Chrome:

Conceptual Architecture

Bits... Please!

Agenda:

1. Introduction
2. Derivation
3. Conceptual Architecture
4. Subsystems
5. Use Case
6. Concurrency Model
7. Current Limitations/ Lessons Learned

Introduction to Chrome

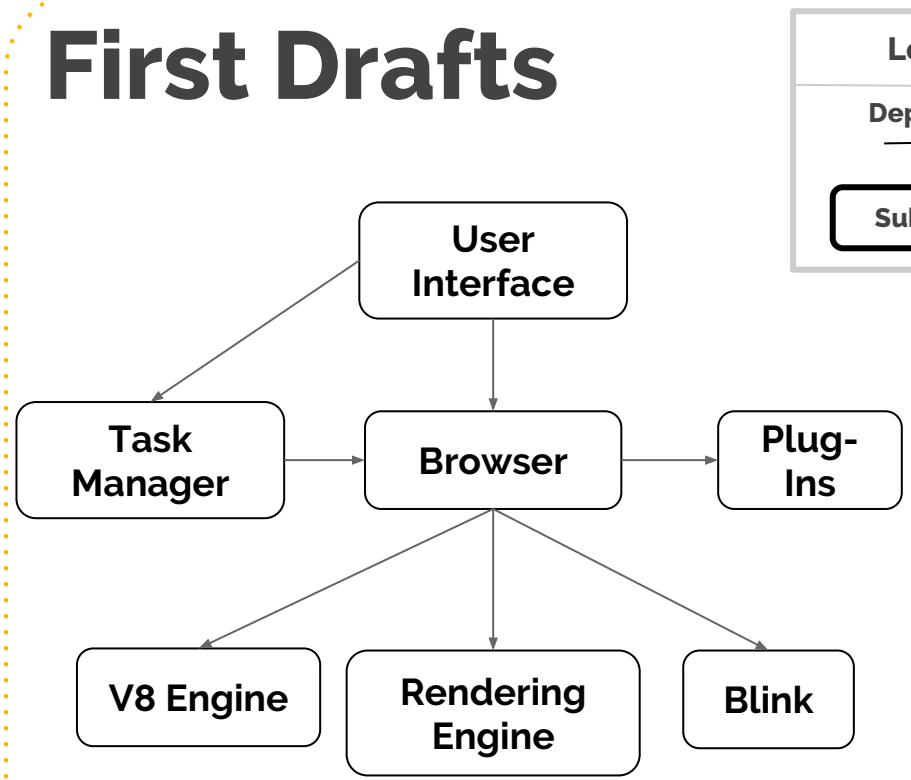
- First released in 2018
- Google built completely from scratch
- Open sourced
- Focused on the 4 S's : **simplicity, speed, security, stability**
- Multi-processor architecture

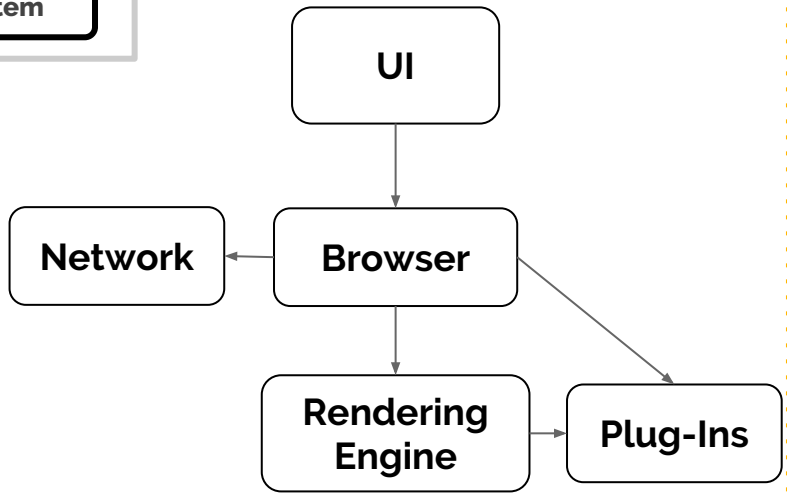
Market Share:

- September 2009: 5.38%
- September 2018: 60.63%

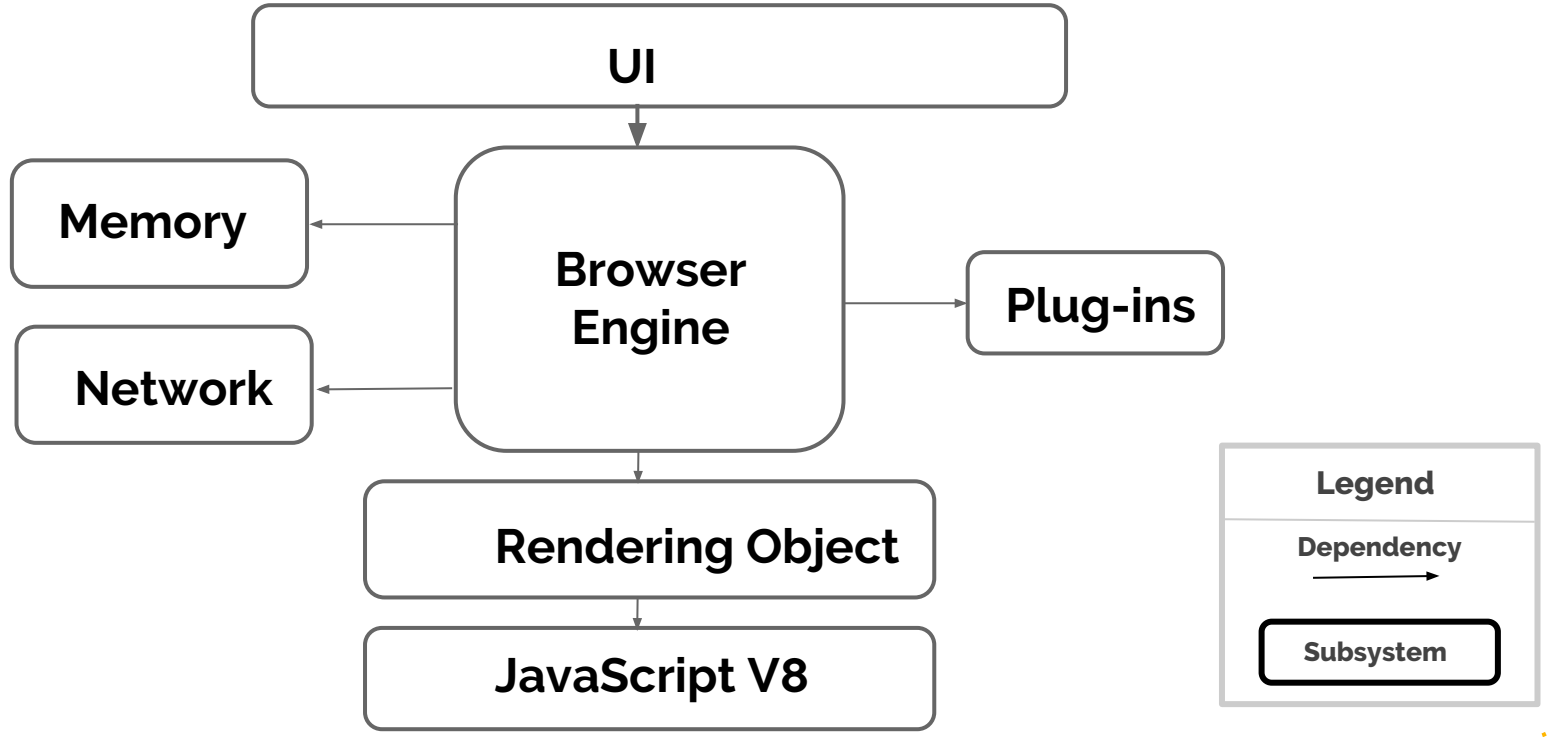
Derivation Process

First Drafts





Conceptual Architecture



Conceptual Architecture Goals

Create a conceptual architecture that will improve the **speed, stability, security, and simplicity of browsing the web**

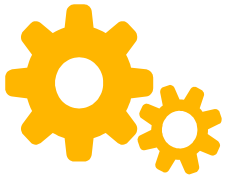
Speed

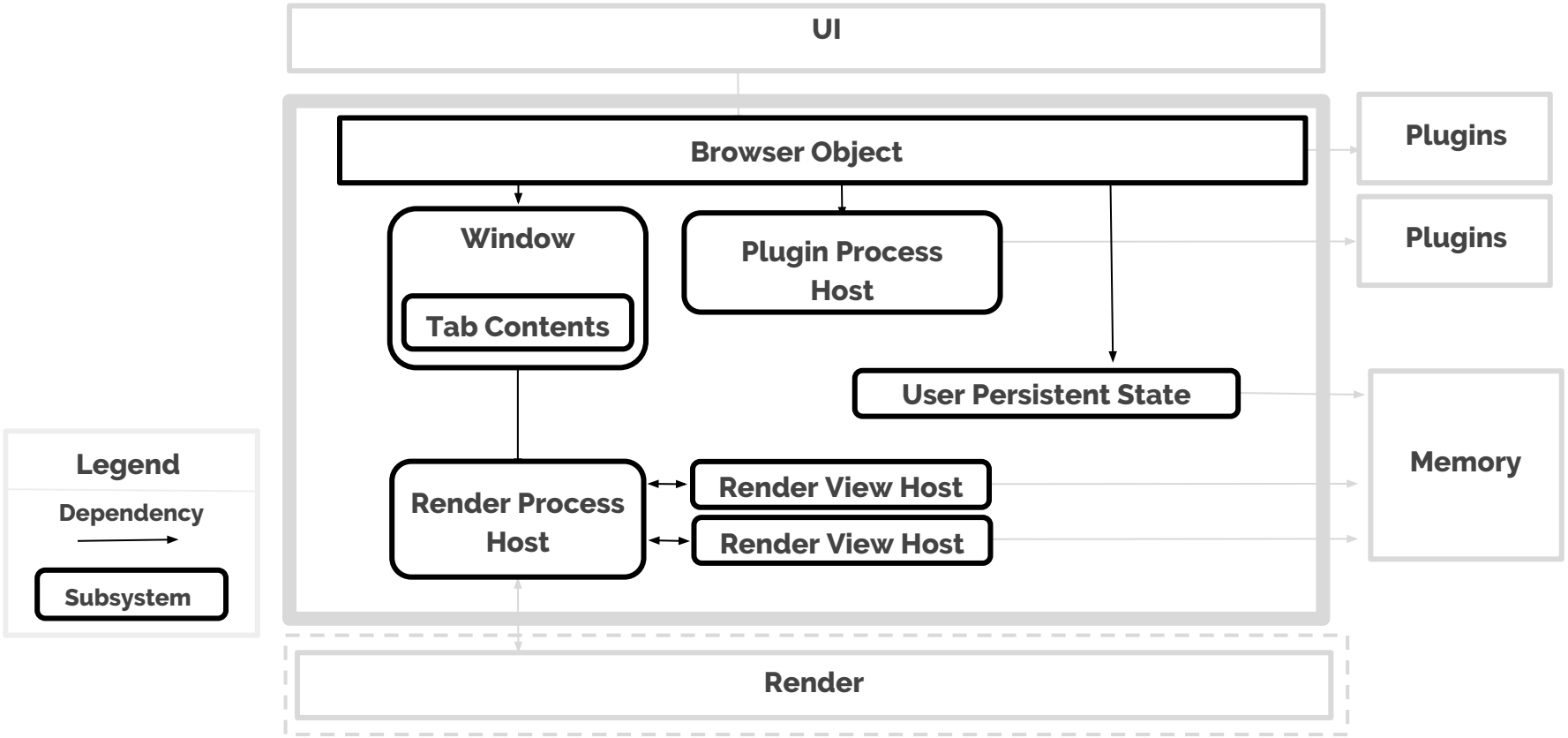
Stability

Security

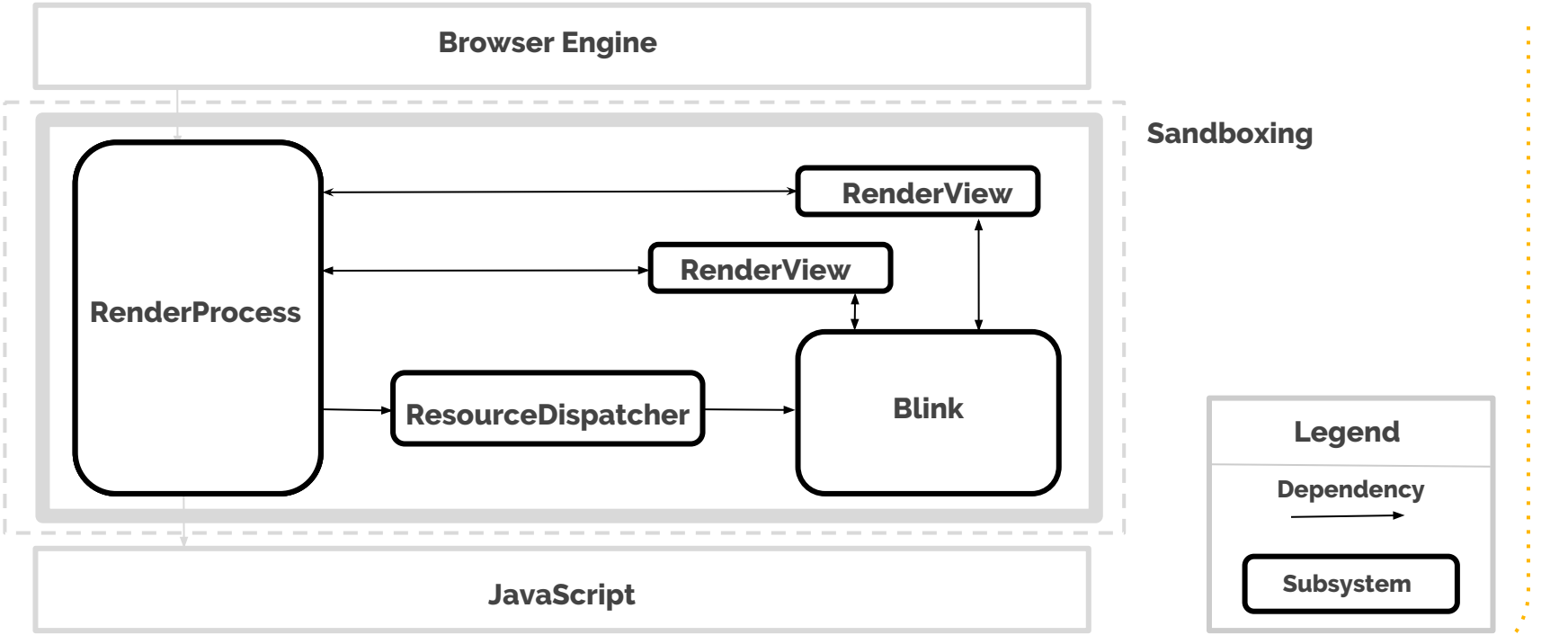
Simplicity

Subsystem: Browser Engine

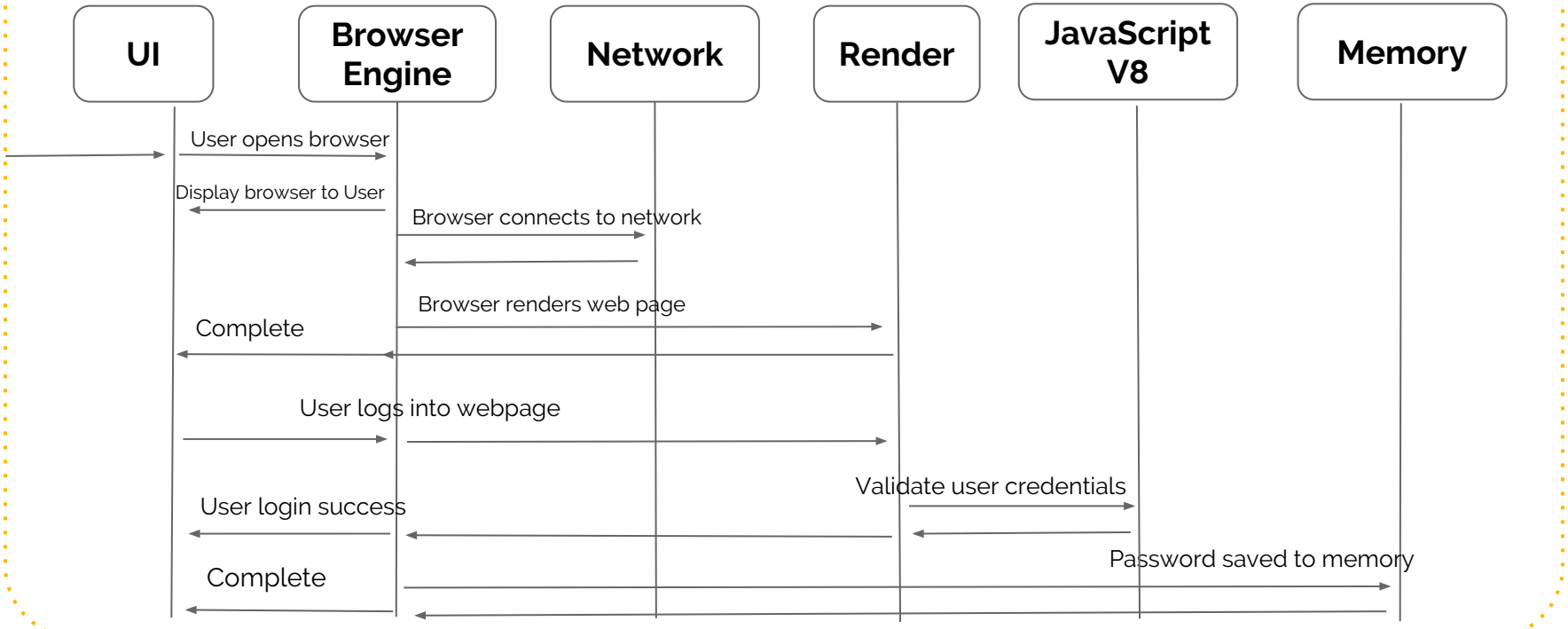




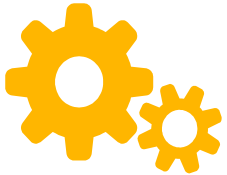
Subsystem: Render



Use Case 1


UI
Browser Engine
Network
Render
JavaScript V8
Memory
User opens browser
Display browser to User
Browser connects to network
Browser renders web page
Complete
User logs into webpage
Validate user credentials
User login success
Password saved to memory
Complete

Chrome's Concurrency Model



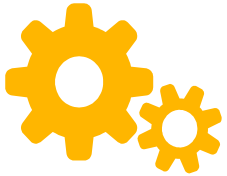
Multiprocess Architecture

- Each tab or plugin has its own process separate from the browser
- Helps protect against rendering failures

Supports multi-threading

- Main Thread
  - Browser Process: updates UI
  - Renderer Process: runs the rendering engine (Blink)
- IO Thread
  - Browser Process: handles the IPCs and network requests
  - Renderer Process: handles the IPCs

Chrome's Concurrency Model



Communication between processes

- Chromium IPC ~ legacy system
- Mojo: message pipes

Implications

- More memory upfront
- Reduces bloat in the long run

Current Limitations and Lessons Learned

Current Limitations

- Not very much high level documentation
- Required quite a bit of research and understanding

Lessons Learned

- Communication
- Set deadlines

Team Issues within Chrome

- Layering Violations
- Dependencies
- Cross Platform Browser System

Conclusion

- Multi-processor architecture
- Layered Architecture at a high level
- Object-oriented and layered at lower levels

Thanks!

Any questions?