Mutating on realtime and non-realtime

- It’s impossible to have multiple threads mutate an object without locking all the involved threads

- Choose a single thread to be in charge of mutating (can be a realtime thread) and other threads pass messages to this thread which describe the changes they want to make

Mutating on realtime and non-realtime

```
struct SourceList {
    std::array<const float*, MAX_SOURCES> buffers = {};
    int numSources = 0;
};

RealtimeMutatable<SourceList> sharedSourceList;
AsyncCaller realtimeThreadCaller;
```

- Realtime audio thread which mixes audio from multiple sources
- User can add/remove sources via GUI (i.e. non realtime thread)
- Sources can also be added/removed from realtime event streams (i.e. realtime thread)