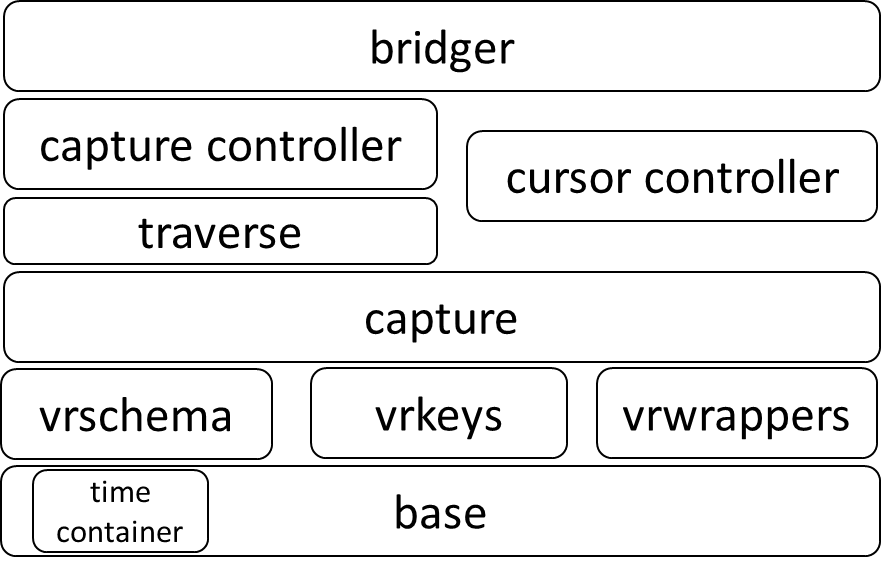
Definitions

Components and responsibilities

|  |  |  |
| --- | --- | --- |
| **component** | **responsibility** | **collaborations and dependencies** |
| base | Lowest level platform interface  range concept  result concept  time containers concept | uses stl/c++ runtime library  uses tbb |
| vr schema | data containers with openvr dependencies | uses timecontainers to model state |
| vr keys | containers to track indices of resources | stl containers |
| vr wrappers | wrappers around openvr apis | uses result concept and openvr apis |
| capture | data structure to snapshot vr state  template to traverse vr\_state | uses vrschema for most of the state  uses vrkeys to supplement the state  uses vrwrappers to abstract interface with openvr apis and errors |
| traverse | update a capture by querying openvr  read and write captures from a file | uses capture’s traverse template |
| capture  controller | api to trigger updates | relies on the caller to inform it of new keys and events  relies on the caller to tell it when to  snapshot new state |
| cursor controller | api to query captures | relies on the capture to set the current time  relies on the capture to know if the structure of the schema has changed |
| bridger | join interfaces/flows  can connect top an openvr\_dll  exports an upstream interface | the downstream interface could be openvr.dll or could be a |



Fileformat (from traverse/capture\_traverser.cpp)

|  |
| --- |
| header |
| save\_summary |
| keys |
| state |
| events |
| time\_stamps |
| key\_updates |
| state\_updates |

Bridger diagram