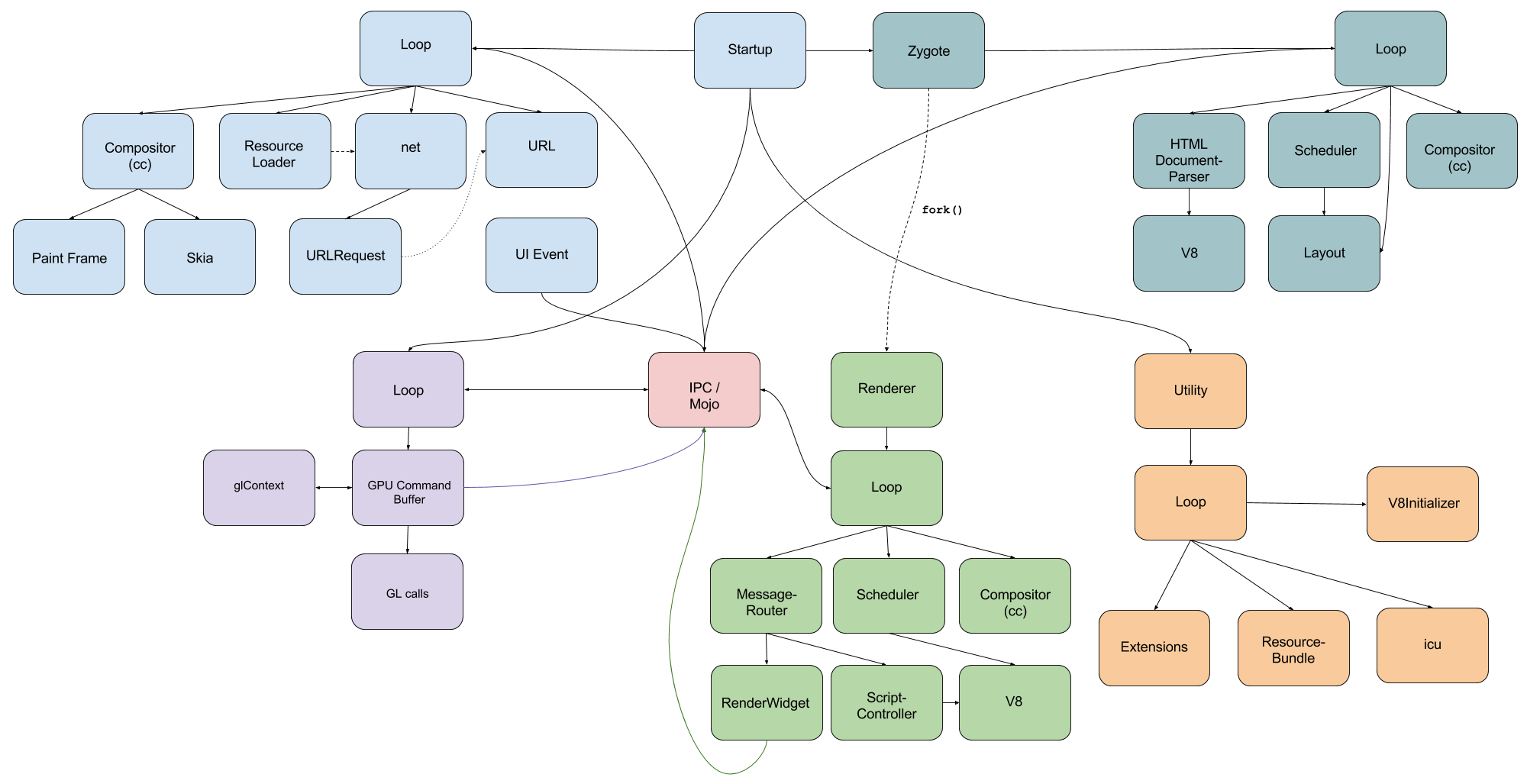
# *Chromium Architecture Overview*

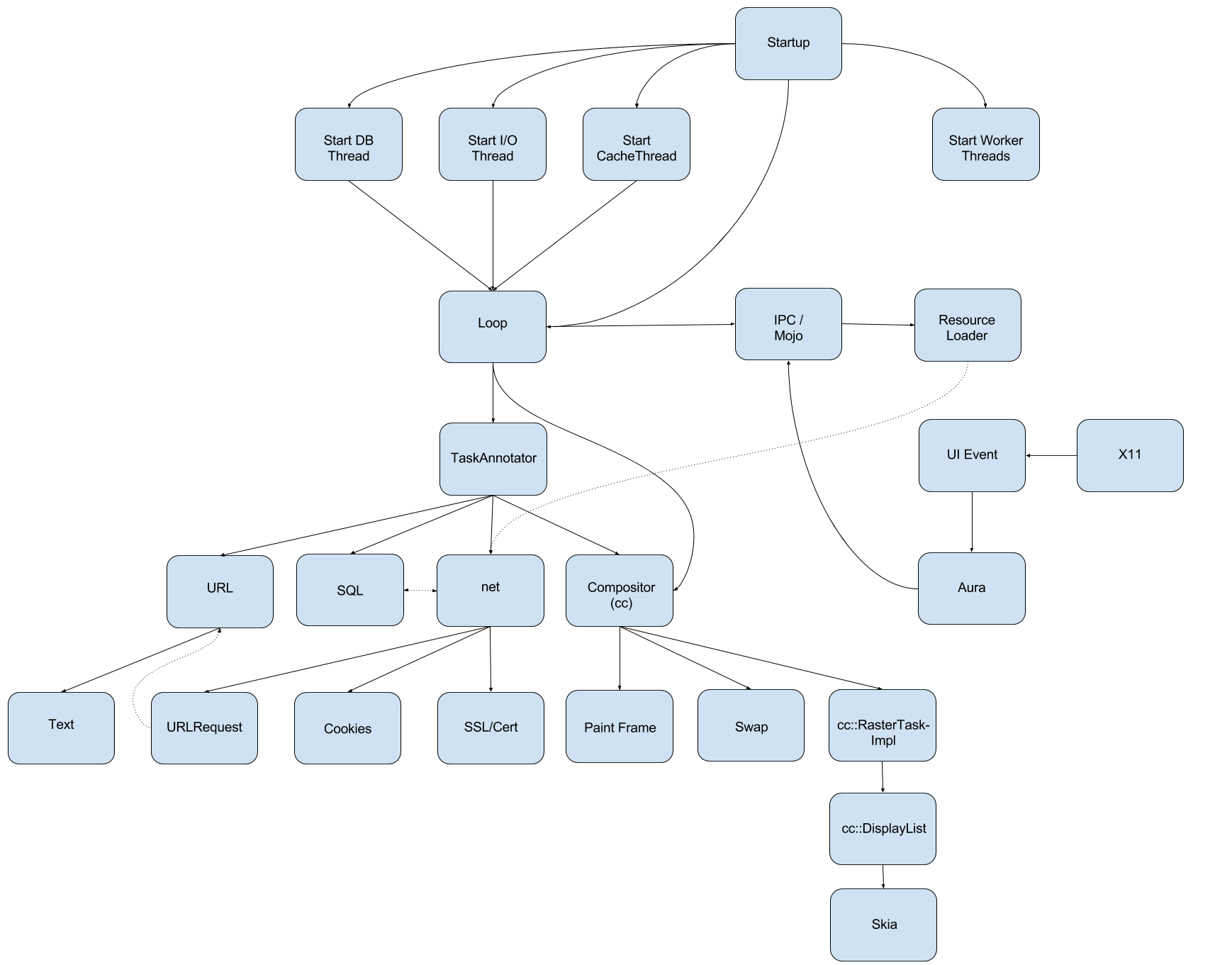
[](https://docs.google.com/drawings/d/1PQlEOLPBIlYdamCaYdM0vaGSVG_uEgtHwtJVMYnBbls/edit?usp=sharing)

Chromium / Chrome is a multi-process browser, unless it is launched with the command line option --single-process. All processes have different responsibilities, but there are some things they all have in common:

* All processes have an inner loop, which deals with task annotation, and communication with other processes.
* The processes communicate with each other through messages. IPC and Mojo frameworks are used to achieve this.

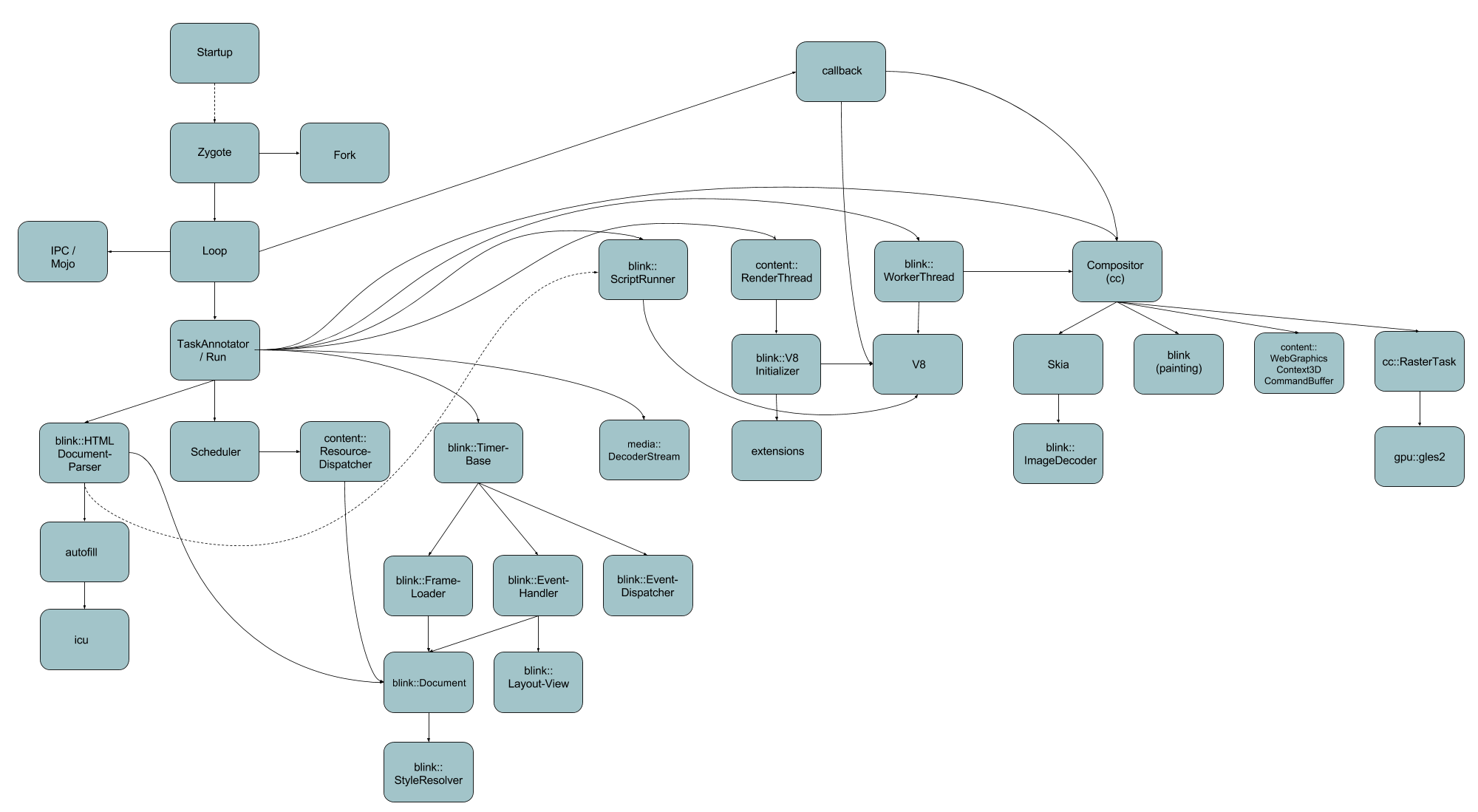
There are 2-5 important process types (depending on the platform). The Browser process is the main process on all platforms. This process is the entry point of the application. The other important process - which is available on all platforms when running in multi-processed mode - is the Renderer process. There is a process called Utility process as well, which is running for a short amount of time but performs important initialization tasks. On Linux-based systems there is also a helper process called Zygote. Last but not least, if hardware acceleration is available on the system, there is an additional process called GPU process.

As mentioned before the Browser process is the startup process. It is also called broker process because it has certain privileges, for example starting other processes or supervising the activities of other processes (for example: sandboxed ones).

[](https://docs.google.com/drawings/d/19bWMqhapTT2qiIYDLZA1IY7kPQik5db0Wz7JBvC8hQw/edit?usp=sharing)

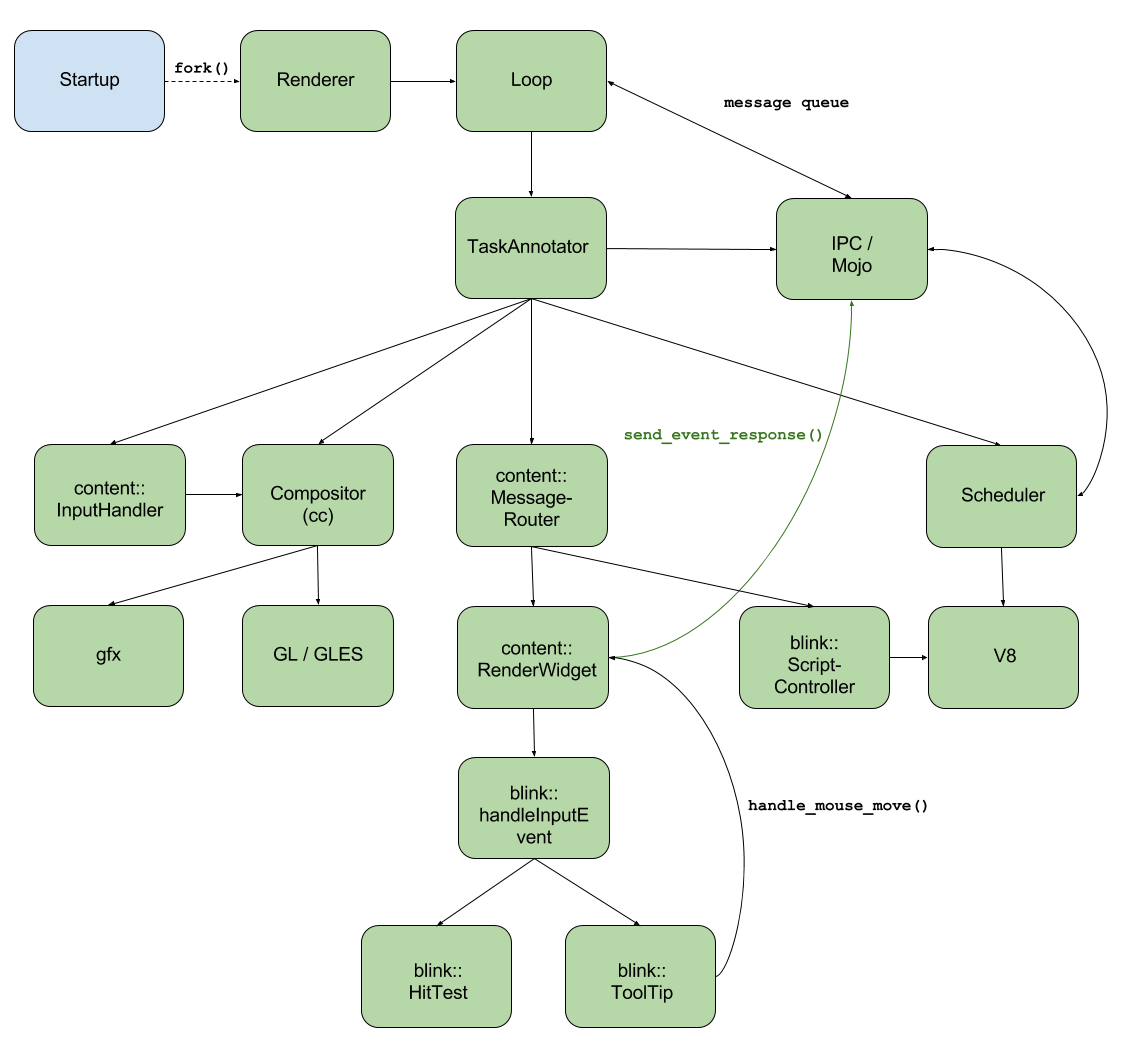
([Detailed description](https://docs.google.com/document/d/16ZV5dwb9SKKpUcUYap7jtWqqueCWfJfCpD4j_sri2CI/edit?usp=sharing))

Browser process spawns separate threads to maintain its main functionalities easier. There is an I/O Thread which is responsible for the communication with the Renderer thread. UI Thread is the main thread of the Browser process which guarantees the responsiveness and event handling of the browser’s UI. DB, File and Cache threads are responsible for handling database connections and queries, file operations and cache management. There is an extra thread: Process Launcher thread, which is kept to spawn or fork new processes, for example the Zygote or GPU process. Besides the previously mentioned threads, Browser process spawns so called worker threads as well. These worker threads deal with necessary tasks (such as DNS lookup) and are part of a WorkerPool thread pool (which is a Linux feature). Besides dealing with processes and threads, Browser process has many other important liabilities. Its most important role is the sandbox management. Browser process hosts the sandbox policy engine service, interception manager and IPC service. It is capable to perform the policy-allowed actions on behalf of the target process. It is also responsible for dealing with network resources, and URL requests.

[](https://docs.google.com/drawings/d/153E_TQxq4NVSqzQyqqUtqkkCHrkEnbVDhqf_Ur6Tr-c/edit?usp=sharing)

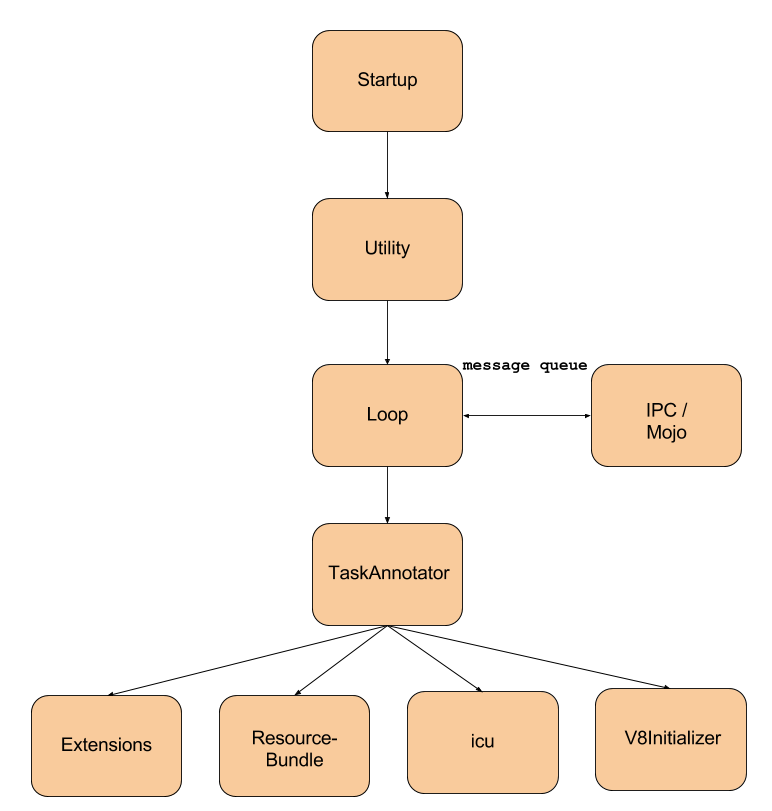
([Detailed description](https://docs.google.com/document/d/1LOS4mJdrS89bCa89kvK9qfC1pxhW2x1F_4jU5M-FSN4/edit?usp=sharing))

As mentioned on Linux there is a special process called Zygote process. It is created at the startup and later the requisite Renderer processes are forked from it. It has an inner loop as any other Chromium process. The performed operations are called from the loop’s TaskAnnotator or TaskRunner object. Zygote process has many responsibilities including document parsing, scheduling tasks, decoding resources, running scripts, starting compositor tasks, etc. On other platforms - where Zygote is not used - these tasks are performed immediately by the Renderer process.

[](https://docs.google.com/drawings/d/1lMECV0oLf741IrXHXf1ENCWzmqhFSe9BdRS0cs8f85k/edit?usp=sharing)

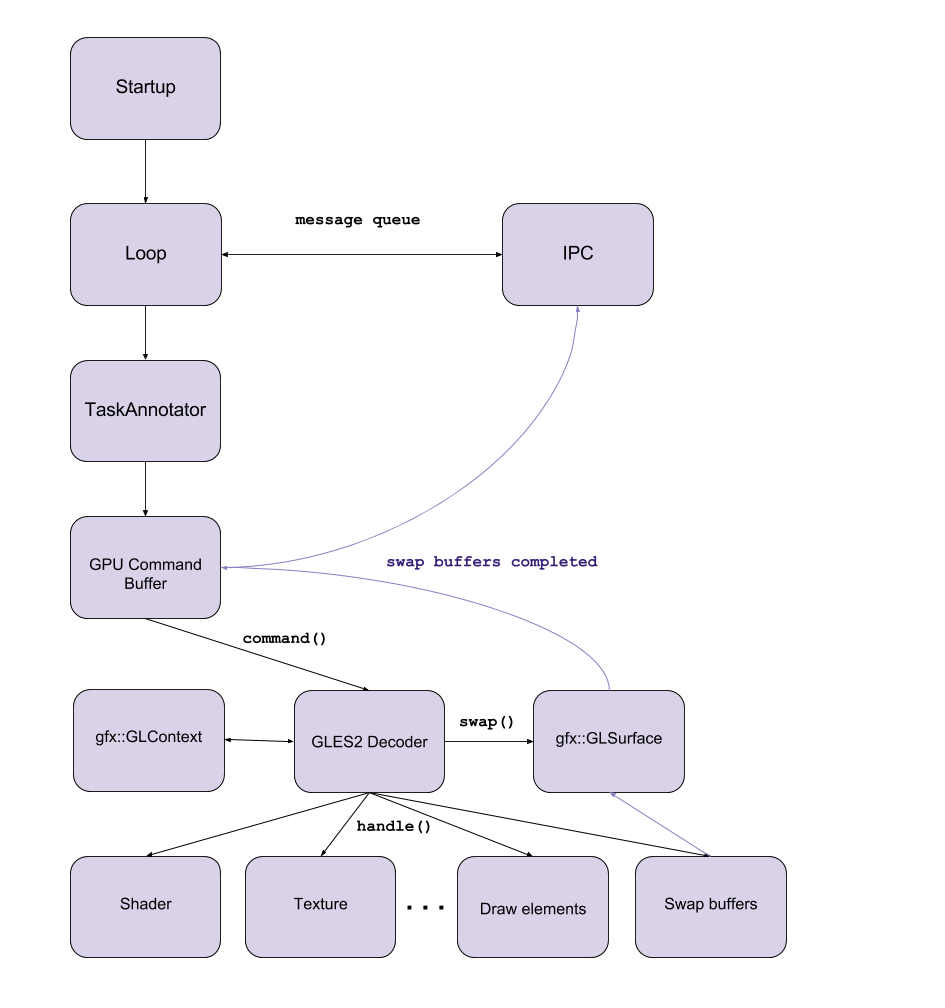
([Detailed description](https://docs.google.com/document/d/1Fej_W7oCWbxup-jYZPe_Cc58hYMJHpRgEVLuASBzDPs))

On Linux based systems the Renderer process is forked from the Zygote process - on other platforms it is spawned separately - it is also referred as target process. More Renderer processes can exist at the same time (in Chrome/Chromium usually one per browser tab). The other important property of the Renderer process is that it is sandboxed, so any potentially dangerous website can make no harm to the system. Renderer process is also multithreaded. It spawns the following threads: I/O thread, Compositor thread and Raster thread. Besides the sandbox related responsibilities (sandbox IPC client, sandbox policy engine client, sandbox interceptions) Renderer process invokes the ScriptController and the V8 JavaScript engine.

[](https://docs.google.com/drawings/d/14uUF0I9ckGi_apywWS7DNEC-9eV9AficoSeovxyNX18/edit?usp=sharing)

([Detailed description](https://docs.google.com/document/d/1gUM_TG4Xgx8Ui7FVuJADJmtRkkhWokSmfnD0MrQGaxs))

On Linux based systems the Utility process is spawned by the Browser process the same way as GPU process. Utility process is created right after the Browser startup. It performs necessary initializations and terminates after completing them. It initializes the ResourceBundle, the icu and the V8 Javascript Engine. It also deals with extension extraction. Utility process can be spawned many times during the browser’s lifetime. For example adding a new extension leads to a newly created Utility process which handles the integration of the new extension into the browser.

[](https://docs.google.com/drawings/d/11zKL9vphVV5aFMaAHZO3NrvxI7BxVY08cwXSC6OmYq4/edit?usp=sharing)

([Detailed description](https://docs.google.com/document/d/1hkk9KmTPv0aNdc-p1SNsxf2B9EQ7X3h3JItu6RA2Jj4))

The fifth process which should be mentioned is the GPU process. It only exists if hardware acceleration is available on the system. This process is responsible for communicating with the GPU, its most important task is to forward the GL / GLES commands to the GPU. Other responsibilities of this process are: dealing with shaders, textures, buffers, etc.

Sources:

<https://www.chromium.org/developers/design-documents/threading>

<https://www.chromium.org/developers/design-documents/sandbox>

<http://www.slideshare.net/HyungwookLee/android-chromium-rendering-pipeline>

<http://www.slideshare.net/HyungwookLee/mobilebrowserinternal-20140122>

<https://code.google.com/p/chromium/codesearch#chromium/src/base/threading/worker_pool_posix.h&q=worker_pool_posix&sq=package:chromium&type=cs&l=1>