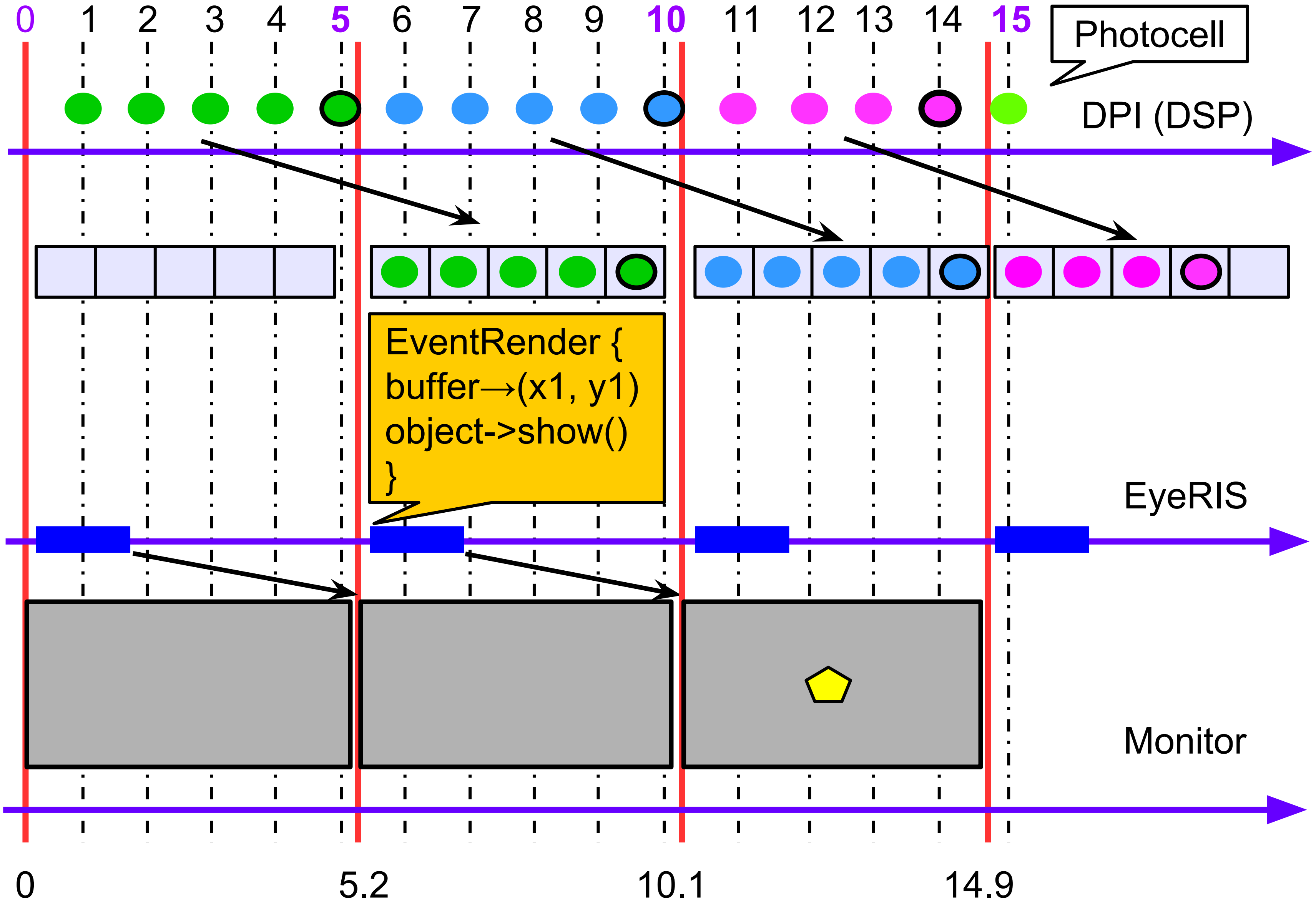
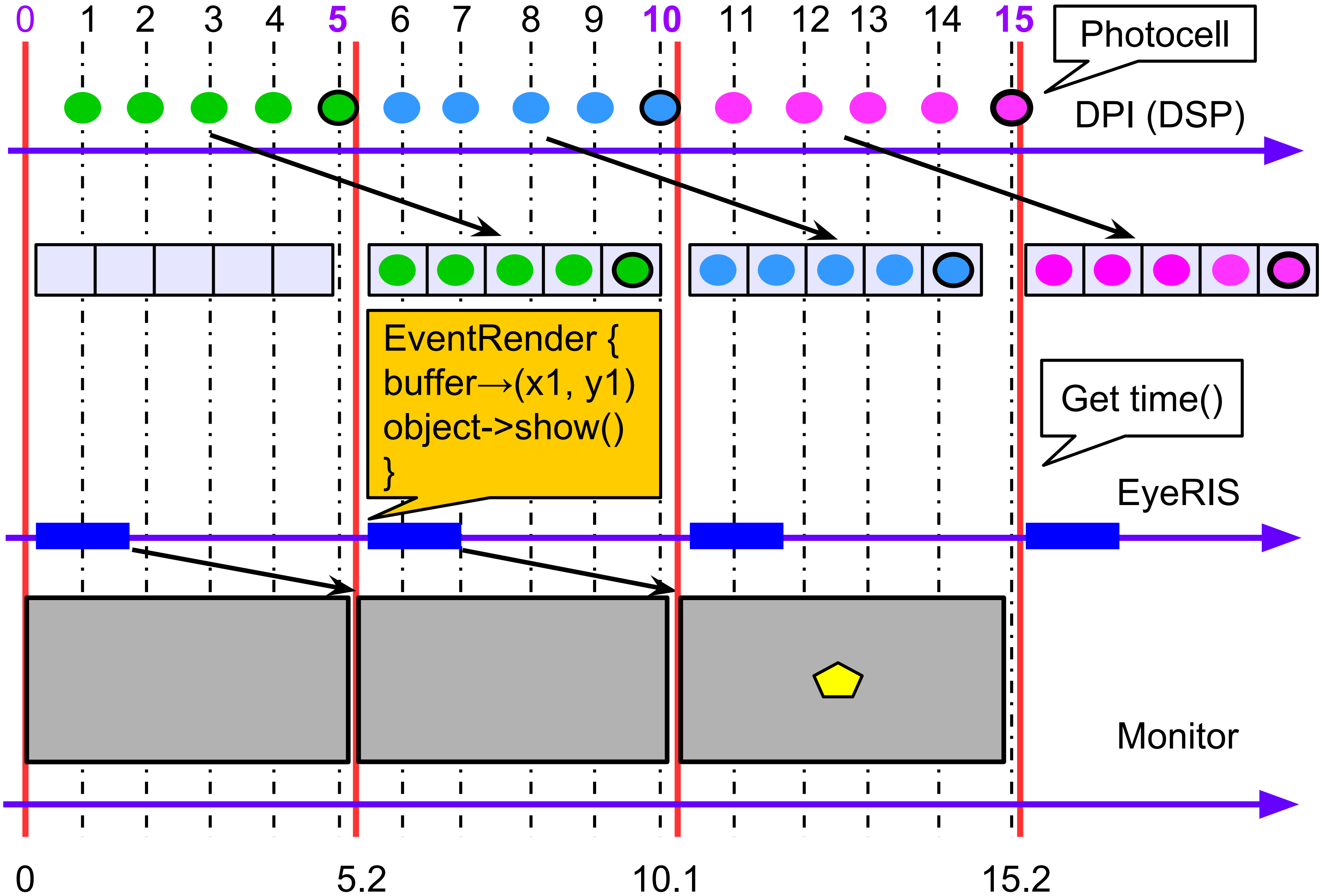
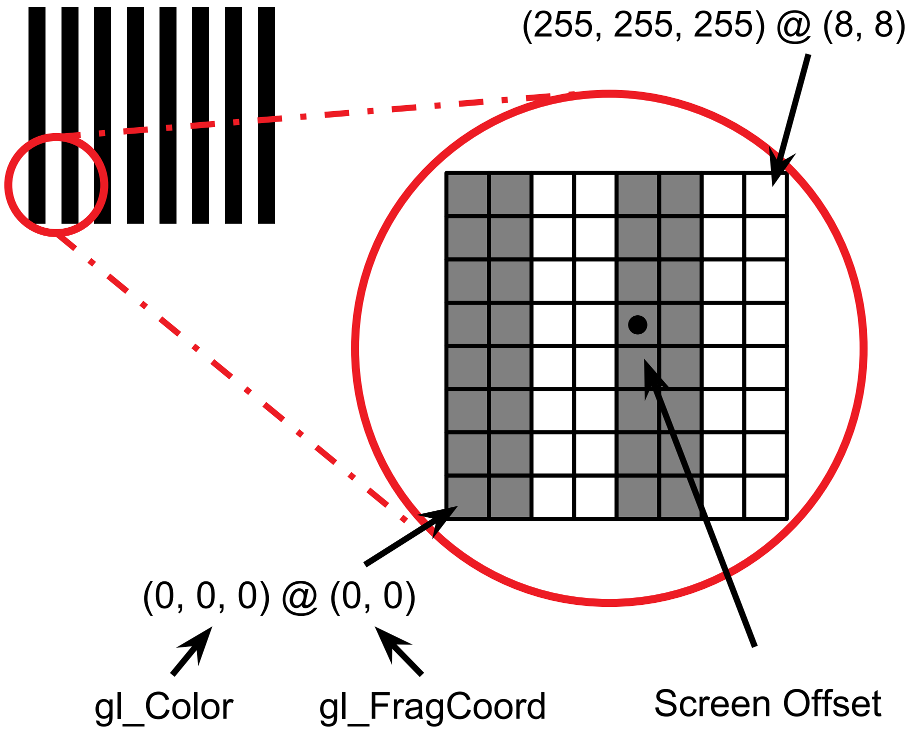


```
EventRender {
buffer→(x1, y1)
object->show()
}
```


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Photocell
DPI (DSP)
EventRender {
buffer→(x1, y1)
object->show()
}
Get time()
EyeRIS
Monitor
0
5.2
10.1
15.2



shader_grating.frag

```
uniform float screenOffsetX;
uniform float screenOffsetY;
uniform float angle;
uniform float sp;
uniform float trasparency;
void main(void) {
    float x = gl_FragCoord.x - screenOffsetX;
    float y = gl_FragCoord.y - screenOffsetY;

    // Calculate the grating
    float phase = (x * cos(angle) + y * sin(angle)) / sp * 2 * PI;
    float scale = (cos(phase) + 1) / 2;
    Vec4 Color = texture2D(texture_0, vec2(gl_TexCoord[0]));
    gl_FragColor.x = Color.x * scale;
    gl_FragColor.y = Color.y * scale;
    gl_FragColor.z = Color.z * scale;
    gl_FragColor.w = trasparency;
}
```

ExpBody.cpp

```
m_shader = new CShader(“shader_grating.frag");
m_shader->setUniformFloat("screenOffsetX", m_pxWidth / 2);
m_shader->setUniformFloat("screenOffsetY", m_pxHeight / 2);
m_shader->setUniformFloat("sp", m_wavelength);
m_shader->setUniformFloat("angle", m_angle / 180 * PI);
m_solid->setShader(m_shader);
```