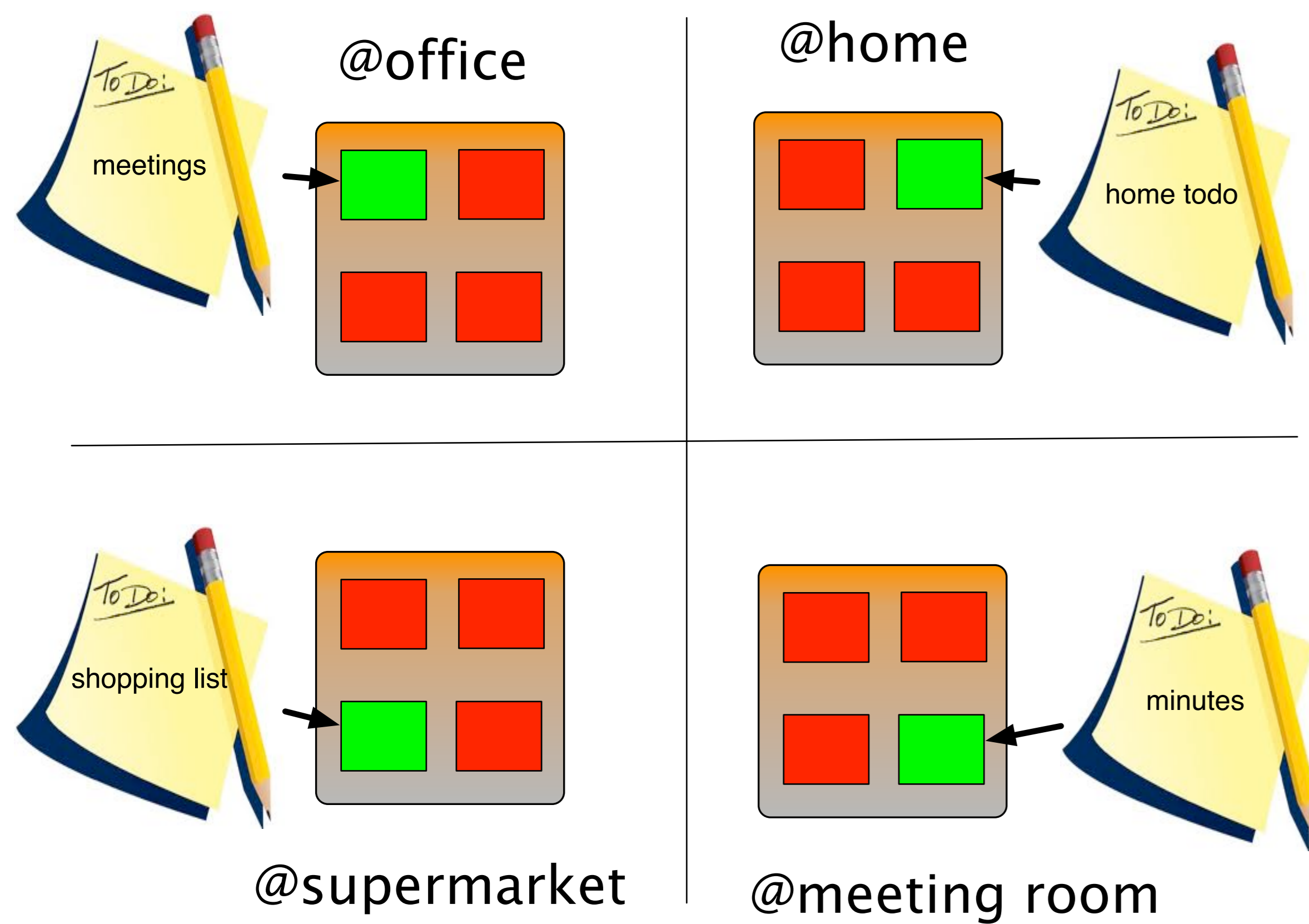


Motivation: a context-aware post-it app

Application behaviour as a user moves about

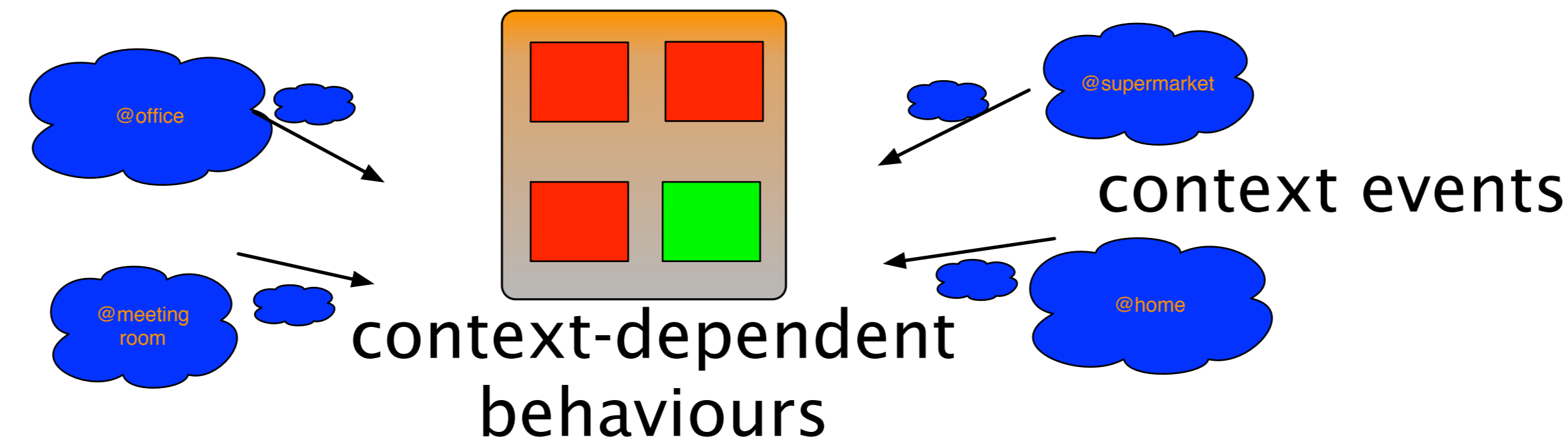


■ active computation ■ suspended computation

Observations

- Computation is mostly driven by context changes
- Dynamic behaviour switching
- Computations arbitrarily stop and resume
- State needs to be preserved between context changes

Problem Statement



Reacting to unpredictable order of context changes

- Use of event handlers leads to inversion of control
- Deciding on which computation behaviour to make available
- Computation behaviours may conflict each other

Manual state management

- Saving application state
- Recovery: reinitialising computation state

```
(define (shopping location)
  (if (supermarket? location)
      ; initialise application
      ; restore/recover application state
      (shopping-list)))
```

► *hard & error prone!*

```
(if (left-supermarket? location)
    ; save application state
    ; stop application )))
```

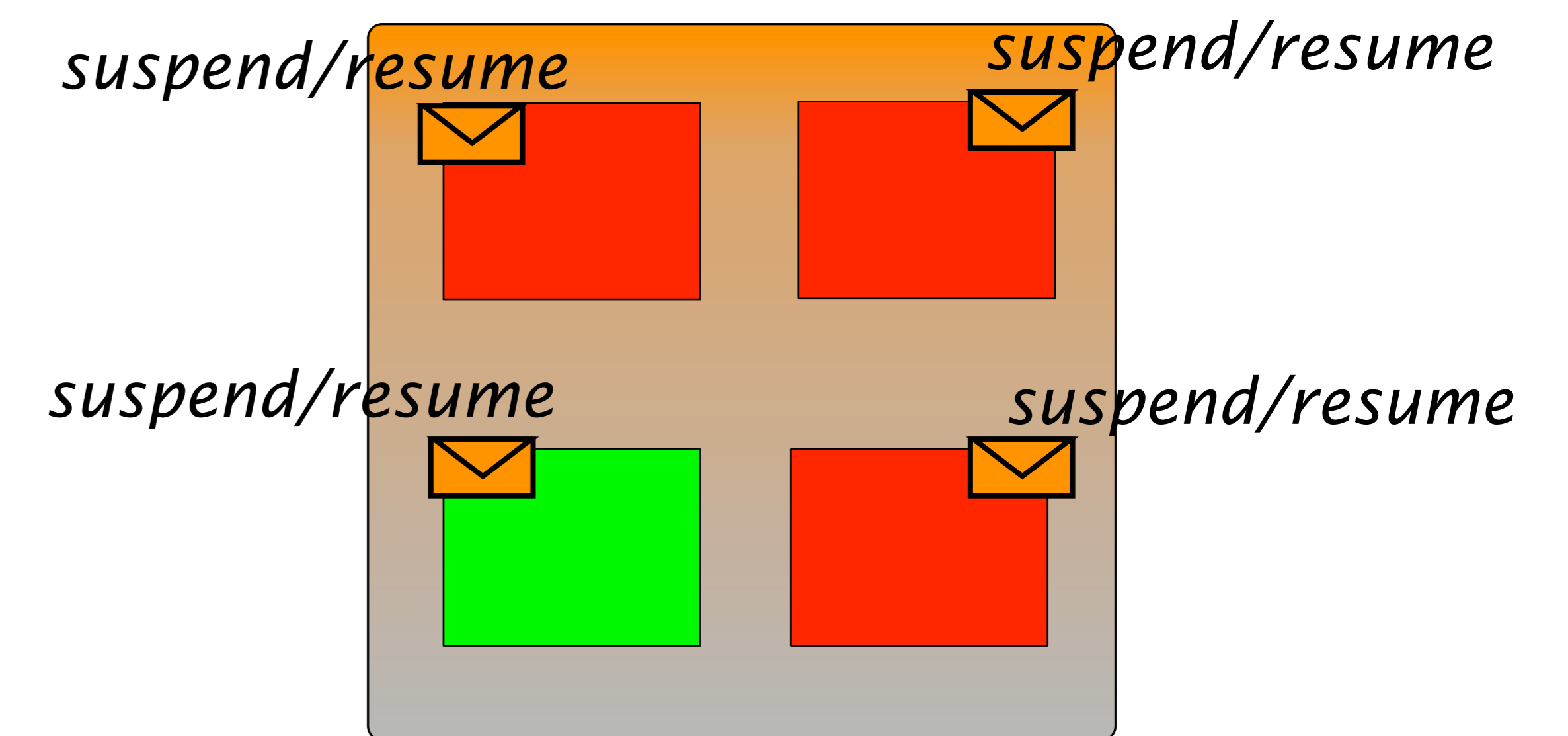
Managing execution state of an ongoing computation



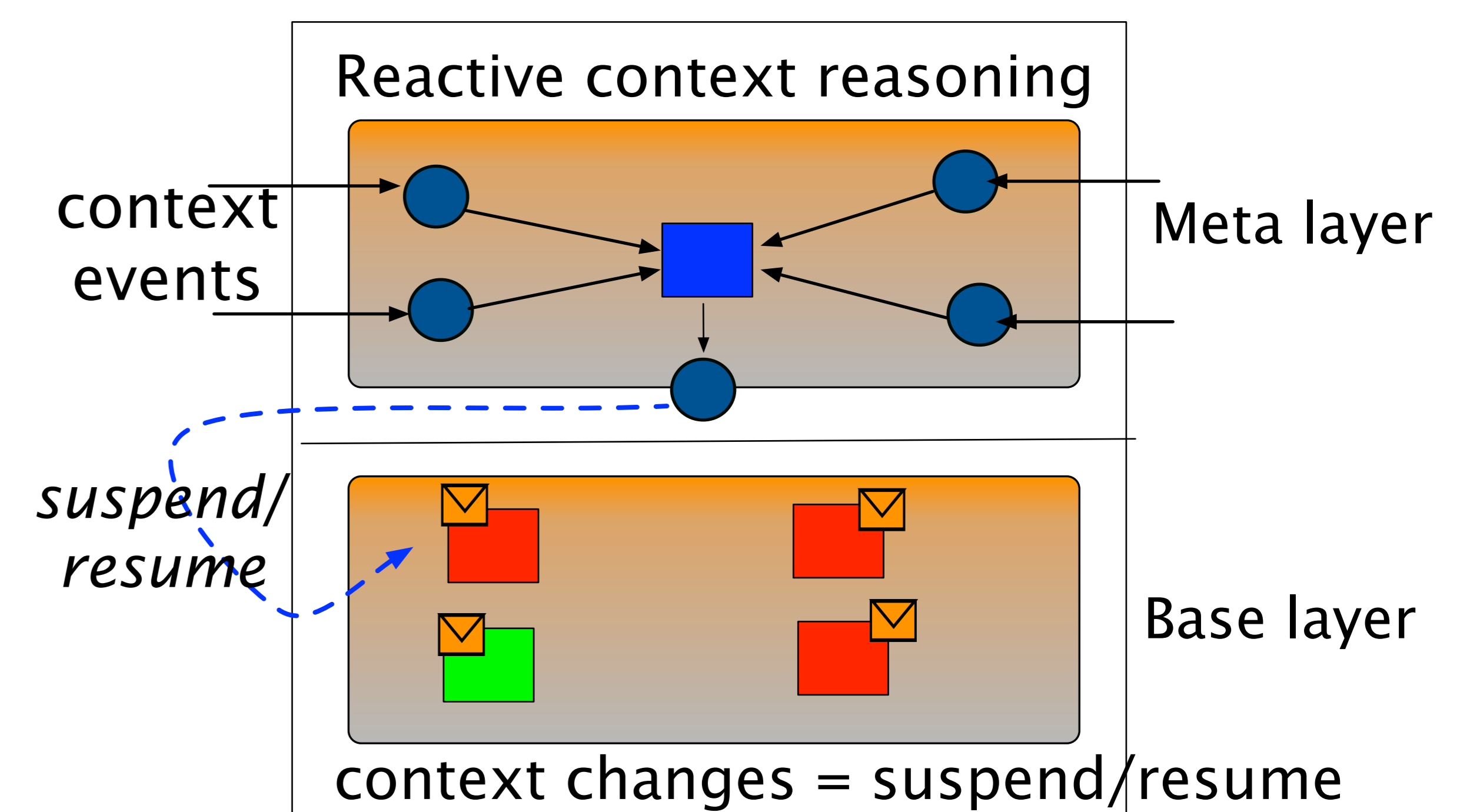
- When the user leaves office: *abort?*

Computations as continuations

- Modularising computations as continuations
- A computation encloses behaviour and state
- A computation responds to resume and suspend messages



Reactive programming+continuations



Implementation platform

- **iScheme[1]**: a Scheme implementation that runs on iPhone/iPod and iPad devices