

# Module Reload Signals

# Module Reload Signals

- Module systems are using various means of hot module reloading
- ESM is unable to propagate updates in the same way due to lack of control of the module cache.

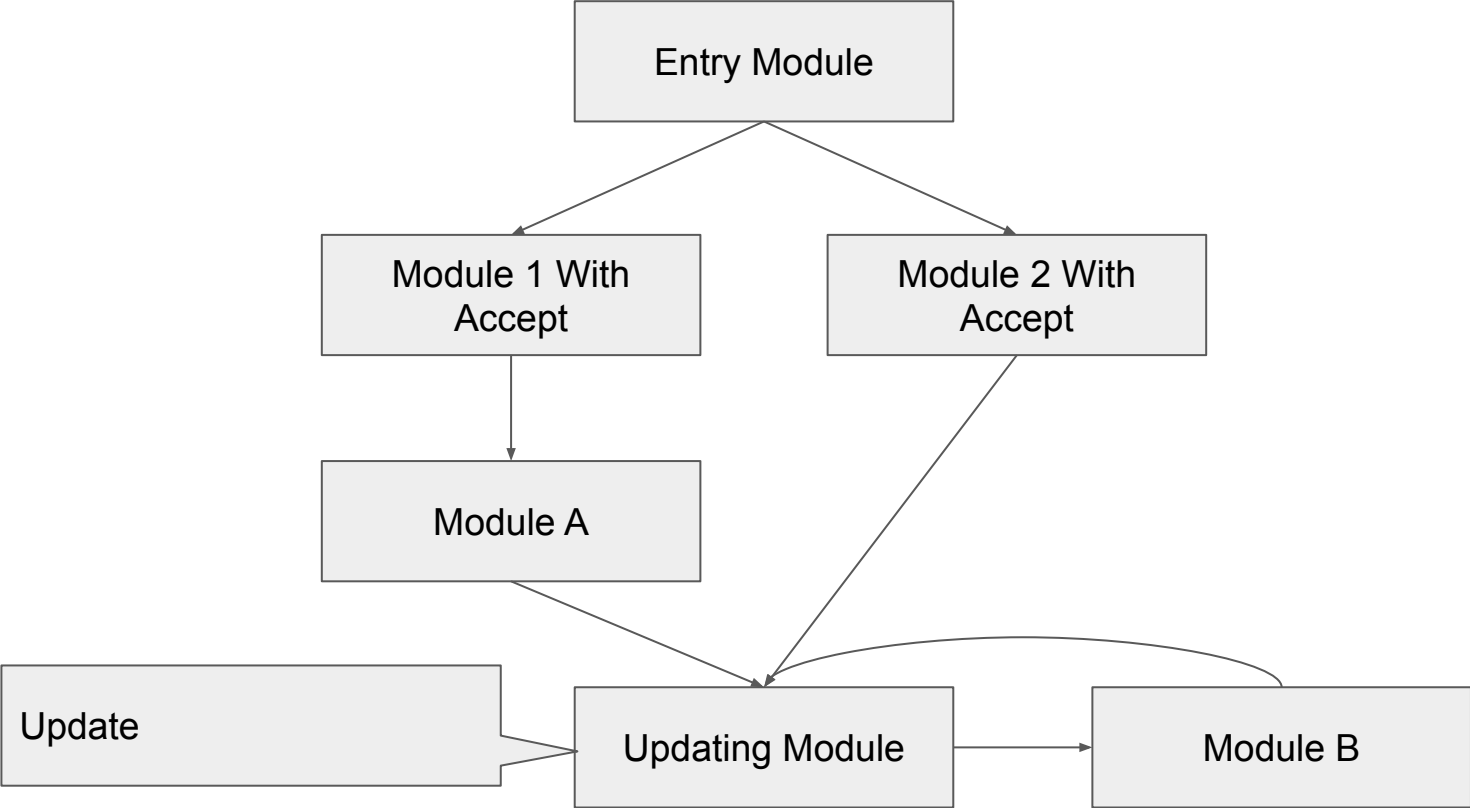
# Example Signal / Handler

- ```
/**  
import.hot adds the module to list of dispatch targets for the intended module  
map key of this module, if the dispatch target already exists, it is a no-op?  
**/  
  
export let foo = 1;  
import.meta.hot((newNamespace) => {  
  foo = newNamespace.foo;  
});
```

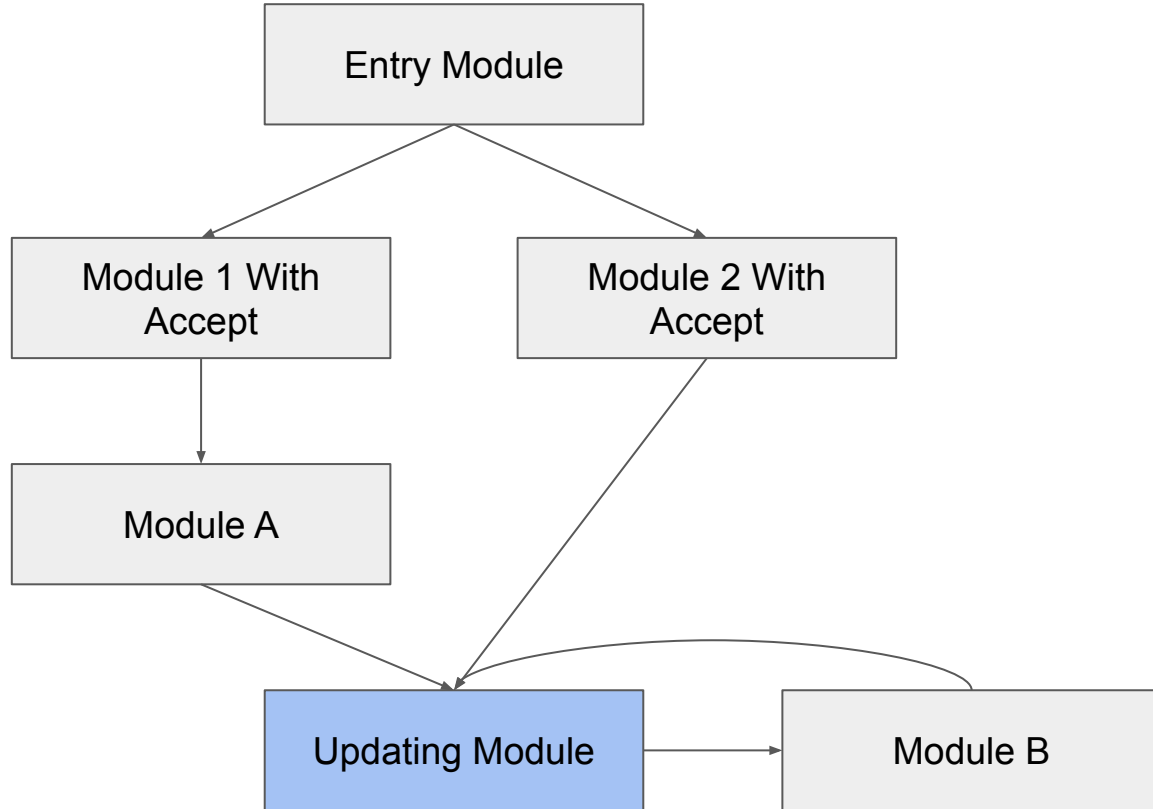
# Example Signal

- Upon a server signalling that a module is needing to be updated, the client starts traversing the graph
- // pseudocode  
await serverSaysModuleUpdated(async (url, version) => {  
 const subgraphToReloadEdges = findAcceptBarrier(url);  
 dispatchHot(url, ns);  
});

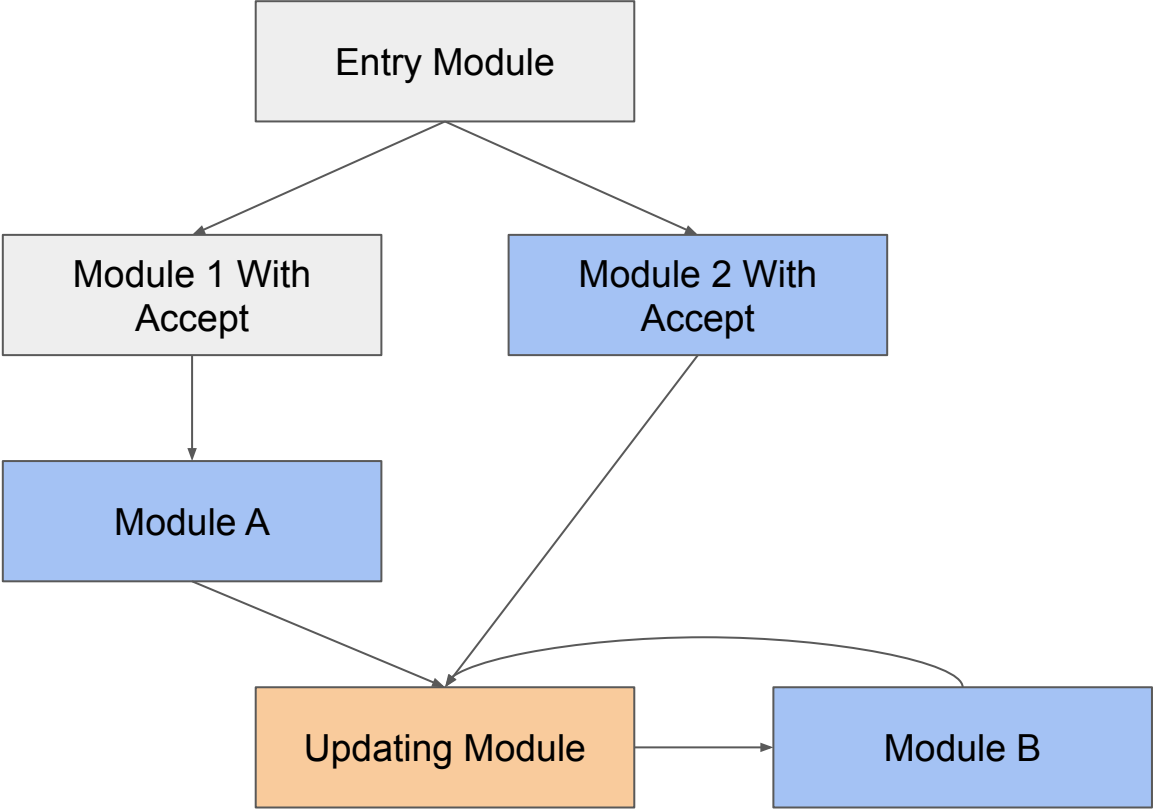
# Example Signal



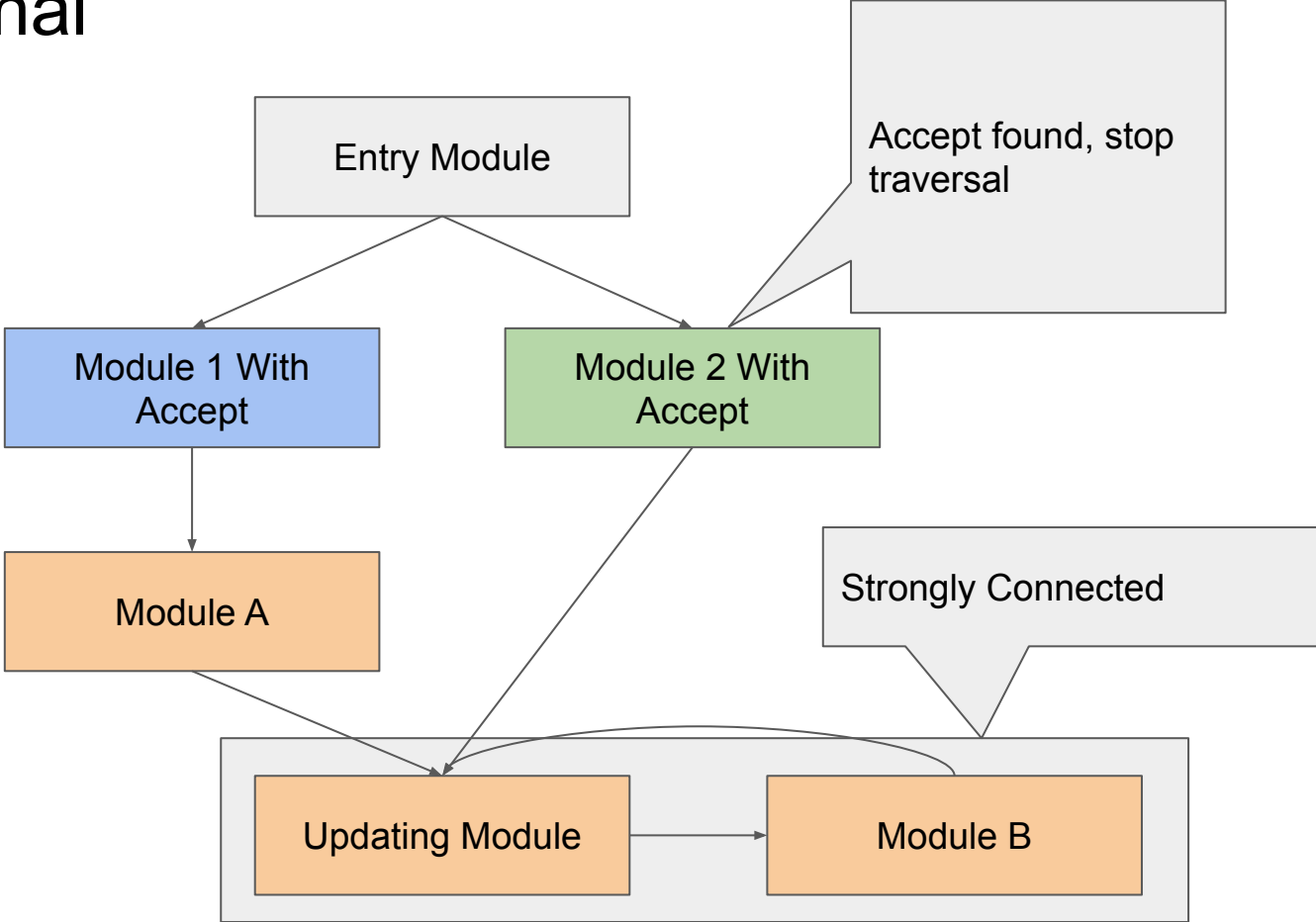
# Example Signal



# Example Signal



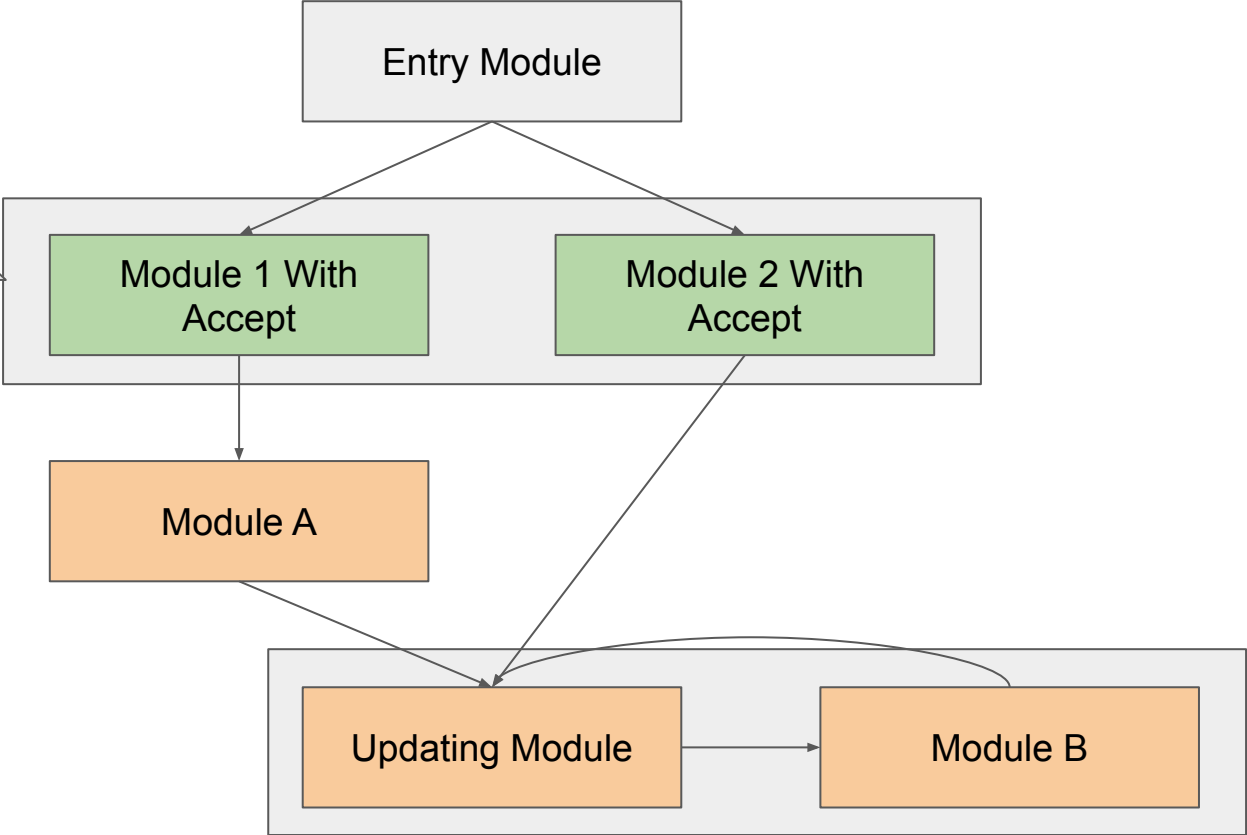
# Example Signal



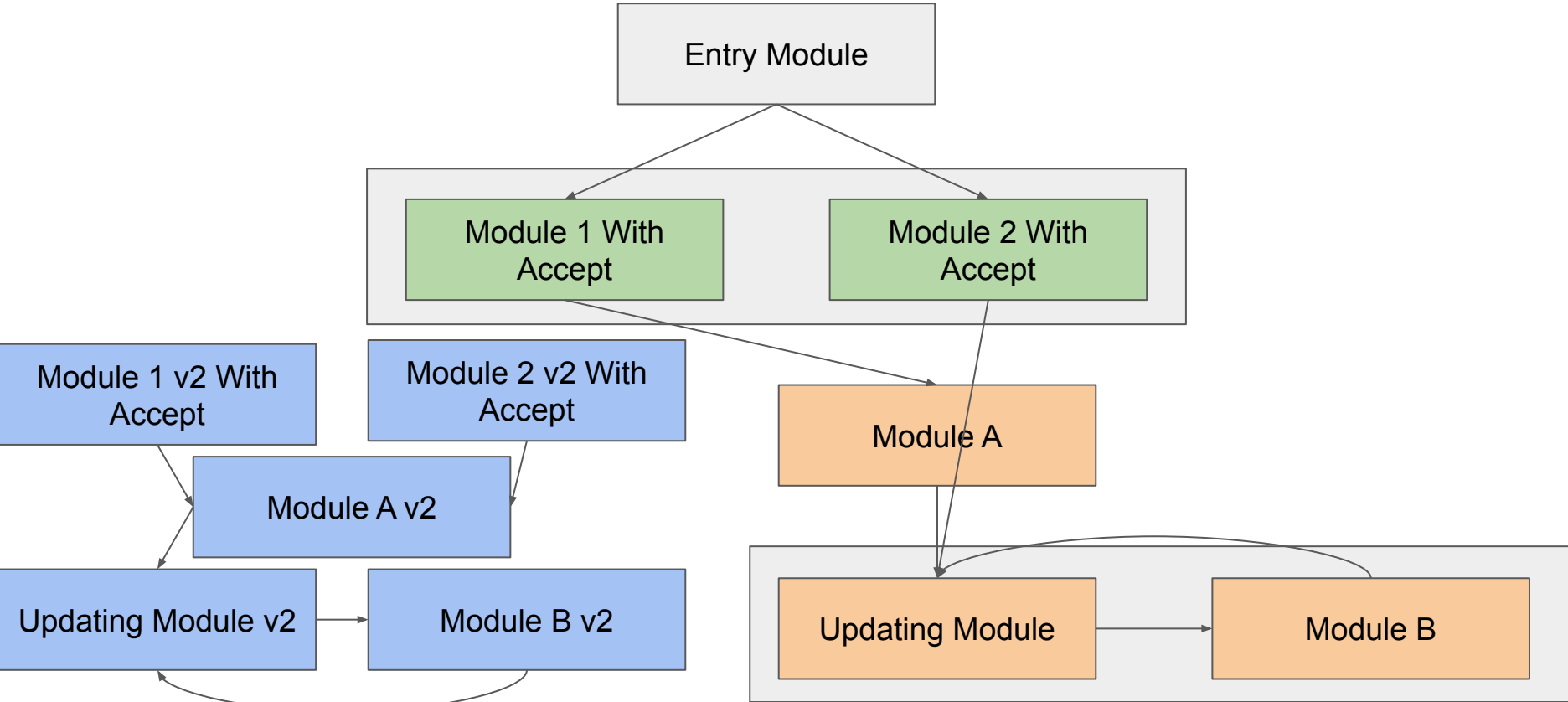


# Example Signal

Subgraph Covered by Accept Found



# Example Signal



# Example Signal

