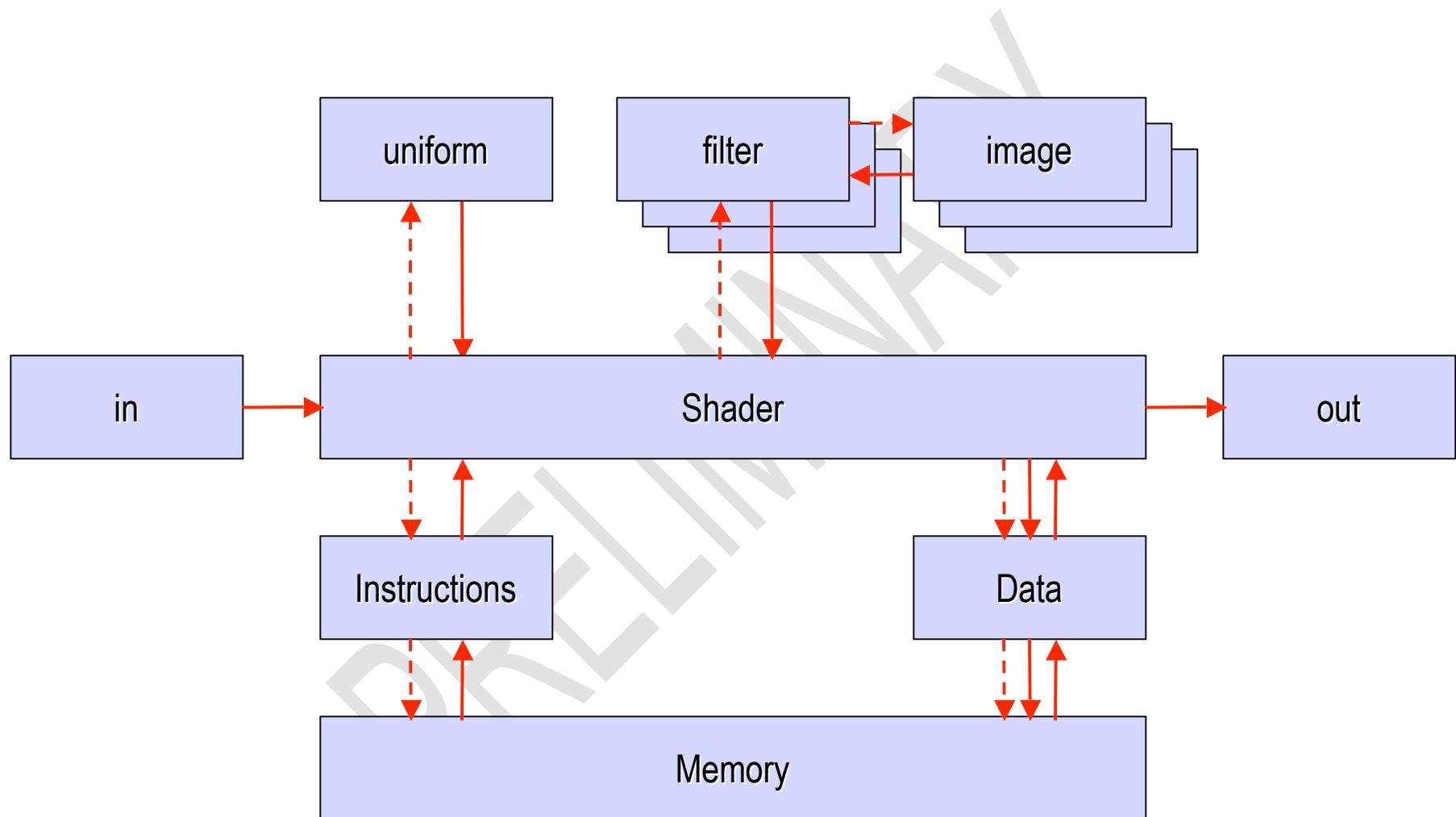
- Contributions to arb-glsl workgroup
 - NOT final, subject to (minor?) changes
- Not exhaustive
 - There is more stuff that will not be covered today
- A couple of BRIEF digressions "(and beyond)"

PRELIMINARY

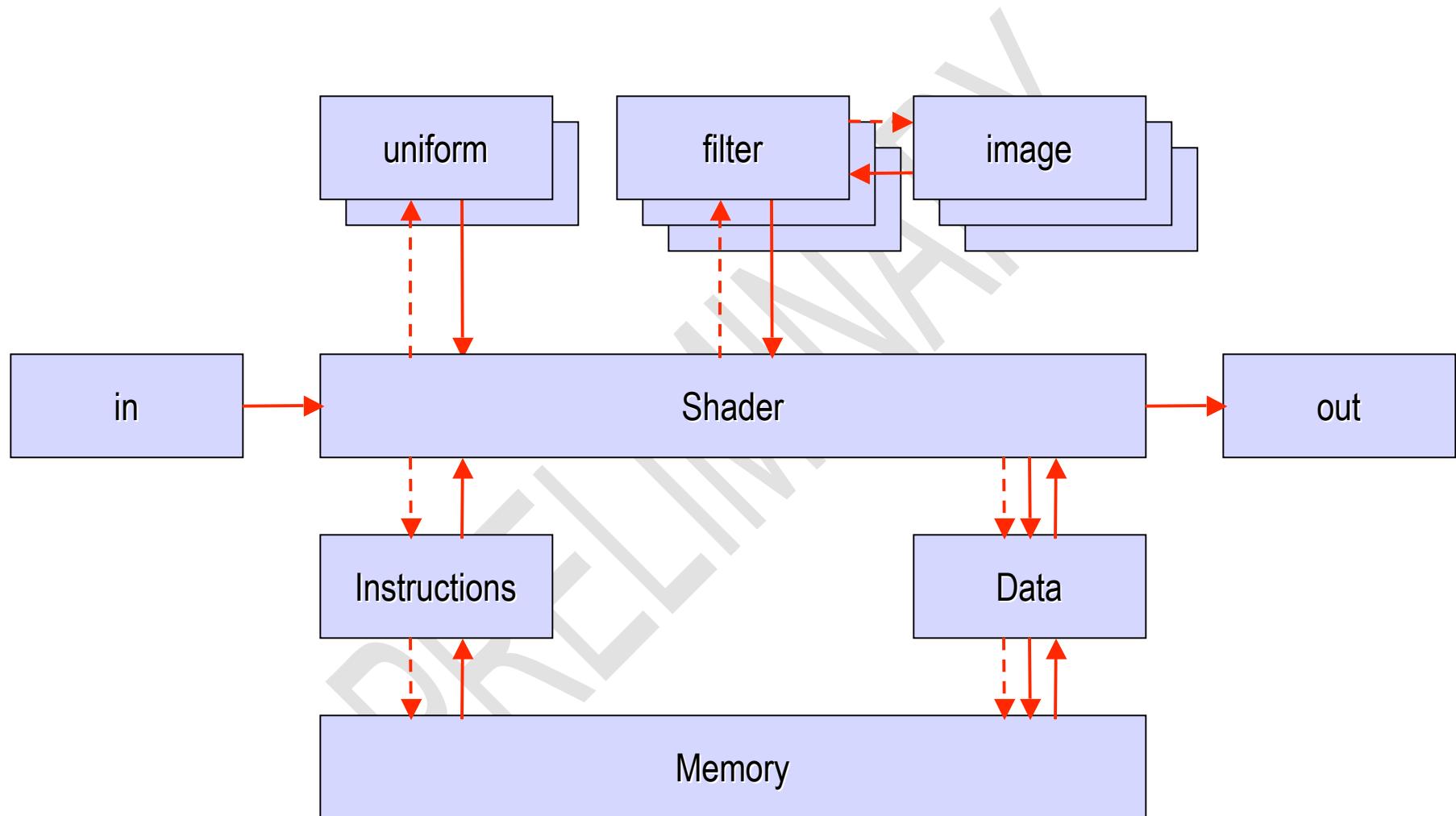
OpenGL 2 Shader



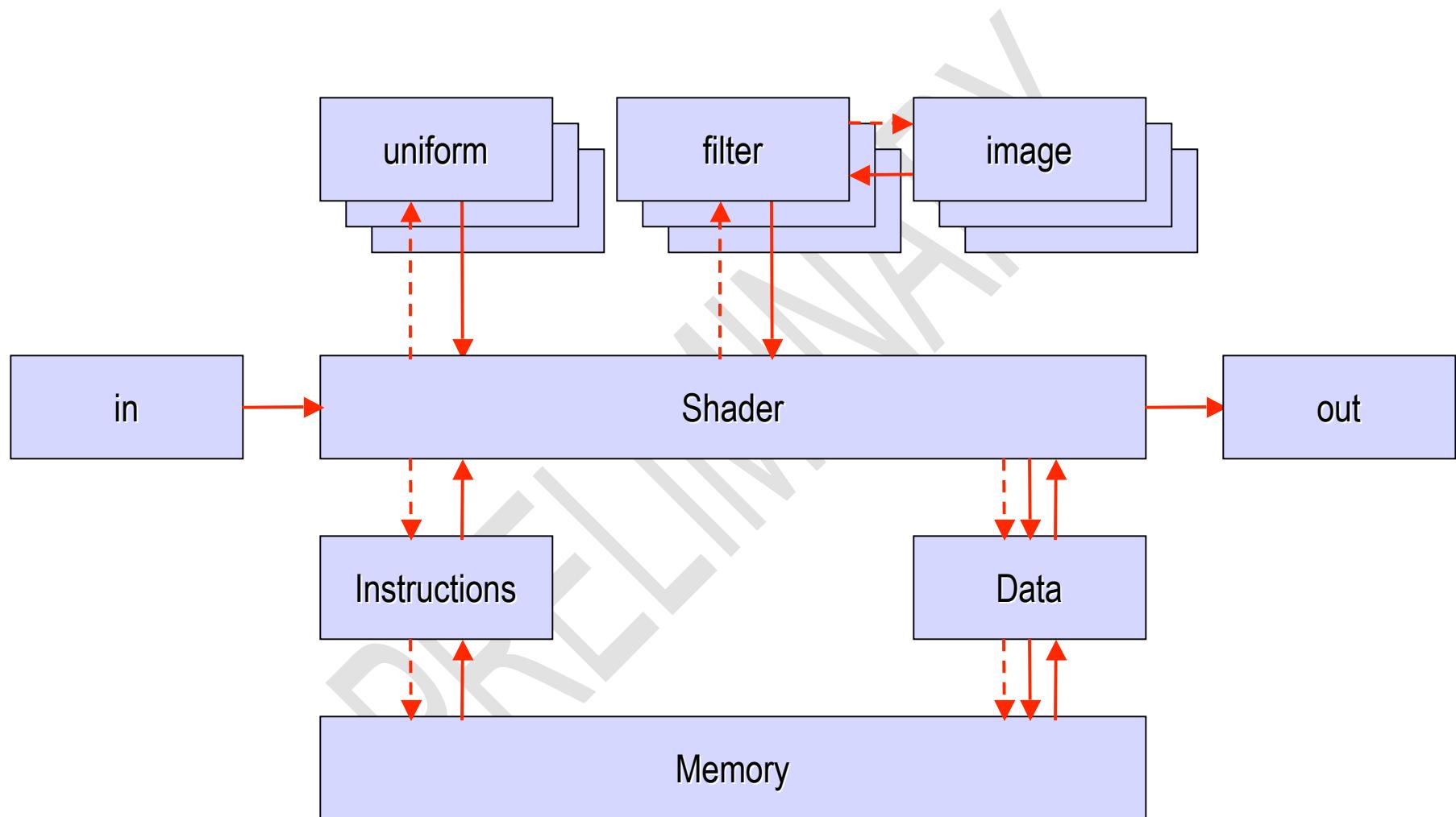
OpenGL 3 Shader



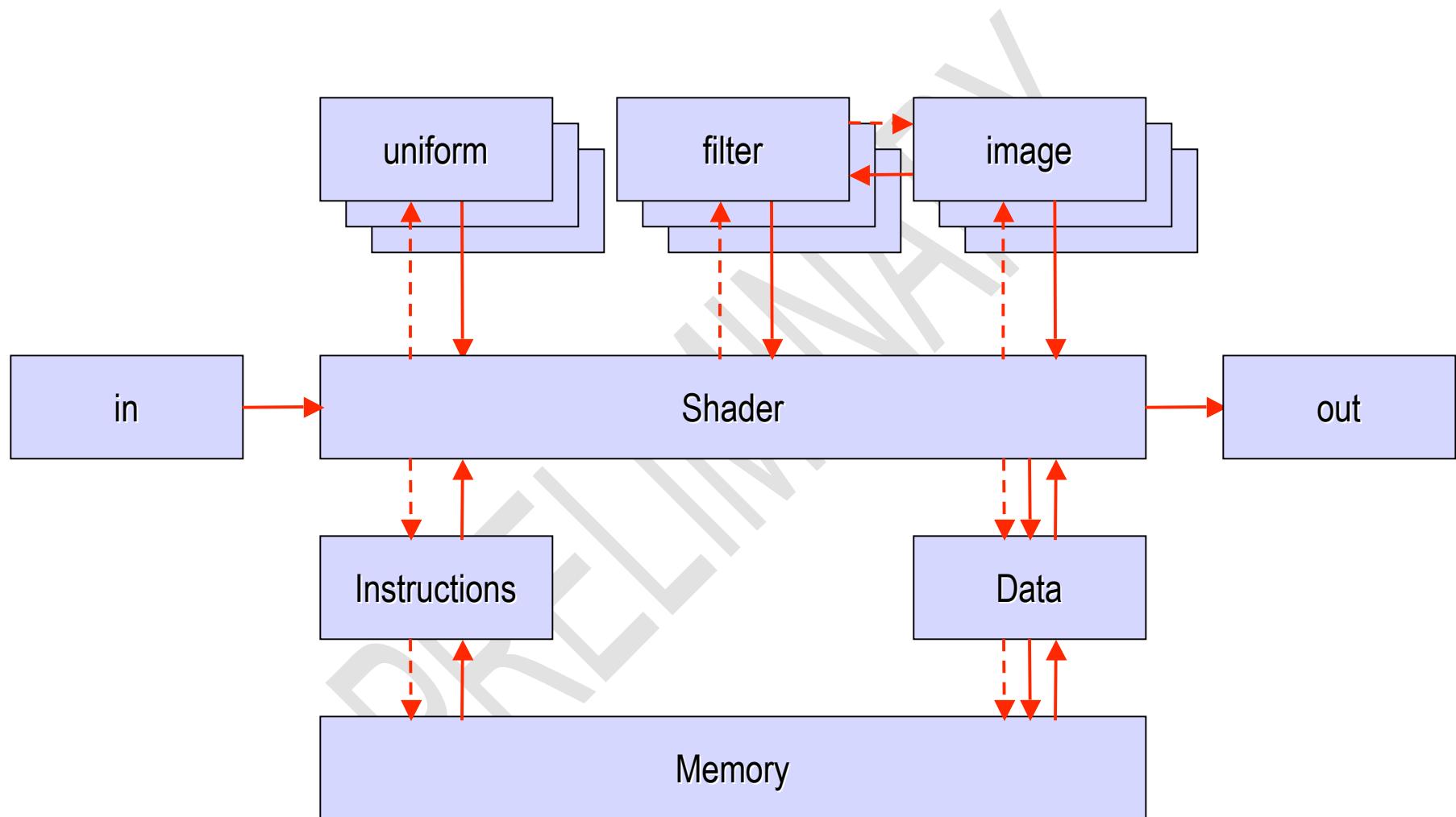
OpenGL 3 Shader



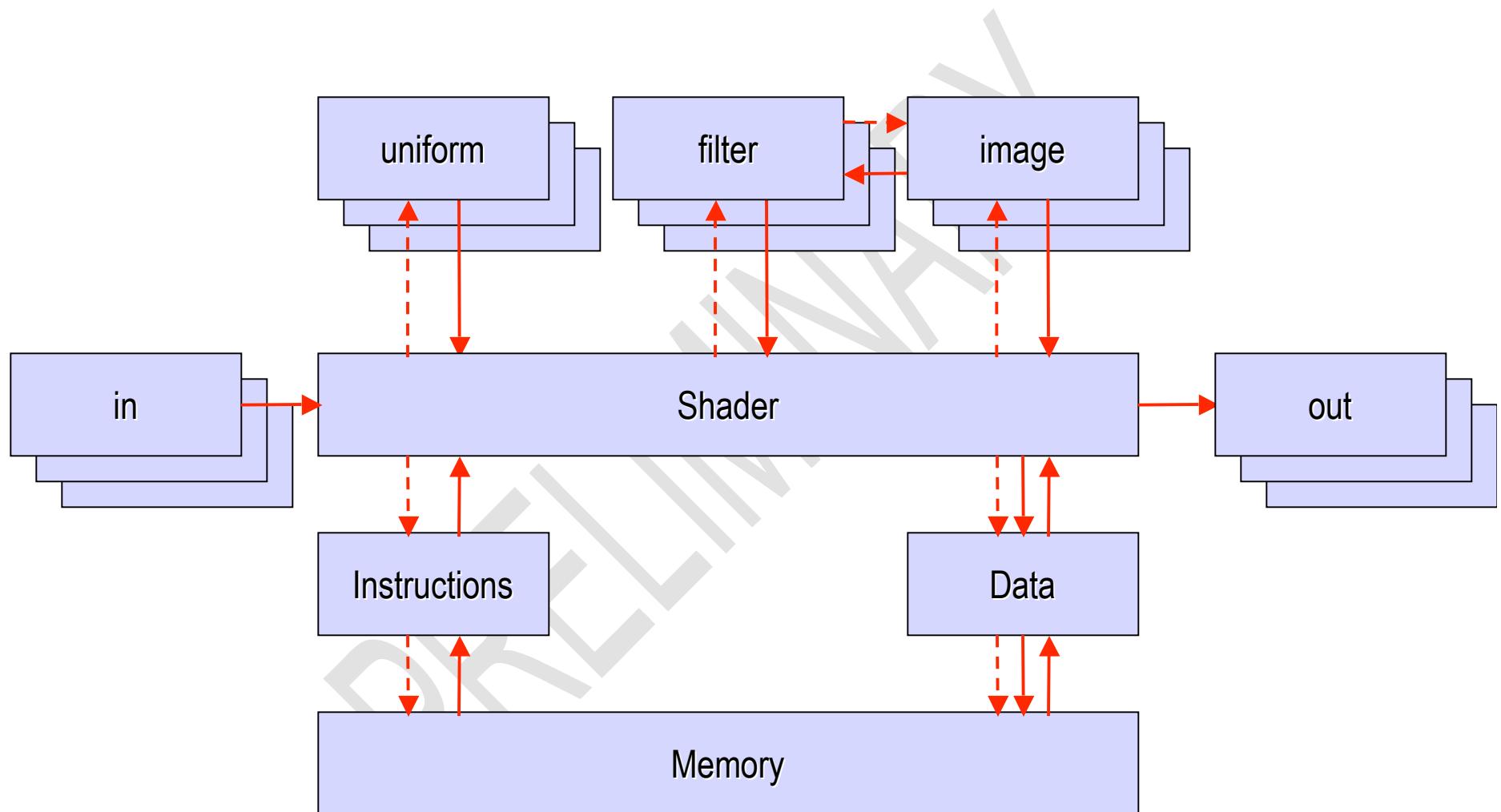
OpenGL 3 Shader (and beyond)



OpenGL 3 Shader (and beyond)



OpenGL 3 Shader (and beyond)



Language Changes

OpenGL 2.0 Shading Language // existing

OpenGL 3.0 Shading Language // proposed

// vertex shader

vertex shader text begins here

and ends...

// fragment shader

fragment shader text begins here

attribute / varying

```
attribute vec4 myNormal;  
varying vec4 vNormal;
```

attribute / varying

```
attribute vec4 myNormal;  
varying vec4 vNormal;
```

```
in vec4 myNormal;  
out vec4 vNormal;
```

attribute / varying

```
attribute vec4 myNormal;  
varying vec4 vNormal;
```

```
in vec4 myNormal;  
out vec4 vNormal;  
  
inout vec4 myNormal;
```

attribute / varying - nota bene

```
// vertex shader  
inout vec4 myNormal;
```

```
// fragment shader  
in vec4 myNormal;
```

attribute / varying - qualifiers

```
centroid varying vec2 myTC;
```

```
// vertex shader
```

```
out centroid vec2 myTC;
```

```
// fragment shader
```

```
in centroid vec2 myTC;
```

attribute / varying - qualifiers

```
varying vec2 myTC;
```

```
// vertex shader
```

```
out smooth vec2 myTC;
```

```
// fragment shader
```

```
in smooth vec2 myTC;
```

attribute / varying - qualifiers

```
// vertex shader  
out smooth vec2 myTC;  
out flat vec4 gl_FrontColor;
```

```
// fragment shader  
in smooth vec2 myTC;  
in flat vec4 gl_Color;
```

qualifiers (and beyond)

```
// vertex shader  
out smooth vec2 myTC;  
out flat float myTemperature;  
  
// fragment shader  
in smooth vec2 myTC;  
in flat float myTemperature;
```

common blocks - uniform buffers

```
common myPerContextData {  
    uniform mat4 MVP;  
    uniform mat3 MVIT;  
    uniform vec4 LightPos[3];  
    // ONLY uniforms, but...  
    // no samplers  
    // no int types  
    // no bool types  
};
```

fixed function state

```
vec4 Peye = gl_ModelViewMatrix * P;
```

```
// "built-in" uniforms  
// are convenience only  
// NO state tracking
```

array of samplers

```
uniform sampler2D BaseMaps[8];
```

```
// deprecated
```

```
// even if hidden in a struct?
```

array of samplers (and beyond)

```
// fragment shader
in flat float Map;
in smooth vec2 MyTC;
// uniform Sampler2D Basemaps[8];
uniform image2DArray Basemaps_image;
uniform filter Basemaps_filter;

vec4 baseColor =
    texture2DArray( Basemaps_image,
                    Basemaps_filter,
                    vec3( myTC, Map ) );
```

preprocessor - extended

##

#include

PRELIMINARY

OpenGL ES Qualifiers

```
lowp      // expect silently ignored  
mediump   // expect silently ignored  
highp     // expect silently ignored
```

Matrix qualifiers

```
uniform row_major mat4x3 bones[10];
```

PRELIMINARY

switch

```
switch ( myUberSelector ) {  
    case 1:  
        // what  
        break;  
    case 2:  
        // ever  
        break;  
    default:  
        // whatever  
}
```

Summary

- **OpenGL 2**
 - attribute / varying
 - centroid varying
 - single uniform global scope
 - built-in is context state
 - built-in uniform state
 - tracked
 - array of samplers
 - preprocessor subset
 - no silently ignored keywords
 - column major only
 - no switch
- **OpenGL 3**
 - in out inout
 - centroid flat smooth
 - global uniforms
 - common is context state
 - built-in uniform state
 - NOT tracked
 - (and beyond) texture2DArray
 - ## and #include
 - ES precision qualifiers
 - row_major qualifier
 - switch



Shading Language Questions?

Bill Licea-Kane

