

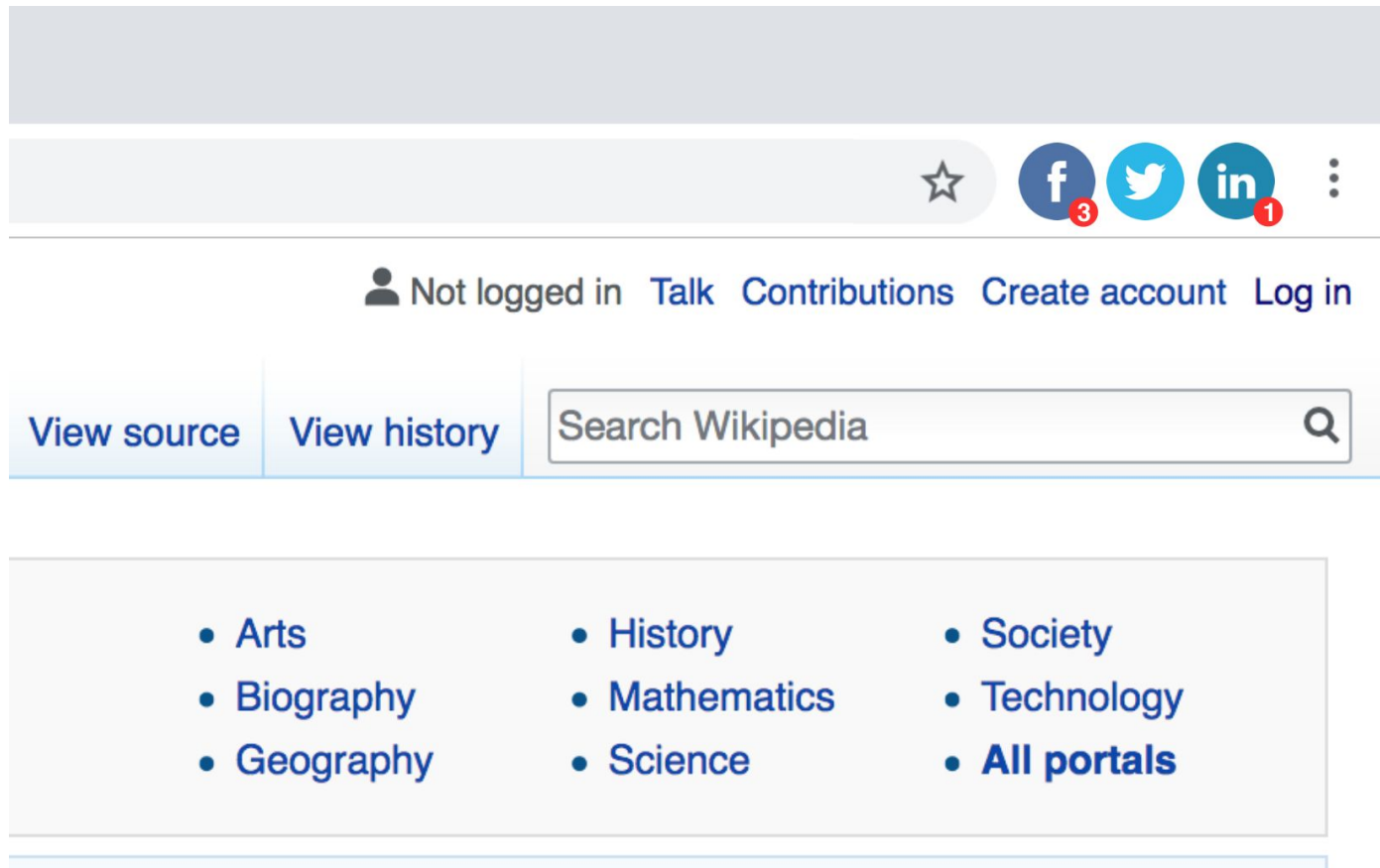
Social Media Hub Feature



The Koalas:

Scarlett Taviss
Aawista Chaudhry
Rita Alsattah
Matthew Eliot
Alexander Salgo
Emily Kaczmarek

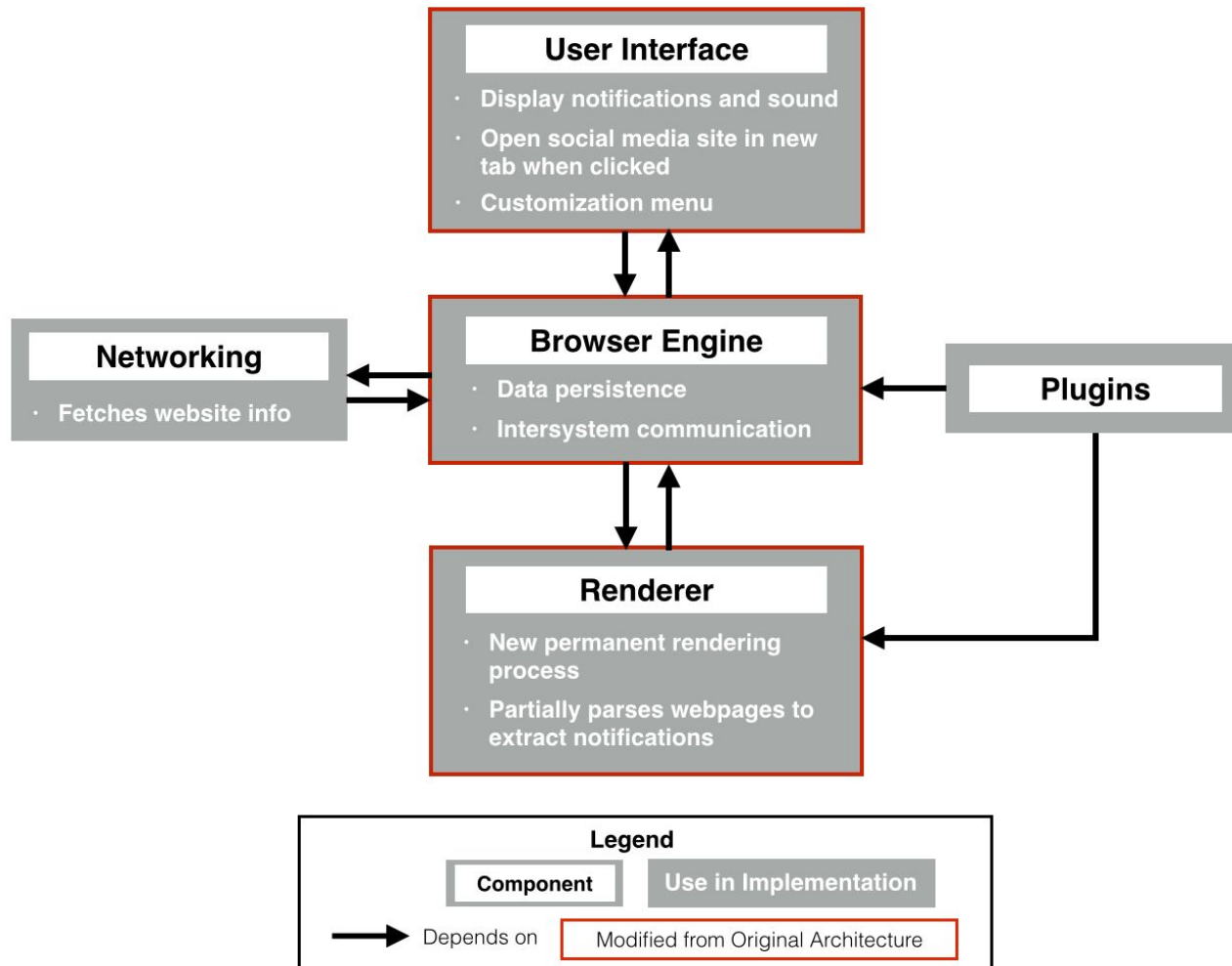
Feature: Social Media Notification Icons



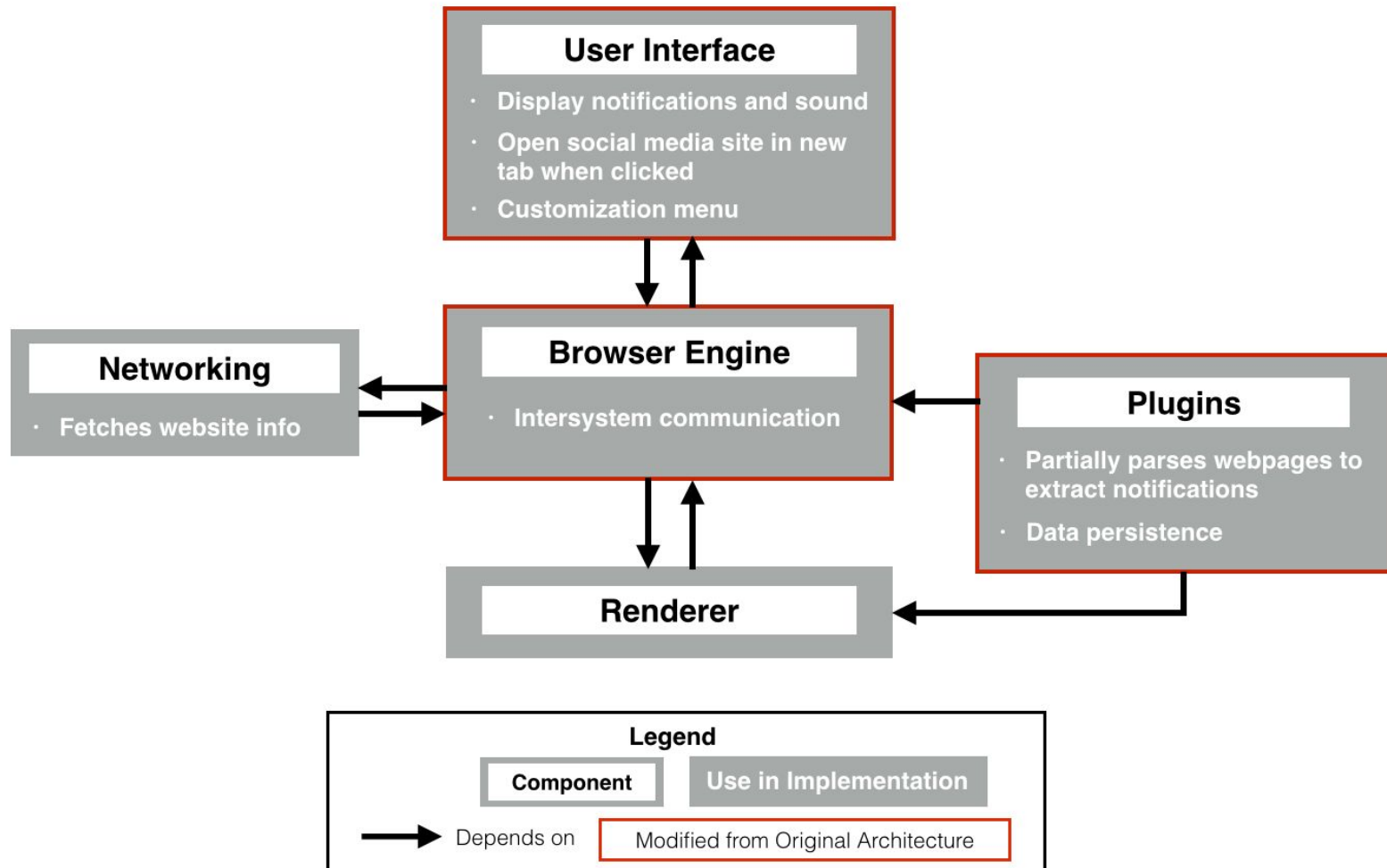
SAAM - Scenarios

1. Display notifications for multiple different social media accounts
2. Access websites every 10 seconds to update notifications
3. User personalizes which social media accounts to display
4. Click icon to open relevant social media website in new tab
5. Play sound when new notification is received

SAAM - Implementation 1



SAAM - Implementation 2



SAAM - Stakeholders & NFRs

Quality Attributes:

- Maintainability
- Evolvability
- Testability
- Performance
- Security

Stakeholders:

- Dev Team
- Chrome
- Users

SAAM - Testing the NFRs

Maintainability: *“99% of updates can be performed without any interruption to user experience.”*

Evolvability: *“New information can be interpreted and displayed from a supported website by a junior developer.”*

Testability: *“Tests run when opening supported social media sites do not result in an increase in page load time of more than 50 ms.”*

Performance: *“95% of newly received notifications result in an icon update within 10 seconds.”*

Security: *“No information about the partially rendered social media sites is unsecured during notification check portion.”*

SAAM - Advantages/Disadvantages

QA		Model 1	Model 2
<i>Maintainability</i>	C	Update = difficult, code distributed within subsystems	A Update = easy, most code within Plugin, less coupling with Chrome
<i>Evolvability</i>	B+	Existing rendering system → easier to interpret/render	C Partial rendering subsystem → may require significant changes
<i>Testability</i>	A	Reuse renderer processes for page loading, sending notification info, etc.	B Page rendered <i>and</i> partially rendered when testing occurs
<i>Performance</i>	B+	Additional renderer process: ↑ overhead (slightly)	A No major issues expected
<i>Security</i>	A	Renderer = sandboxed, only notification info sent outside	F Pages rendered in non-sandboxed environment → major security risk

Impacted Subsystems

Maintains object-oriented architecture style.

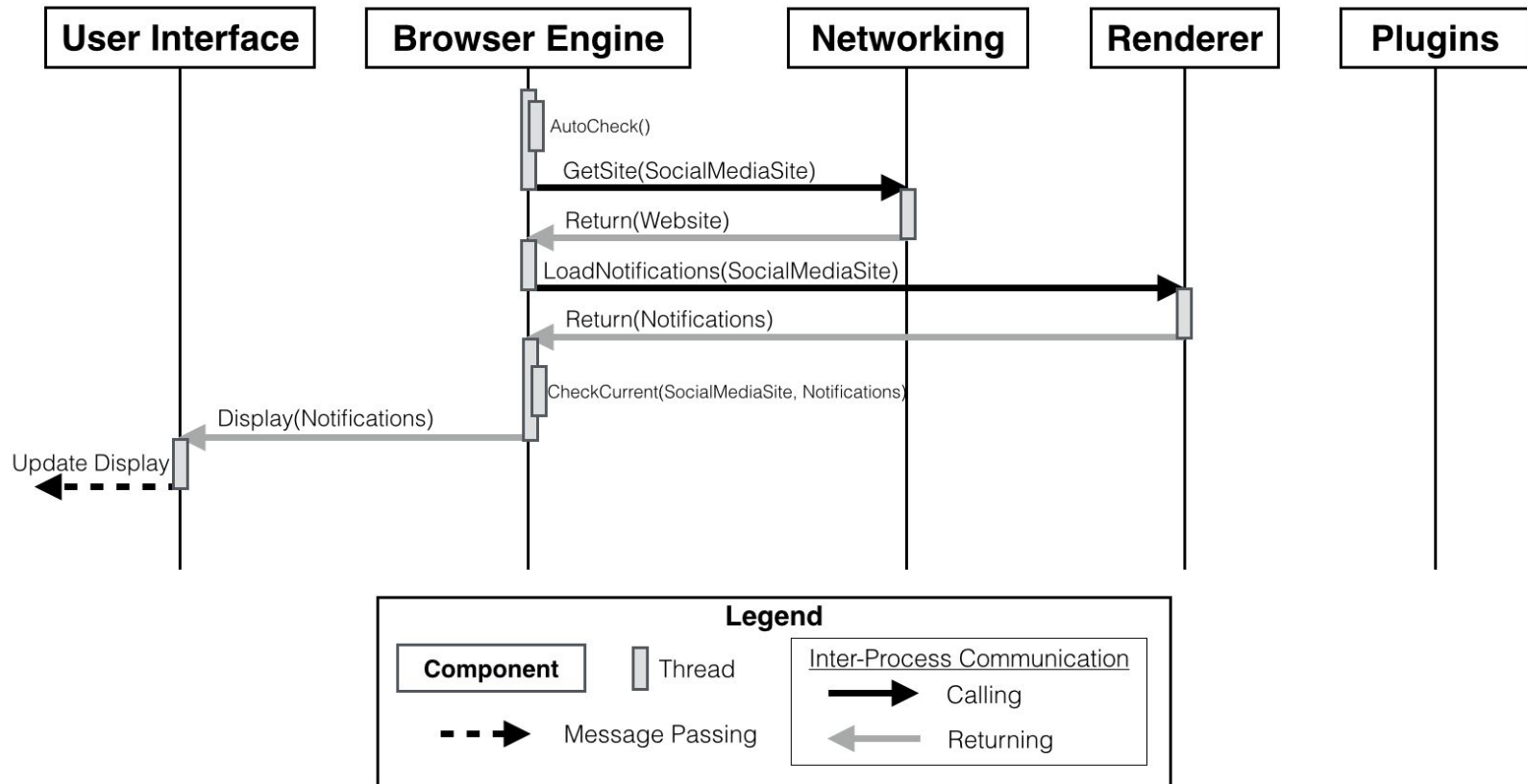
User Interface - Used to display the social media notifications.

Browser:

- Performs auto checks on each social media site, calling to the renderer every 10 seconds.
- Compares and updates notifications

Renderer - Retrieves the notifications from the social media site.

Sequence Diagram



Risks/Limitations of Approach

- Difficult to update
- Choking the Network
- Increase in resources used
- Reduced concurrency

Concurrency

- Multi-process architecture
- Reduces system concurrency by using a renderer
 - Larger impact on smaller machines
- Multiple threads
 - Allows for testing

Testing Impact of Interactions

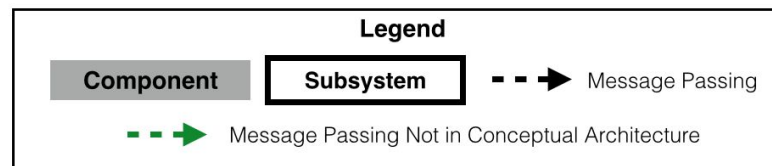
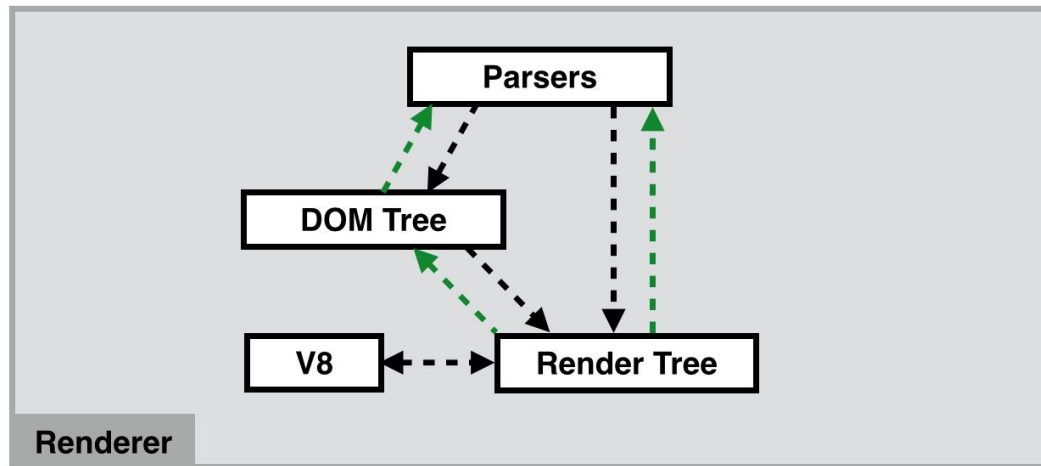
- User receives notification:
 - Notification appears on screen
 - Only one icon is updated
- User clicks notification:
 - New tab is opened
 - Notification number matches webpage
- User updates social media preferences

Team Issues

Core team to implement feature: Renderer

Secondary team: DOM

Teams to Consult: UI, Browser



Concrete Architecture of Renderer Subsystem

Limitations & Lessons Learned

- Differentiating between quality attributes
(Maintainability vs Evolvability)
- Implementing a new feature in Chrome requires a lot of consideration and testing
- Thinking like a developer: SAAM, Testing, Implementation

Conclusion

- Motivation: Enhanced user experience
- SAAM Analysis of 2 possible implementations helped us determine final approach
- Impacted subsystems: UI, Browser, Renderer
- Immense amount of considerations taken in developing a new feature in Chrome

Questions?