Example problem

Let’s walk through a possible concurrent solution

The statement

- We need to make a ***browser*** (original idea… right?)
  - The browser needs access to many remote resources (HTML pages, images, etc…)
  - We can’t block the operations invoked by the user
  - The user wants to have a system that is ALWAYS responsive
  - The browser needs to accommodate plugins and satisfy all the “working requirements”

What could be the components we need?

(Answer)

What could be the components we need?

Browser interface

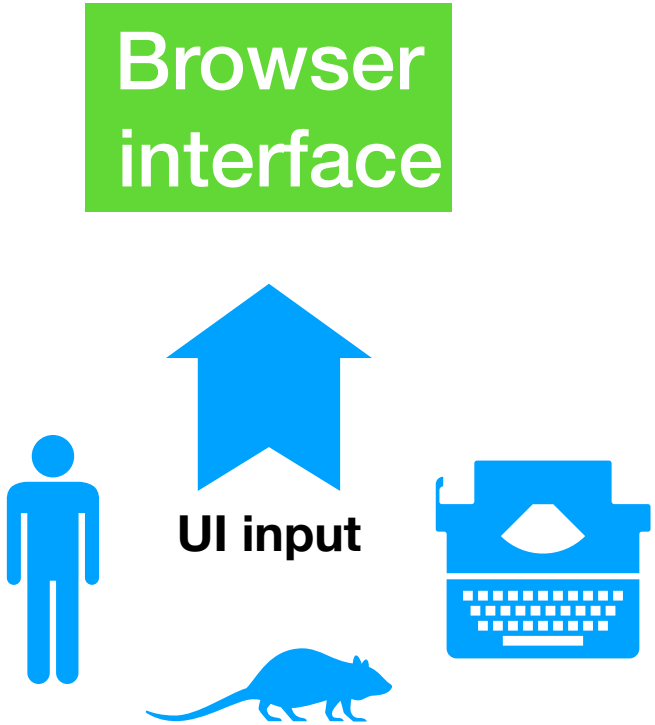
Fetching mechanism for remote resources

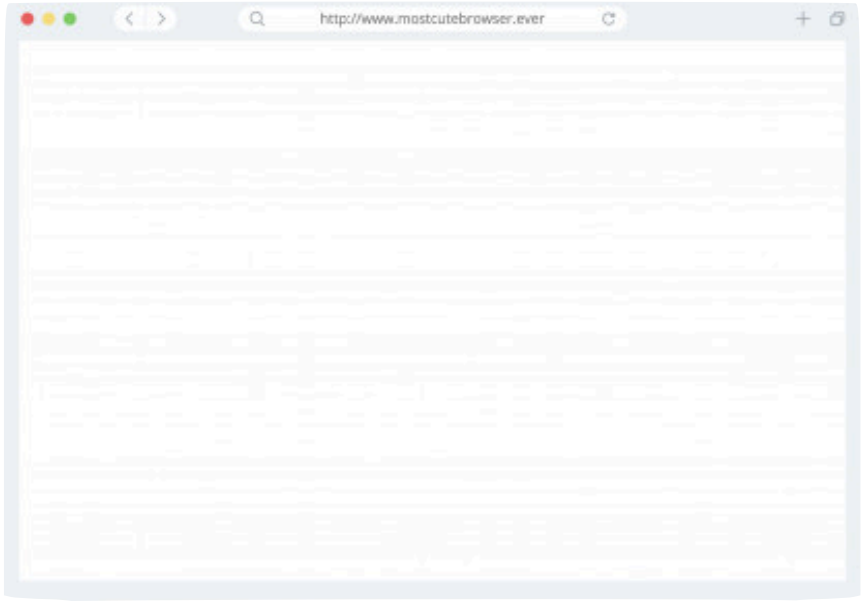
Plugins

Do we need more?

Ok… browser interface

- Needs to be responsive
- Needs to render the resources fetched from other components

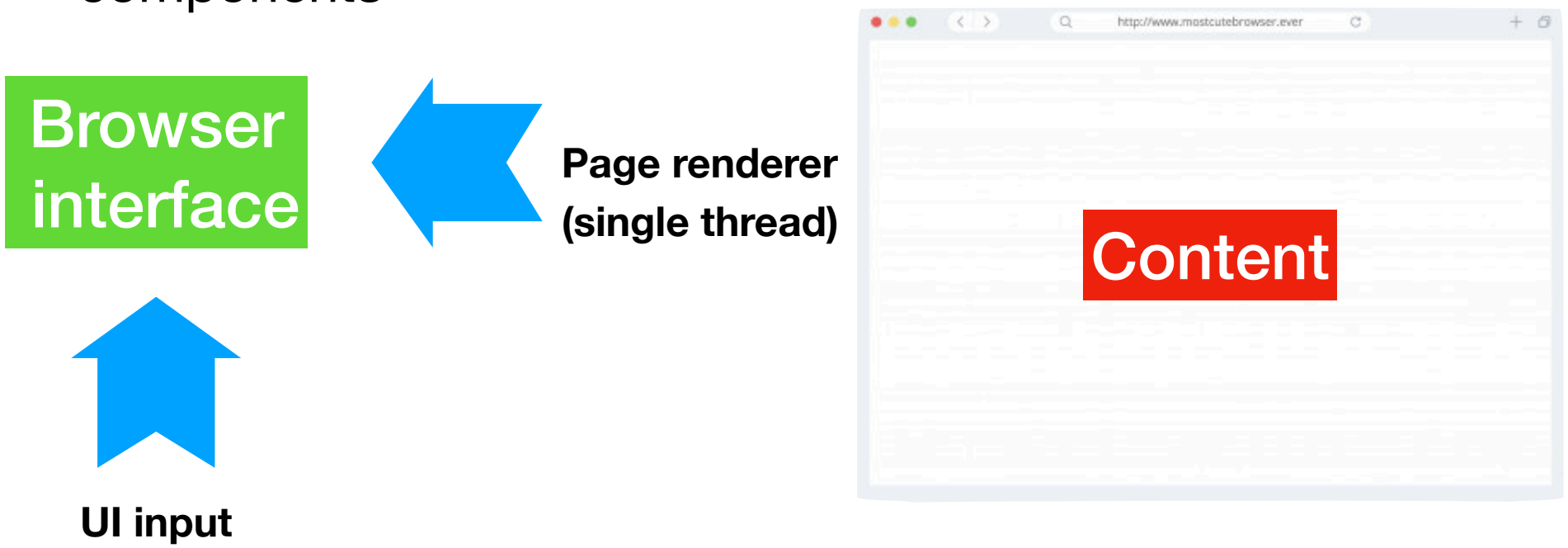




Who is writing inside this window?

Browser interface …
continued

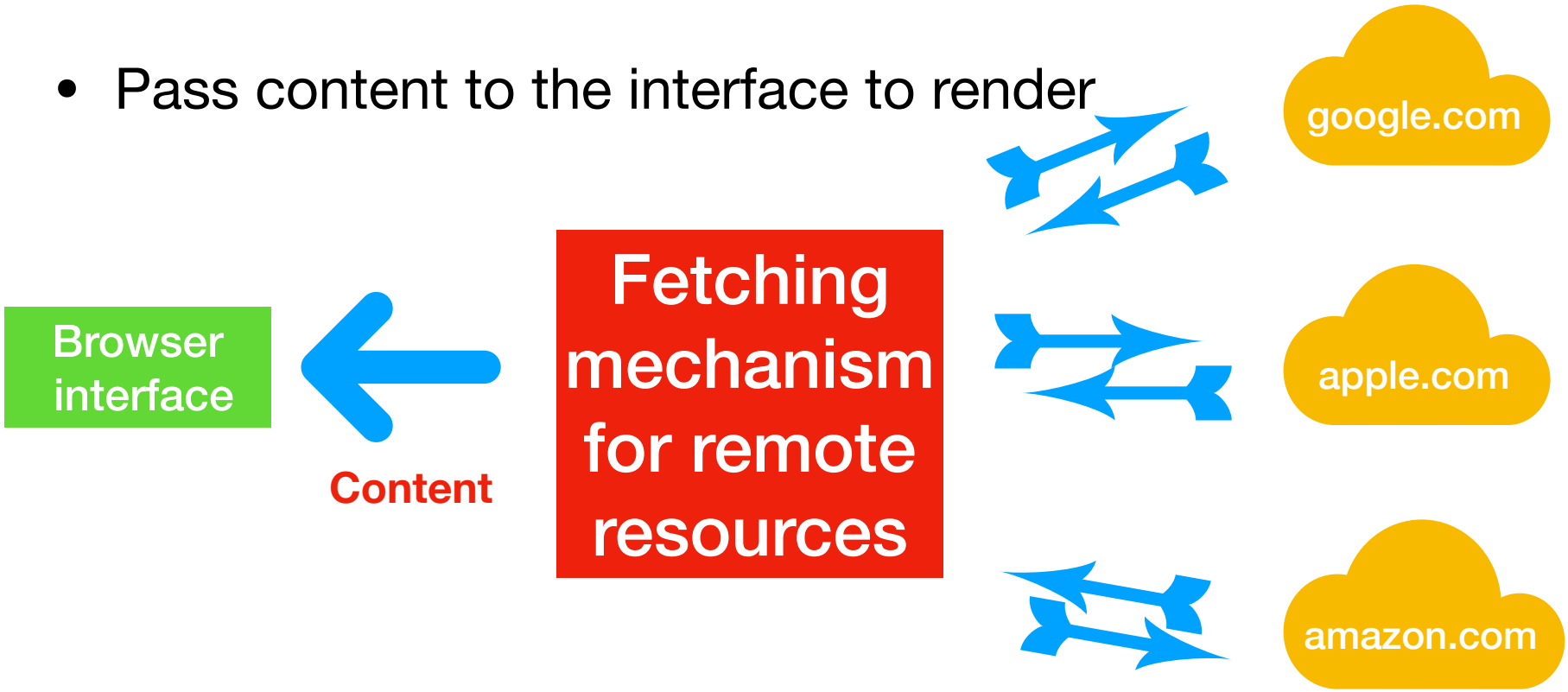
- Needs to be **responsive**
- Needs to render the resources fetched from other components



Who is writing inside this window?

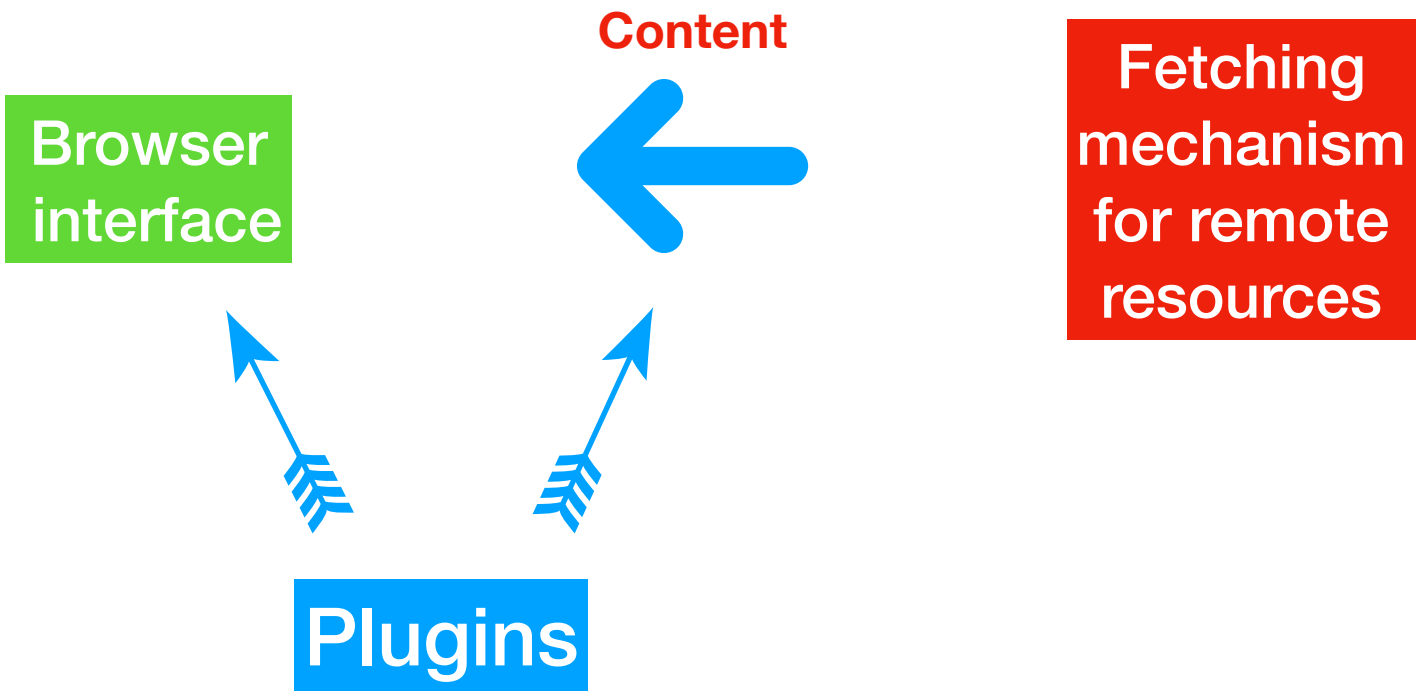
Fetching module

- Needs to fetch the info from remote sources (**many**)
- Pass content to the interface to render



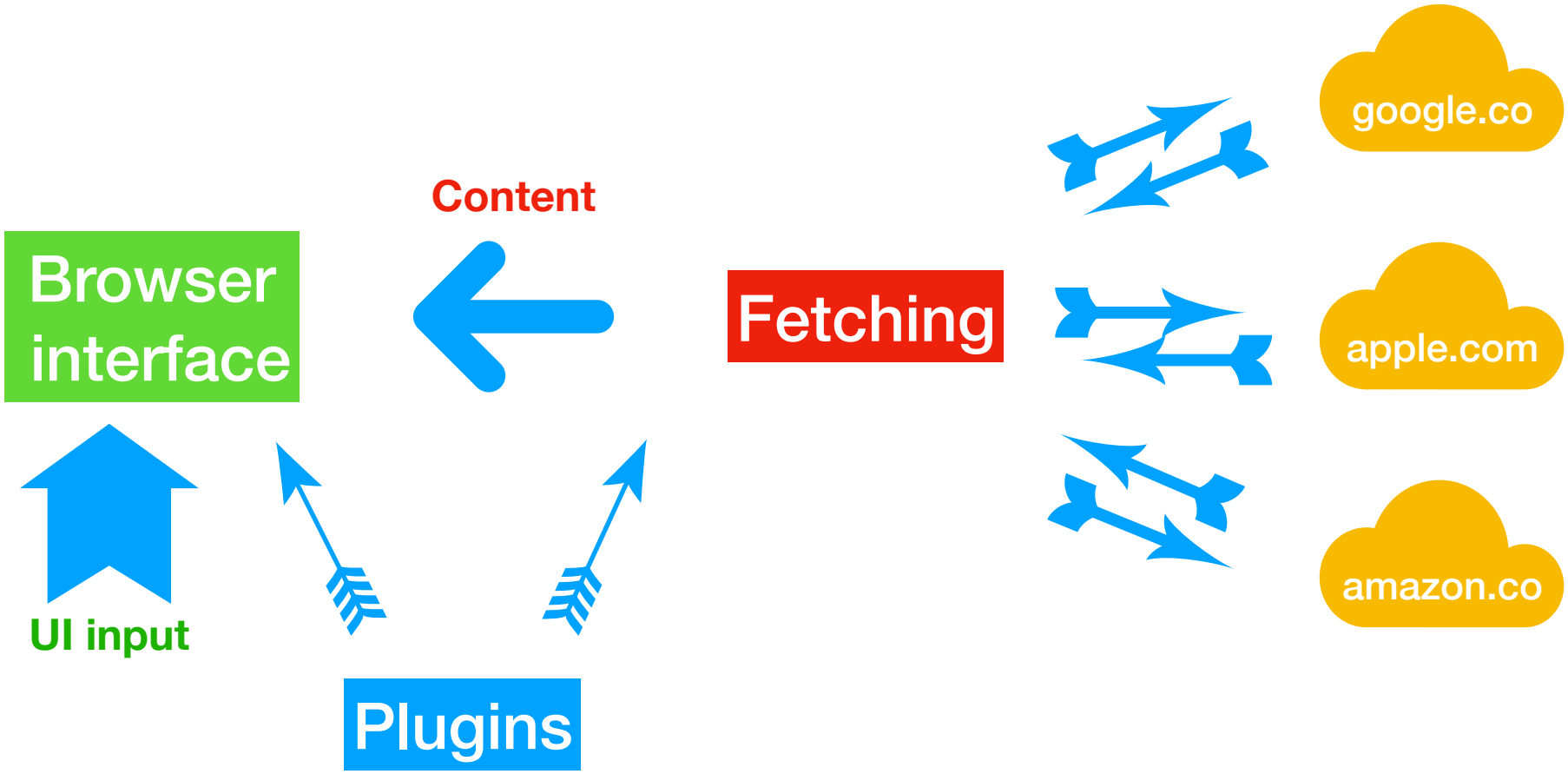
Plug-ins module

- Needs to **access** content and the interface
- Can **alter** content and interface



All together now…

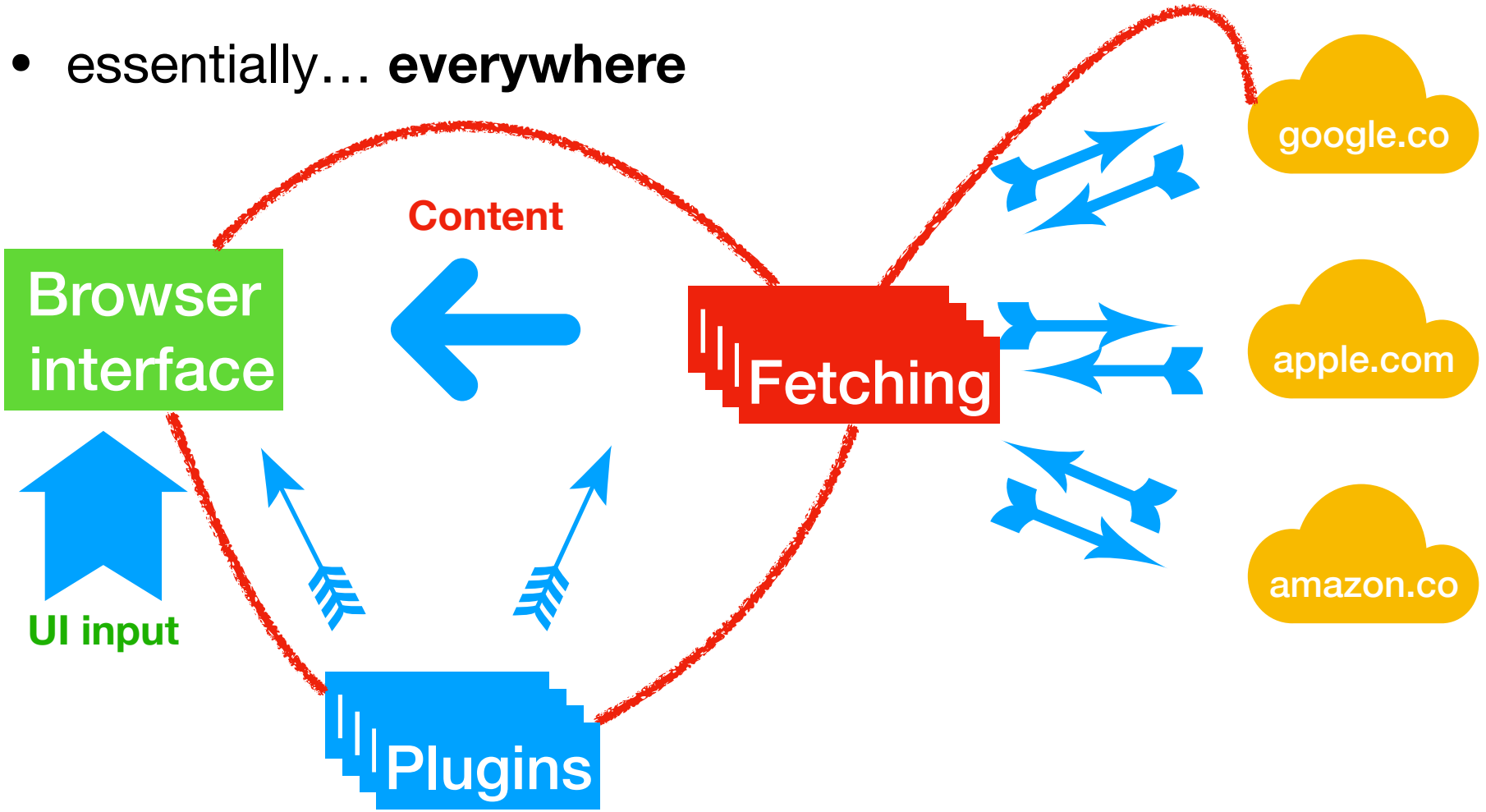
- Can you address the **concurrent** challenges?



(Answer)

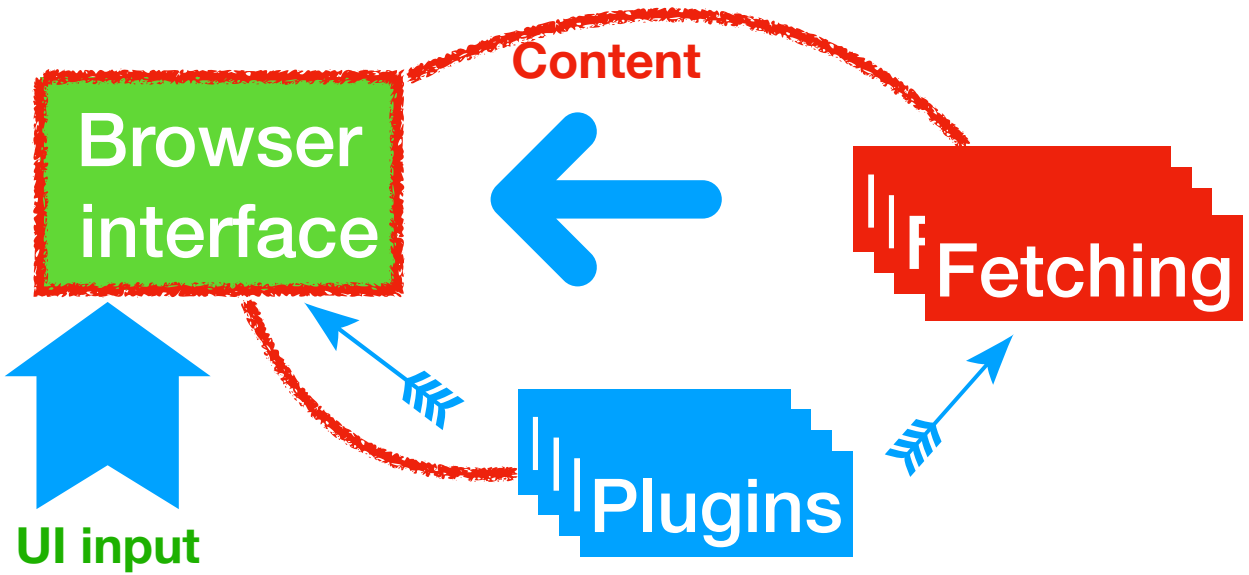
All together now…

- Can you address the **concurrent** challenges?
- essentially… **everywhere**



Let’s get some detailed problem solving…

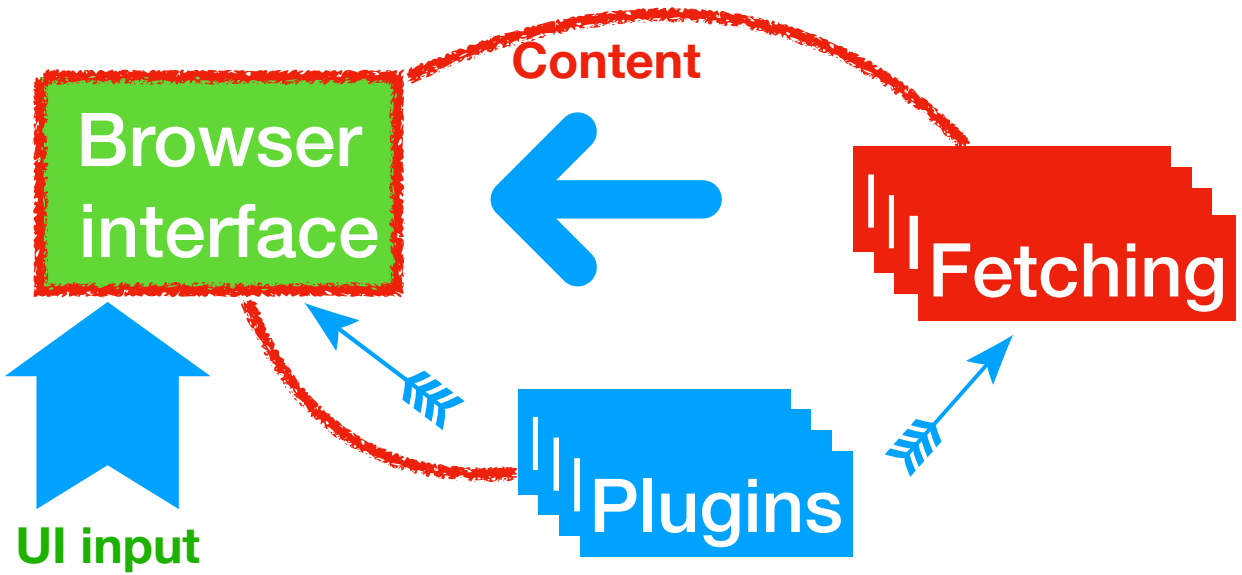
- ONLY one thread can access the rendering, Solution?



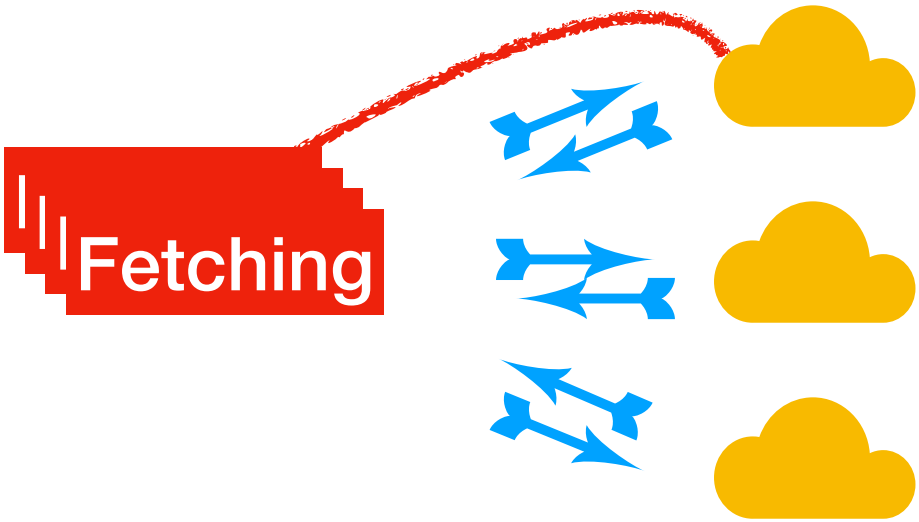
(Answer)

Asynchronous calls to render the content

- ONLY one thread can access the rendering, Solution?
- **Asynchronous** calls to the main thread (or deliver a message and leave it to go…, go ahead for your work) and queue whatever else is needed for that.
  - Message passing manager?



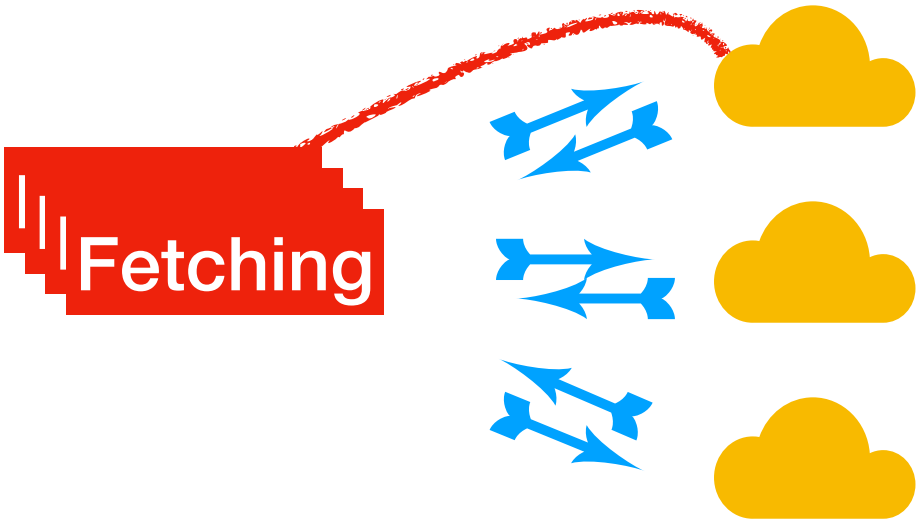
Remote fetching



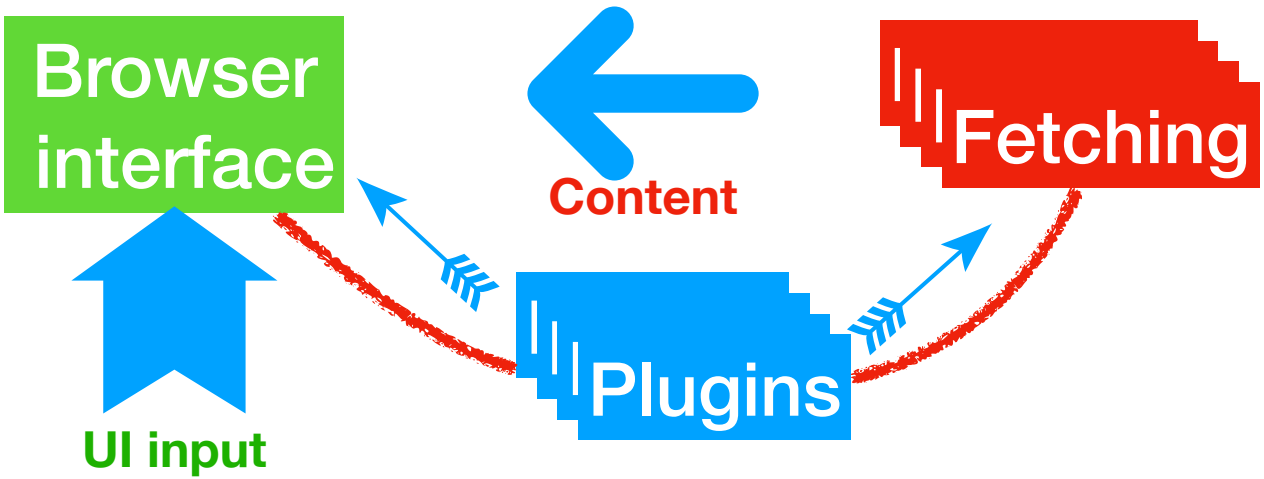
(Answer)

Synchronous blocking

- Fetching takes remote resources
- We can make a thread for each resource (blocking calls to the resource), but each thread is synchronous to the resource, so it is stuck there.
- And wait from the other side with **async** calls.



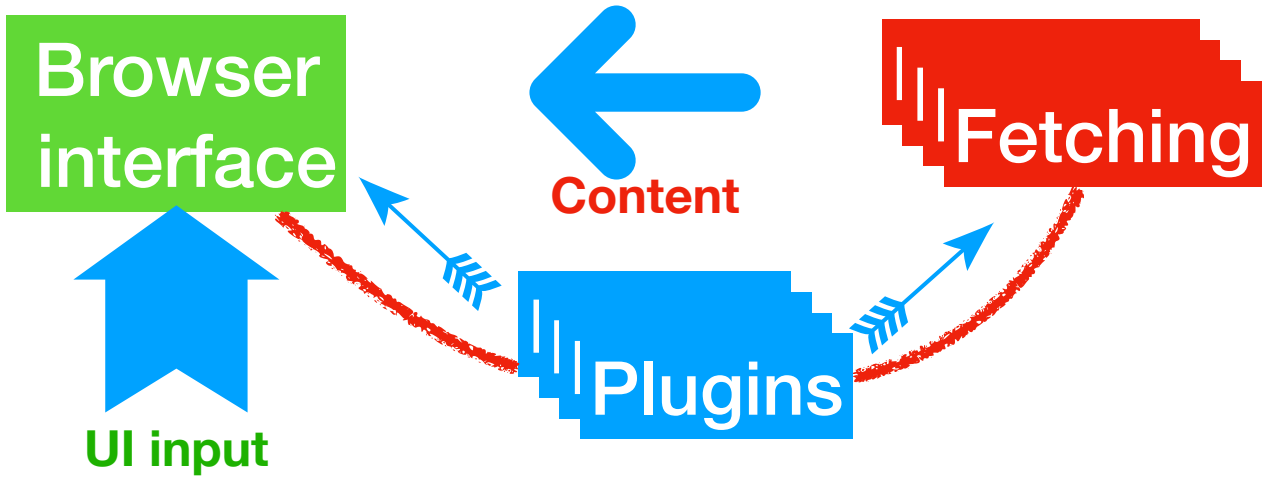
Plugins access



(Answer)

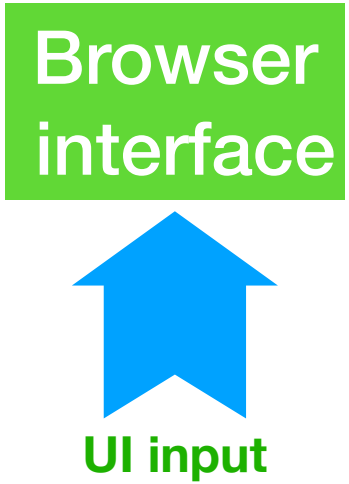
Locking the content

- Multiple plugins can change/access the content of the page (or generate a page).
- The plugins must access concurrently with other plugins.
- The browser interface must put a lock/mutex for sharing the information of the rendered content



UI concurrency

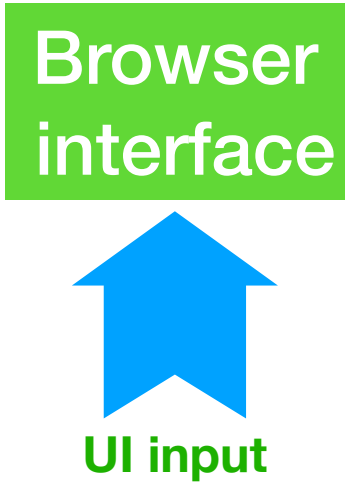
- The browser must wait for:
  - The plugins
  - The content to fetch
  - UI interrupts
- Solution?



(Answer)

UI concurrency

- The browser must wait for:
  - The plugins
  - The content to fetch
  - UI interrupts
- Solution?
  - All asynchronous calls (call back systems for buttons, async calls for content and plugin, etc…)



Let’s review some exercises….

- NOW :P