Gregory Terzian  
Servo TSC member (maintainer)  
Modular Servo:  
What does it mean, where do we stand, and where are we going?  
May 6th, 2024
Servo, the embeddable, independent, memory-safe, **modular**, parallel web rendering engine

**modularity**(*Wikipedia*): the degree to which a **system**'s components may be separated and recombined, often with the benefit of:

1. flexibility and variety in use.
2. reduction in complexity (hide complexity of component behind interface)

**system**(*Wikipedia*): a group of interacting or interrelated elements that act according to a set of rules to form a unified whole
Servo, the embeddable, independent, memory-safe, **modular**, parallel web rendering engine

1. Modularity of internal components
2. As a modular component embedded in another system
3. Components of Servo used in other systems (and Servo using other system’s components)
Modularity of components:

1. Do components hide complexity from each other?
2. Can components be separated and recombined?

Example: Image cache
Servo as a modular system: a group of interacting elements, components (layout, script, networking, etc...), forming a web engine.
Modularity: 1: internal complexity hidden from other components? Yes: only interfaces is shared (bonus: faster compilation).
**Modularity**: 2: Can components be separated and recombined? No: listener concept(hidden in script component) introduces IPC dependency.

Part of the interface:

```
IpcSender<PendingImageResponse>
```

But almost: just need to abstract away means of communication.
As a modular embedded component:

1. Do components hide complexity from each other? Yes: embedding API
2. Can components be separated and recombined? Yes: see various embedding examples

Examples: ServoShell, Tauri WebView, KDAB Qt WebView
Tauri WebView: project funded by NLnet.

“The web ecosystem lacks a cross-platform, non-corporate controlled system for running web content. Tauri is a system for distributing cross-platform applications that relies on engines present on a system - effectively those owned by Apple, Google, and Microsoft. These permit varying levels of user control. The Servo project is a cross-platform, open source web engine.”

Source: https://nlnet.nl/project/Tauri-Servo/
At KDAB we managed to embed the Servo web engine inside Qt, by using our CXX-Qt library as a bridge between Rust and C++. This means that we can now use Servo as an alternative to Chromium for webviews in Qt applications.

“With the browser inherently being exposed to the internet, it is usually the biggest attack vector on a system. Naturally this makes Servo very attractive as an alternative browser engine, given that it is written in a memory-safe language.”

Source: https://www.kdab.com/embedding-servo-in-qt/
Use of independent components:

1. Do components hide complexity from each other?
2. Can components be separated and recombined?

On both points: still a work in progress....

Examples: Spidermonkey, Webrender, WGPU
Example: Spidermonkey

Script execution engine: JS and Wasm.

Some complexity hidden through safe Rust interfaces, but much use of low-level unsafe bindings still present.

Recent blog:

https://servo.org/blog/2024/04/15/spidermonkey/

Example: WGPU

Cross-platform, safe, pure-rust graphics api.

Used to implement WebGPU: DOM objects implemented by Servo in the script component, with a “backend” service running wgpu-core. Modular, again with the exception of some leaking: IPC communication.

Plans to re-use this infra to implement 2d canvas through Vello(Rust renderer using wgpu).
Example: Stylo-Blitz from Dioxus Labs

HTML and CSS renderer using Servo components: stylo(CSS resolution), html(html5ever) and css parsers(rust-cssparser).

“fulfill the long-held dream of many Rustaceans that Servo could power a native GUI library for Rust”,

https://github.com/jkelleyrtp/stylo-dioxus
Example: Rust-url

URL parser for Rust.

Used both in Gecko networking stack (Necko), Servo, and about 300k cargo installs a month.

https://github.com/servo/rust-url
Servo, the embeddable, independent, memory-safe, modular(?) parallel web rendering engine

Sometimes, and with ongoing efforts…
THANK YOU

More information available at:

@gterzian
@servo

servo.org
Thank you