

# Delay Clipboard Rendering

Anupam Snigdha

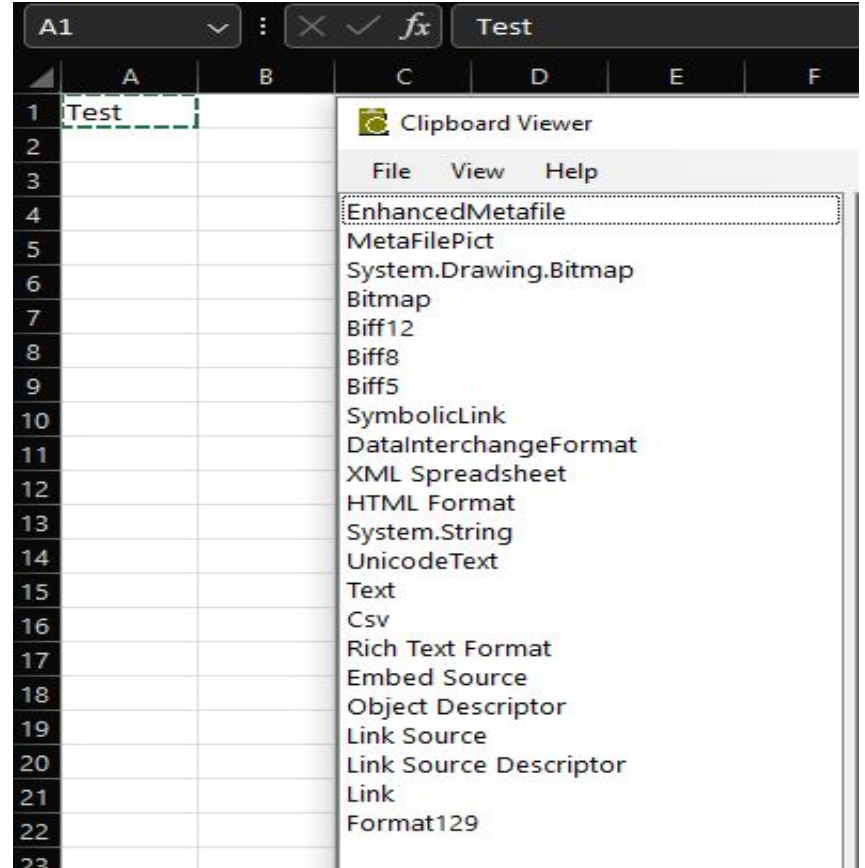
Microsoft

BlinkOn 18

10/19/2023

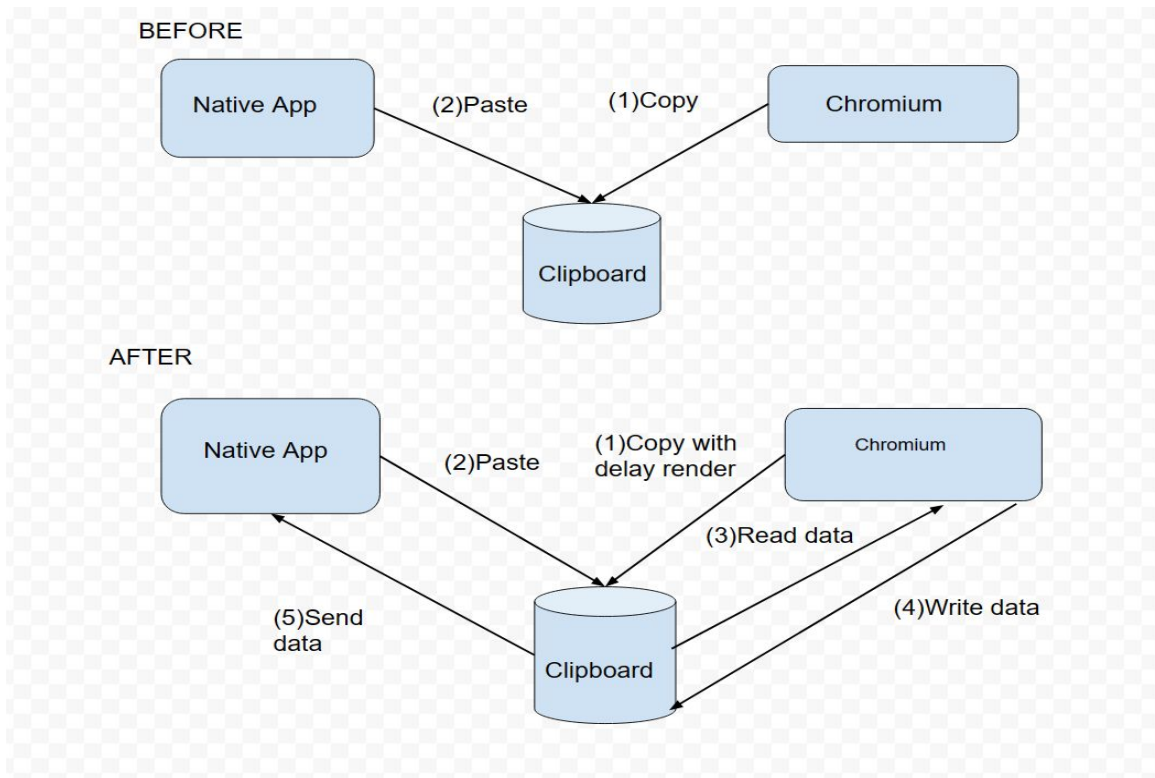
# Problems

1. Not all clipboard format are used.
2. Formats can be expensive to generate.
3. Chromium may support more built-in formats.



# Overview

Delayed clipboard rendering is the ability to delay the data generation of a copied clipboard format until it is needed by the target applications during paste.



# Proposal

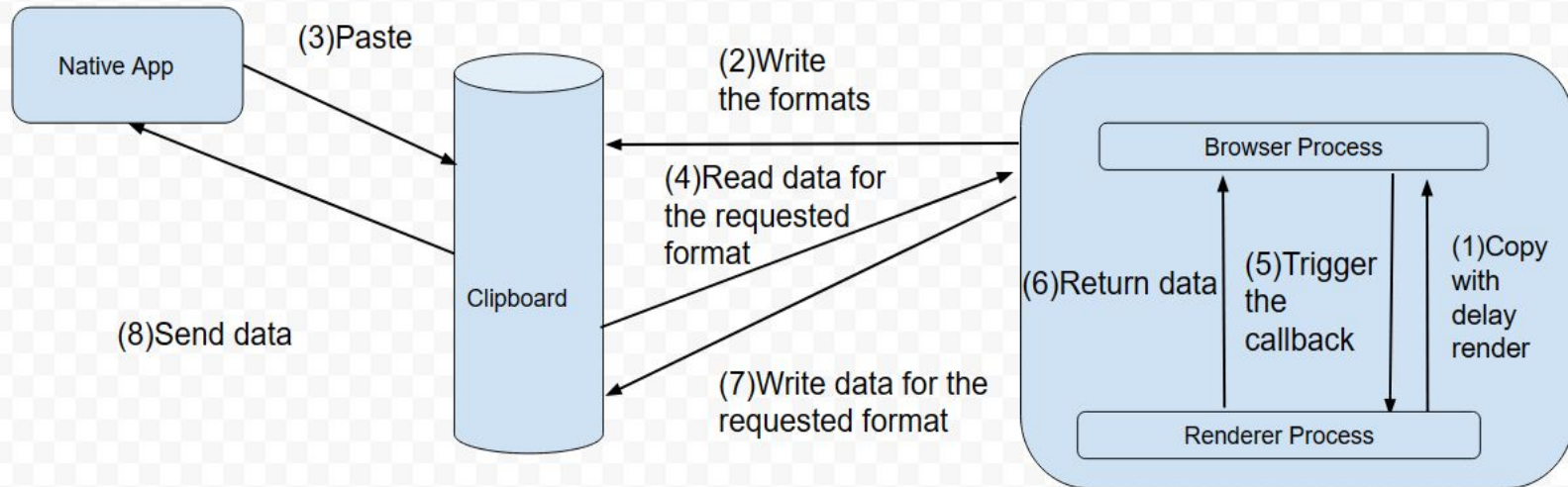
Web API changes in clipboard\_item.idl:

```
callback DelayedBlobCallback = Blob();  
typedef Promise<Blob> PromiseBlob;  
typedef (PromiseBlob or  
DelayedBlobCallback) ClipboardItemData;  
...  
constructor(record<DOMString,  
ClipboardItemData> items);  
...
```

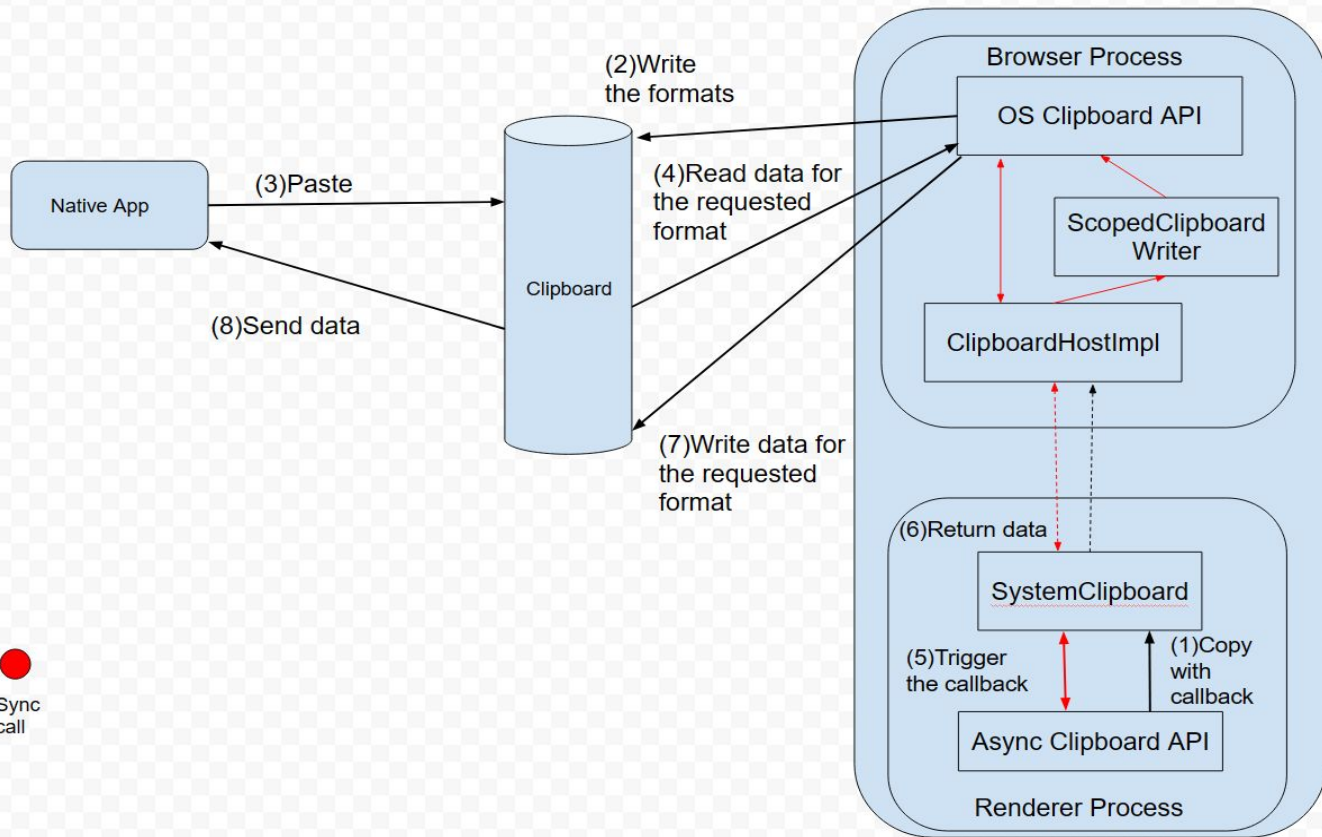
## JavaScript Example

```
const textBlob = new Blob(['Hello, World!'], {type:  
  'text/plain'});  
  
const clipboardItemInput = new  
ClipboardItem({'text/plain': Promise.resolve(textBlob),  
  'web application/custom-format':  
    generateExpensiveBlob});  
  
navigator.clipboard.write([clipboardItemInput]);  
  
function generateExpensiveBlob() {  
  var inputData = ...;  
  return new Blob([inputData], {type: 'web  
application/custom-format'});  
}
```

# Delay clipboard rendering message flow

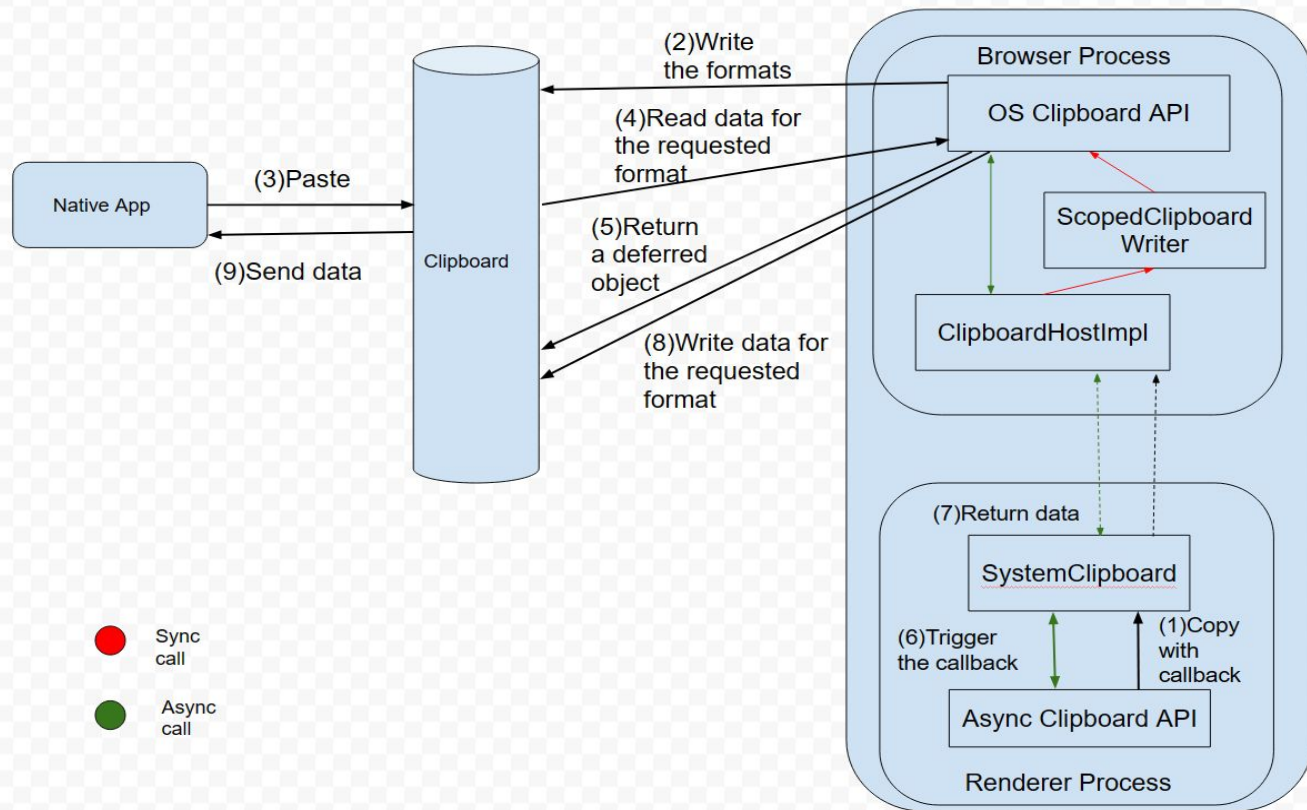


# Option 1: Synchronous read/write.



# Option 2: Asynchronous read/write.

No delay rendering on platforms that don't have support for async data population.



# Tab/Browser close

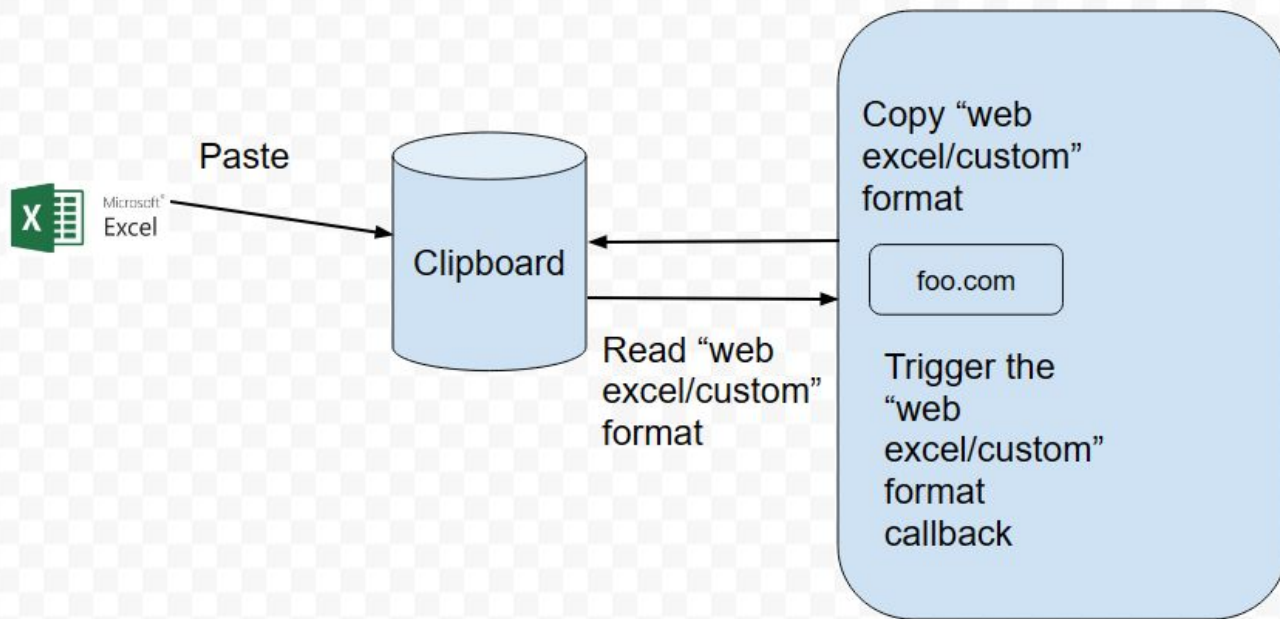
- A format is delay rendered and user closes the Browser/Tab or navigates away from the page.
- Problem: Data loss on paste
- Solution: One built-in format needs to be populated upfront.
- More details -

<https://docs.google.com/document/d/1H6ow7RWa4MeycKP3OQBoMhLuMaJct3jSCWW6apcVu9U/edit?usp=sharing>



# Privacy concern

Web custom format exposes the target app where the user is pasting the data.



Possible mitigation:

Support delay rendering of only the built-in formats.

- Makes the API less useful.

# Contact & Github issues

Email: [snianu@microsoft.com](mailto:snianu@microsoft.com)

IDL proposal: <https://github.com/w3c/editing/issues/423>

Feedback: <https://github.com/w3c/editing/issues/417>

Credit:

Thanks to Josh Bell, Evan Stade, Sanket Joshi and our partners for the feedback and helping with the design.