



Google Chrome A1: Conceptual Architecture

By Sushi Overflow

Introduction

Google Chrome is a free **web browser** largely based on Chromium, which is an **open-source software** developed and maintained by The Chromium Project first released in 2008.

Four main goals of Chrome: **Speed** **Security** **Stability** **Simplicity**

EX) *Internet Explorer 2008*



- Around **62%** of people in the world use Chrome in their daily life. (from data collected on Sept, 2018)

Derivation Process

Step 1: Researched about the history of Chromium, to make sure we understand the principles behind.

Step 2: Looked at the multiprocessor architecture of chromium

- i.e. learned about browser, render, and connections between them

Step 3: Examined the reference architecture of web browser.

- Figured out that there are some significant differences between the conceptual architecture of Chrome and the reference architecture

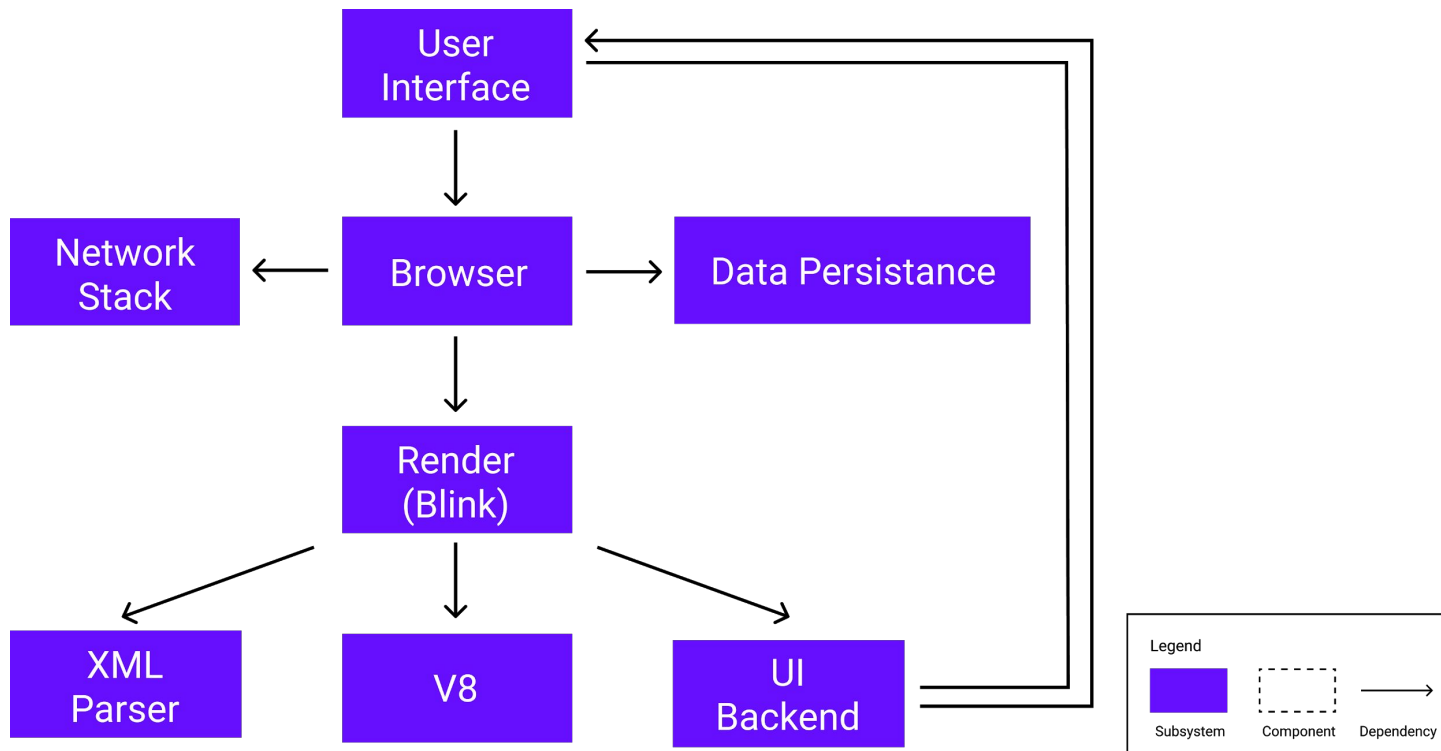
- such as the dependencies of subsystems, and some subsystems are not for Chrome.

Step 4: Based on what we had for the conceptual architecture, we came up with a sequence diagram.

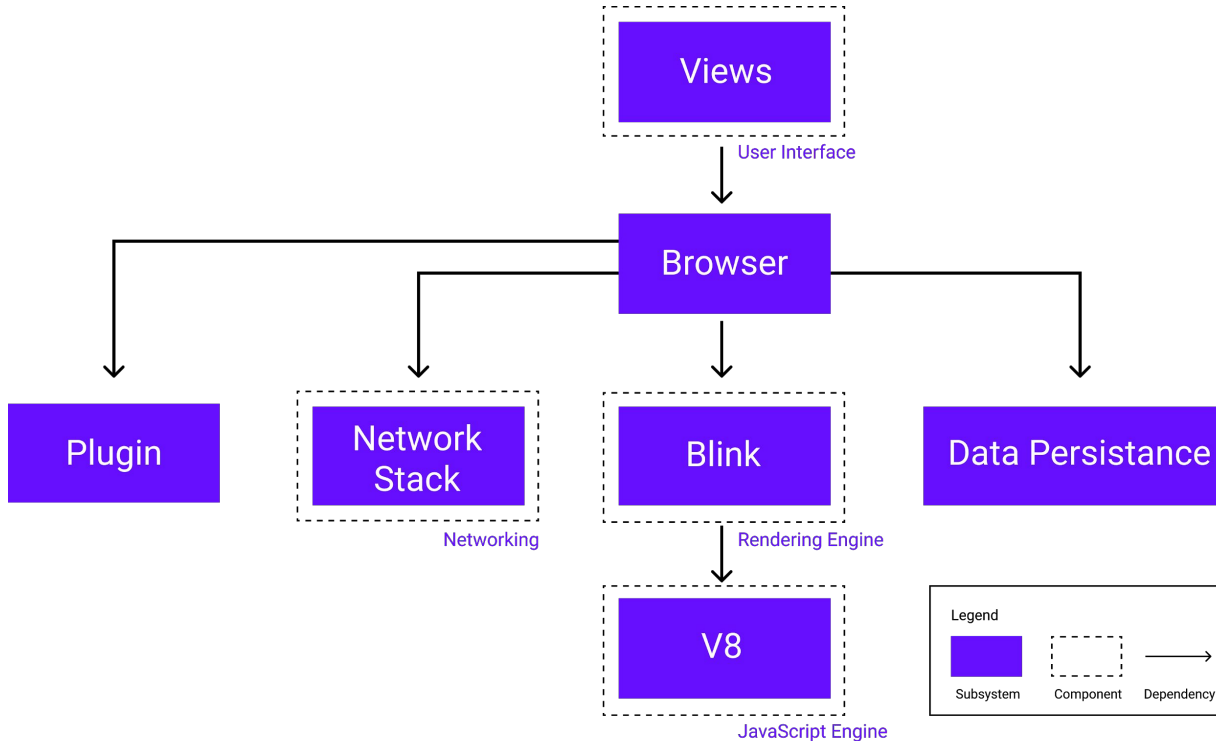
- To match the sequence diagram, we modified our conceptual architecture back and forth to make sure it matches.

Step 5: Showed our work to TA to confirm that our <opinions> matches our architecture.

Alternatives: Before Finalization



Final Architecture Diagram



Architecture Style:
Hybrid of Layered & Object-Oriented style

Advantage:

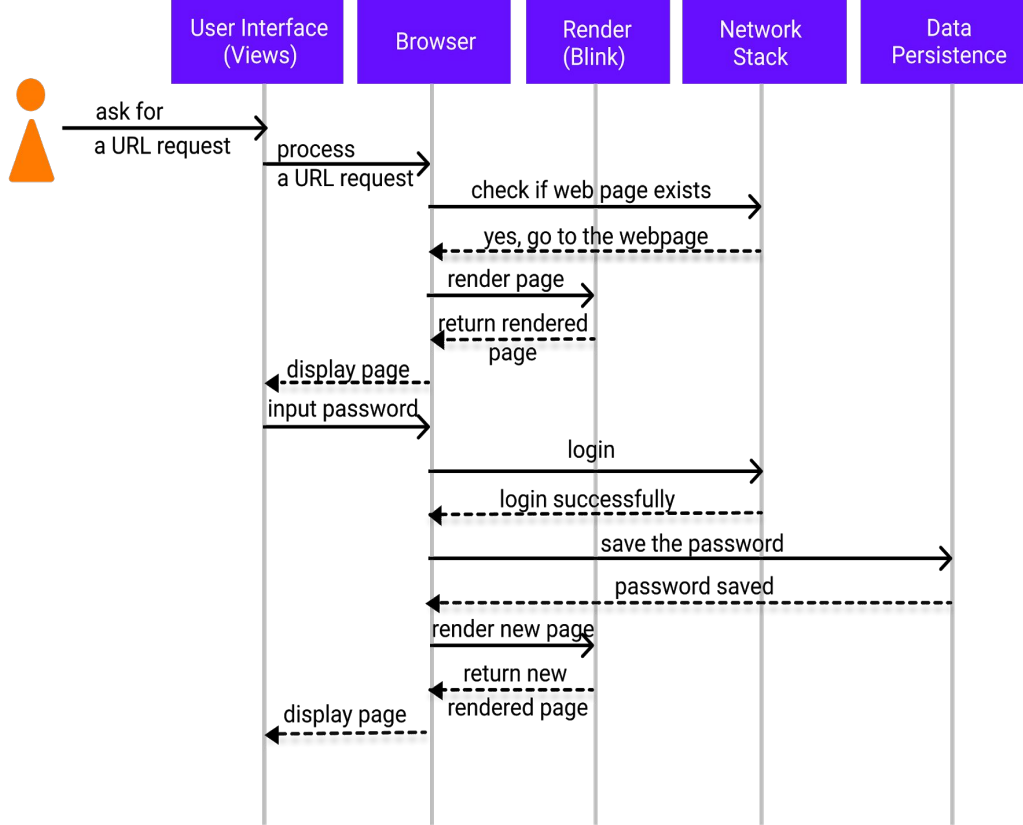
- Reuse

Disadvantage:

- Coupling

Features in Chrome:

- Extension/Plug-in
- Multi-Processor Rendering Engine



Sequence Diagram

for the use case of; logging in to a website and succeed, Chromium saves the password

Legend



subsystem



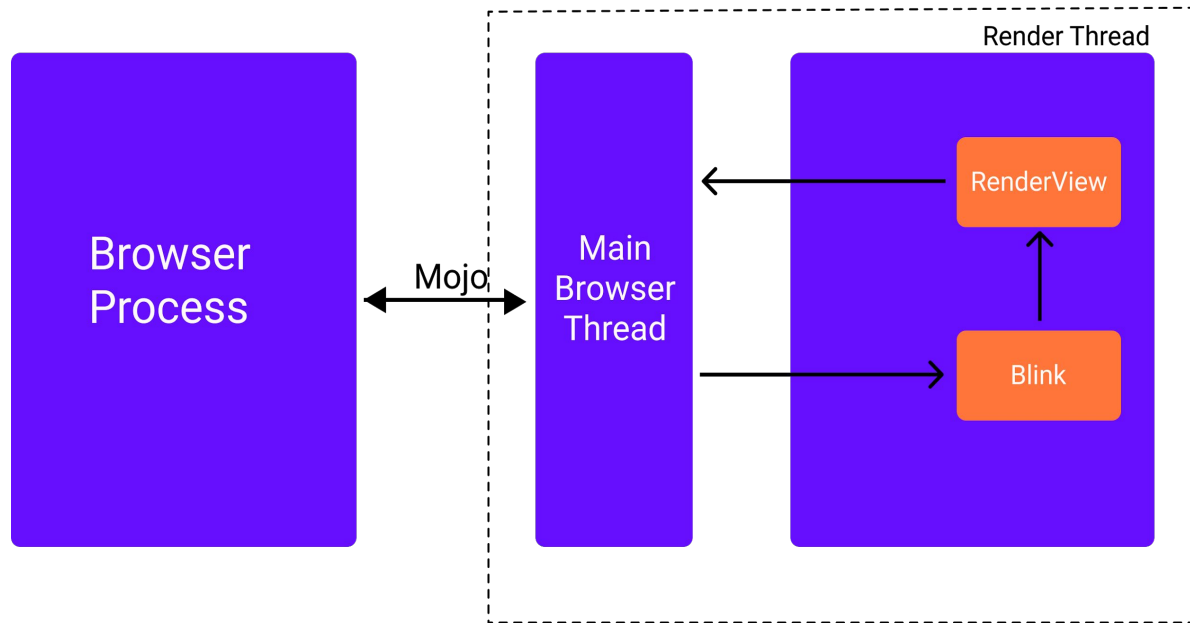
action



feedback

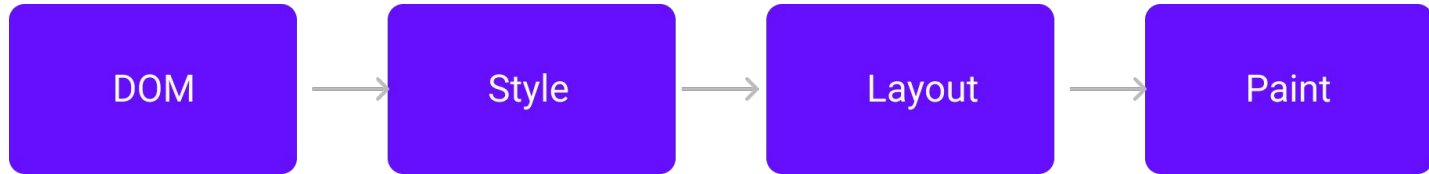
Deep Dive: Rendering Engine

The rendering engine is responsible for constructing web pages. A single browser process manages n separate render processes running **Blink**, each of which are sandboxed.



Deep Dive: Rendering Engine

Blink running in each rendering process.



Effects of Concurrency

Definition: Concurrency improves the overall speed of the execution in multi-processor system.

In Chromium, there are separate processes for browser tabs; restrict rendering access.

Sandbox

: The sandbox is a C++ library that allows the creation of sandboxed process

To remove privileges and disable some groups before launching the render process

Site Isolation

Separate processes for each browser tabs
Why one process for a group of tabs in reality?
=> trade robustness for low memory

MOJO (IPC in the past)

Blink can use Mojo to interact with the services and the browser process.

Chrome Team Issues

Architecture impact on the teams inside Chromium and their cooperations:

- Multithreading: main thread & multiple worker threads.
- Memory Reduction: multiprocessor requires memory & teams' cooperation.

Architecture impact on the teams between the Chromium and other companies:

- Multi-process causes difficulty on debugging of Windows.
- Multi-process allows for flexibility doing security (sandboxing on MacOS).

Lessons Learned

Balance & Priorities

After hours of researching and understanding the goals of the product, coming down to a list of the core subsystems of the requirements were important to building the conceptual architecture.

Importance of Long-term Goal Thinking

The Chrome Team's miss on considering cross-platform issues caused them to recreate from scratch for MacOS, Linux, and mobile, which could have been prevented if they recognized long-term possibilities during the initial stage of the architecture development.

Sequence Diagram

Sequence Diagram played a significant role in visualizing the real-life use case of the architecture, which led to help solidify the conceptual architecture.

Conclusion

Google Chrome was developed to focus on content, adding new features like draggable tabs, and extensions were minimized as much as possible to ensure simplicity, speed, security, and stability.

Chrome is best suited with hybrid of layered & object-oriented style architecture, because it is modularized to instantiate processes. Moving from WebKit to Blink, Chrome could develop a faster and scalable rendering engine to further improve on speed and stability.