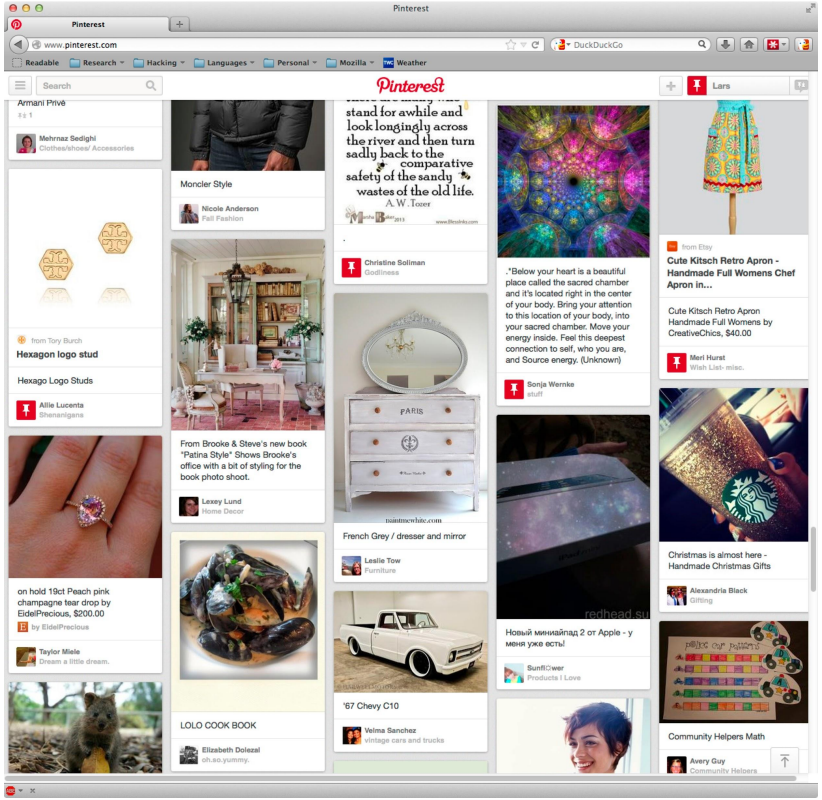
SERVO: TODAY & TOMORROW

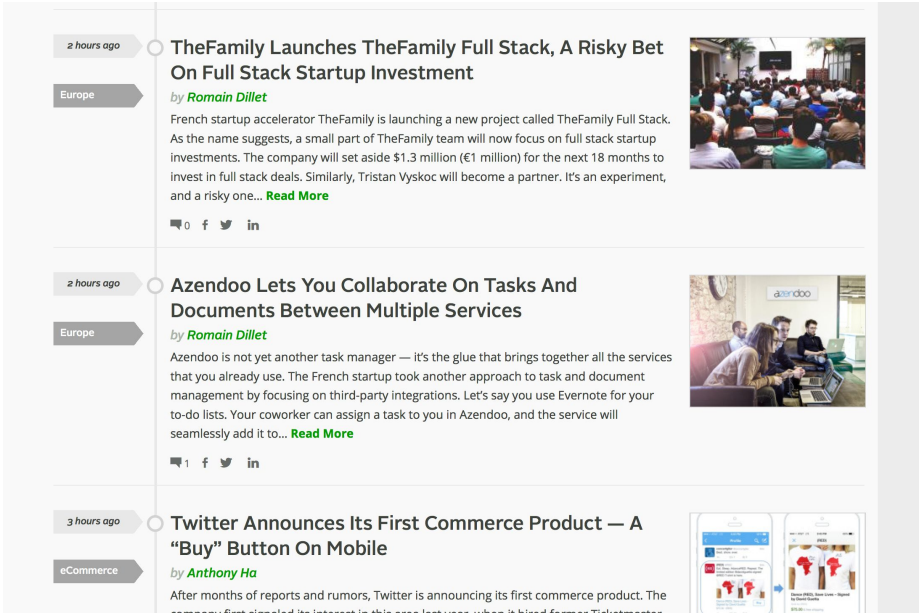
Jack Moffitt // Mozilla Research // September 26, 2016

Goal

Create a new browser engine making a generational leap in *performance* and *robustness*.

Performance Through Parallelism

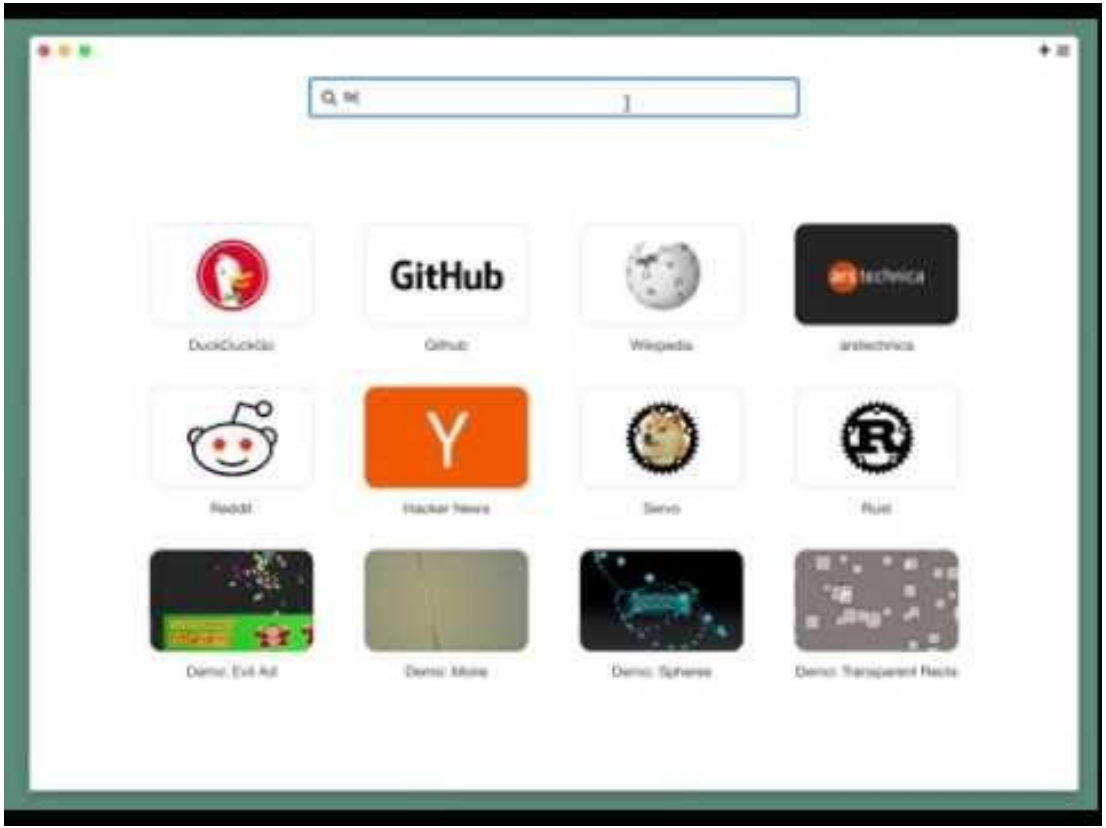




Robustness From Rust

- WebAudio implementation in Firefox had 34 security critical bugs
- All were array out of bounds or use after free errors
- All would have been prevented by the Rust compiler

Servo Nightly - download.servo.org



Servo Nightly Reception

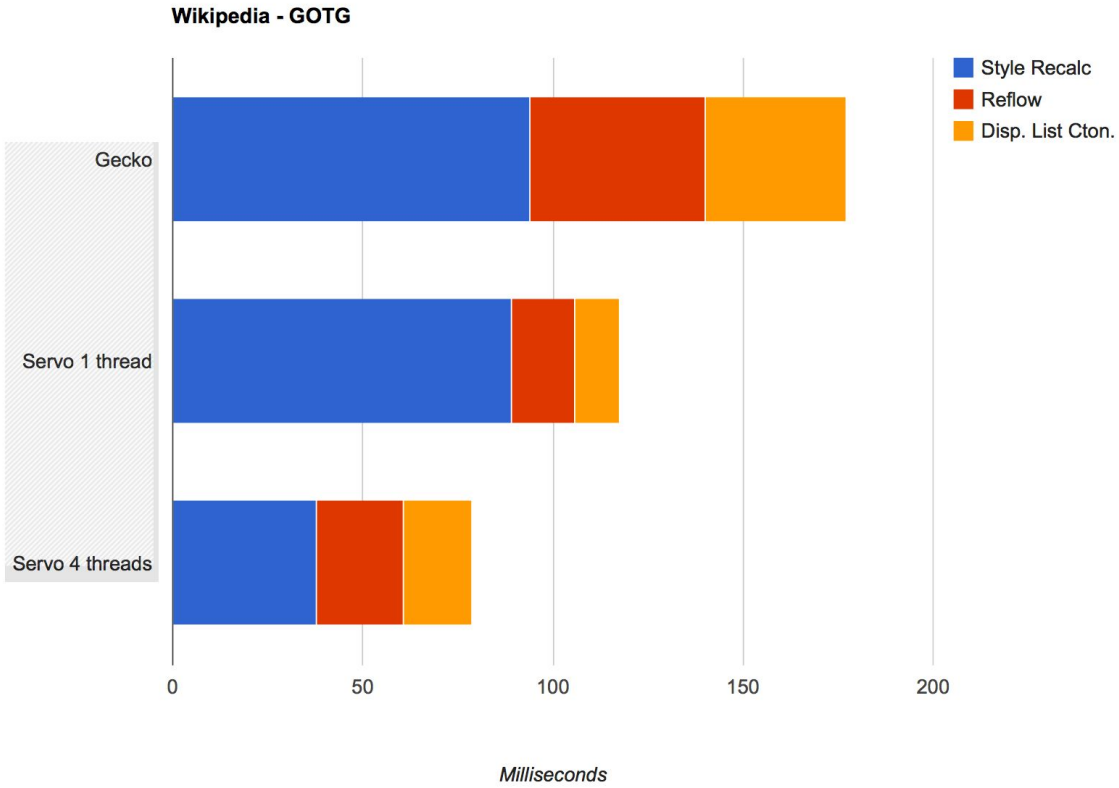
*“It's definitely slower to actually load pages — I'm guessing that there are still a lot of network optimizations that more mature codebases have accrued that Servo hasn't yet — **but holy cow are pages buttery smooth** once they do load (and even while they're loading, which is unusual). **Comparing Chrome and Servo in terms of UI jank felt pretty shocking**, in Servo's favor. Kudos, it looks like Rust and WebRender have paid off.” - reissbaker*

The Branches

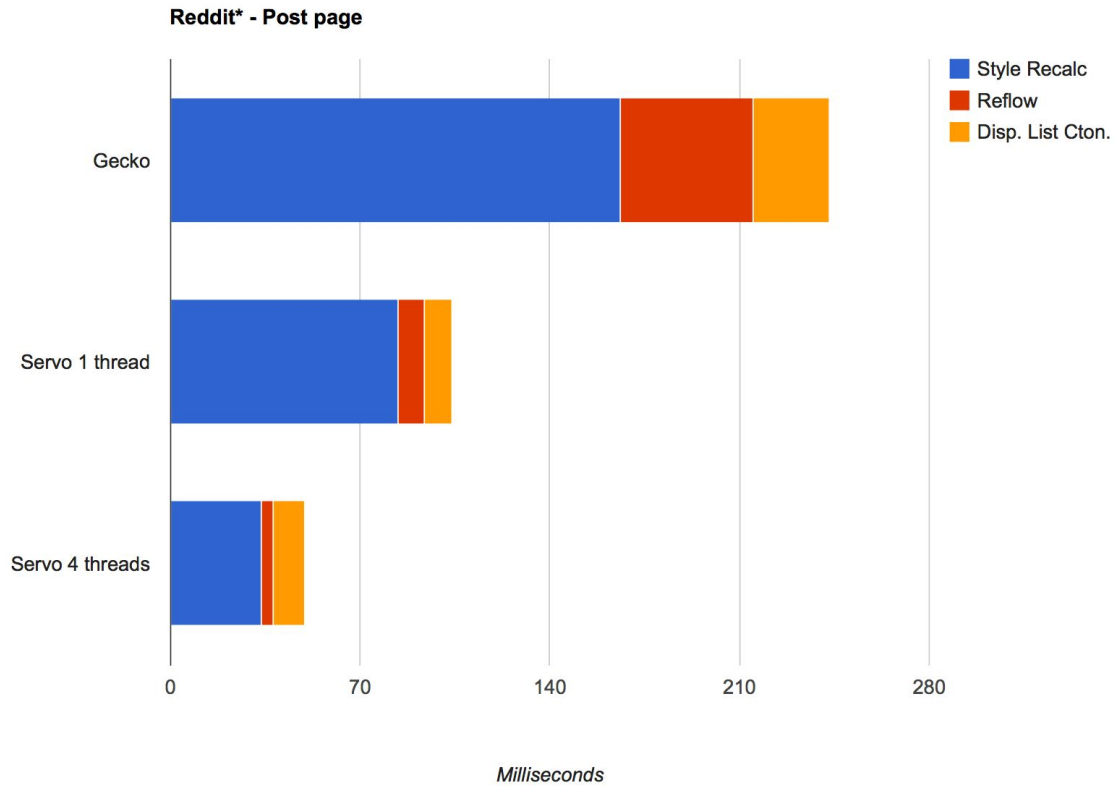
- Parallel layout
- Parallel styling
- WebRender
- Constellation

PARALLEL STYLE & LAYOUT

Servo vs. Gecko



Servo vs. Gecko



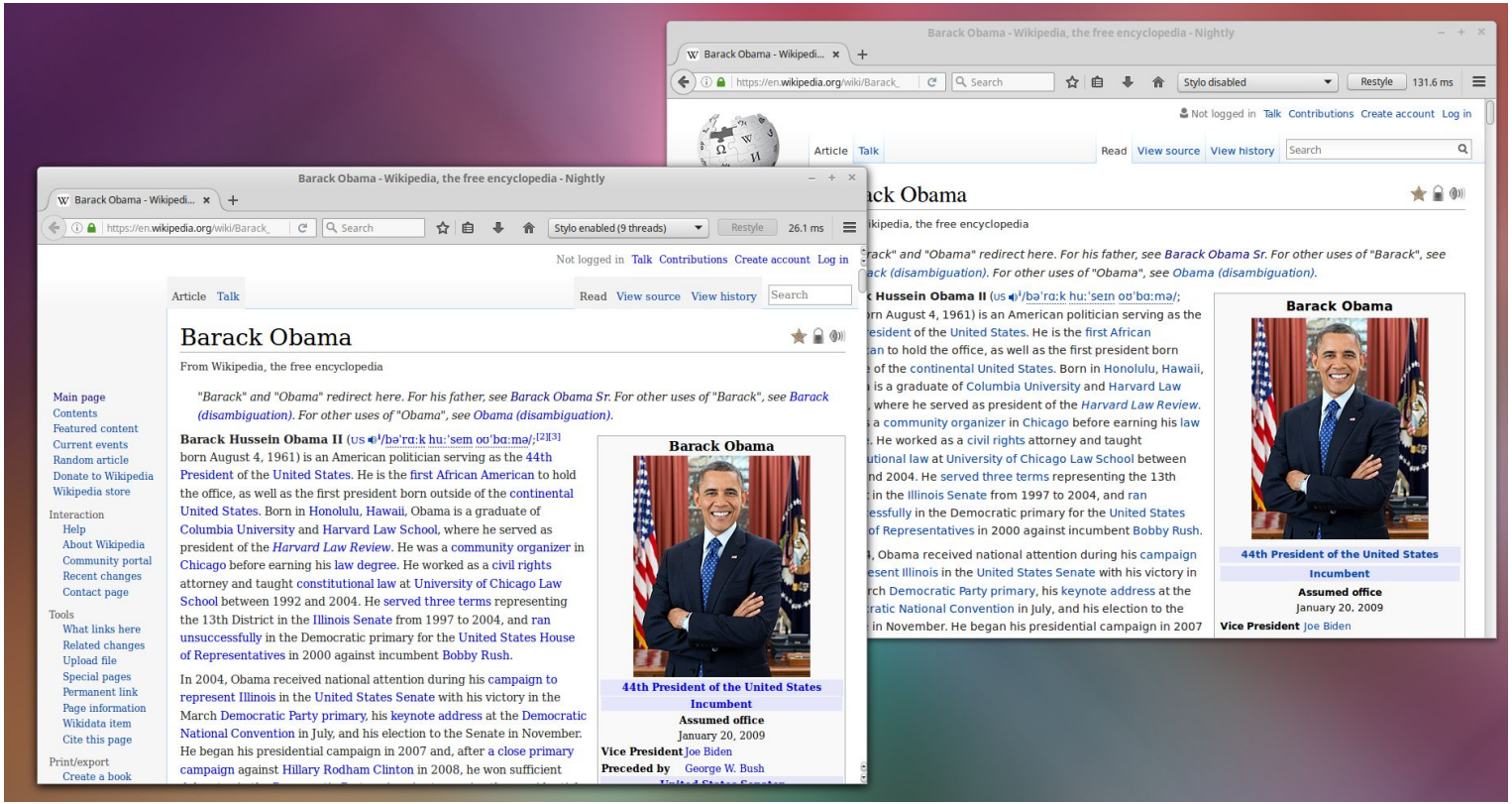
Power Performance

- Running cores at a lower voltage reduces performance by 30% but power usage by 40%
- Servo can make up this performance loss through parallelism, achieving the same performance with 40% less power

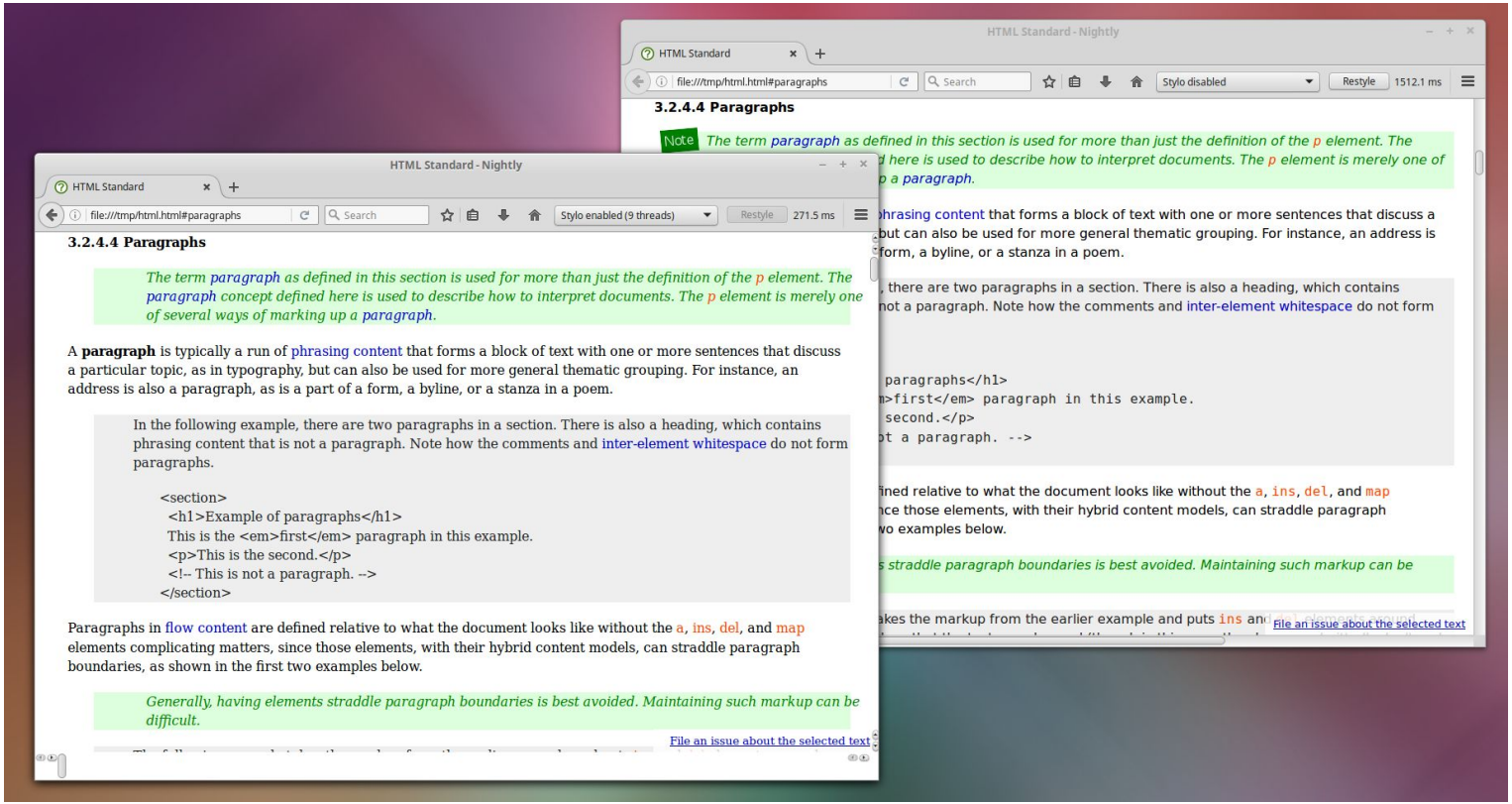
Stylo

- Goal: Replace the CSS styling system in Gecko with the parallel version from Servo
- Initial experiments showed same performance gain when used in Gecko
- Telemetry shows ~95% of Firefox users will get at least 2x speedup, and ~50% will get at least 4x.
- First major uplift of Servo technology into Firefox

Stylo on Wikipedia

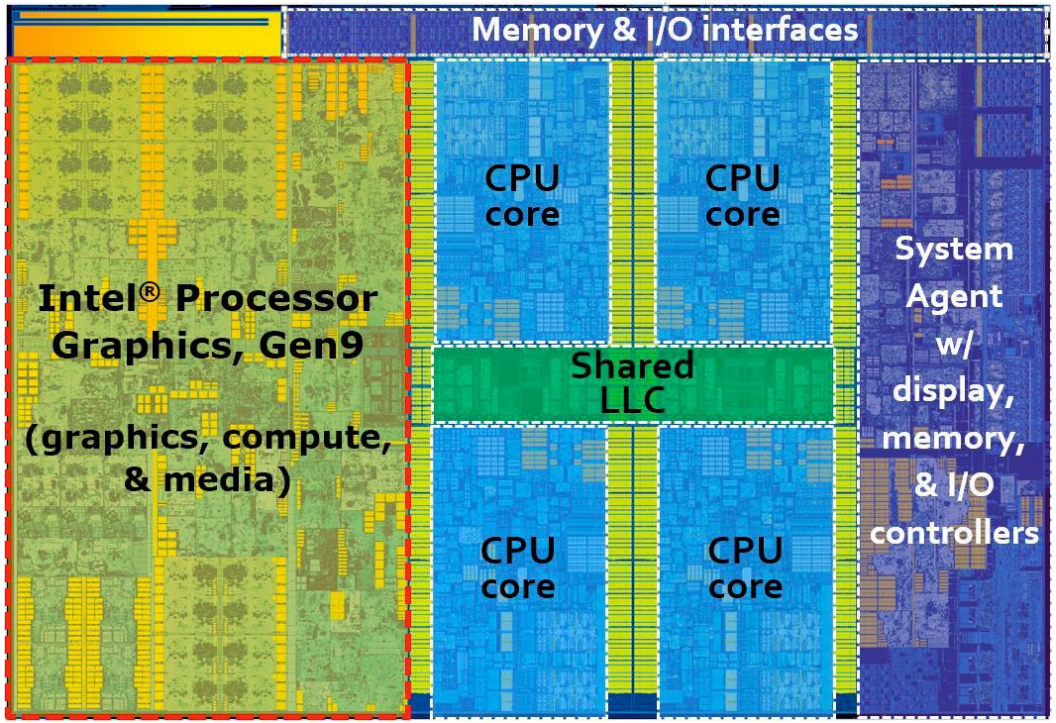


Stylo on HTML5 Spec



WEBRENDER

WebRender Motivation



WebRender in a Nutshell

- Hard to optimize immediate mode
- Retained mode graphics are better suited to GPUs
- Web pages are basically scene graphs, which can be sent to the GPU all at once
- Parallel CPU side preprocessing for some things (batch creation, font rasterization, border-radius, etc)

WebRender Progress

- Landed in Servo
- Optimized and fixed bugs
- Shipped in Servo nightly in June
- New design prototyped
- New design finished and landed

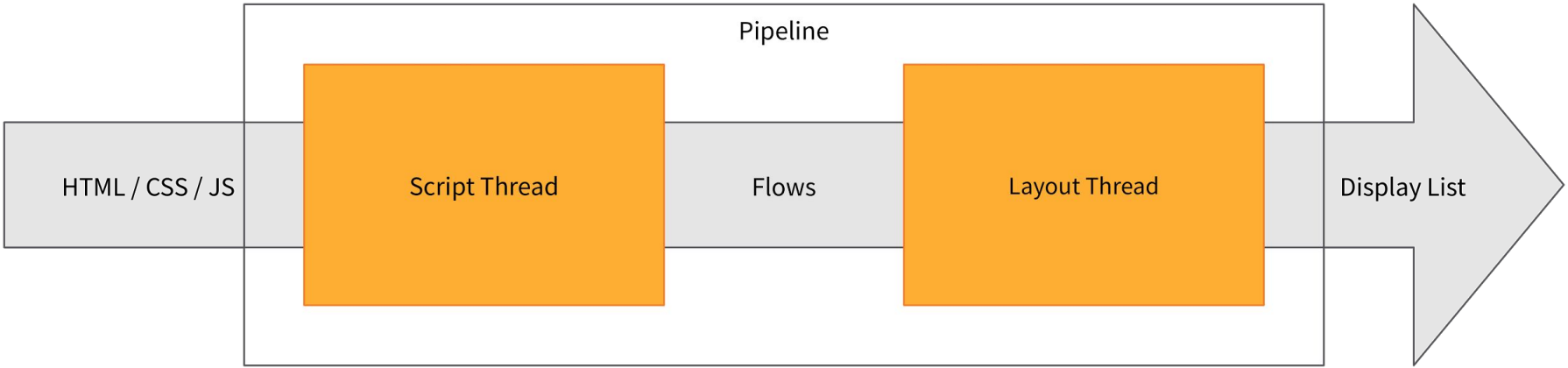
Firefox

Servo

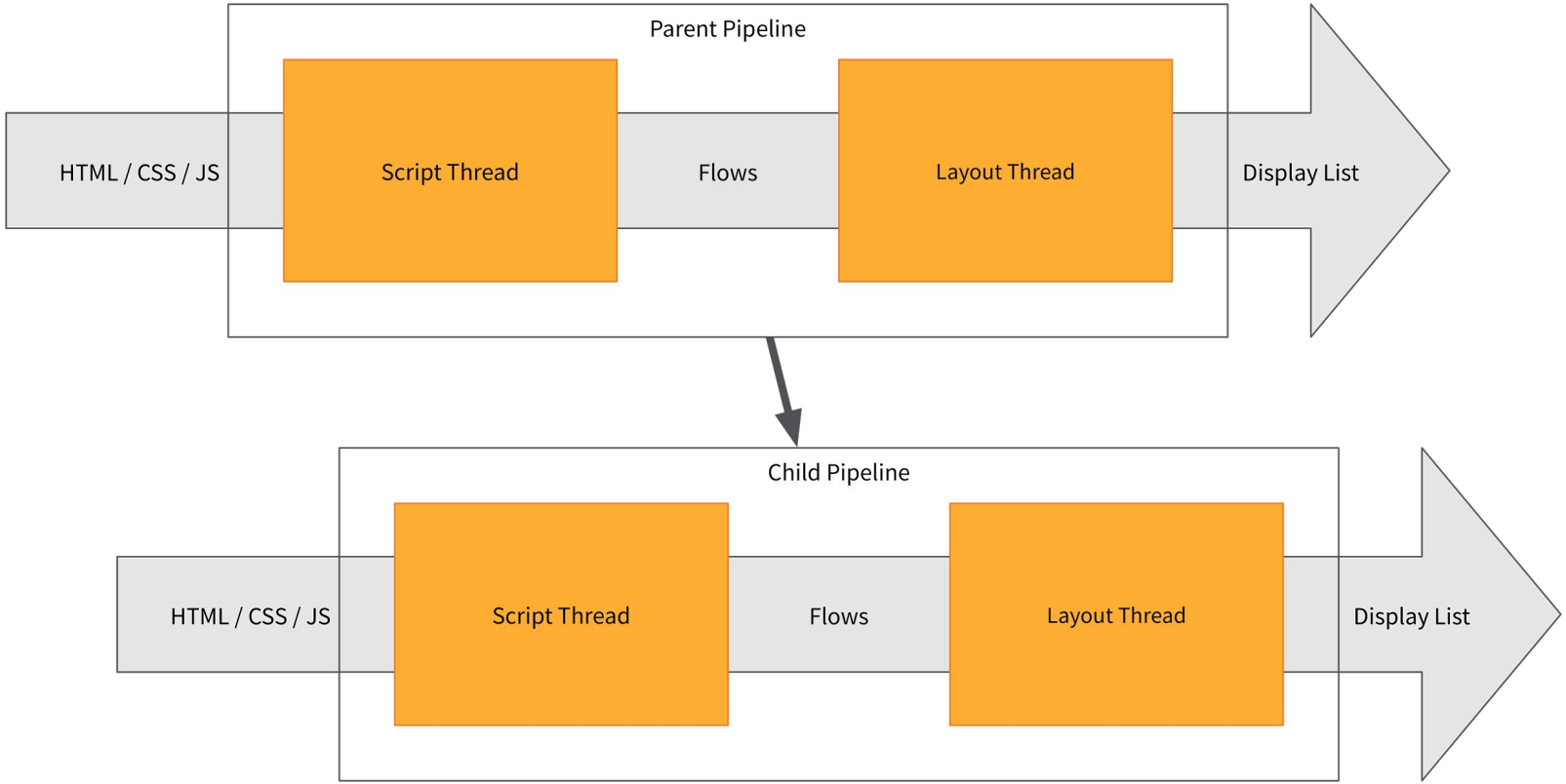

Apple WebKit Nightly - 5fps

CONSTELLATION

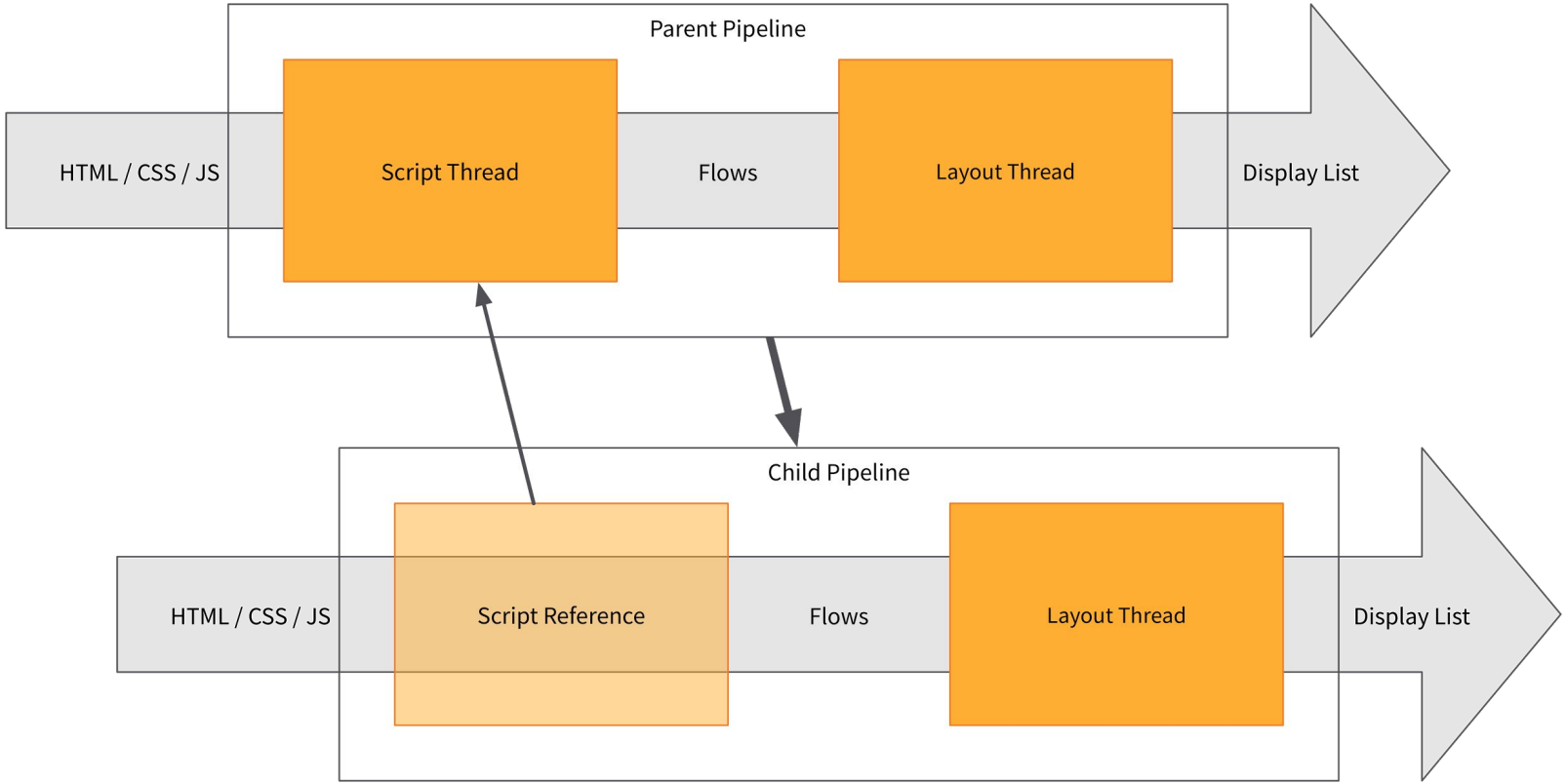
SINGLE PIPELINE



PIPELINE WITH CROSS-DOMAIN IFRAMES



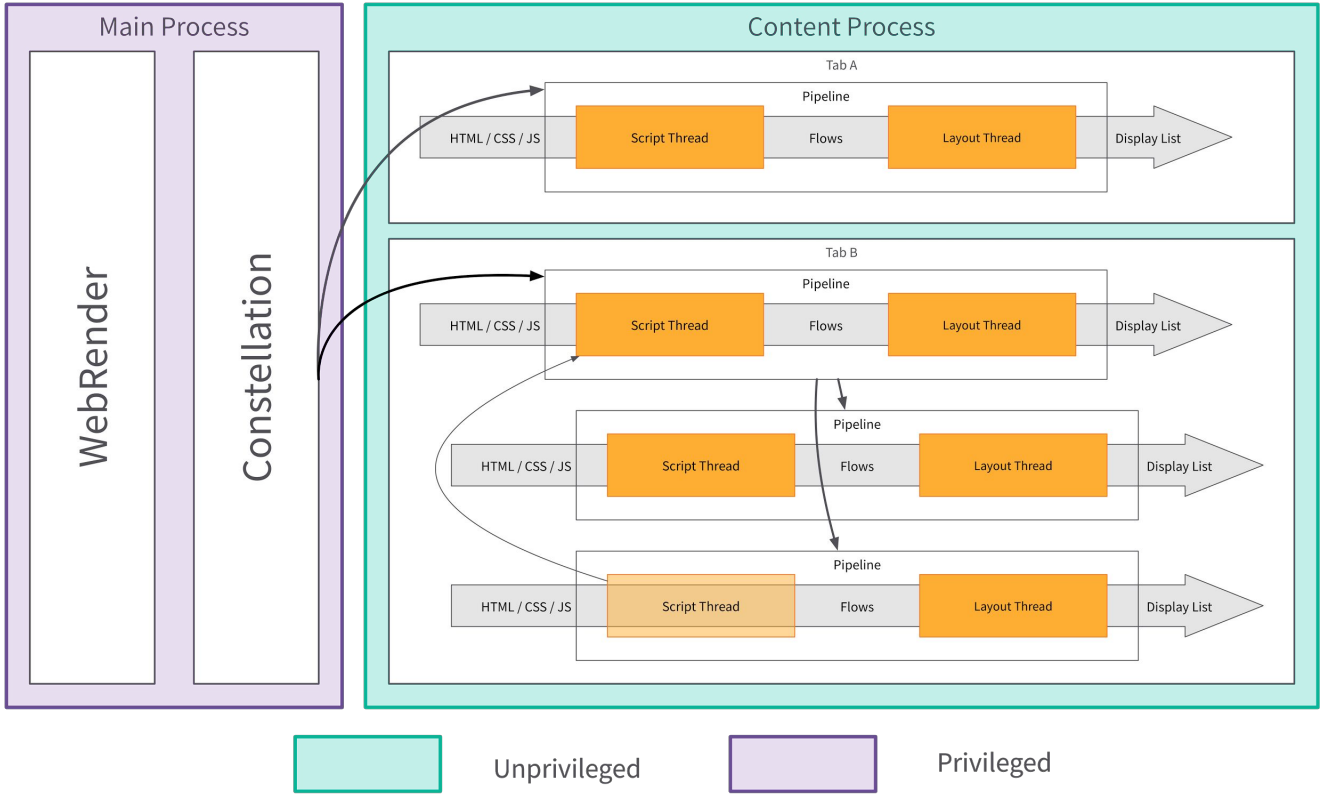
PIPELINE WITH SYNC IFRAMES



BENEFITS OF PIPELINES

- iframes don’t block their parent page or each other
- Layout and JavaScript don’t always block each other
- Many failures can be handled independently

MULTIPROCESS CONSTELLATION



The New Branches

- Constellation
- Parallel layout
- Parallel styling
- WebRender
- **Magic DOM**
- **Parallelize MORE**

Magic DOM

- Explore ways to achieve significant performance wins in JS / DOM integration
- Discussed for years, but a concrete proposal presented last year seemed fairly promising
- Fusing reflectors and DOM objects
- Self hosted JS

Parallelism Experiments

- JPEG decode (GPU)
- PNG decode
- Glyph rasterization (GPU)

External Research Collaborations

- Machine learning (Rohit Zambre at UC Irvine)
- Power management (Connor Imes at U. Chicago)
- Incremental computation (Kyle Headley at U. Colorado)
- WebBluetooth (Gabor Loki et al, University of Szeged)
- Software transactional memory (Alan Jeffrey)

QUESTIONS