

Recoil Deep dive

Basic

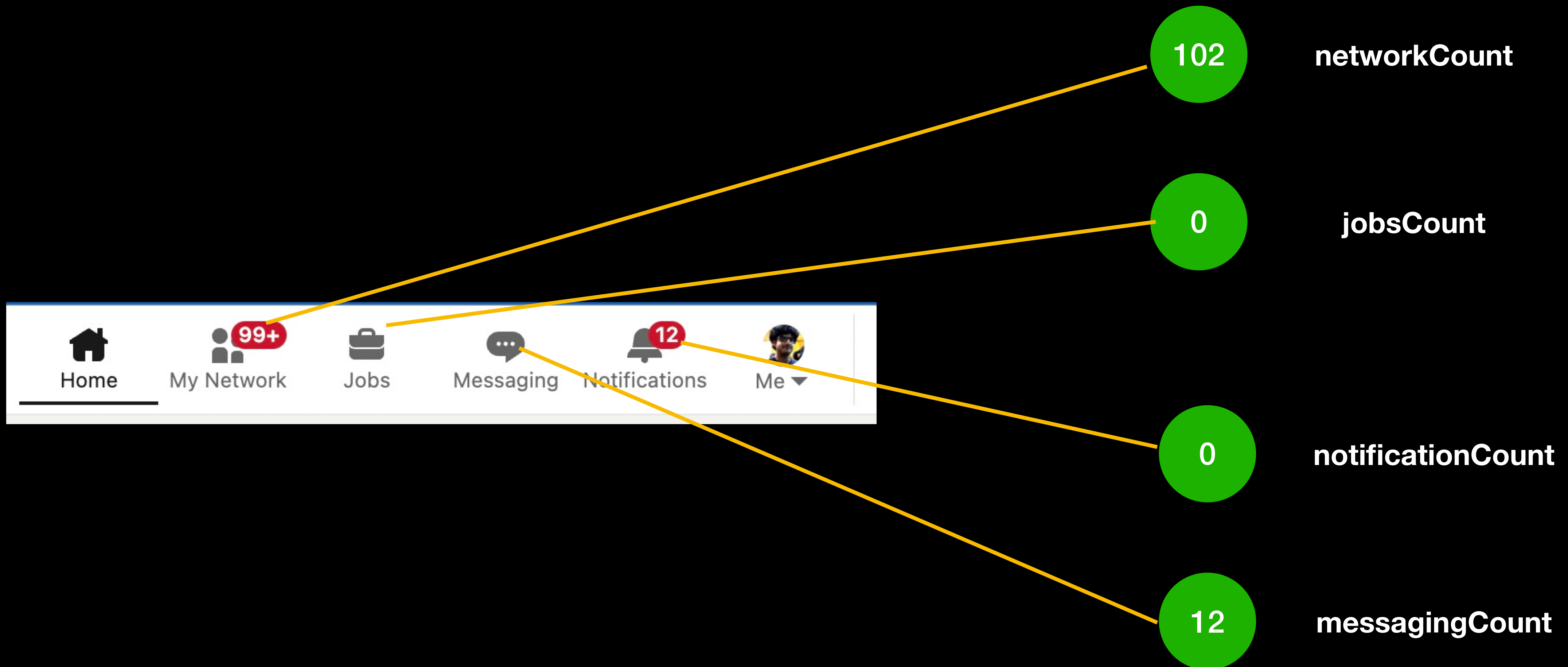
Atoms, selectors, Asynchronous data queries
useRecoilState, useRecoilValue, useSetRecoilState

Advance

atomFamily, selectorFamily
useRecoilStateLoadable, useRecoilValueLoadable

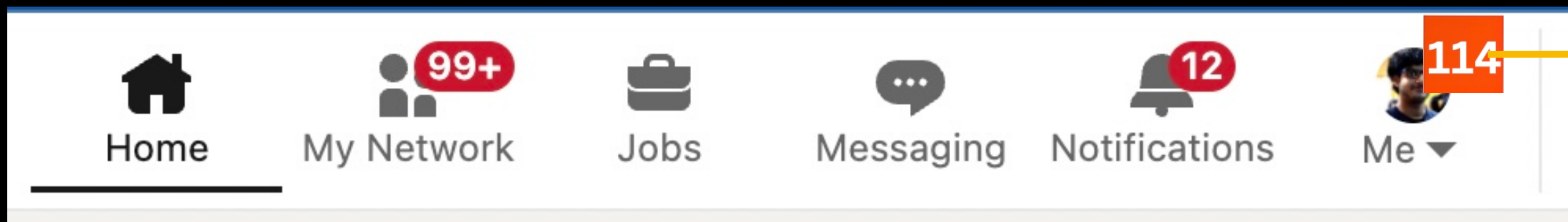
Code - <https://github.com/100xdevs-cohort-2/7.4>

Atom



Selectors

Lets say you want to show all notification counts above your avatar



102

0

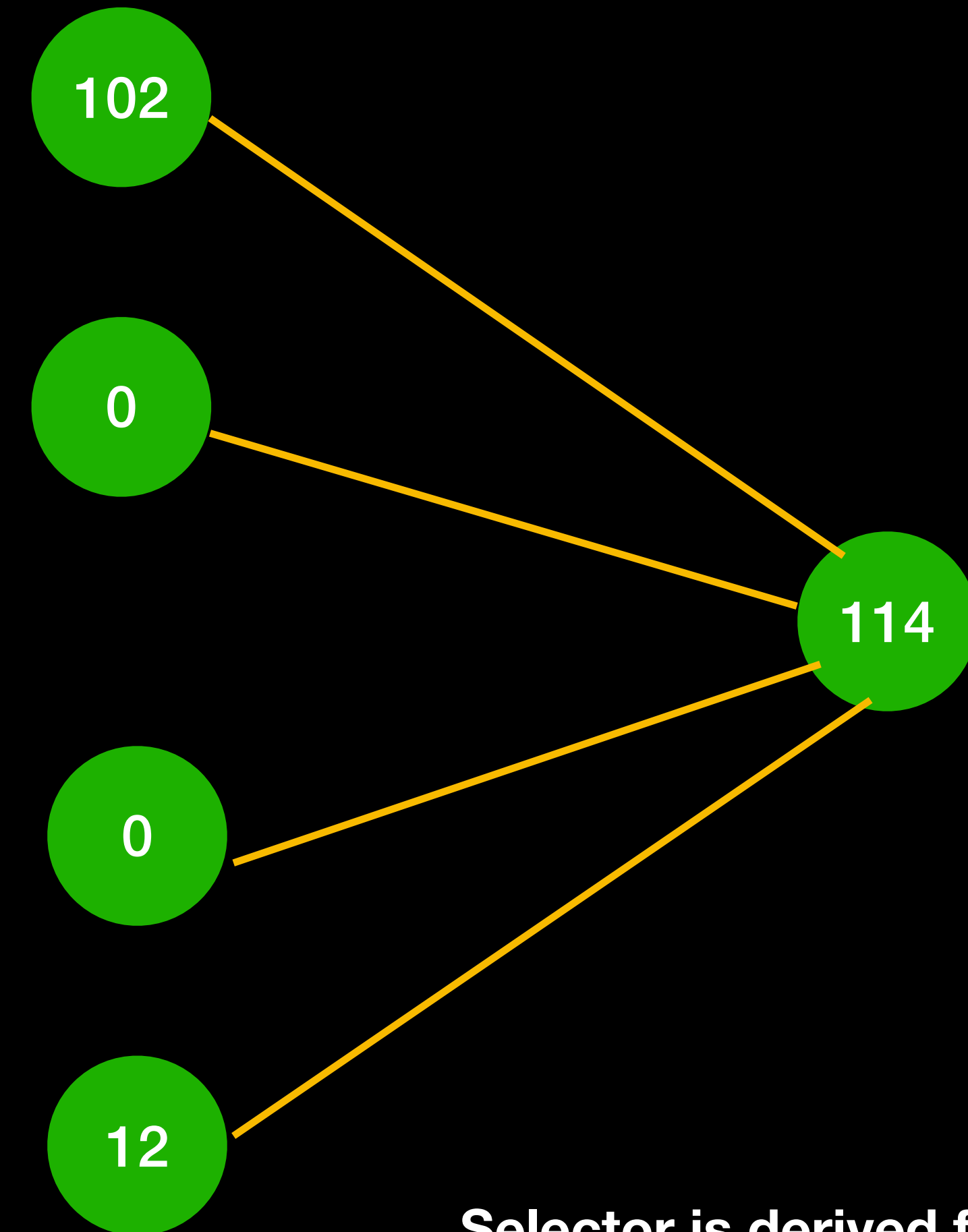
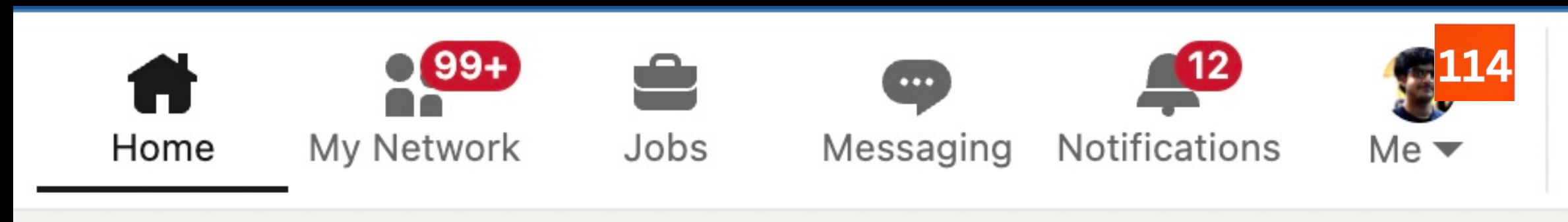
114

0

12

Selectors

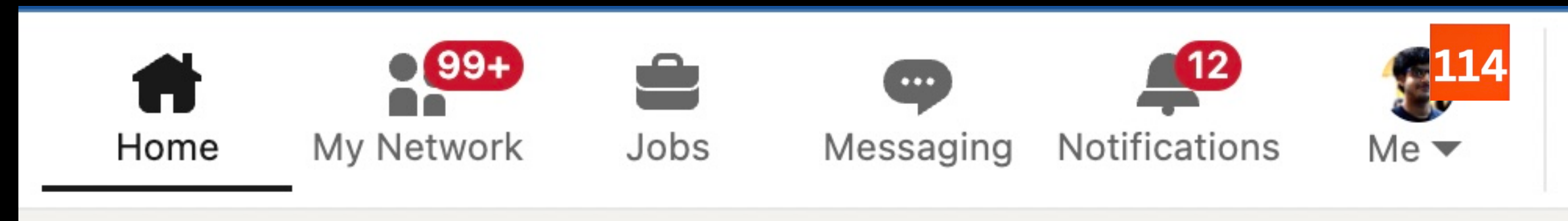
Lets say you want to show all notification counts above your avatar



Selector is derived from other atoms

Atoms and Selectors

Lets build this Appbar component (hardcode values)



Recoil Deep dive

Basic

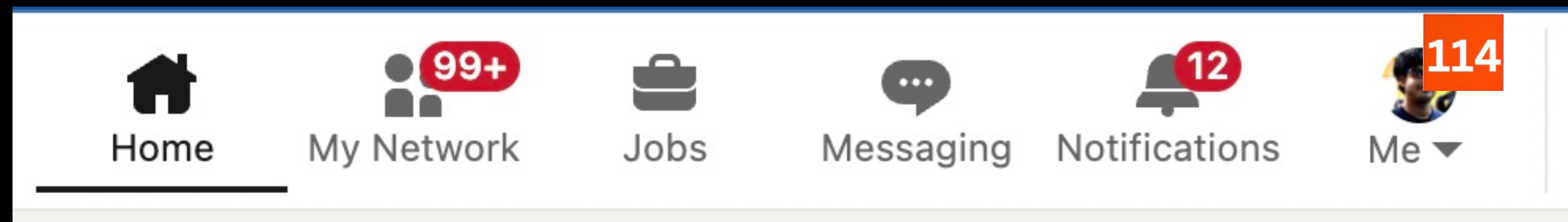
Atoms, selectors, Asynchronous data queries
useRecoilState, useRecoilValue, useSetRecoilState

Advance

atomFamily, selectorFamily
useRecoilStateLoadable, useRecoilStateLoadable

Asynchronous data queries

Lets build this Appbar component (values coming from backend)



<https://sum-server.100xdevs.com/notifications>

Recoil Deep dive

Basic

Atoms, selectors, Asynchronous data queries
useRecoilState, useRecoilValue, useSetRecoilState

Advance

atomFamily, selectorFamily
useRecoilStateLoadable, useRecoilStateLoadable

atomFamily

Problem - Sometimes you need more than one atom for your use case

For example - Creating a todo application

atomFamily

Question - Create a component that takes a todo id as input, and renders the TODO

You need to store the Todo in an atom (can't use useState)

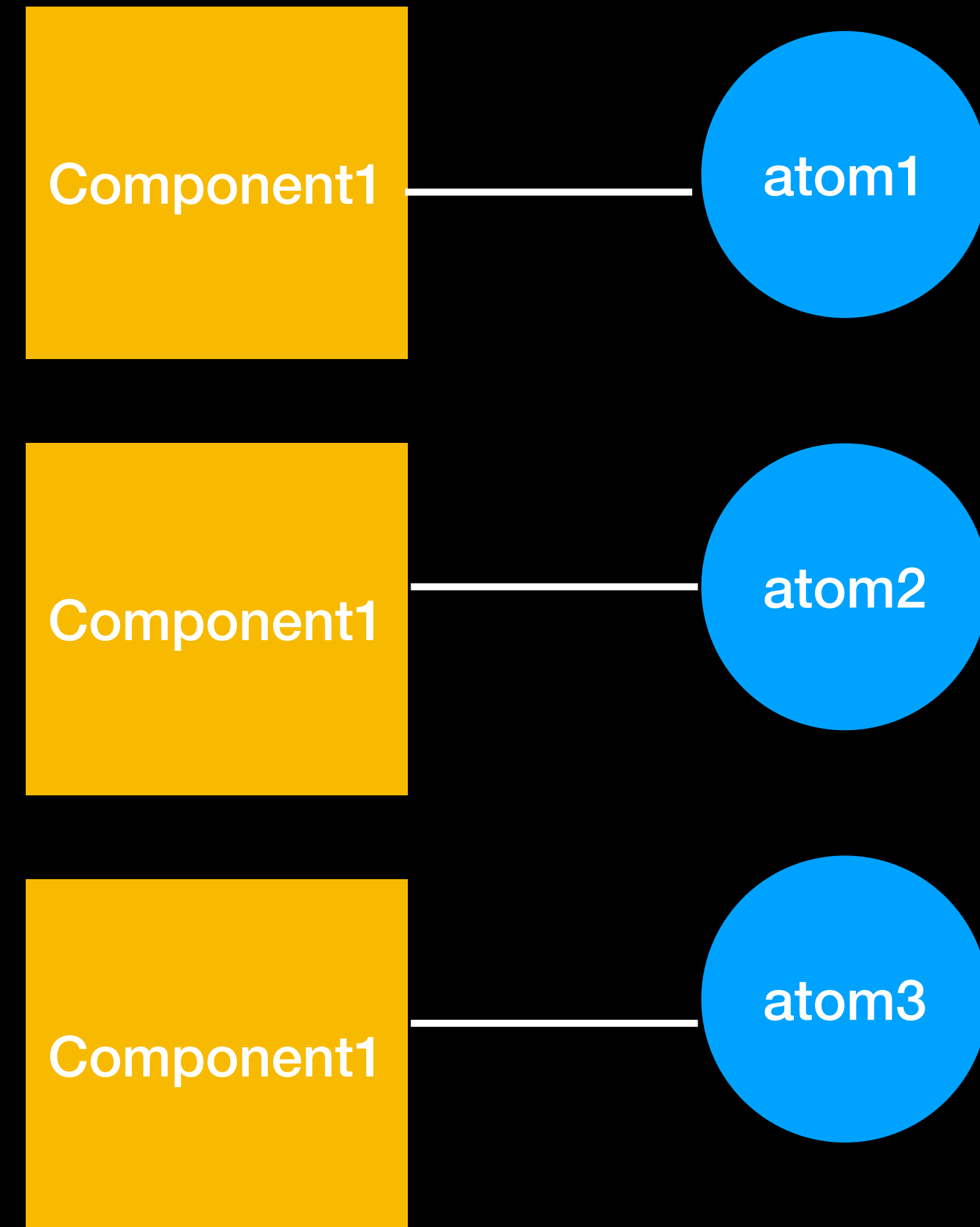
All the TODOs can be hardcoded as a variable



```
{
  "todo": {
    "id": 2,
    "title": "Todo 2",
    "description": "This is todo 2",
    "completed": false
  }
}
```

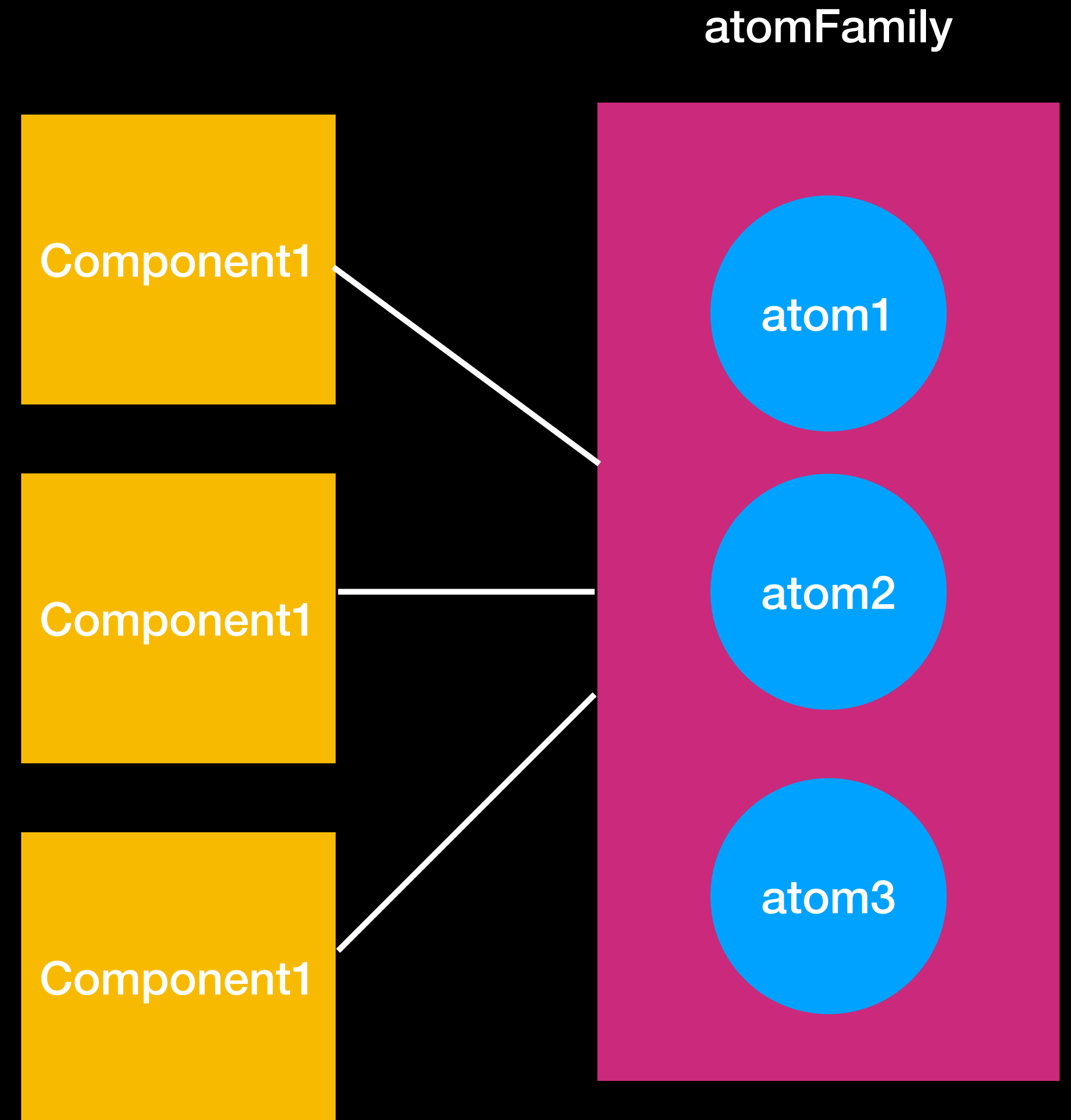
atomFamily

Would you have a single atom ?
Would you have one atom per todo?
How would you create (and delete) todos dynamically?



atomFamily

Would you have a single atom ?
Would you have one atom per todo?
How would you create (and delete) todos dynamically?



atomFamily

Let's code the todo app using an atom family

```
src > JS atoms.js > ...
1  import { atomFamily } from "recoil";
2
3  const TODOS = [{
4    id: 1,
5    title: "Go to Gym",
6    description: "Hit the gym from 7-9"
7  }, {
8    id: 2,
9    title: "Go to eat food",
10   description: "Eat food from from 9-11"
11  },]
12
13  export const todosAtomFamily = atomFamily({
14    key: 'todosAtomFamily',
15    default: id => {
16      return TODOS.find(x => x.id === id)
17    },
18  });
```

Recoil Deep dive

Basic

Atoms, selectors, Asynchronous data queries
useRecoilState, useRecoilValue, useSetRecoilState

Advance

atomFamily, selectorFamily
useRecoilStateLoadable, useRecoilValueLoadable

selectorFamily

In the TODO application, lets say you are supposed to get TODOs from a server

`https://sum-server.100xdevs.com/todo?id=1`

selectorFamily

In the TODO application, lets say you are supposed to get TODOs from a server

<https://sum-server.100xdevs.com/todo?id=1>

```
selectorFamily / src / 10 atoms.ts / ...
1  ∨ import { atomFamily, selectorFamily } from "recoil";
2  import axios from "axios";
3
4  ∨ export const todosAtomFamily = atomFamily({
5    key: 'todosAtomFamily',
6    ∨ default: selectorFamily({
7      key: "todoSelectorFamily",
8      ∨ get: (id) => async ({get}) => {
9        const res = await axios.get(`https://sum-server.100xdevs.com/todo?id=${id}`);
10       return res.data.todo;
11     },
12   })
13 });
```


Recoil Deep dive

Basic

Atoms, selectors, Asynchronous data queries
`useRecoilState`, `useRecoilValue`, `useSetRecoilState`

Advance

`atomFamily`, `selectorFamily`
`useRecoilStateLoadable`, `useRecoilValueLoadable`

Code - <https://github.com/100xdevs-cohort-2/7.4>

useRecoilStateLoadable, useRecoilValueLoadable

What happens when the values aren't loaded immediately?

For example, the TODOs that are coming back from the server?

How can we show loader on screen when that happens rather than an empty state?