Assignment 2

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

Intro

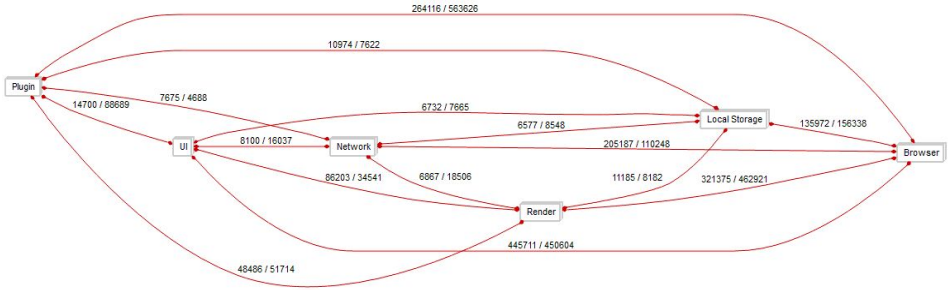
0. Intro
1. Derivation Process
2. Concrete Architecture
3. Justification
4. Sequence Diagram
5. Concrete Architecture of the UI Component
6. Concurrency
7. Components
8. A3 Feature
9. Lessons learned
10. Chrome-clusion

Derivation Process

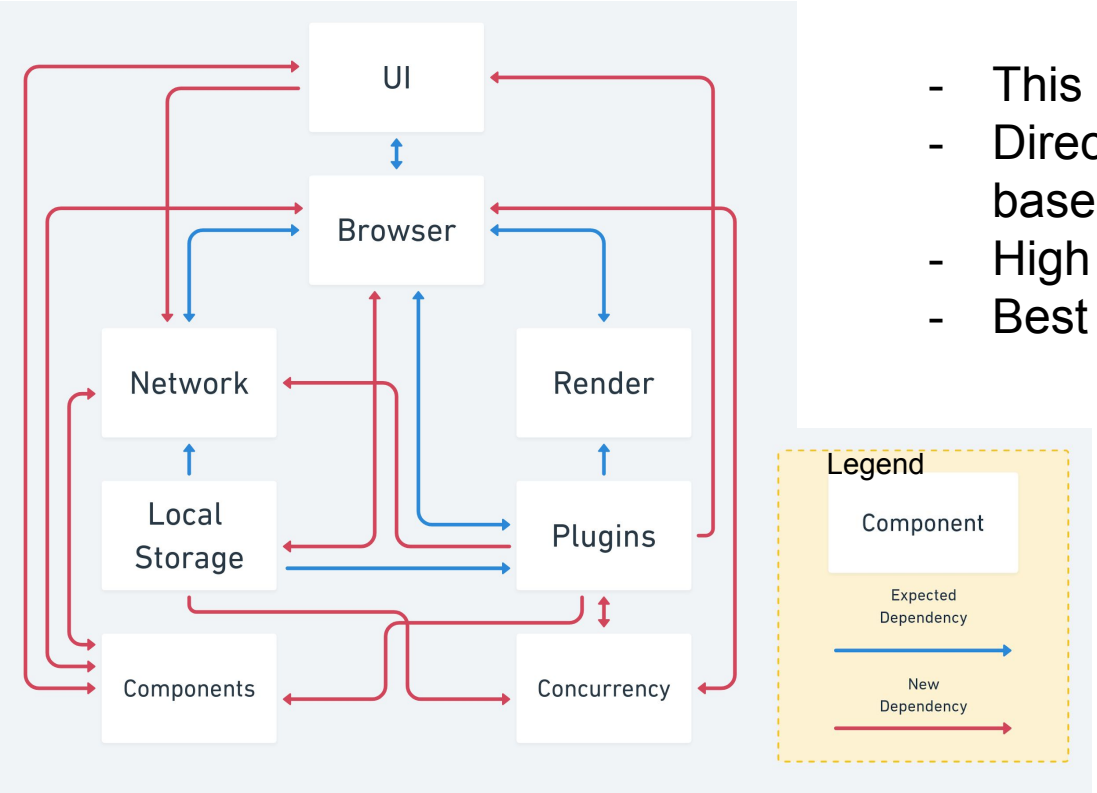
- To maximize the cohesion of our architecture, we looked through each directory in Understand one by one.
- This optimization for cohesion came with a price in the form of coupling. Each element is fully connected with high numbers of dependencies.
- This violates the benefits of our architecture style, as this level of coupling makes it impossible to change the implementation of one subsystem without affecting other subsystems
- It became clear to us that our architecture needed to change to change to correct this violation

Derivation, Continued

- To reduce coupling, we changed our approach
- Focusing on the purpose of the classes within the directory, we moved directories as units, choosing the most common purpose for each
- Although lower cohesion, we believe that these changes better reflect the concrete architecture
- Created two new subsystems...



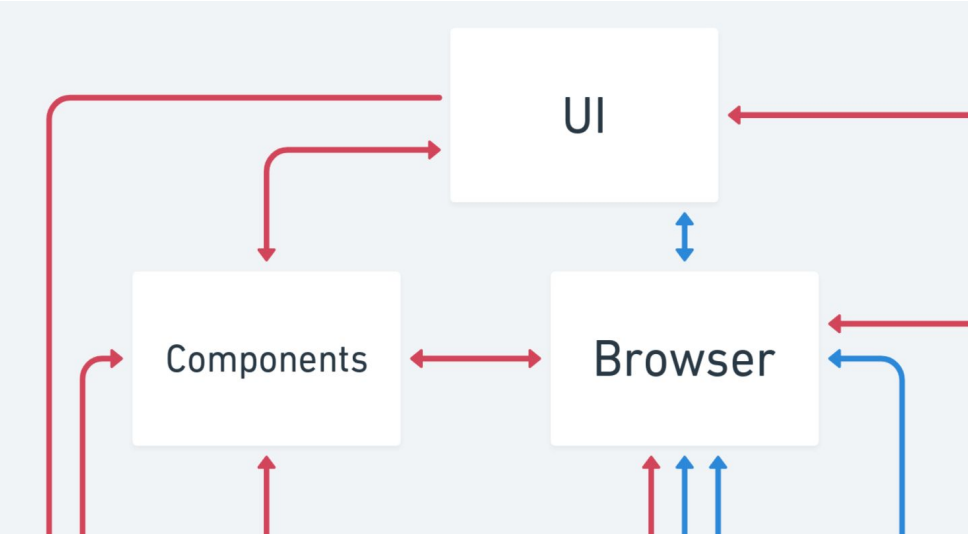
Concrete Architecture

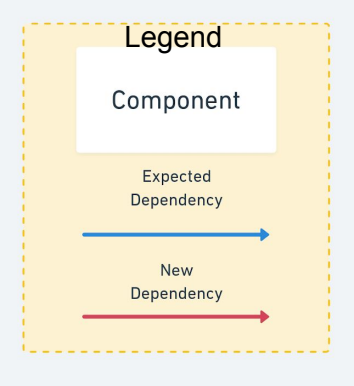


- This is the concrete architecture
- Directories were searched and sorted based on relations and services
- High coupling and high cohesion
- Best reflects Chrome as it is today

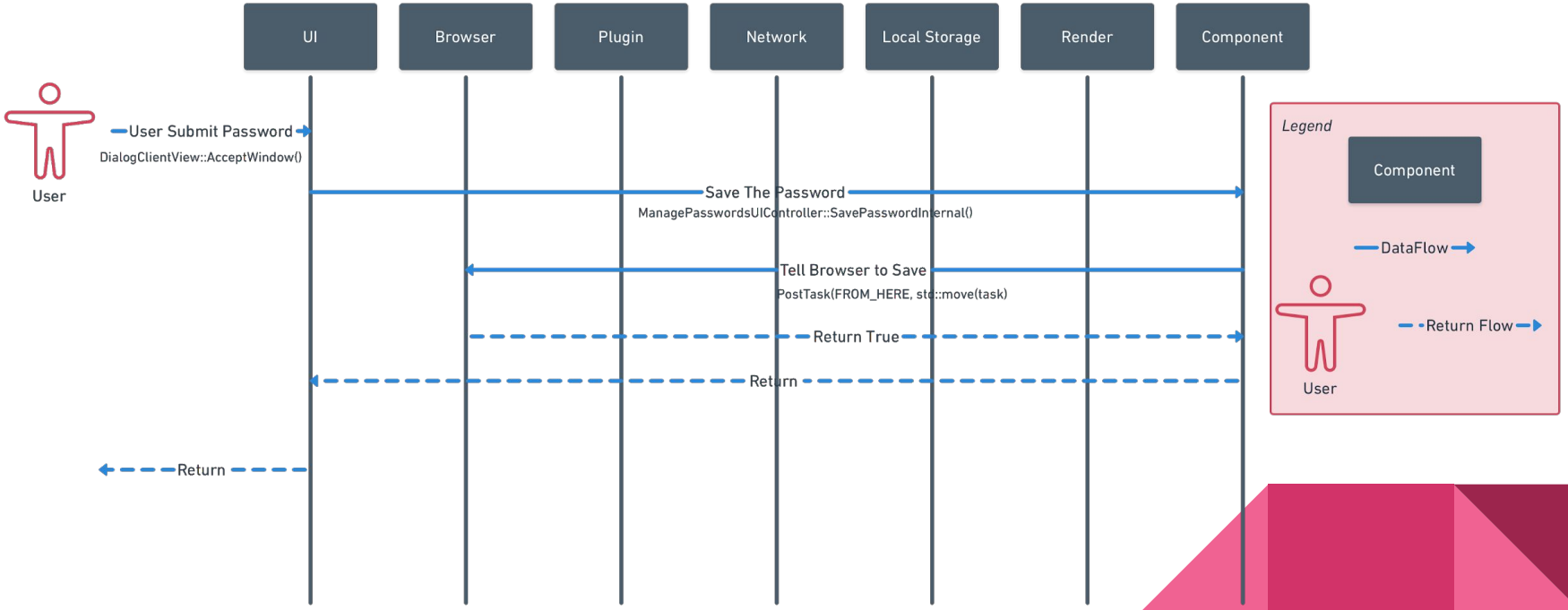
Justification

- The concrete is object-oriented and is *not* layered
  - Too many dependencies
- UI -> Browser becomes UI <-> Browser
  - Platforming handling
- New subsystem Components is codependent on UI

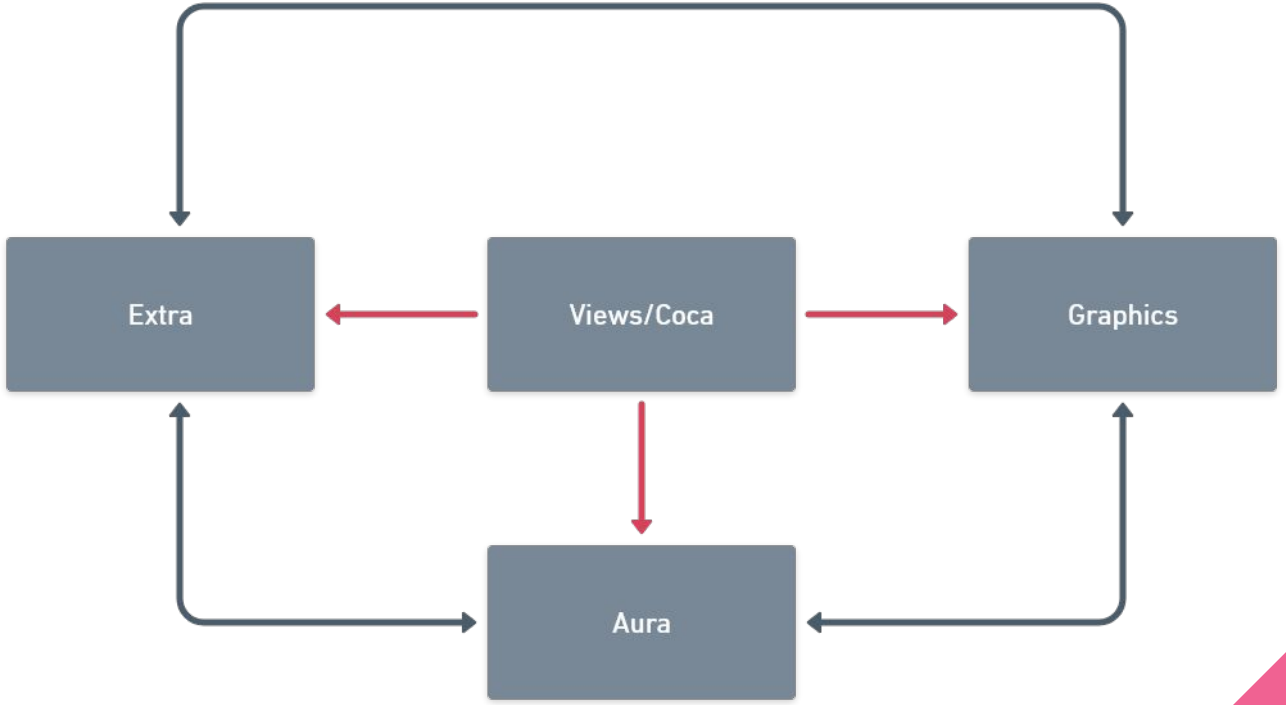


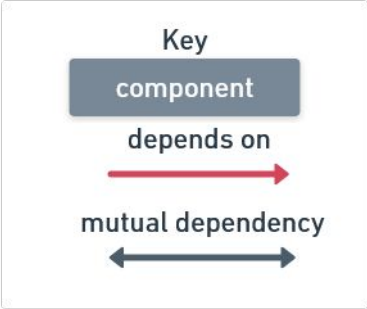


Sequence Diagram



Concrete UI


Extra
Views/Coca
Graphics
Aura


Key
component
depends on
mutual dependency

Concurrency

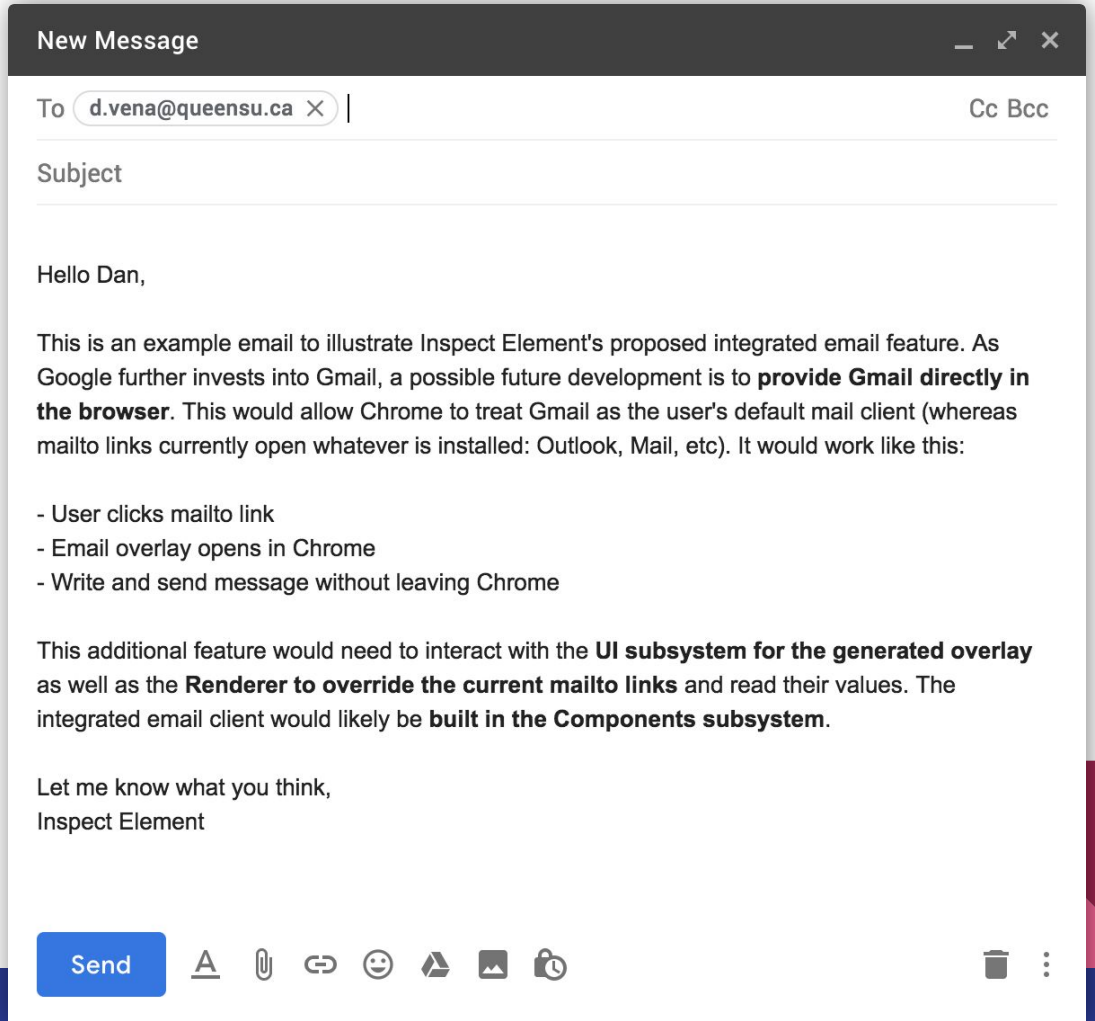
- We originally required a “centralized concurrency controller”
- Previously determined to be a part of Browser subsystem
- Upon researching the concrete architecture, we discovered that this was its own subsystem
- Our previous report had noted that Mojo was the system they were using
- Thus, we created the “Concurrency” subsystem
- Main dependencies:
  - Plugin: for controlling sandboxed plugins, broker interfaces
  - Browser: for handling tabs, windows, iframes

Components

- Our previous report suggested multiple extensions assessing the plugin subsystem
- We neglected to include these in any subsystem
- After viewing the codebase, the components directory housed most of these extensions, however were given more permissions
- Main dependencies:
  - UI: for interaction with the application window (bookmarks' 'star')
  - Network: for network access & network interruptions

A3 Idea

d.vena@queensu.ca



Lessons Learned

Thick coupling & Low Cohesion

- Not layered style, all object oriented
- Hack-y files and classes

New subsystems & new dependencies

- Components & Concurrency added
- Local Storage component does not work as expected

Concrete Architecture of Chrome has difficult modifiability

- Interfaces vs. Huge interdependent codebase
- Not layered, less modifiable

Lessons Learned: **Chrome Team Issues**

Amount of interfaces creates modifiability difficulties

- Comes from thick coupling

Open Source issues expanded

- Created a huge codebase
- Some poorly documented subdirectories included

Object oriented style side effects in a huge codebase

- Next to impossible to test in the concrete architecture

Lessons Learned: **Limits of Findings**

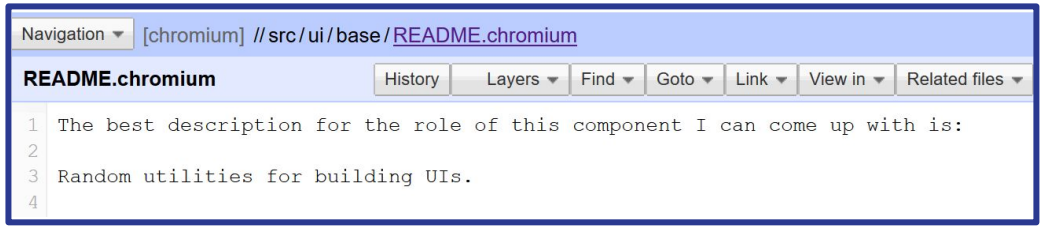
Owners of the code are listed only by emails

- Possible to search codebase for email

Comments on lots of main directories do not include README.md

- Certain README.md files are just not helpful

Understand software can be slow on such a huge codebase



Lessons Learned: **Process of Using Understand**

The Good: **What works**

- Familiar aesthetic interface
- Functional
  - Good visual representation of systems & subsystems

The Bad: **Inconveniences**

- Long wait times
- Workarounds needed
- Exporting/Importing

The Ugly: **Ruins functionality**

- Crashes and saving

Lessons Learned: **Our Team Issues**

Achievements:

- Regular meetings
- Individual work then discussion

Working on:

- Better concurrency
- Working consistently

Chrome-clusion

- Chrome is a very large piece of software making it hard to *Understand*
- Documentation was still hard to follow and dive through (occasionally no ReadMe's)
- Chrome has very high coupling within its Concrete Architecture
- Chrome's Concrete Architecture contains two more components than we previously thought
- More frequent group meetings