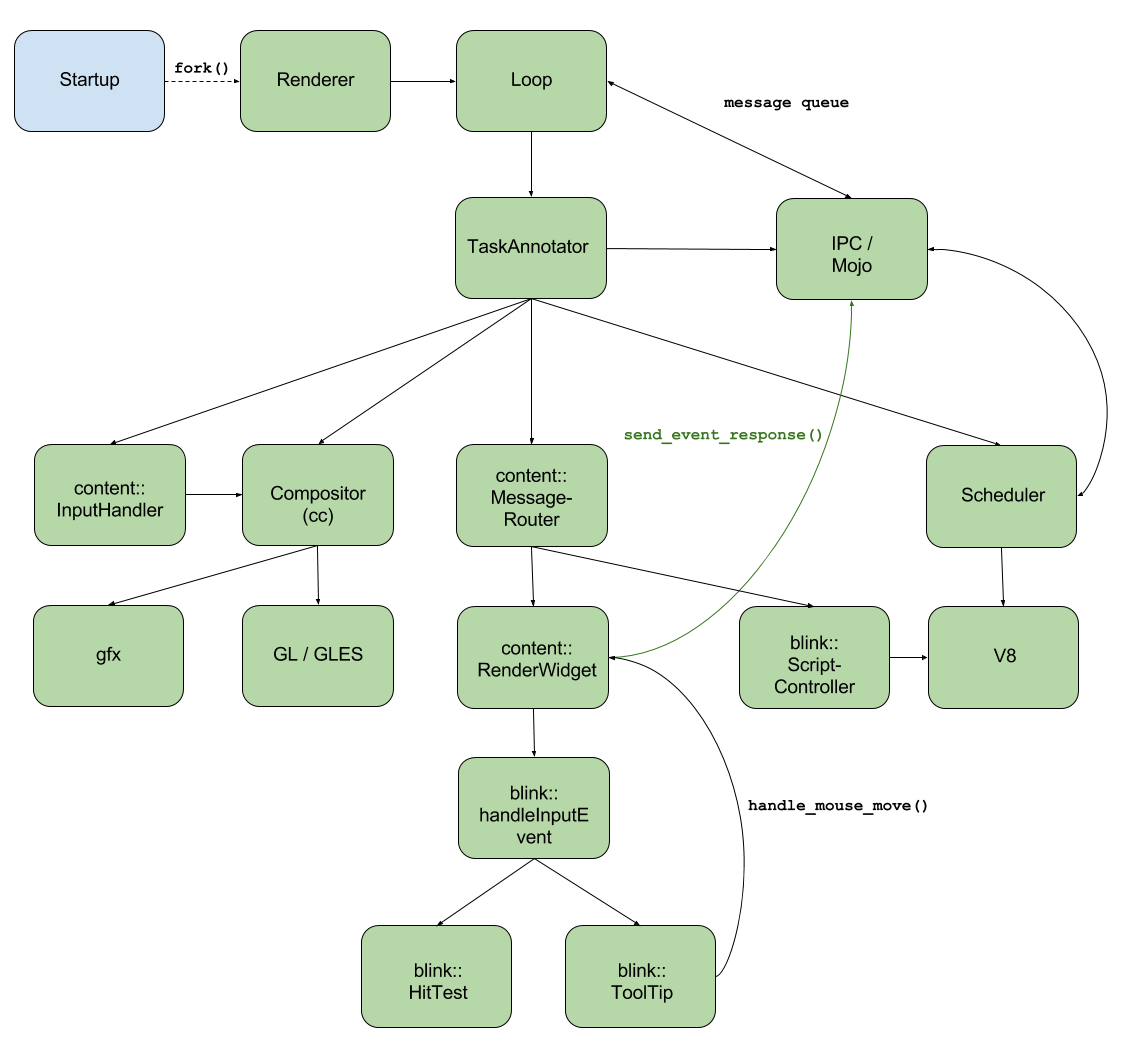
# ~*Renderer process - detailed architecture overview*

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

* **Startup**
  + startup box represents the parent process’s startup procedure
  + after Browser process startup, Zygote process is created
* **Renderer**
  + renderer box represents that the Renderer process is forked from the Zygote process
* **Loop**
  + MainMessageLoop
    - is used to process events for a particular thread
    - puts the incoming messages, tasks to a queue
    - pops a task from the queue and starts it
    - strong relationship with the IPC communication framework
    - has task reentrancy protection
      * second task cannot be started until first task finishes
* **IPC / Mojo**
  + framework which is used for inter-process communication
  + connects directly to the MainMessageLoop
  + provides communication channels through which the messages can be sent
  + message creating, sending and receiving
  + asynchronous message handling
* **TaskAnnotator**
  + all incoming tasks are going through a TaskAnnotator which annotates the task before the execution
  + implements common debug annotations for posted tasks. This includes data such as task origins, queueing durations and memory usage
  + runs a previously queued task
* **Scheduler**
  + package which contains multiple classes regarding task schedule
  + TaskQueueManager
    - The task queue manager provides N task queues and a selector interface for choosing which task queue to service next. Each task queue consists of two sub queues
      * Incoming task queue
      * Work queue
* **content::MessageRouter**
  + the MessageRouter handles all incoming messages sent to it by routing them to the correct listener
  + routing is based on the Message's routing ID
  + since routing IDs are typically assigned asynchronously by the browser process, the MessageRouter has the notion of pending IDs for listeners that have not yet been assigned a routing ID
* **content::RenderWidget**
  + RenderWidget provides a communication bridge between a WebWidget and a RenderWidgetHost, the latter of which lives in a different process.
  + handles incoming Message in OnMessageReceived method
  + handles input Events across a chain to **blink::handleInputEvent**
  + in case of a mouse event, the **tooltip** gets set and also **hit tests** should be executed
  + sends response back through IPC
* **content::InputHandler**
  + **content::InputHandlerManager**
    - manages InputHandlerProxy instances for the WebViews in this renderer.
  + **content::InputHandlerProxy**
    - this class is a proxy between the content input event filtering and the compositor's input handling logic. InputHandlerProxy instances live entirely on the compositor thread. Each InputHandler instance handles input events intended for a specific WebWidget.
  + calls specific **Compositor** functions as a result for the input event
* **Compositor (cc)**
  + compositor calls specific **gfx** and **GL/GLES** functions in order to perform the correct drawing
* **blink::ScriptController**
  + evaluates JavaScript, and gets the return value through a V8ScriptRunner object
* **V8**
  + Blink’s JavaScript engine
  + JavaScript
    - parsing
    - compiling
    - executing
  + callbacks to modify DOM tree, etc.
  + deals with
    - heap allocation
    - garbage collection

Sources:

<https://code.google.com/p/chromium/codesearch#chromium/src/components/scheduler/base/task_queue_manager.h&q=task_queue_manager.h&sq=package:chromium&type=cs&l=5>

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

<https://code.google.com/p/chromium/codesearch#chromium/src/base/message_loop/message_loop.h>

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

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

<https://code.google.com/p/chromium/codesearch#chromium/src/content/renderer/input/input_handler_manager.h&q=input_handler_manager.&sq=package:chromium&type=cs&l=1>

<https://code.google.com/p/chromium/codesearch#chromium/src/content/renderer/input/input_handler_proxy.h&q=input_handler_pr&sq=package:chromium&type=cs&l=1>