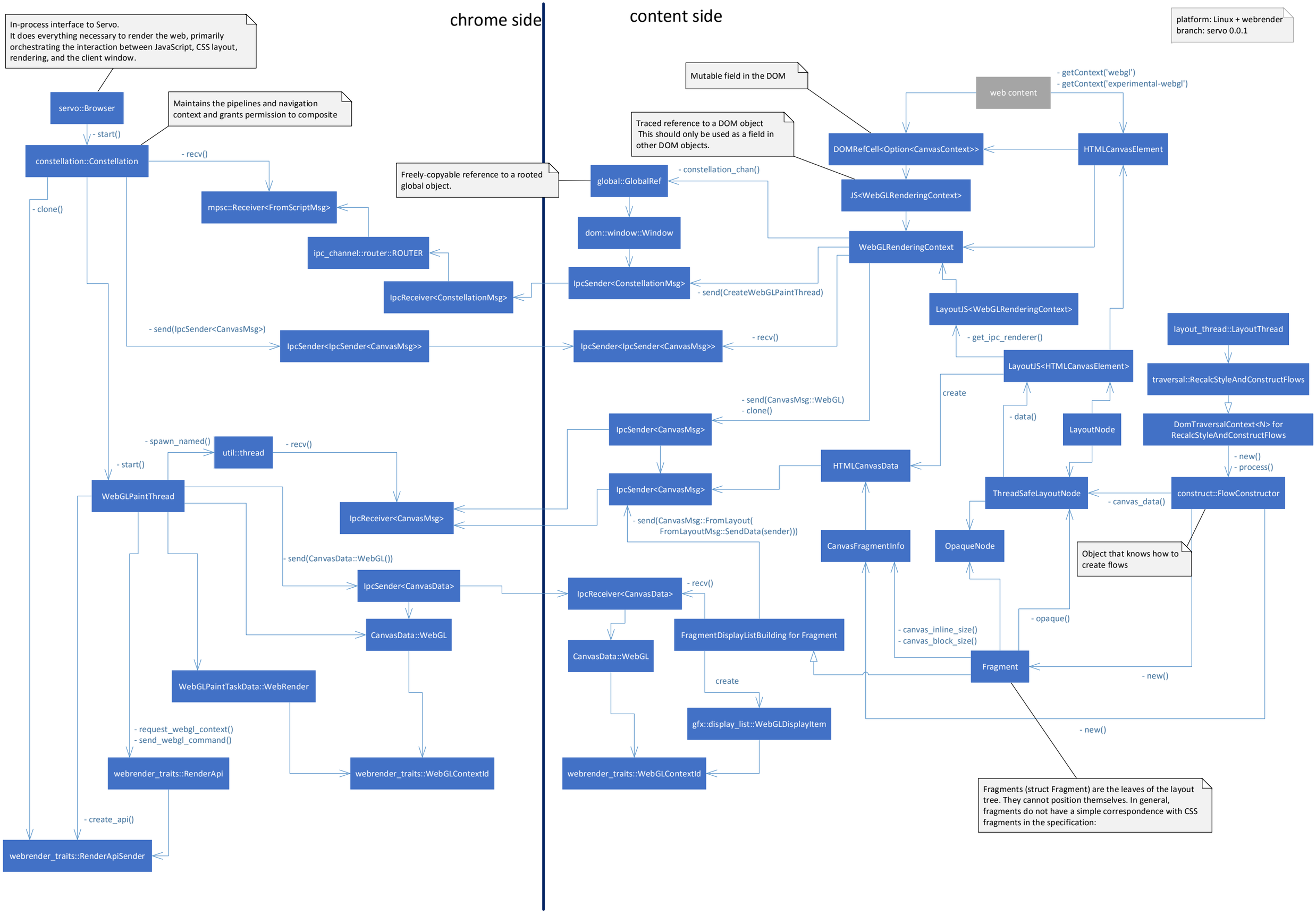
chrome side
content side
platform: Linux + webrender
branch: servo 0.0.1
In-process interface to Servo.
It does everything necessary to render the web, primarily orchestrating the interaction between JavaScript, CSS layout, rendering, and the client window.
servo::Browser
- start()
Maintains the pipelines and navigation context and grants permission to composite
constellation::Constellation
- recv()
- clone()
mpsc::Receiver<FromScriptMsg>
ipc_channel::router::ROUTER
IpcReceiver<ConstellationMsg>
- send(IpcSender<CanvasMsg>)
IpcSender<IpcSender<CanvasMsg>>
- spawn_named()
util::thread
- recv()
- start()
WebGLPaintThread
IpcReceiver<CanvasMsg>
- send(CanvasData::WebGL())
IpcSender<CanvasData>
CanvasData::WebGL
WebGLPaintTaskData::WebRender
- request_webgl_context()
- send_webgl_command()
webrender_traits::RenderApi
webrender_traits::WebGLContextId
- create_api()
webrender_traits::RenderApiSender
Freely-copyable reference to a rooted global object.
global::GlobalRef
- constellation_chan()
dom::window::Window
IpcSender<ConstellationMsg>
- send(CreateWebGLPaintThread)
IpcSender<IpcSender<CanvasMsg>>
- recv()
- send(CanvasMsg::WebGL)
- clone()
IpcSender<CanvasMsg>
IpcSender<CanvasMsg>
- send(CanvasMsg::FromLayout(
FromLayoutMsg::SendData(sender)))
IpcReceiver<CanvasData>
- recv()
CanvasData::WebGL
FragmentDisplayListBuilding for Fragment
create
gfx::display_list::WebGLDisplayItem
webrender_traits::WebGLContextId
Mutable field in the DOM
Traced reference to a DOM object
This should only be used as a field in other DOM objects.
web content
- getContext('webgl')
- getContext('experimental-webgl')
DOMRefCell<Option<CanvasContext>>
HTMLCanvasElement
JS<WebGLRenderingContext>
WebGLRenderingContext
LayoutJS<WebGLRenderingContext>
- get_ipc_renderer()
LayoutJS<HTMLCanvasElement>
create
- data()
LayoutNode
HTMLCanvasData
ThreadSafeLayoutNode
OpaqueNode
CanvasFragmentInfo
- canvas_inline_size()
- canvas_block_size()
Fragment
- opaque()
- new()
- new()
layout_thread::LayoutThread
traversal::RecalcStyleAndConstructFlows
DomTraversalContext<N> for RecalcStyleAndConstructFlows
- new()
- process()
construct::FlowConstructor
- canvas_data()
Object that knows how to create flows
Fragments (struct Fragment) are the leaves of the layout tree. They cannot position themselves. In general, fragments do not have a simple correspondence with CSS fragments in the specification: