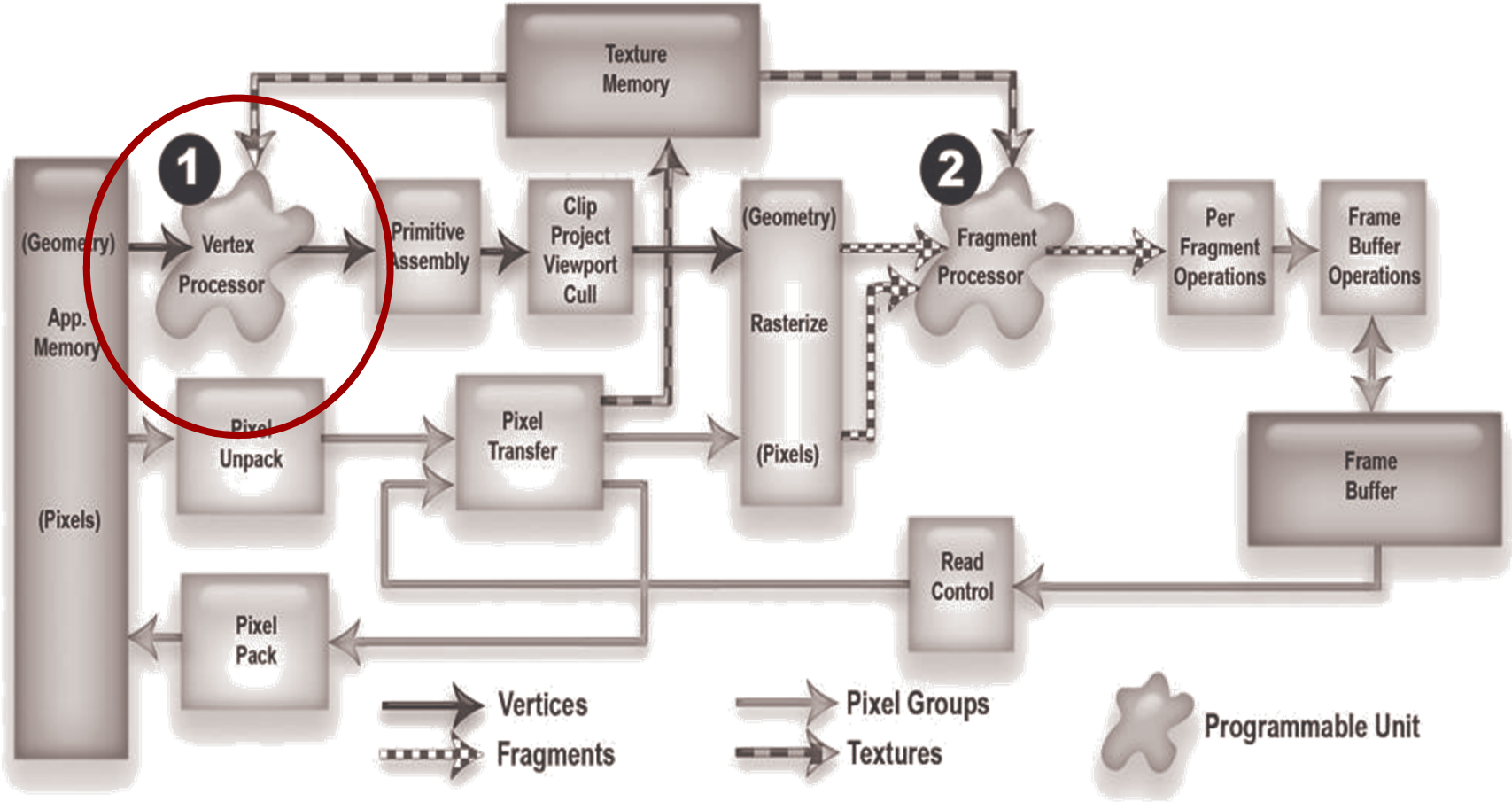
Vertex shaders i Fragment shaders

Carlos Andújar

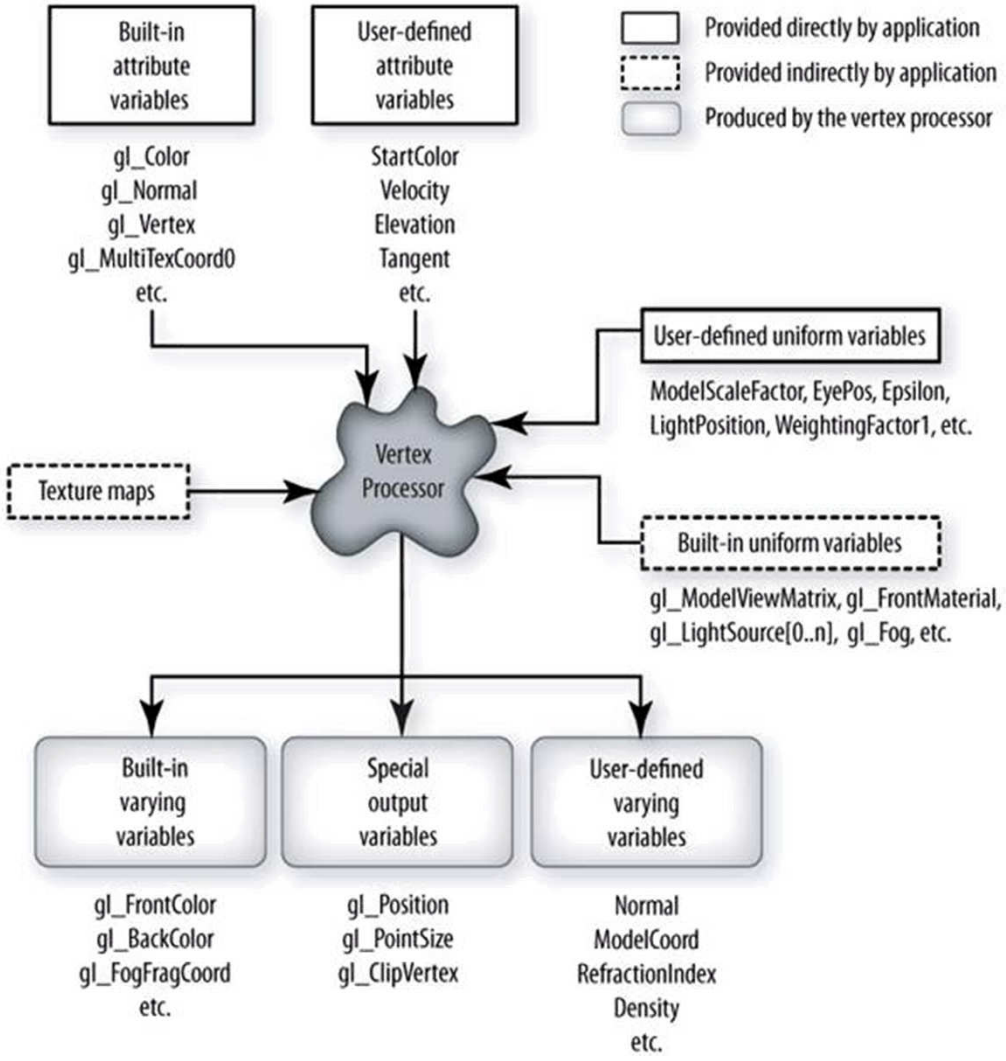
Setembre 2013

VERTEX SHADERS

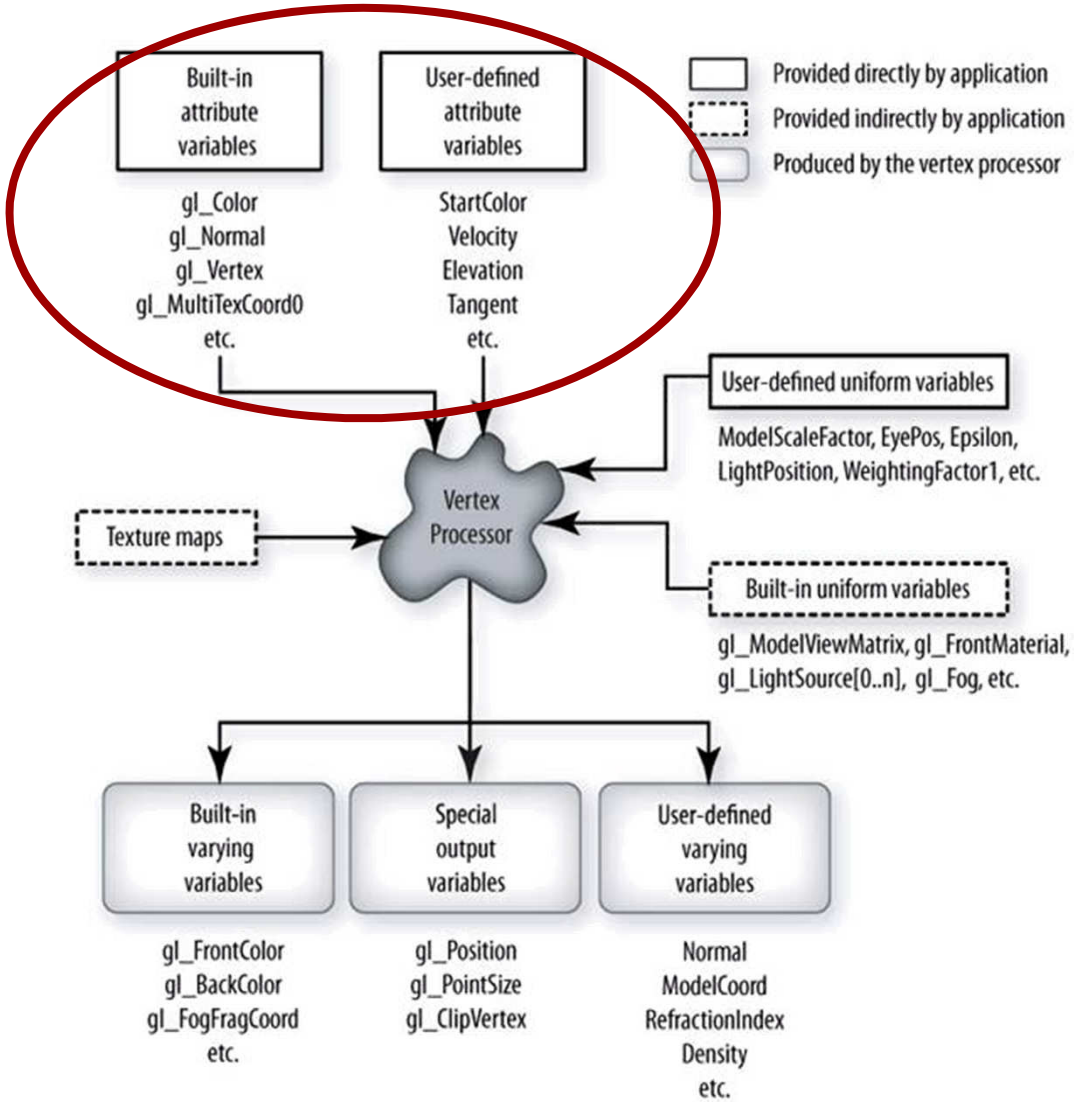
Vertex shaders



Vertex shaders



Vertex shaders



Vertex shaders

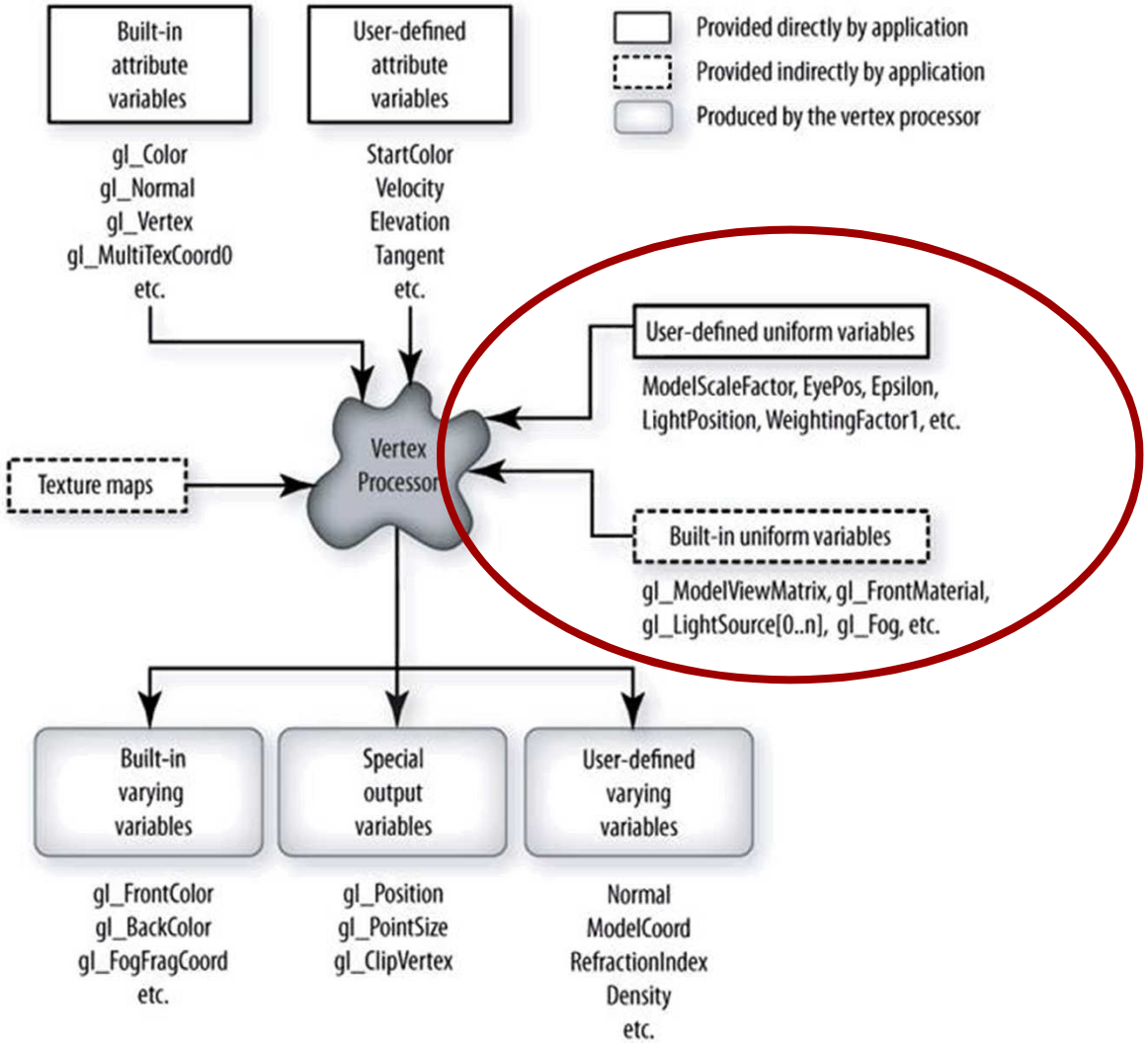
- **Attribute** variables: són variables que representen els *atributs d'un vèrtex*. Poden canviar de valor per cada vèrtex d'una mateixa primitiva.
  - **Built-in** attributes (no cal declarar-los)
    - Des de l'aplicació s'envien amb glColor, glNormal...
    - Des del shader s'accedeixen amb gl_Color, gl_Normal, gl_Vertex...
  - **User-defined** attributes: (cal declarar-los)
    - Des de l'aplicació s'envien amb glVertexAttrib i es lliguen a un nom amb glBindAttribLocation.
    - Des del shader s'accedeixen amb un nom arbitrari definit per l'usuari: velocitat, etc.

Vertex shaders

Atributs predefinits:

vec4 **gl_Color**;	glColor()
vec3 **gl_Normal**;	glNormal();
vec4 **gl_Vertex**;	glVertex();
vec4 **gl_MultiTexCoord0**;	glTextCoord();
vec4 **gl_MultiTexCoord1**;	// . . . up to N
…	

Vertex shaders



Vertex shaders

- **Uniform** variables: són variables que canvien amb poca freqüència. Com a molt poden canviar un cop *per cada primitiva* (però no pas per cada vèrtex de la primitiva).
  - **Built-in variables**: variables d'estat OpenGL
    - Des del shader s'accedeixen amb gl_ModelViewMatrix, glLightSource[0..n], etc
  - **User-defined variables**: cal declarar-les
    - Des de l'aplicació s'envien amb glUniform i es lliguen a un nom amb glGetUniformLocation.
    - Des del shader s'accedeixen amb un nom arbitrari definit per l'usuari: EyePos, etc.

Vertex shaders

Llista variables **uniform predefinides** més importants:

```
// Matrix state
uniform mat4 gl_ModelViewMatrix;
uniform mat4 gl_ProjectionMatrix;
uniform mat4 gl_ModelViewProjectionMatrix;

// Derived matrix state that provides inverse and transposed versions
uniform mat3 gl_NormalMatrix; // transpose of the inverse of 3x3 MV
uniform mat4 gl_ModelViewMatrixInverse;
uniform mat4 gl_ProjectionMatrixInverse;
uniform mat4 gl_ModelViewProjectionMatrixInverse;
```

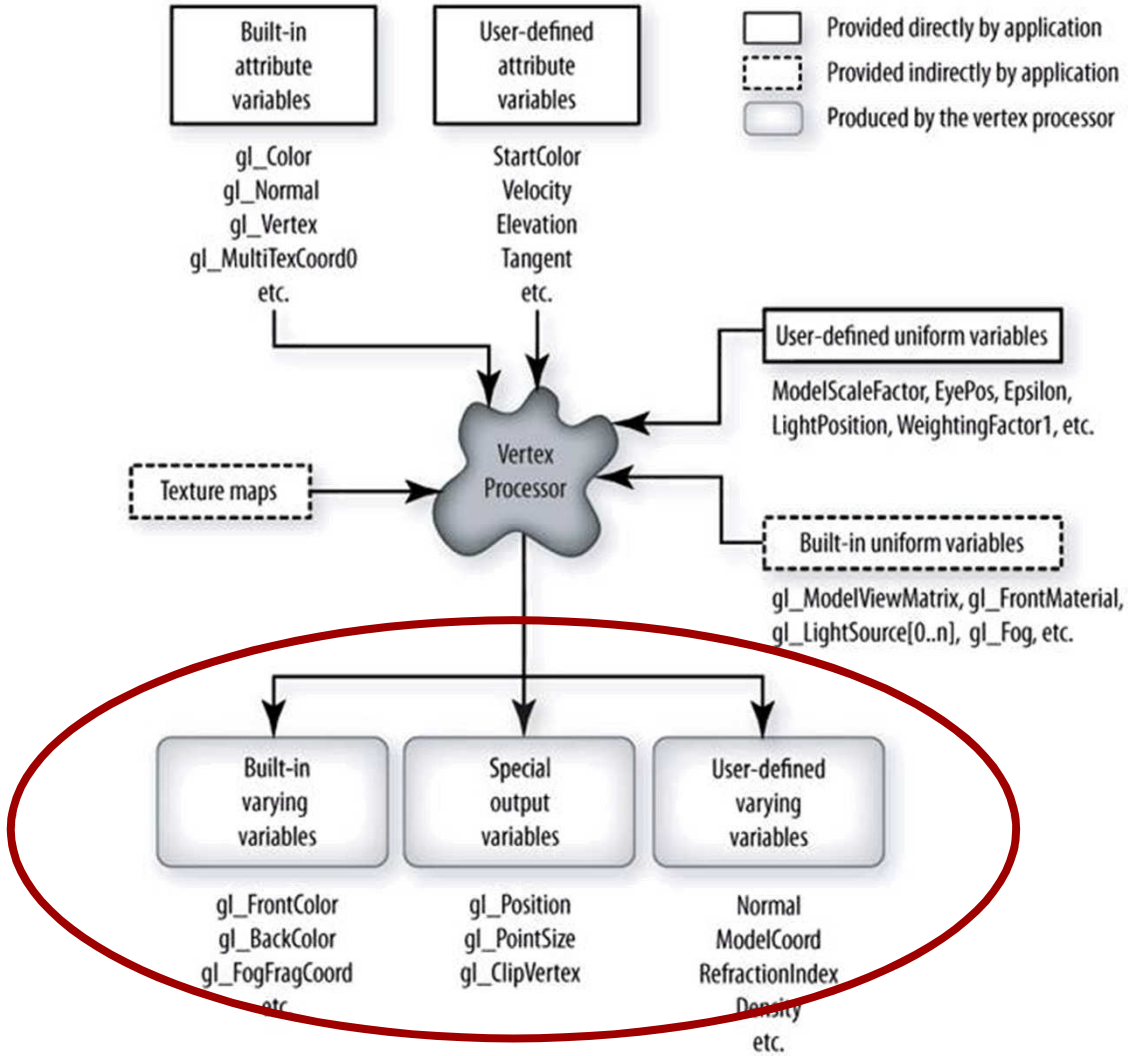
Vertex shaders

```
// Material State
struct gl_MaterialParameters
{
    vec4 emission;
    vec4 ambient;
    vec4 diffuse;
    vec4 specular;
    float shininess;
};

uniform gl_MaterialParameters gl_FrontMaterial;
uniform gl_MaterialParameters gl_BackMaterial;
```

Vertex shaders

```
// Light State
struct gl_LightSourceParameters
{
    vec4 ambient;
    vec4 diffuse;
    vec4 specular;
    vec4 position;
    vec3 spotDirection;
    ...
};

uniform gl_LightSourceParameters gl_LightSource[gl_MaxLights];
```

Vertex shaders



Vertex shaders

- **Varying** variables: són variables que es passen del vertex program al fragment program.
  - Pel vertex program són de sortida.
  - Pel fragment program són d'entrada, i es calculen per interpolació.
  - **Built-in** (gl_FrontColor...), **User-defined** (Normal...)

Vertex shaders

Llista variables **varying** predefinides:

```
varying vec4  gl_FrontColor;
varying vec4  gl_BackColor;
varying vec4  gl_TexCoord[];
…
```

- Els valors escrits a gl_FrontColor, gl_BackColor, … s'usen, un cop es determina si la primitiva es backface o no, per calcular gl_Color (entrada pel fragment shader).

Vertex shaders

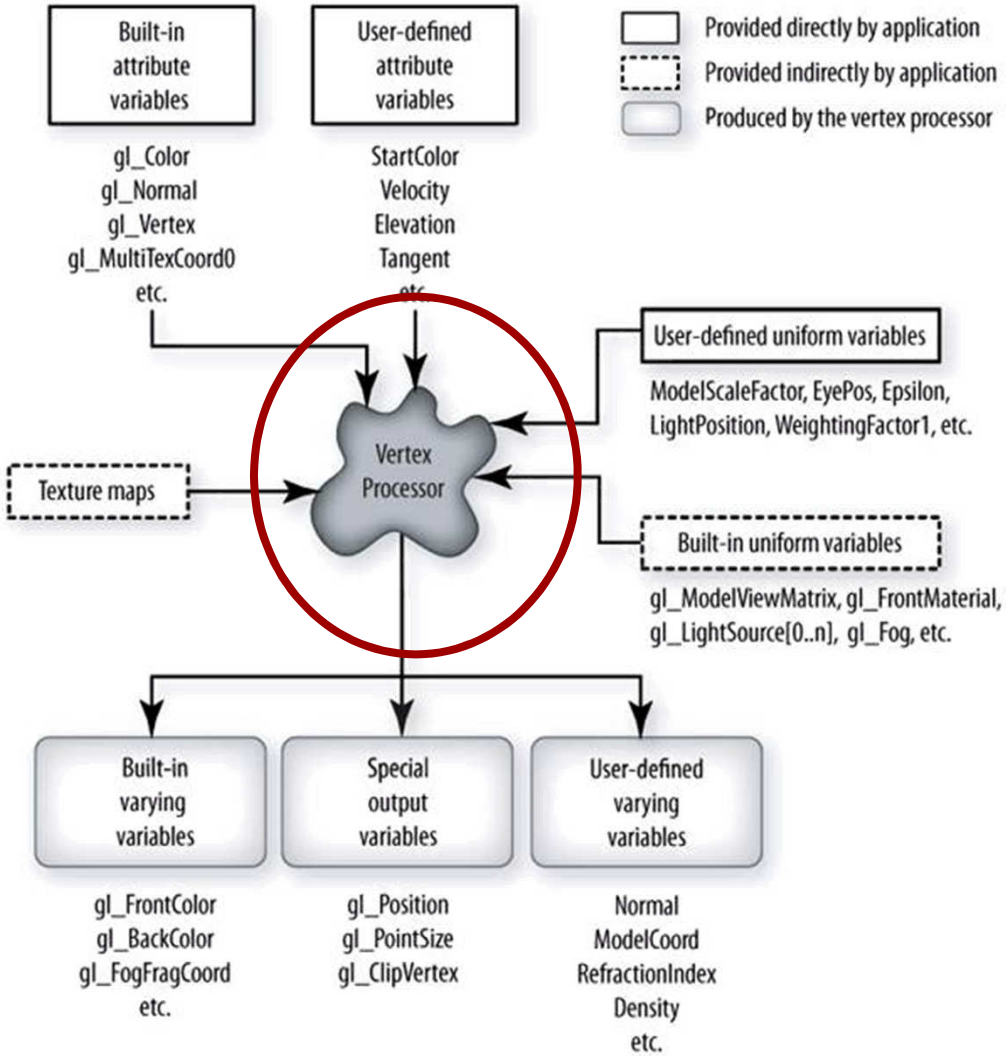
Un vertex shader sempre ha d'escriure a

vec4 **gl_Position**

les coordenades del vèrtex en clip space.

Normalment ho farà multiplicant el vèrtex per les matrius Modelview i Projection.

Vertex shaders

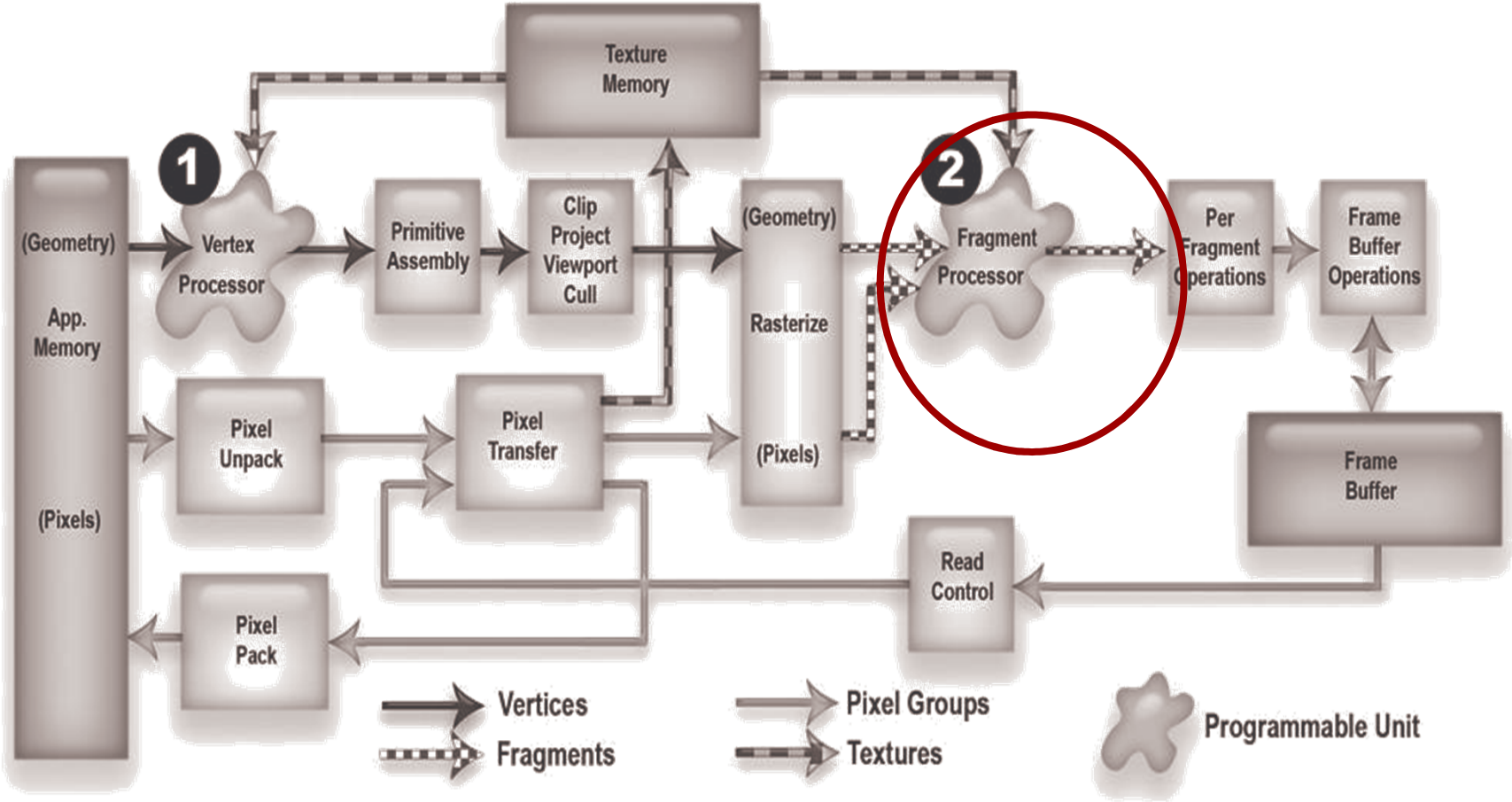


Vertex shaders

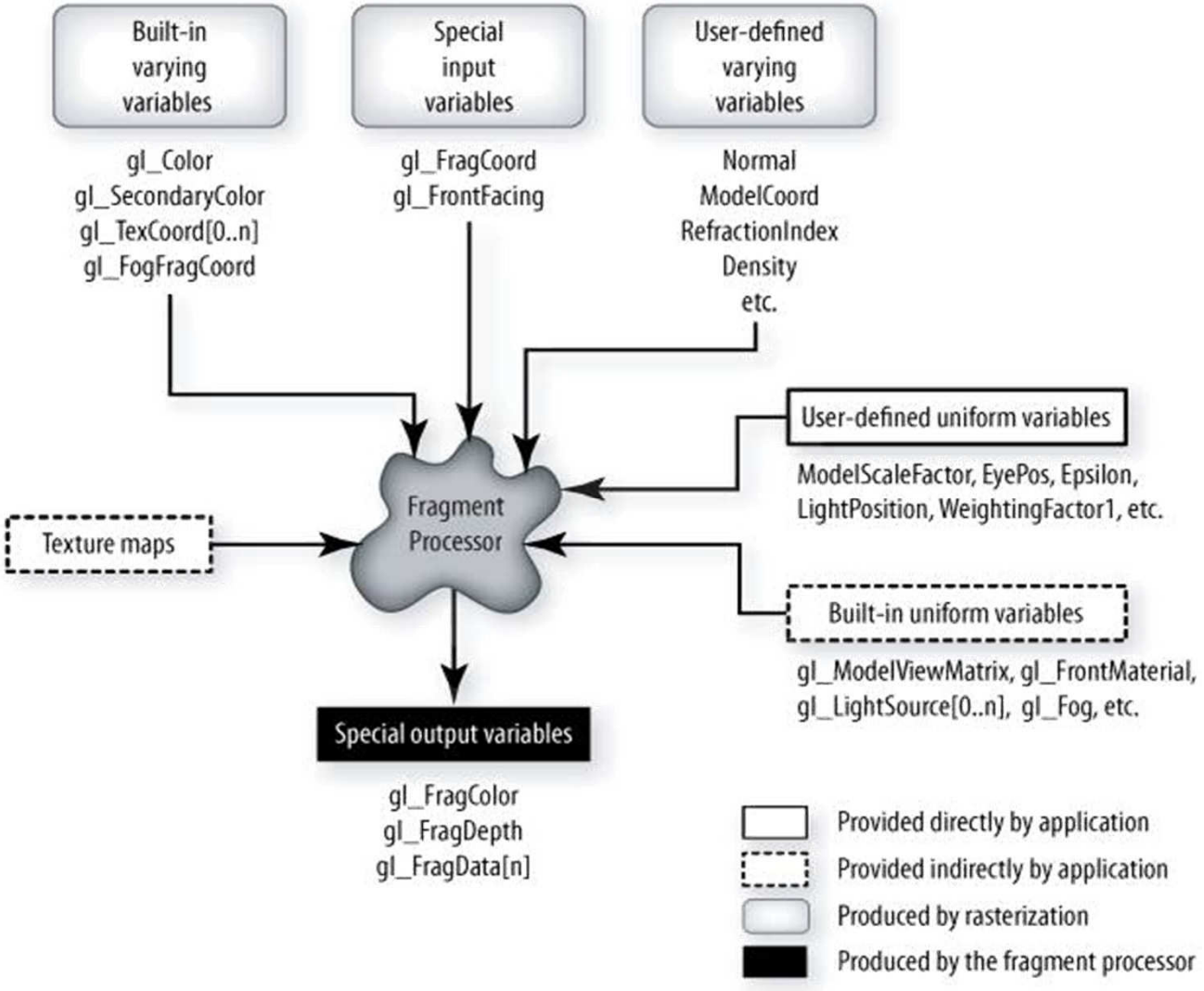
- El vertex shader s'executa per cada vèrtex que s'envia a OpenGL.
- Les tasques *habituals* d'un vertex program són:
  - Transformar el vèrtex (object space → clip space)
  - Transformar i normalitzar la normal (eye space)
  - Calcular la il·luminació del vèrtex
  - Generar coordenades de textura del vèrtex
  - Transformar les coords de textura

FRAGMENT SHADERS

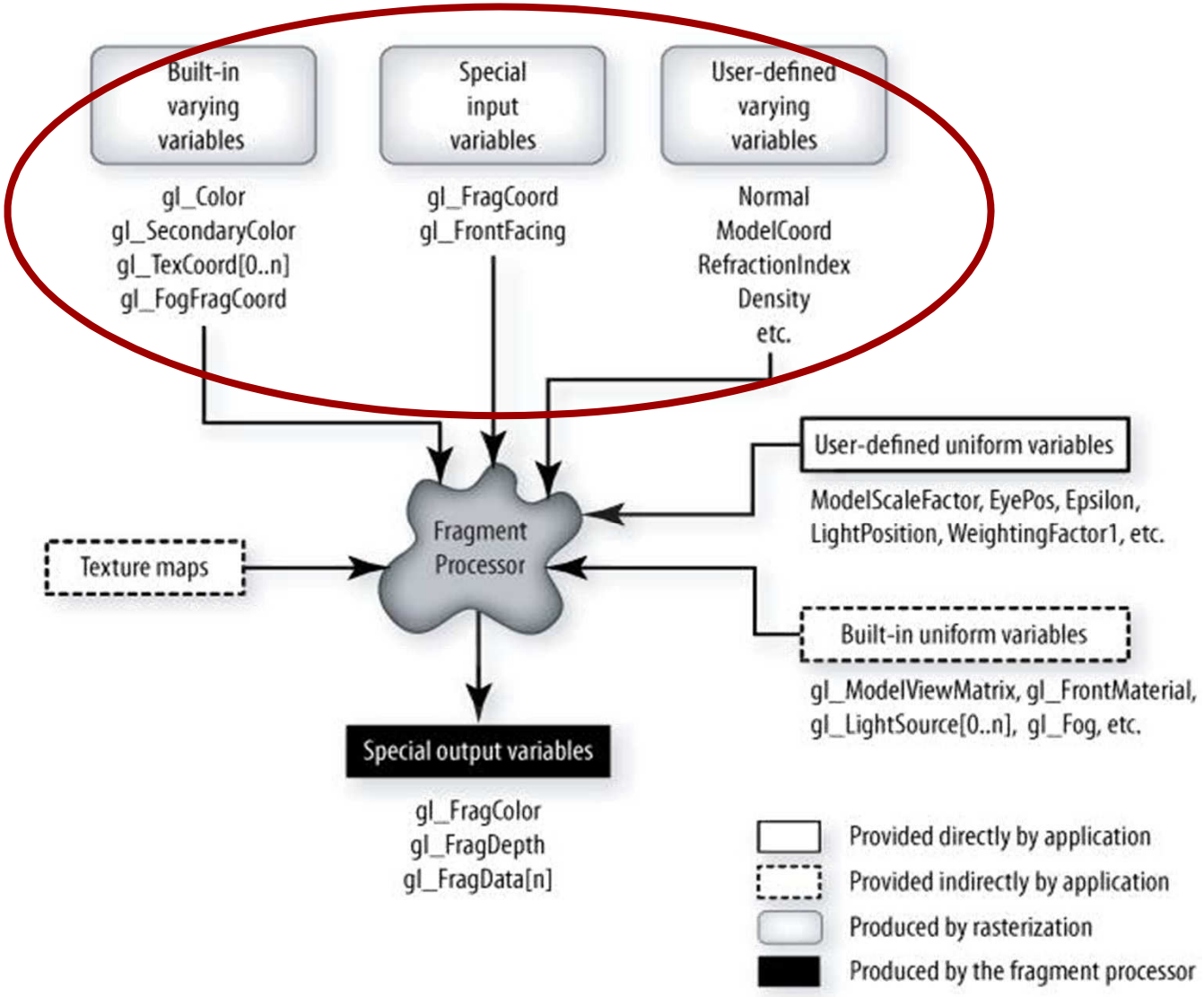
Fragment shaders



Fragment shaders



Fragment shaders



Fragment shaders

- **Varying** variables: són variables que es calculen al vertex shader i arriben interpolades al fragment shader.
- Varying predefinits:

  varying vec4 **gl_Color**;

  varying vec4 **gl_TexCoord**[];

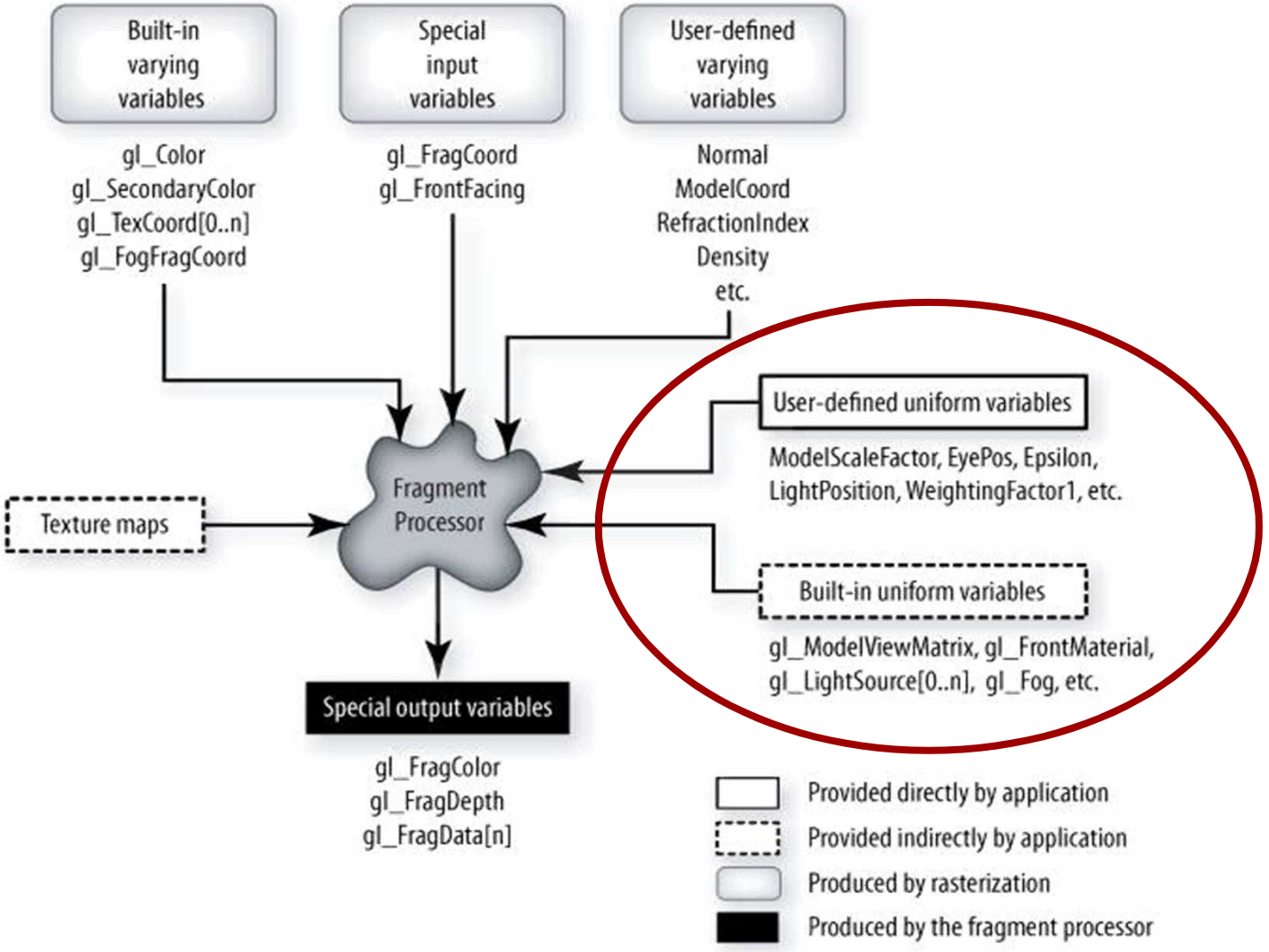
  ...

Fragment shaders

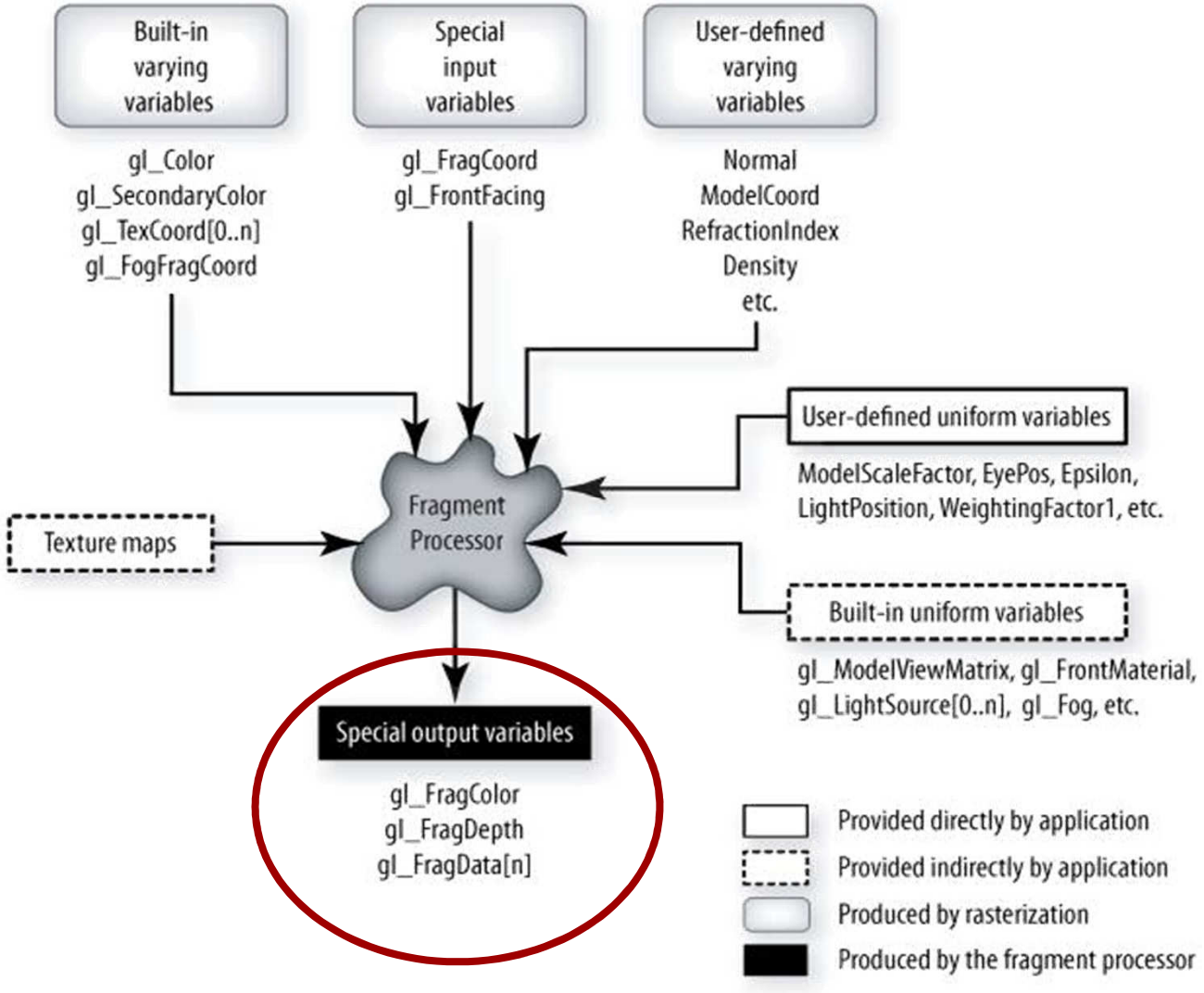
Special input variables: calculats per OpenGL de forma automàtica; es poden llegir al fragment shader:

```
vec4 gl_FragCoord;    // coordenades del fragment (window space)
bool gl_FrontFacing;  // true si el fragment és d'un polígon frontface
```

Fragment shaders



Fragment shaders



Fragment shaders

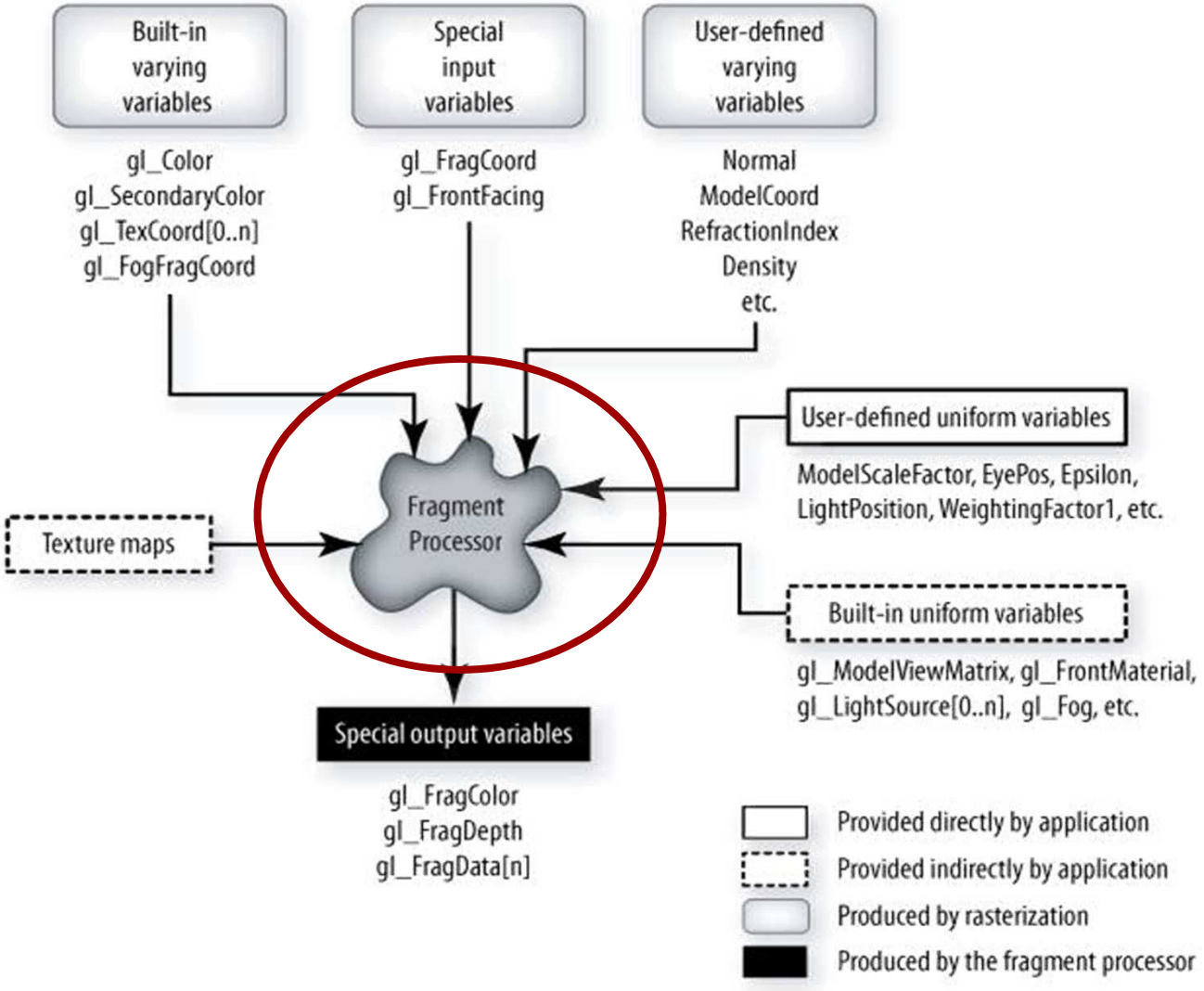
Special output variables: són els valors que ha de calcular el fragment shader:

vec4 **gl_FragColor** // color del fragment (abans de blending)

float **gl_FragDepth** // depth final del fragment (pel z-buffer)

vec4 gl_FragData[] // usat per MRT (glDrawBuffers)

Fragment shaders



Fragment shaders

- Un fragment shader s'executa per cada fragment que produeix cada primitiva.

- Les tasques habituals d'un fragment shader són:
  - Accedir a textura
  - Incorporar el color de la textura
  - Incorporar efectes a nivell de fragment (ex. boira).

- I el que no pot fer un fragment shader:
  - Canviar les coordenades del fragment (sí pot canviar **gl_FragDepth**)
  - Accedir a informació d'altres fragments (tret de dFdx, dFdy)