CISC 322 Assignment 1:
Conceptual Architecture of Google Chrome

Inspect Element (Jonathon Gallucci, Joseph Gravenor, Jack Guinane, Maxwell Keleher, Matthew Pollock, Roberto Ruiz De La Cruz)

Overview

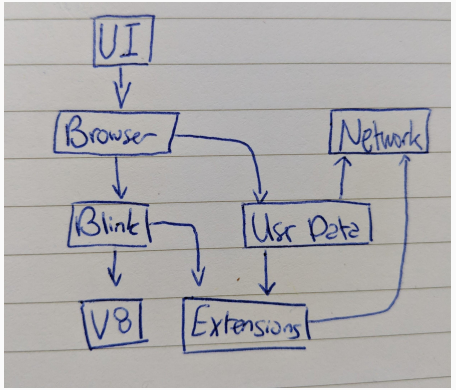
0. Intro
1. Derivation Process
2. Actual Arch
3. Walk through the actual arch
4. Dig into 1 subsystem
5. Concurrency
6. Team Issues
7. Lessons learned
8. Chrome-clusion

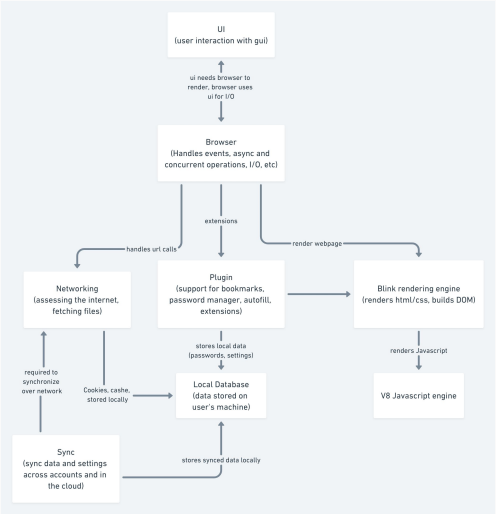
What is Google Chrome?

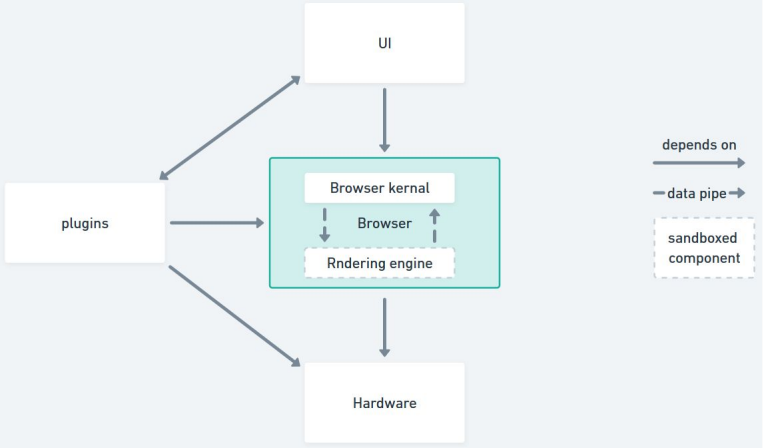
- Developed by Google as part of the open-source Chromium project
- Built with C++
- Minor updates every 2-3 weeks and major updates every 6 weeks
- As of 2018 Chrome has a 56% market share across all platforms
- Focused on speed, stability, and security

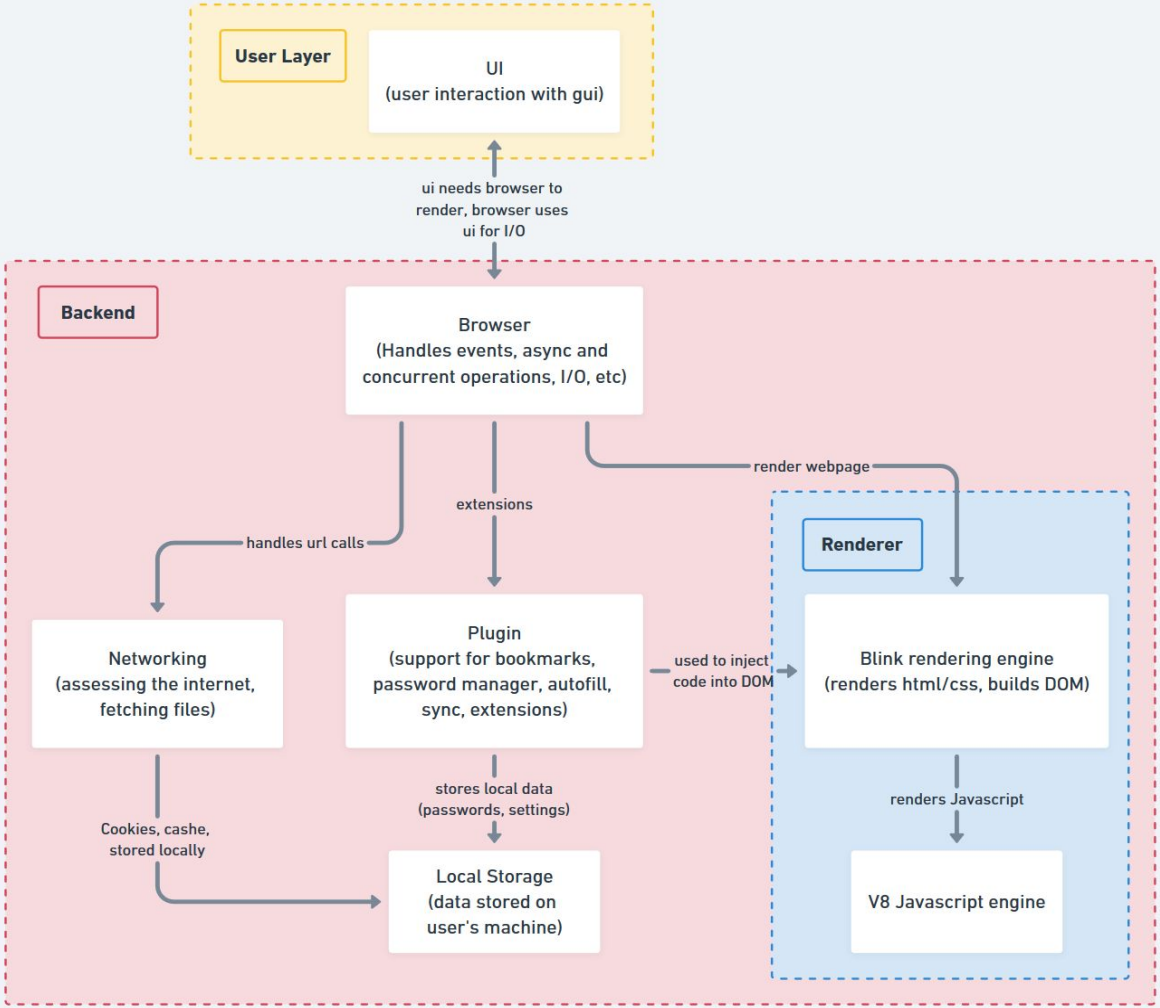
Derivation Process

1. Drafts in Pairs
2. Combine the drafts
   a. Accept Similarities
   b. Discuss Differences
3. Build Final Version
4. Verify Architecture









Conceptual Architecture

Architecture is object oriented and layered

Allows the sandboxing of elements (such as 3rd party plugins) for security

Easier to update only one subsystem, allowing chrome to have quicker development cycles than IE or Firefox

Browser

- Main backend subsystem
- Handles events, I/O, calling renderer, etc.
- What all the subsystems interact with
- Allows for the sandboxing of other systems (check network calls for viruses, keep extensions from overstepping)

Network Stack

- Fetches url requests and possibly user filesystem for local requests (ie file:///…)
- Written to be cross-platform, to help eliminate bugs in native networking and improve speeds
- URLRequestContext contains all the associated context necessary to fulfill the URL request, such as cookies, host resolver, proxy resolver, cache, etc.
- URLRequest, as indicated by its name, represents the request for a URL
- Requires Local Storage for storing cache and cookies

Plugin

- The plugin subsystem allows for ‘plugins’ to interact and improve the browser
- Includes Chrome Extensions, Bookmarks, Password manager, Google sync
- Dedicated subsystem allows 3rd party plugins to be sandboxed
- Allows each extension to be developed discreetly
- Depends on Local Storage to save data (settings, bookmarks, passwords, etc)
- Depends on Blink to read the DOM and inject code into the webpage

Local Data

- Subsystem for interacting with the user’s filesystem
- Used to store setting, passwords, cookies, etc
- Synced with user’s other machines via Sync plugin
- Separate subsystem allows all data to be kept together (easier for security and sync)

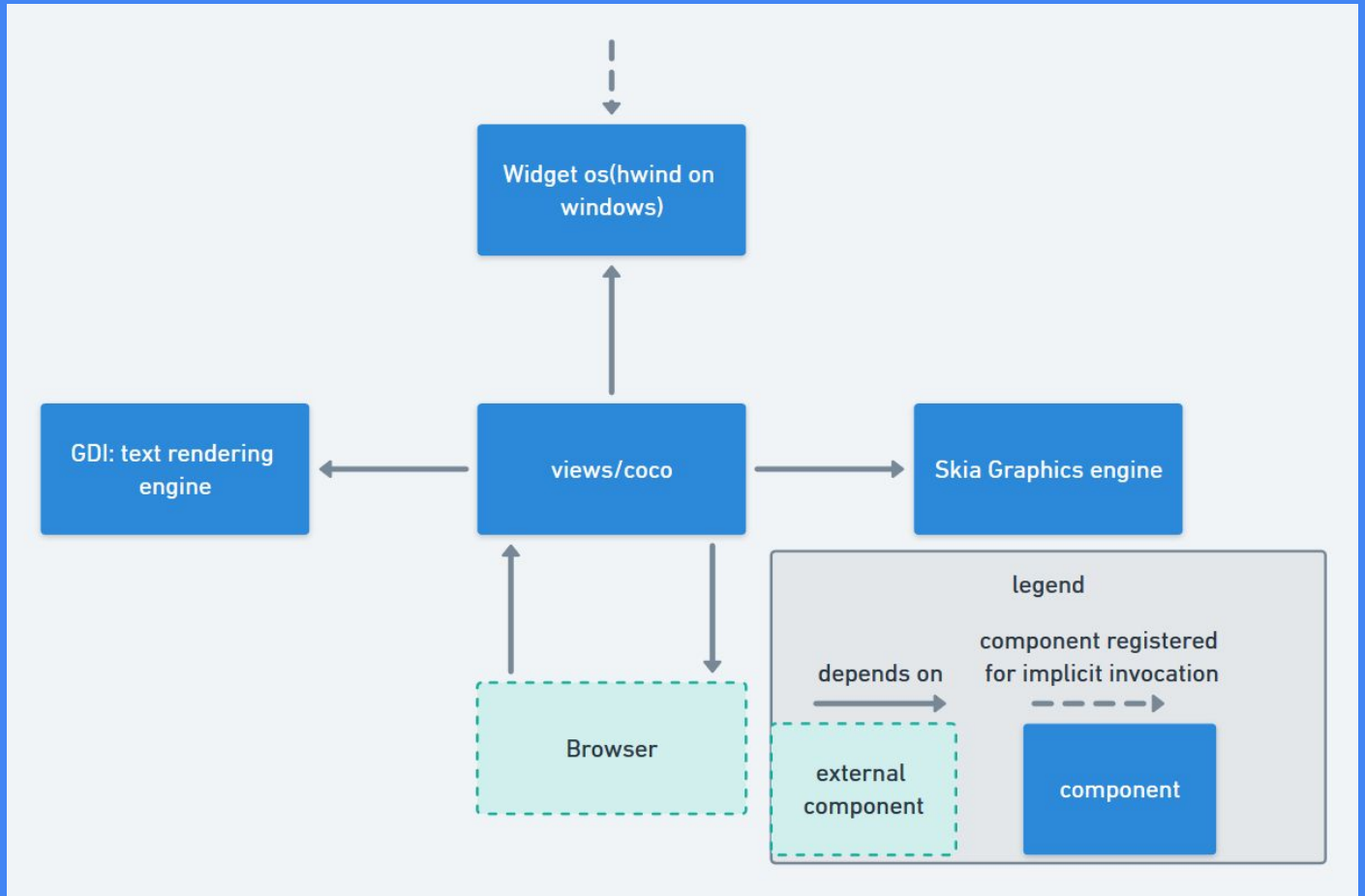
Blink

- Blink is the Rendering Engine responsible for rendering and HTML and CSS
- Builds the DOM Tree and the Layout for the Browser to render
- Depends on V8 to run any Javascript in the HTML
- By Being separated this can allow Blink to be ported to other browsers and helps with coupling

V8

- This just executes Javascript Or Webassembly and passes what needs to be rendered to Blink
- While technically bundled in Blink it’s its own subsystem and it doesn't require blink
- Treated as an isolated sandbox where blink passes javascript V8 executes

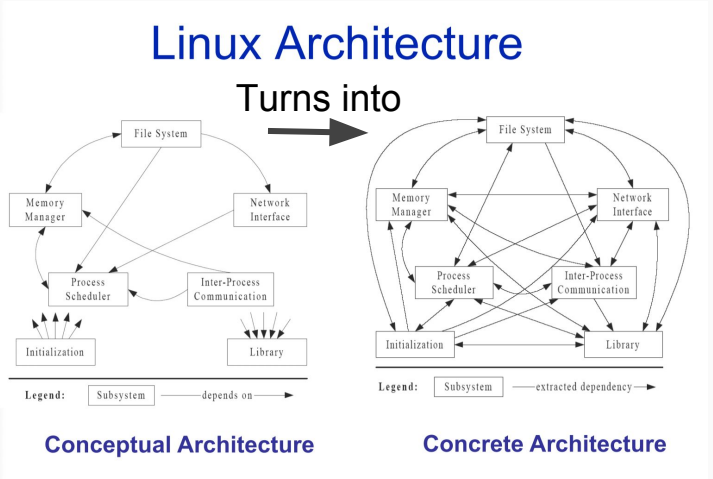
Subsystem Architecture: UI



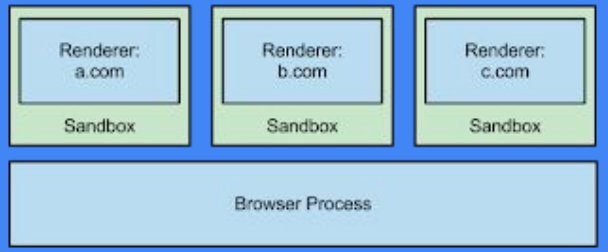
Problems with Development

Problems with our Architecture:

- Architecture proposed is an object oriented, layered style
  - Performance concern as events go through layers
  - It is hard for the design to follow the architecture.
- Open source
- Plugin architecture used to build browser add ons
- Concurrency uses substantial resources

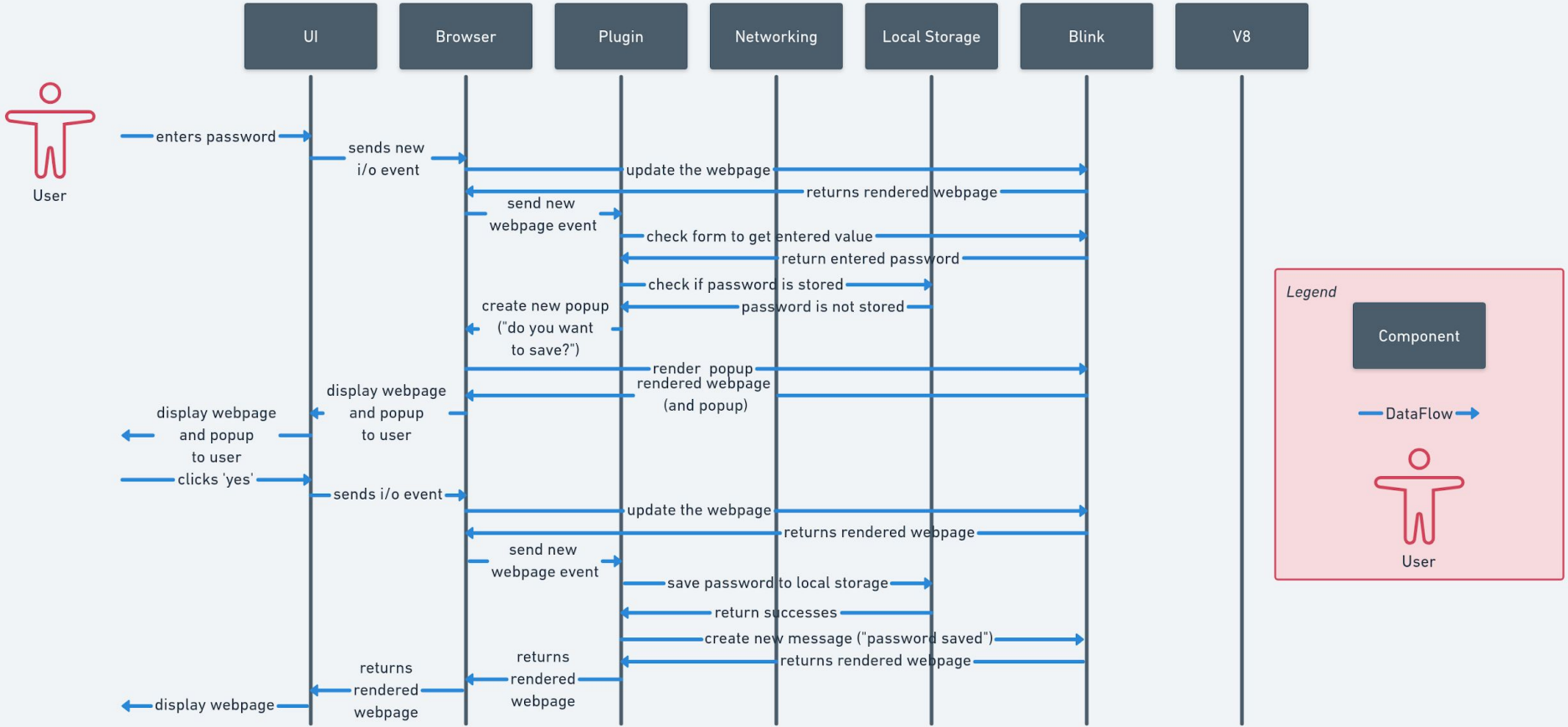


Concurrency



- Chrome concurrent works through a process called Site Isolation
- Site Isolation run each web frame and Iframe in its own sandbox renderer
- This uses Chromium Process-per-site-per-instance model
- Every subsystem is concurrent to allow for this to be possible
- This increases security as each site cannot talk to other sites
- Ex. Iframe embedded in site can’t access the DOM for that site reducing XSS attack vectors
- Because each subsystem is discrete component you can have multiple instance of require one without duplicate unnecessary ones
- Centralized browser also chrome to control the creation of each instance of rederer

Sequence Diagram



In Chrome-clusion

- Difficult to find accurate data due to out of data documentation.
- Documentation was dense.
- Highly modular which allows for quick development cycle.
- Portability maintained by default OS input fields.
- The high RAM use is the result of the multi-processing architecture.
- High amount of sandboxing can cause performance issues, but improve security issues.