



View Transitions

a.k.a. Shared Element Transitions



Overview

- Quick intro ([Jake Archibald's S.E.T. I/O Talk](#))
- Implementation model
- A couple of demos

What the majority of navigation look like

- Click to navigate
- Maybe wait a bit
- Old content is instantly replaced with new content



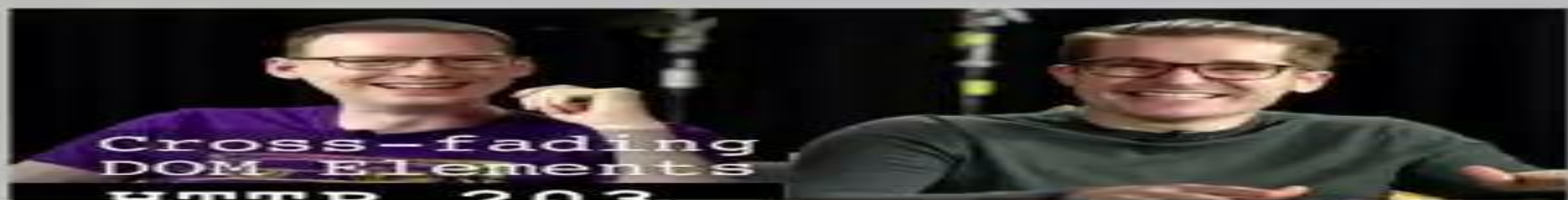
HTTP 203



2022-03-29



2022-03-15

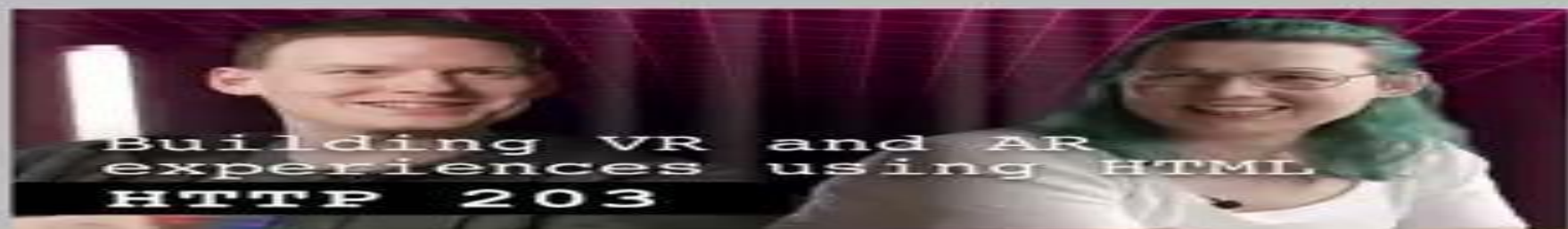




HTTP 203



2022-03-29



2022-03-15



```
async function changeContent() {  
    /* Do the necessary DOM mutations to "navigate" */  
}  
  
async function clickHandler() {  
    await changeContent();  
}  
  
thumbnail.addEventListener("click", clickHandler);  
  
/* You can also use navigation API to handle back navigation  
   and forward navigation, but it's outside of the scope of  
   this talk */
```

```
async function deprecatedClickHandler() {  
    await changeContent();  
}
```

```
async function experimentalClickHandler() {  
    let t = document.startViewTransition(changeContent);  
    await t.finished;  
}
```

Pseudo DOM Structure

- **html::view-transition** – view transition root “element”
 - **::view-transition-group(root)** – grouping element with transform and sizing
 - **::view-transition-image-pair(root)** – effect isolation element
 - **::view-transition-old(root)** – replaced element with old content snapshot
 - **::view-transition-new(root)** – replaced element with new content snapshot
- Default UA stylesheet
 - Creates some default animations


```
<style>
```

```
::view-transition-old(root) ,
```

```
::view-transition-new(root) {
```

```
    animation-duration: 5s;
```

```
}
```

```
</style>
```

```
async function experimentalClickHandler() {
```

```
    let t = document.startViewTransition(changeContent);
```

```
    await t.finished;
```

```
}
```


view-transition-name

```
<style>
.header {
  view-transition-name: header;
  contain: layout; /* or contain: paint */
}

.header-text {
  view-transition-name: header-text;
  contain: layout;
}

</style>
```

header-text

header

root



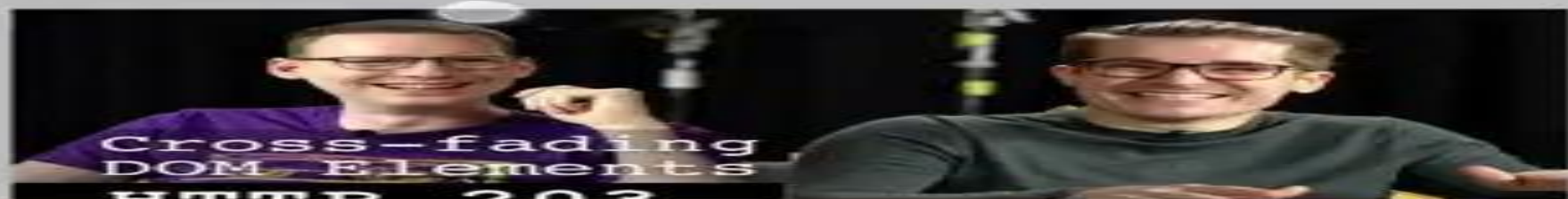
HTTP 203



2022-03-29



2022-03-15



Pseudo DOM structure

- **::view-transition**
 - **::view-transition-group(header-text)**
 - **::view-transition-image-pair(header-text)**
 - **::view-transition-old(header-text)**
 - **::view-transition-new(header-text)**
 - **::view-transition-group(header)**
 - **::view-transition-image-pair(header)**
 - ...
 - **::view-transition-group(root)**
 - ...

main thread

compositor thread

GPU/viz process

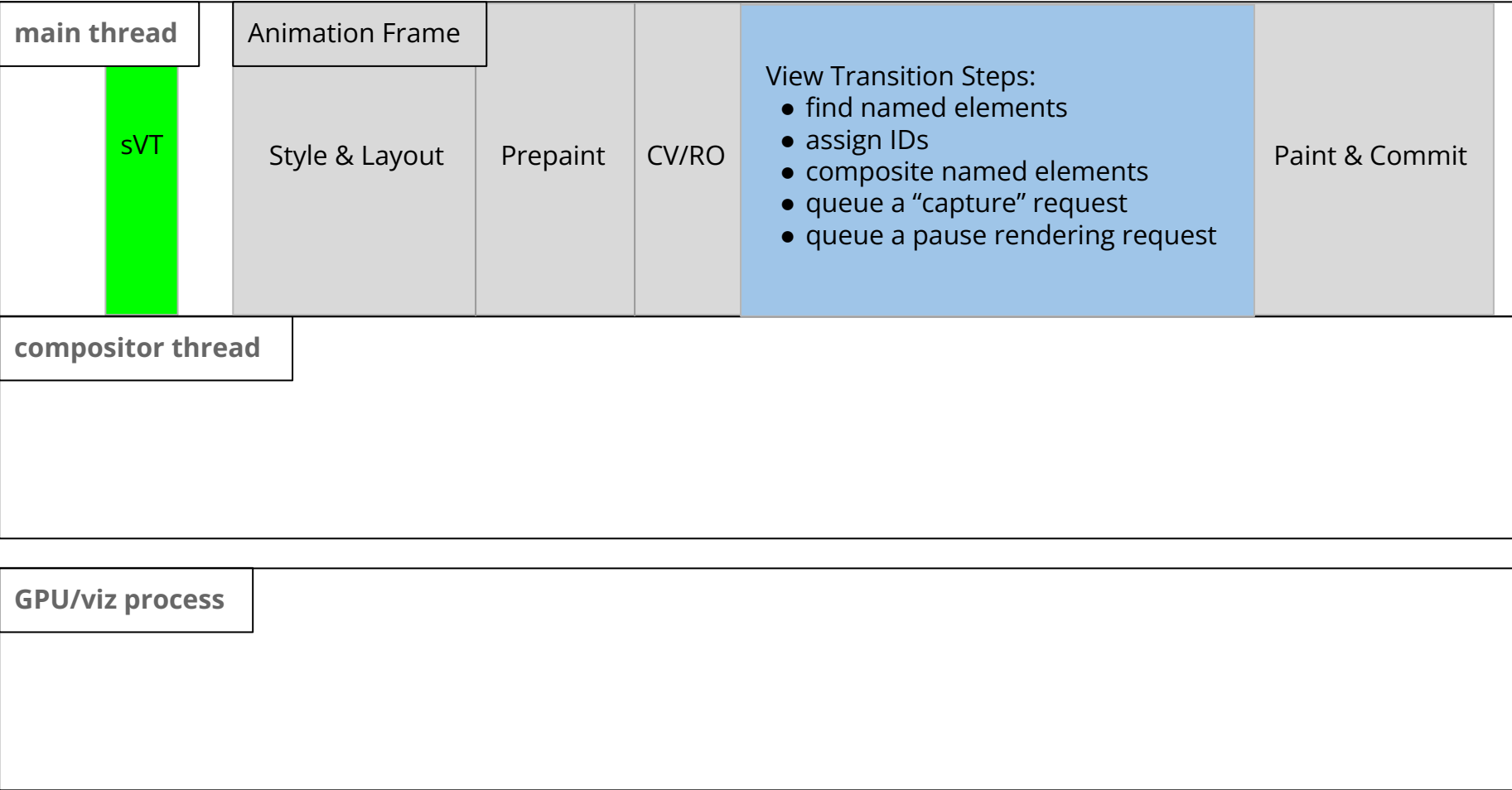
main thread

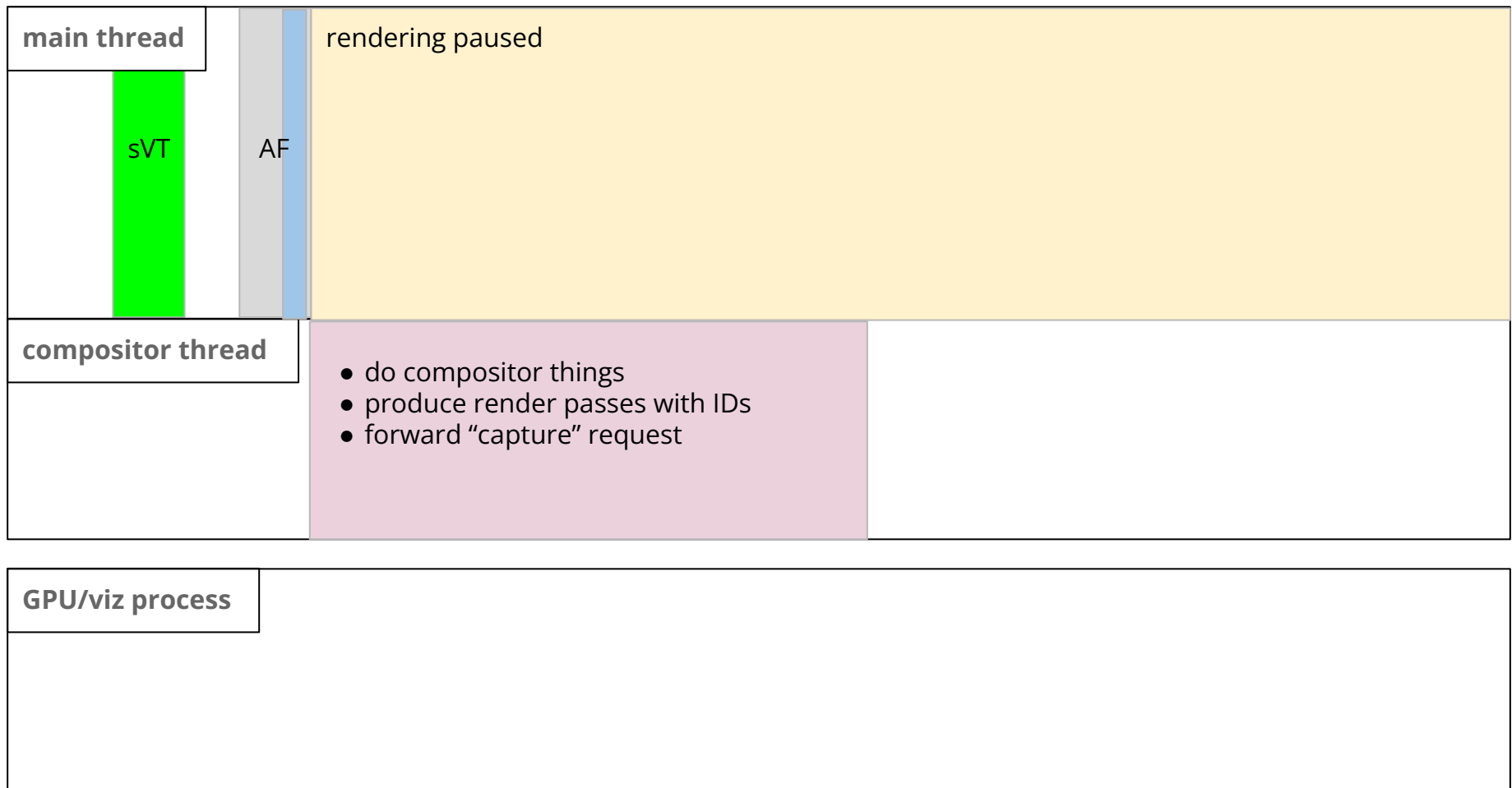
startViewTransition(cb)

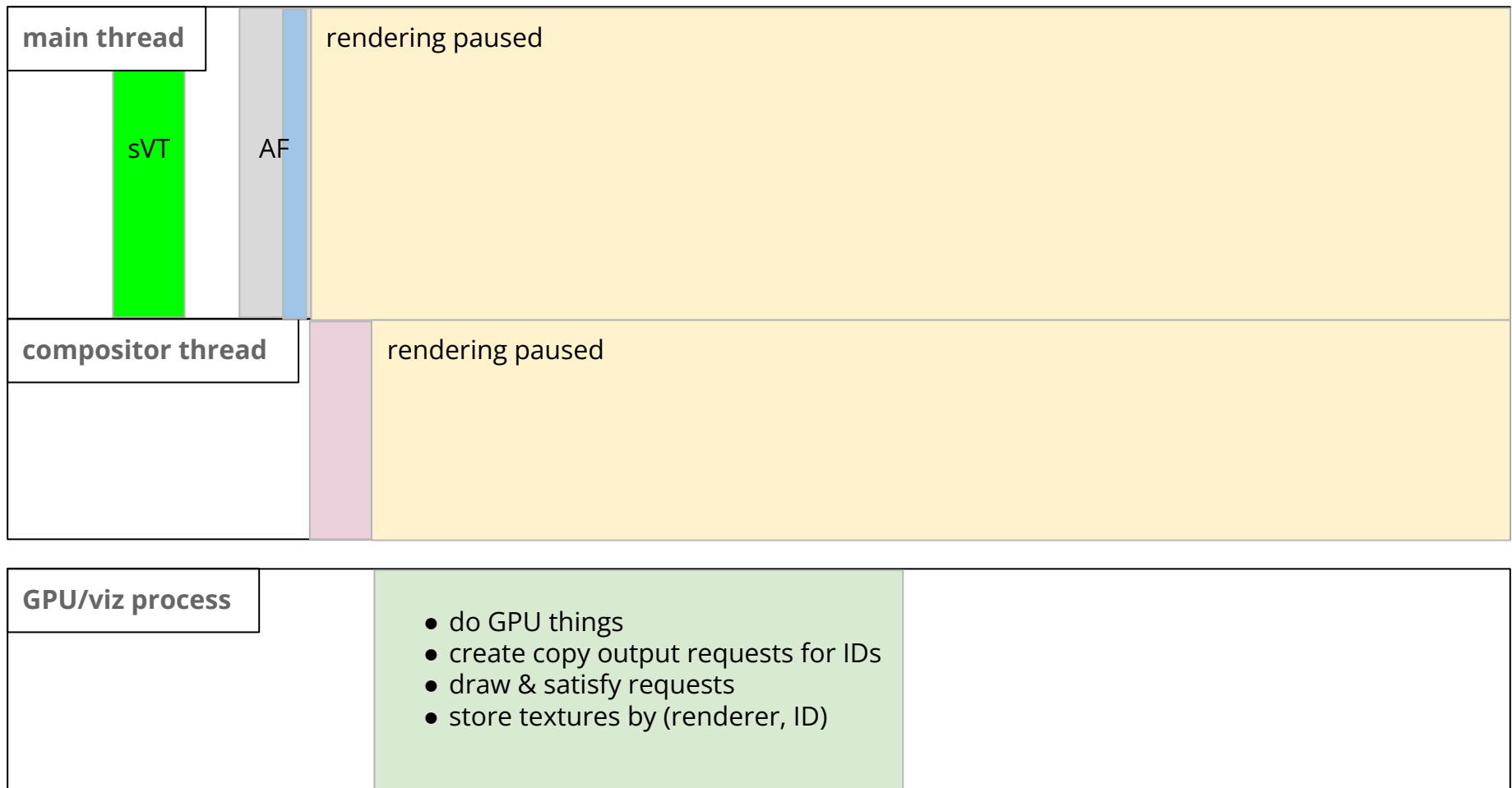
- create ViewTransition
- set up init state
- schedule Anim. Frame
- return object

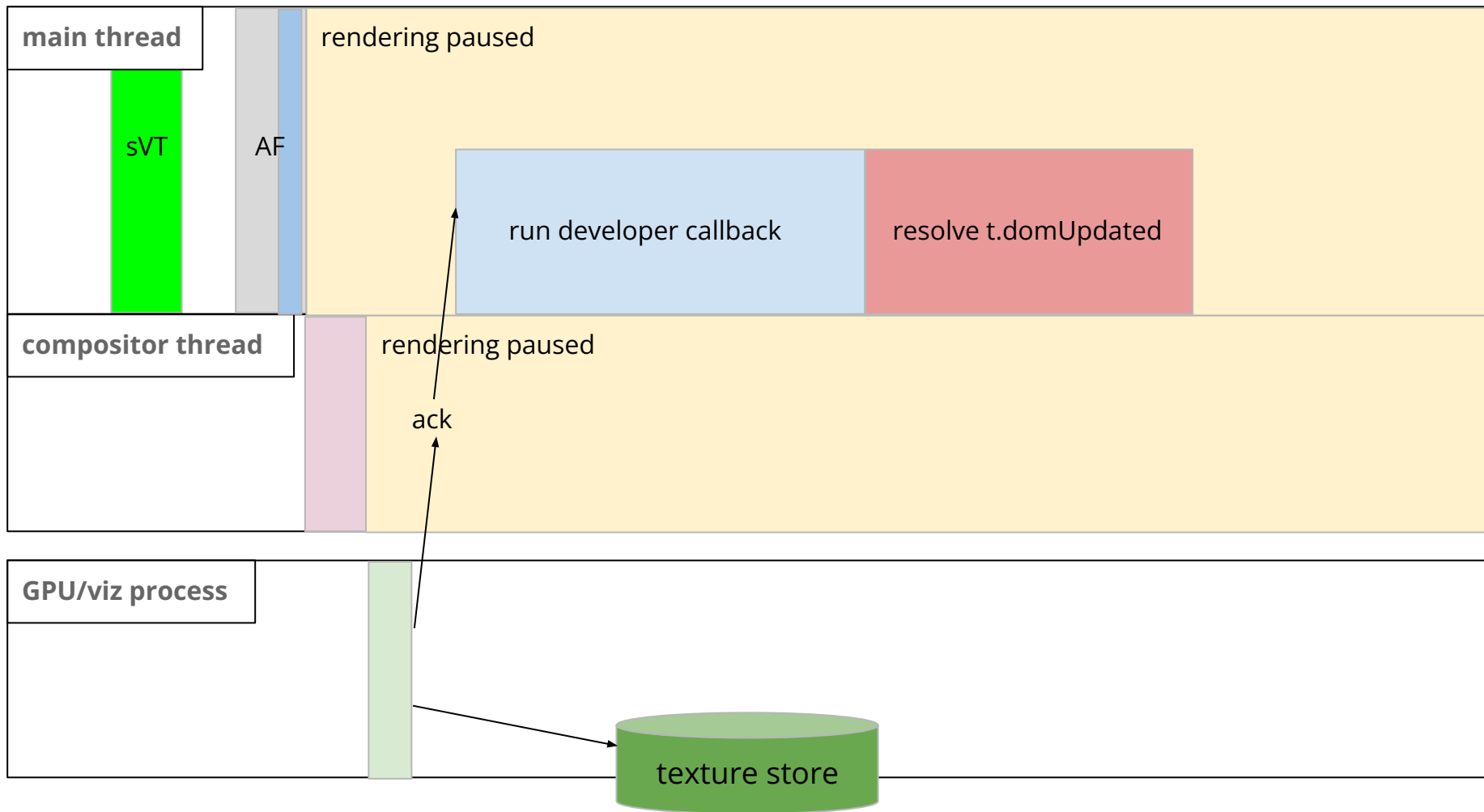
compositor thread

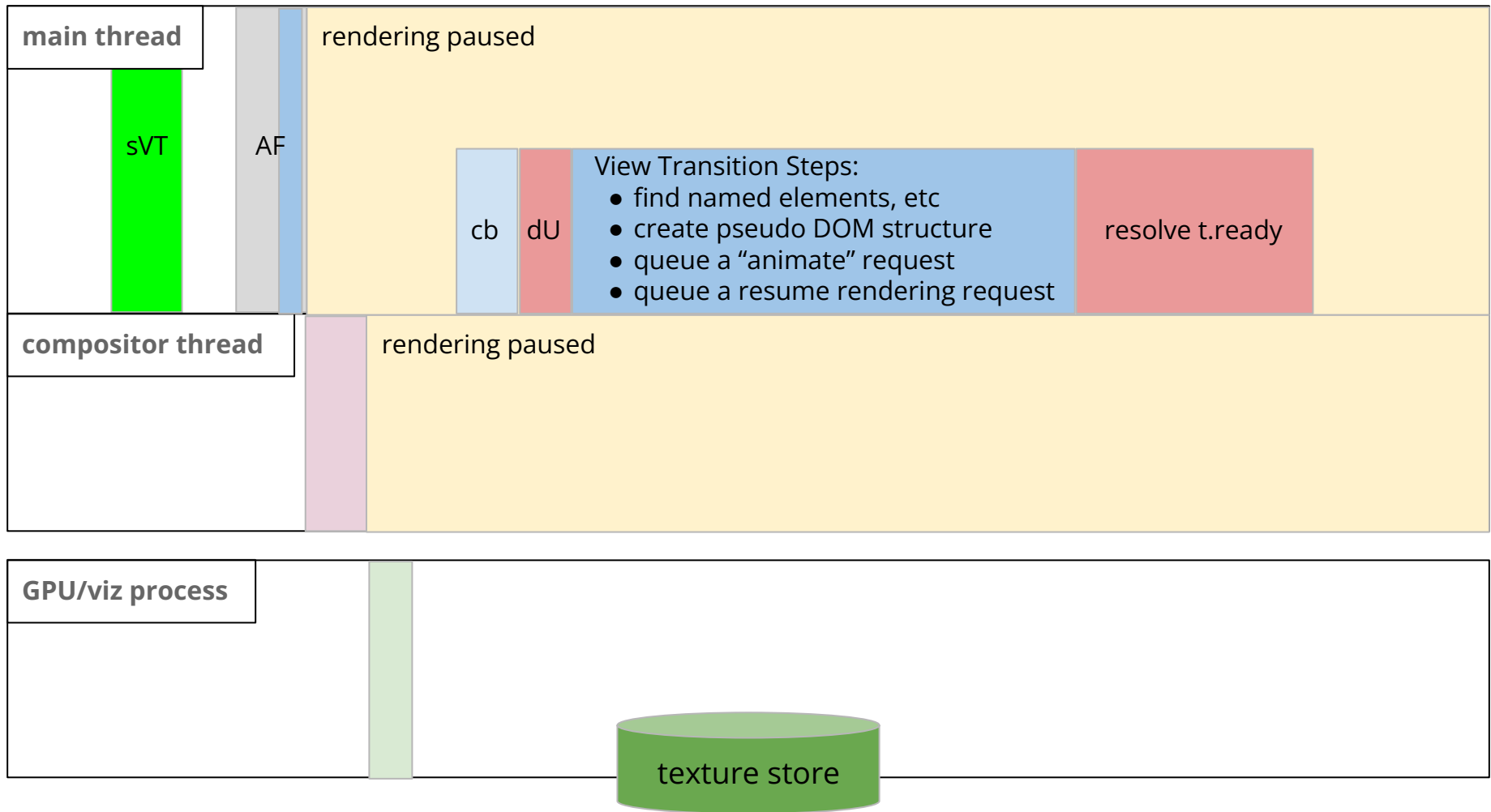
GPU/viz process

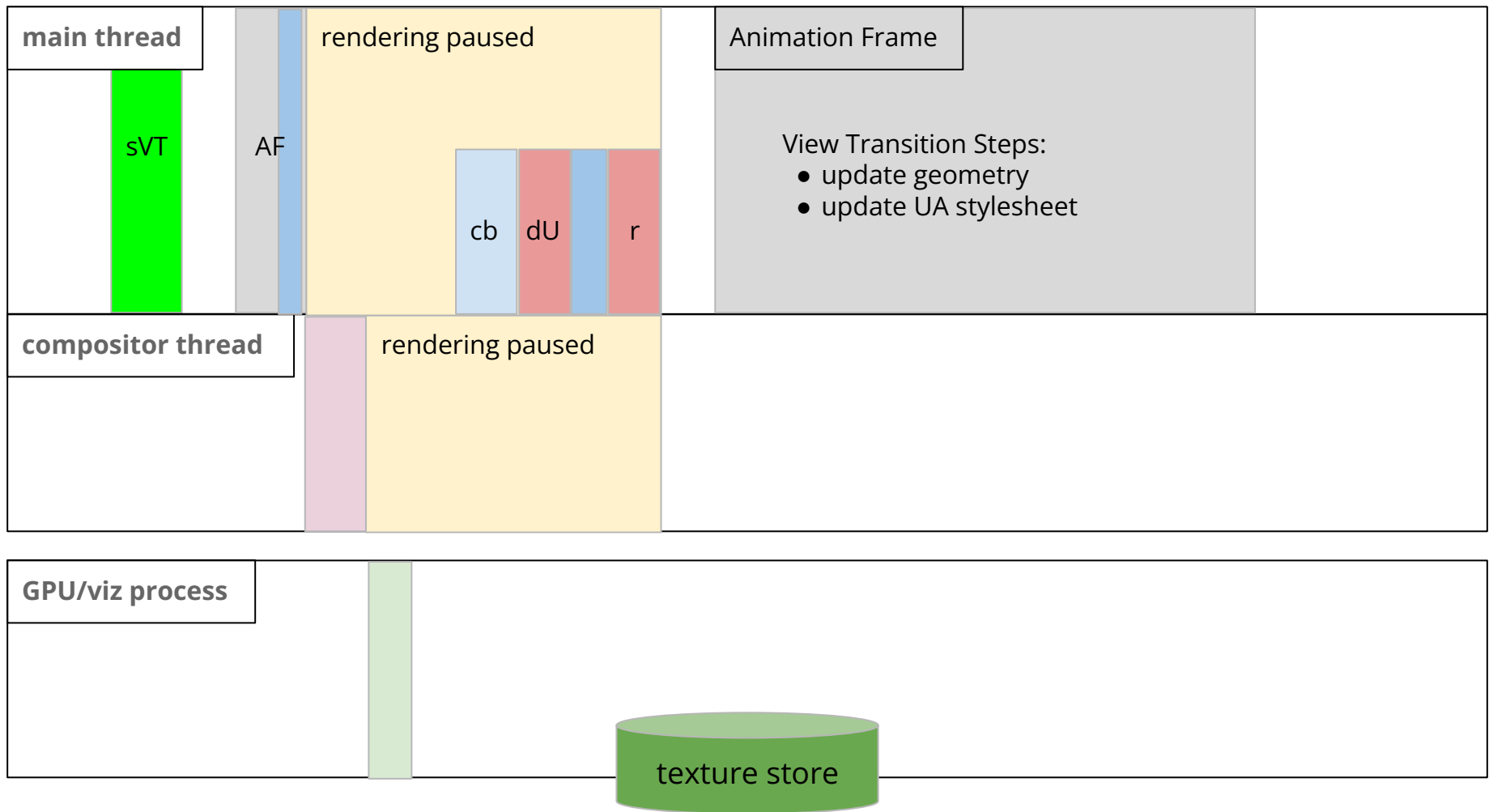


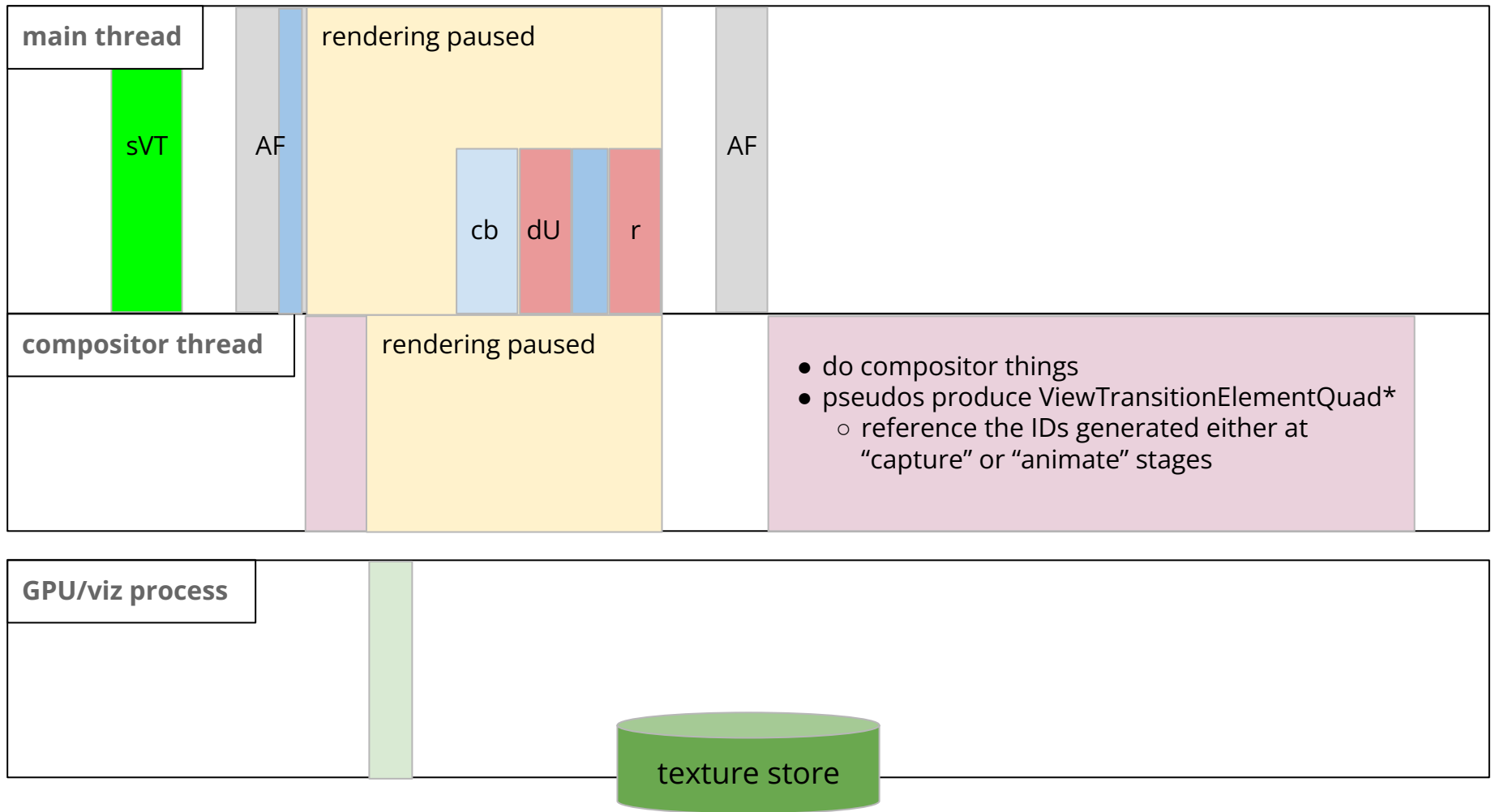


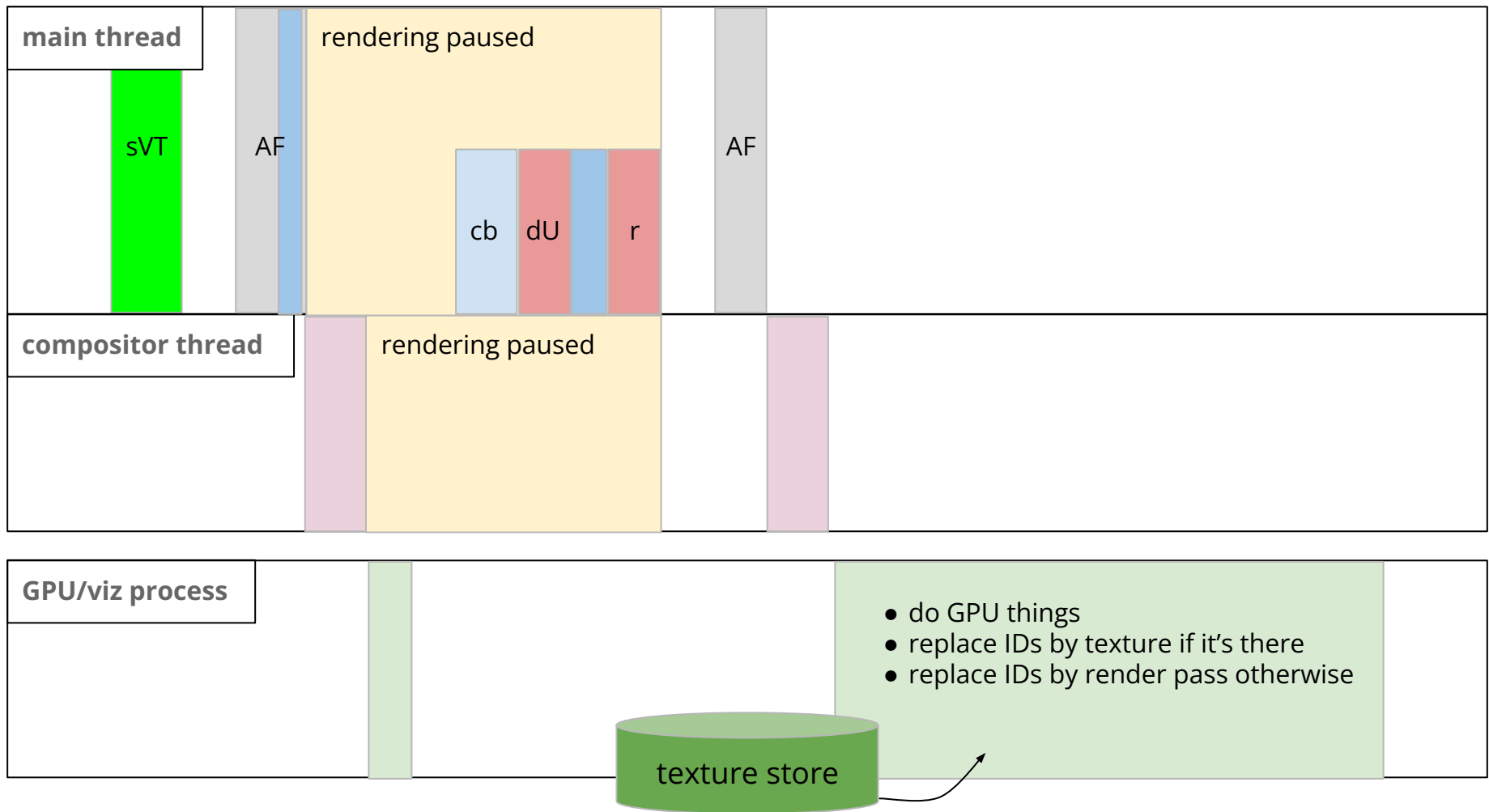


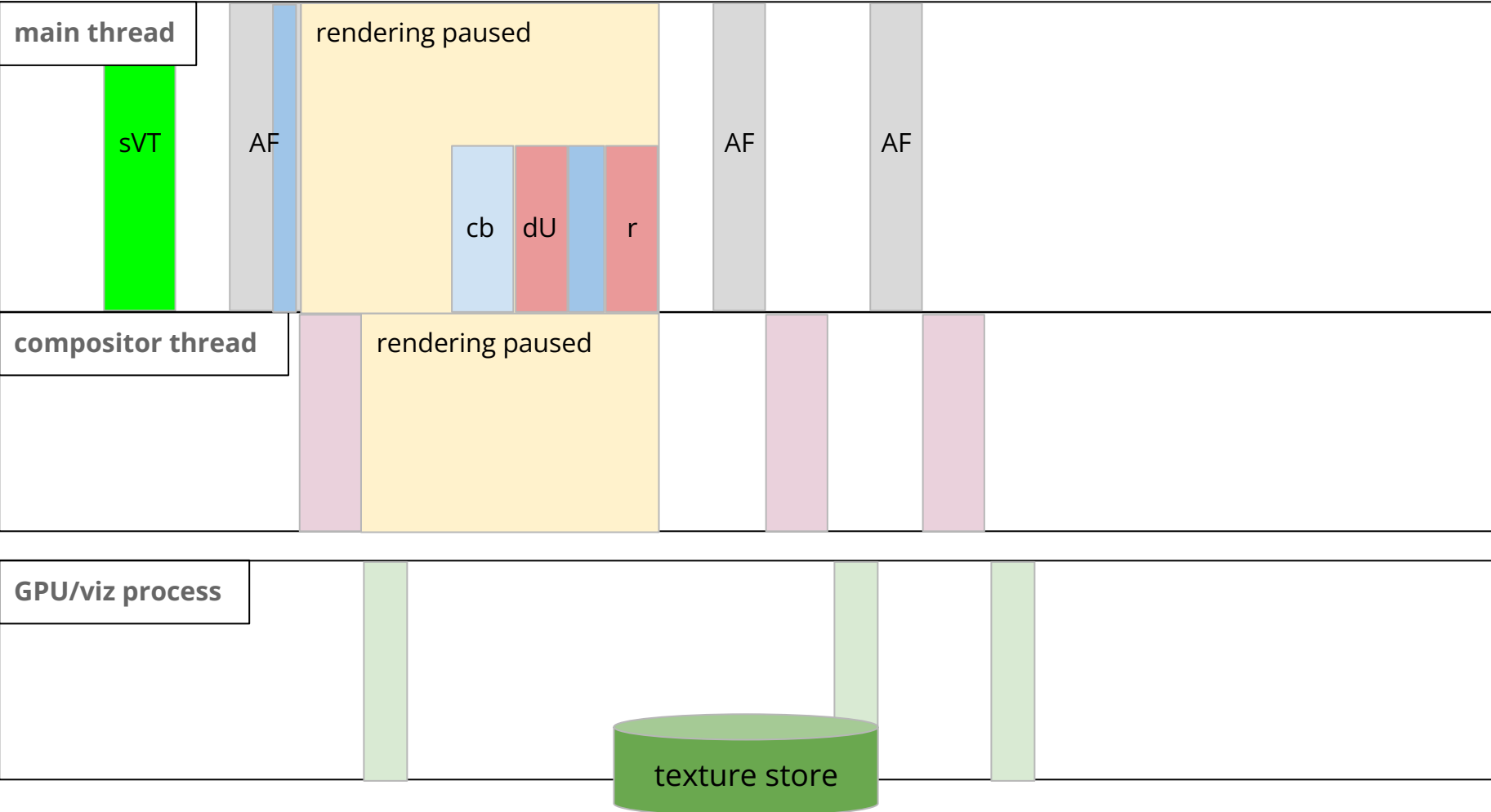


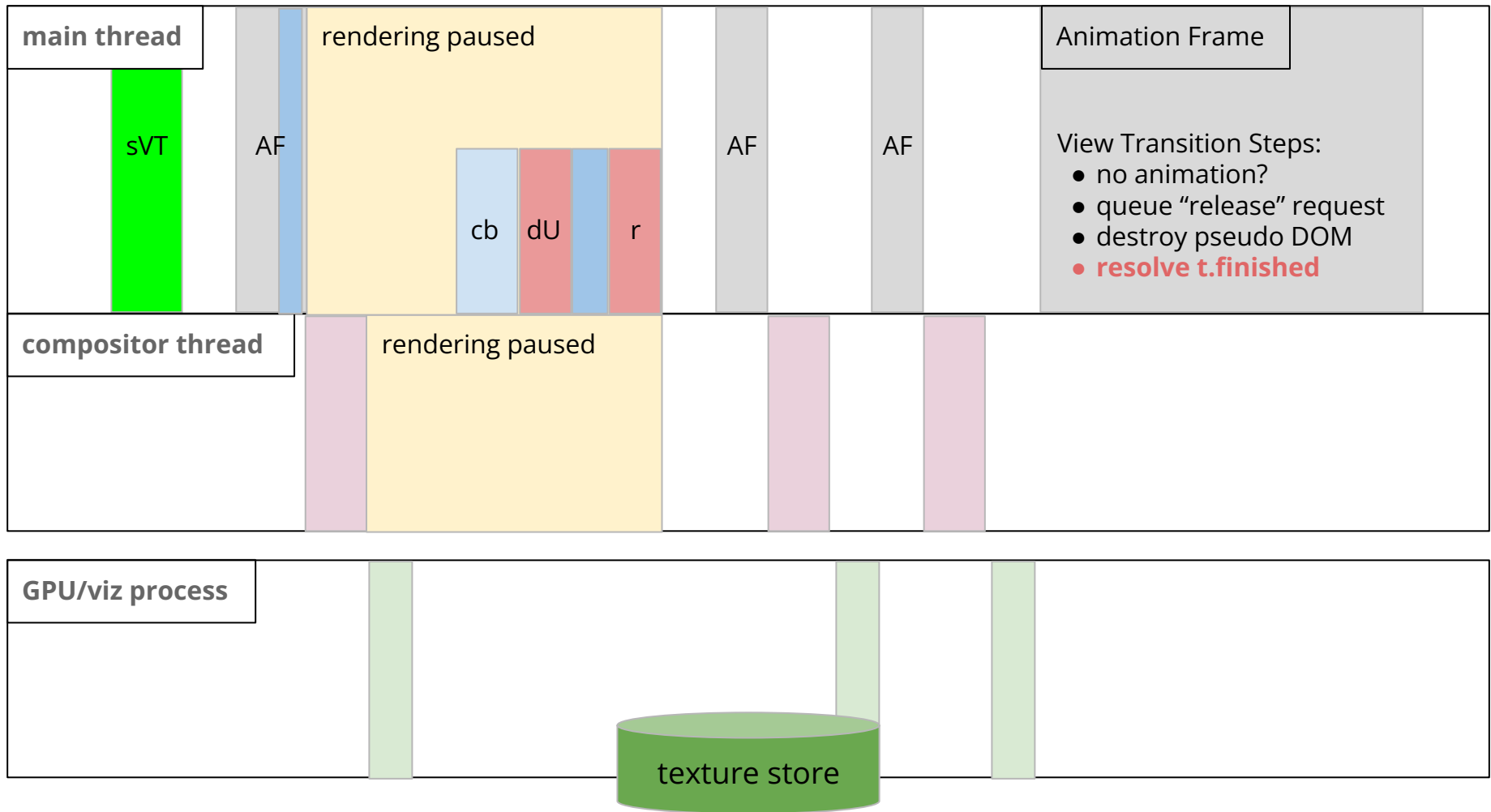


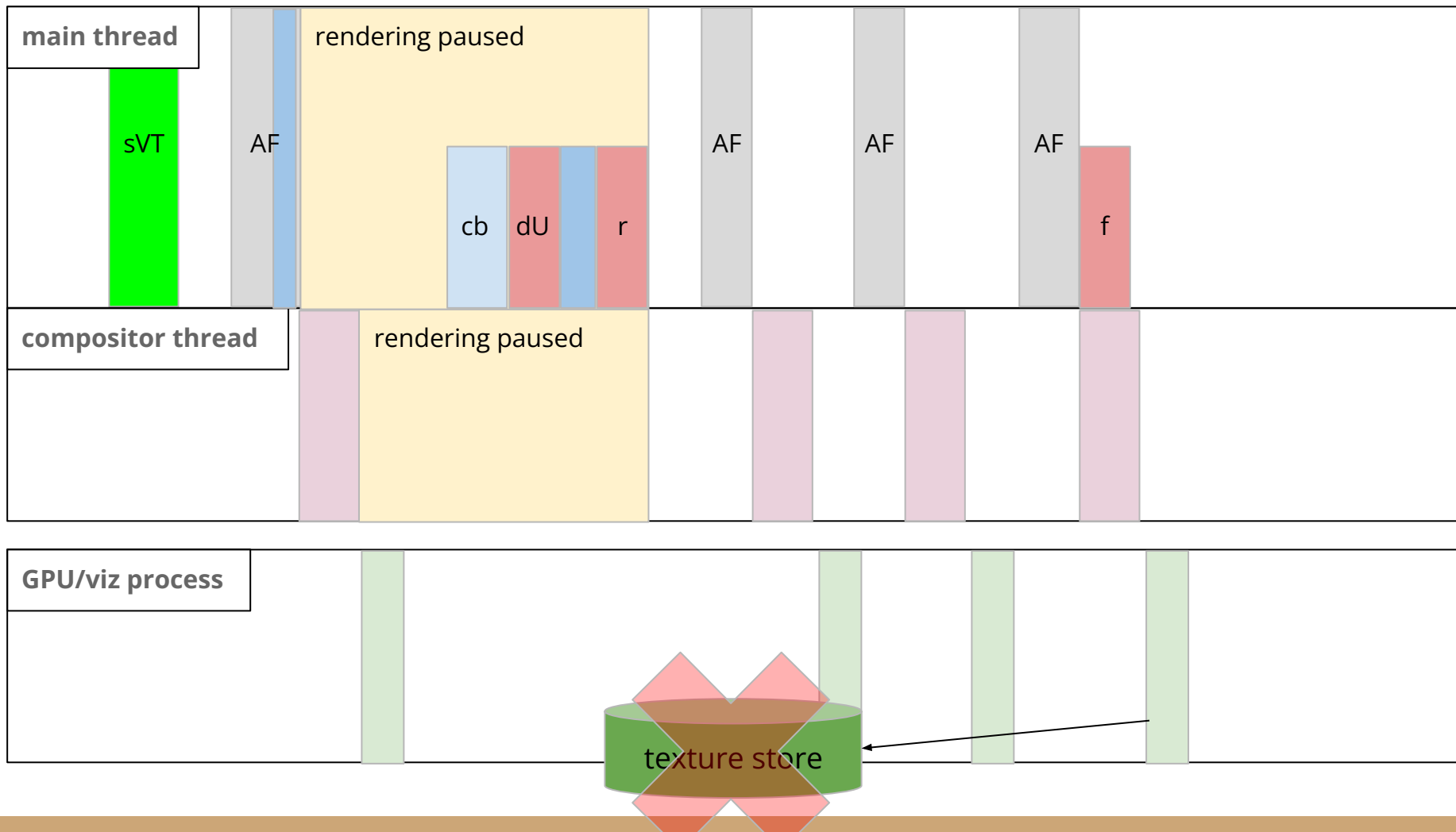


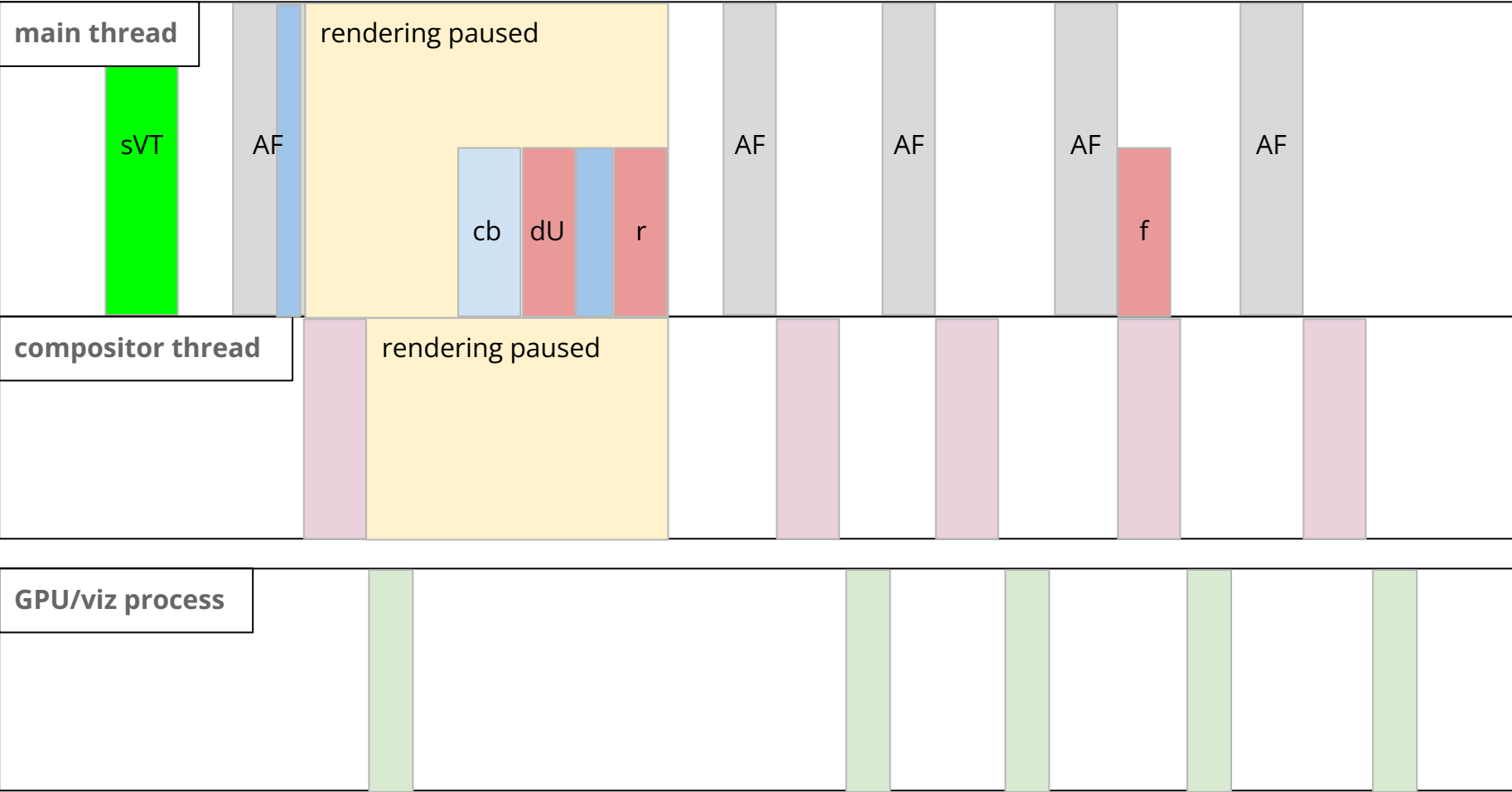


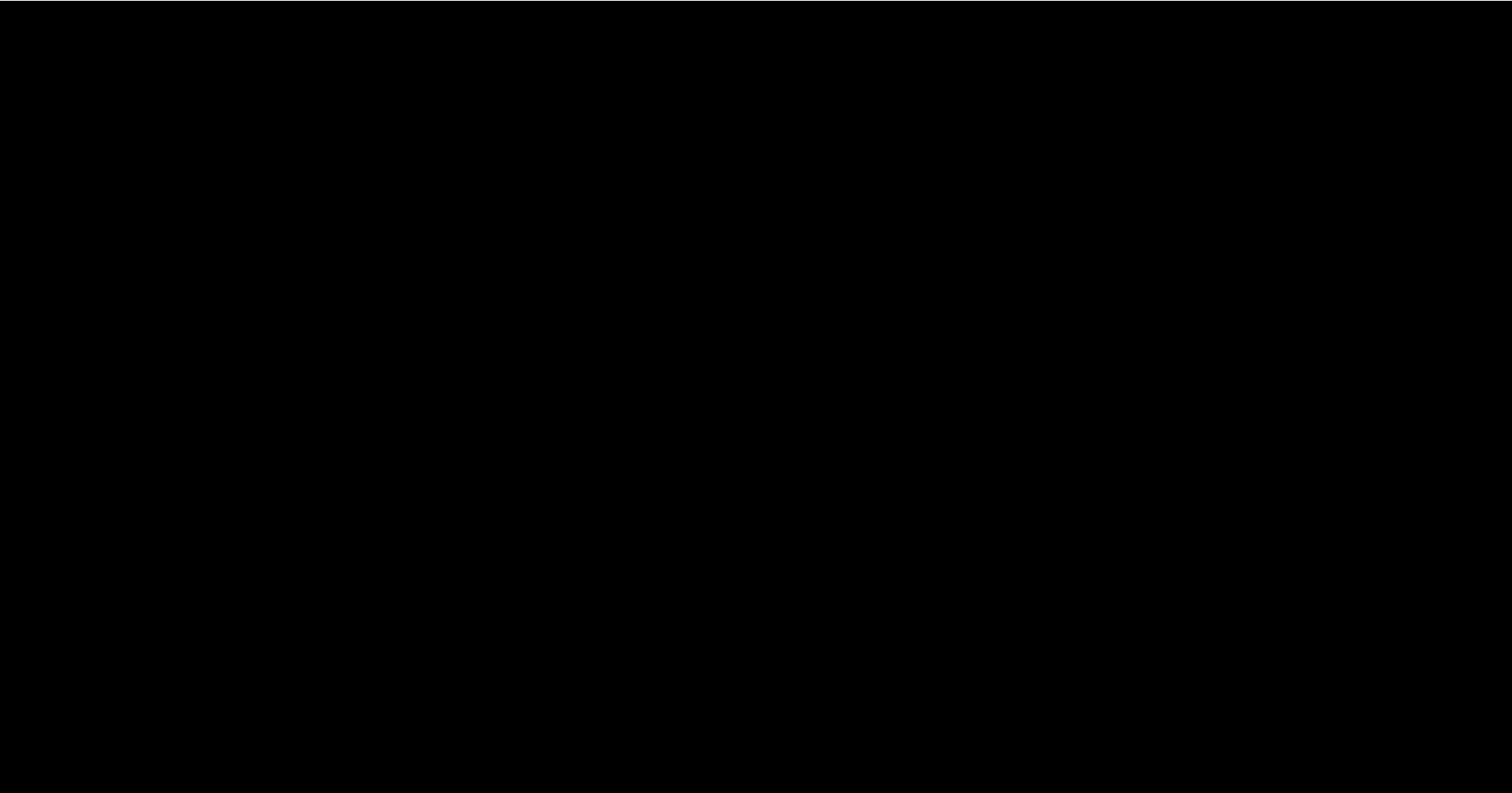


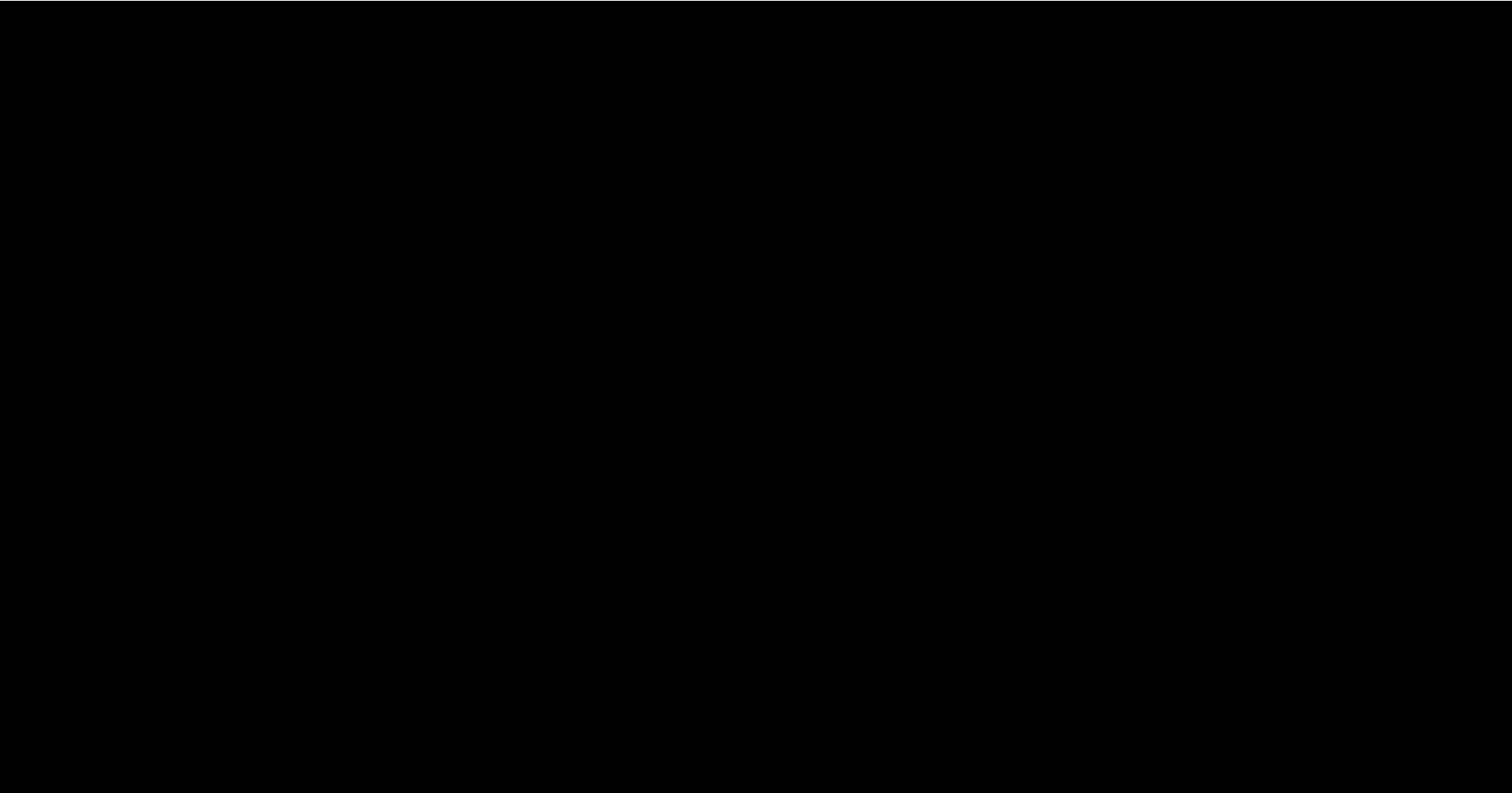












Thanks! Here's a roadmap

- Single-Page App support
 - getting close to ready
 - <https://drafts.csswg.org/css-view-transitions-1/>
 - `chrome://flags/#view-transition`
- Multi-Page App support
 - being prototyped
 - `chrome://flags/#view-transition-on-navigation`
- Scoped transitions
 - host the pseudo structure on any element
 - sometime in 2023
- Other enhancements