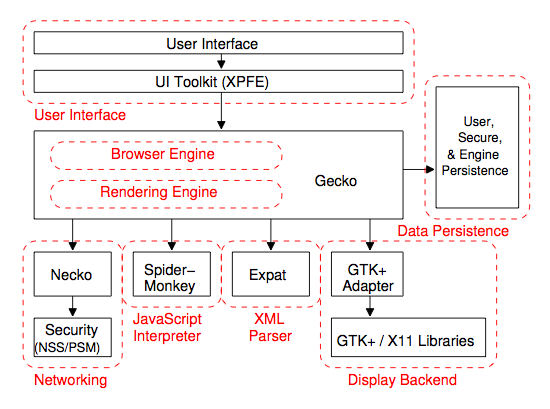
**Webrender Issue**

**Code : B2g porting on LS5030 baseline**

**What is Webrender ?**

* WebRender is a GPU-based 2D rendering engine it could be used as a backend for any sort of 2D graphics that CSS can describe
* Takes a post layout CSS display list and draws it to the screen as quickly as possible
* It's a new rendering engine (the part of Gecko that draws stuff to screen) that's designed from the ground up to make better use of GPUs and multiple CPU cores.
* It currently uses the OpenGL API internally. OpenGL is a cross-platform graphics API that specifies a standard software interface for 3D graphics processing hardware.
* WebRender works directly at a lower and GPU-centric level, dealing in terms of batches, draw calls and shaders.

**Architecture :**



* **User Interface:** provides the methods with which a user inter-acts with the Browser Engine.This includes the address bar, back/forward button, hyperlinks,bookmarking menu, etc.
* **Browser Engine:** marshals actions between the UI and the rendering engine. This provides a high-level interface to the Rendering Engine. The Browser Engine provides methods to initiate the loading of a URL and other high-level browsing actions (reload, back, forward). It also provides the User interface with various messages relating to error messages and loading progress
* **Rendering Engine:** produces the visual representation of a given URL. The Rendering Engine interprets the HTML, XML, and JavaScript that comprises a given URL and generates the layout that is displayed in the User Interface. A key component of the Rendering Engine is the HTML parser
* Different browsers use different rendering engines: Internet Explorer uses Trident, Firefox uses Gecko, Safari uses WebKit.Chrome and Opera uses WebKit (before is Blink). 6
* **UI Backend:** – Used for drawing basic widgets like combo boxes and windows –Underneath it uses operating system user interface methods.

**Rendering Engine – Basic flow**

**Step 1**: Parsing the HTML document and convert elements to DOM nodes in a tree called the “content tree” – HTML Parser

**Step 2:** Parse the style data, both in external CSS files and in style element together with visual instructions in HTML will be used to create another tree, call “render tree”

**Step 3:** After the construction of the render tree it goes through a “layout" process. This means giving each node the exact coordinates where it should appear on the screen

**Step 4:** The next stage is painting–the render tree will be traversed and each node will be painted using the UI backend layer – Painting

**Path for code :**

* **LS5030/Uranus/Linux/android/gecko/b2g/chrome/content/default-apps/homescreen$ ls**

**index.html style**

* **In style directory it contains index.css file**

>> In log it will print like (Example)

Gecko : WebViewChild:

Gecko : Window created: chrome://b2g/content/settings/index.html

Gecko : Got metaChanged: (viewport) width=device-width, initial-scale=1

Gecko : Got afterpaint event: chrome://b2g/content/settings/index.html

**Code path : gecko/toolkit/modules/WebViewChild.jsm**

**Issue faced in LS5030 code :**

webrender::device::gl: Failed to compile shader: brush\_image

webrender::device::gl: Vertex shader compilation failed.

webrender::device::gl: ERROR: 0:14: 'samplerExternalOES' : requires extension GL\_OES\_EGL\_image\_external to be enabled

**Code path : gecko/gfx/we/webrender/gl.rs**

**FIX :**