```cpp
class RealTimeAsyncUpdater::RealTimeAsyncUpdateDispatcher : private Timer
{
public:
    RealTimeAsyncUpdateDispatcher()
    {
        startTimerHz (25);
    }

    void add (RealTimeAsyncUpdaterMessage&);
    void remove (RealTimeAsyncUpdaterMessage&);

private:
    void timerCallback() override
    {
        serviceUpdaters();
    }

    void serviceUpdaters();

    CriticalSection lock;
    Array<RealTimeAsyncUpdaterMessage*> updaters;
};
```

```cpp
class RealTimeAsyncUpdater::RealTimeAsyncUpdateDispatcher : private Timer
{
public:
    RealTimeAsyncUpdateDispatcher()
    {
        startTimerHz (25);
    }

    void add (RealTimeAsyncUpdaterMessage&);
    void remove (RealTimeAsyncUpdaterMessage&);

private:
    void timerCallback() override
    {
        serviceUpdaters();
    }

    void serviceUpdaters();

    CriticalSection lock;
    Array<RealTimeAsyncUpdaterMessage*> updaters;
};
```

```cpp
void RealTimeAsyncUpdater::RealTimeAsyncUpdateDispatcher::add (RealTimeAsyncUpdaterMessage& m)
{
    const ScopedLock sl (lock);
    jassert (! updaters.contains (&m));
    updaters.add (&m);
}

void RealTimeAsyncUpdater::RealTimeAsyncUpdateDispatcher::remove (RealTimeAsyncUpdaterMessage& m)
{
    const ScopedLock sl (lock);
    updaters.removeFirstMatchingValue (&m);
}

void RealTimeAsyncUpdater::RealTimeAsyncUpdateDispatcher::serviceUpdaters()
{
    const ScopedLock sl (lock);

    for (auto updater : updaters)
        updater->serviceMessage();
}
```