



Calendar Integration with Event Kit

A story of a boy, a girl, and a universe

Ed Voas

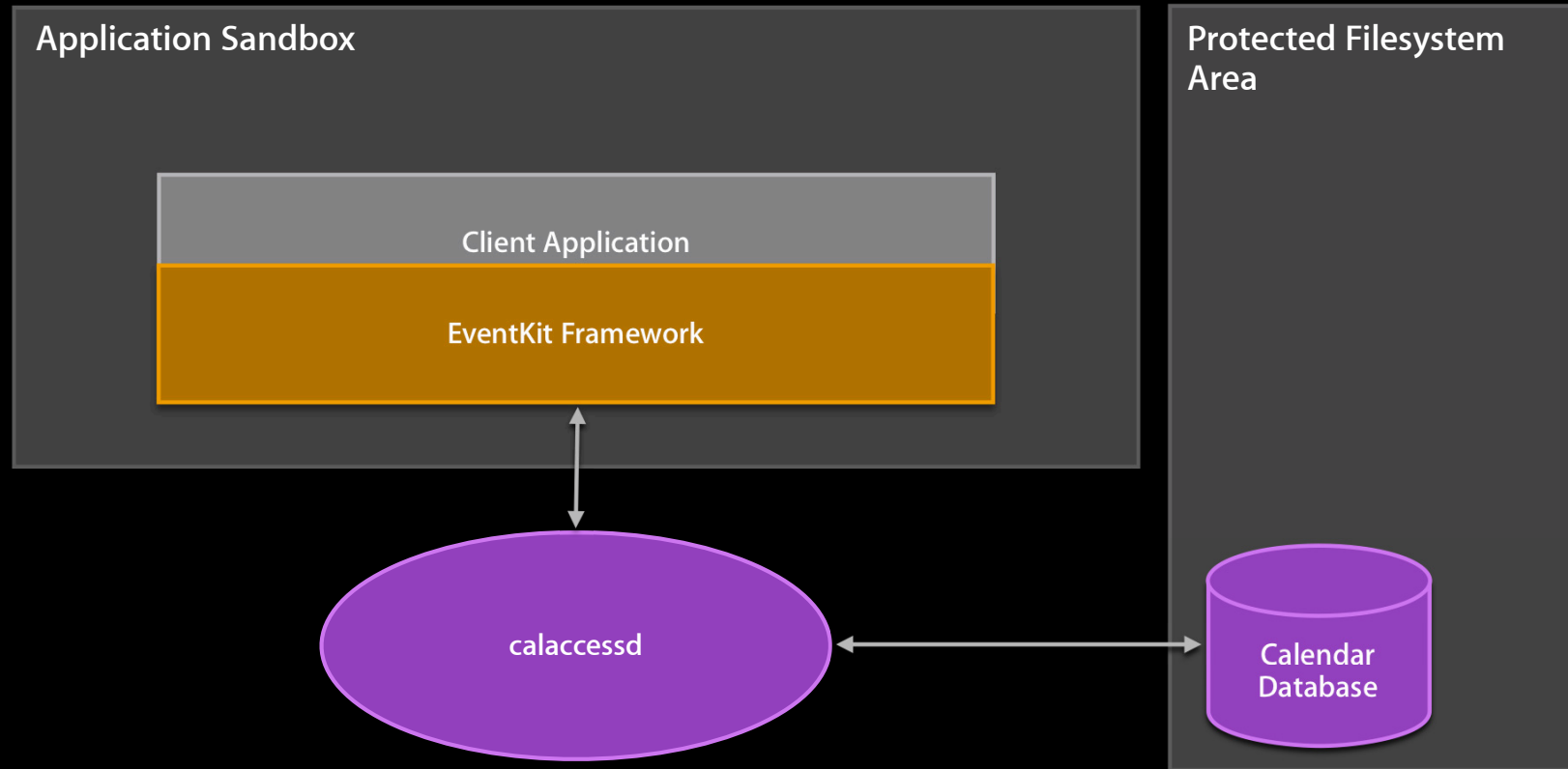
Senior Software Engineer
iOS — Calendar

What Is EventKit?

- High-level API to access calendar data on device
- Two sets of APIs
 - Calendar access (EventKit)
 - View Controllers (EventKitUI)
- Not a low-level sync API
 - But, sync is automatic

Always be clear about when you are changing calendar data.

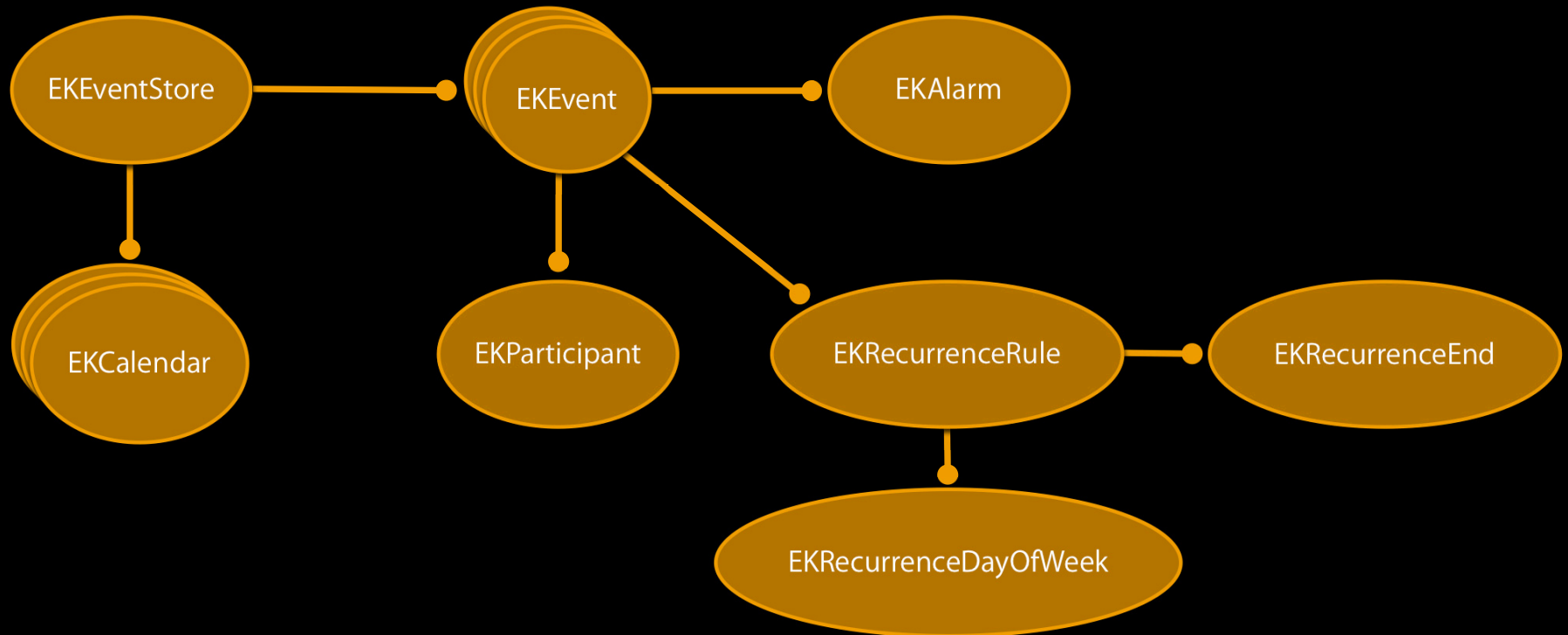
Architecture



API and Classes

UIKit Framework

EventKit Classes

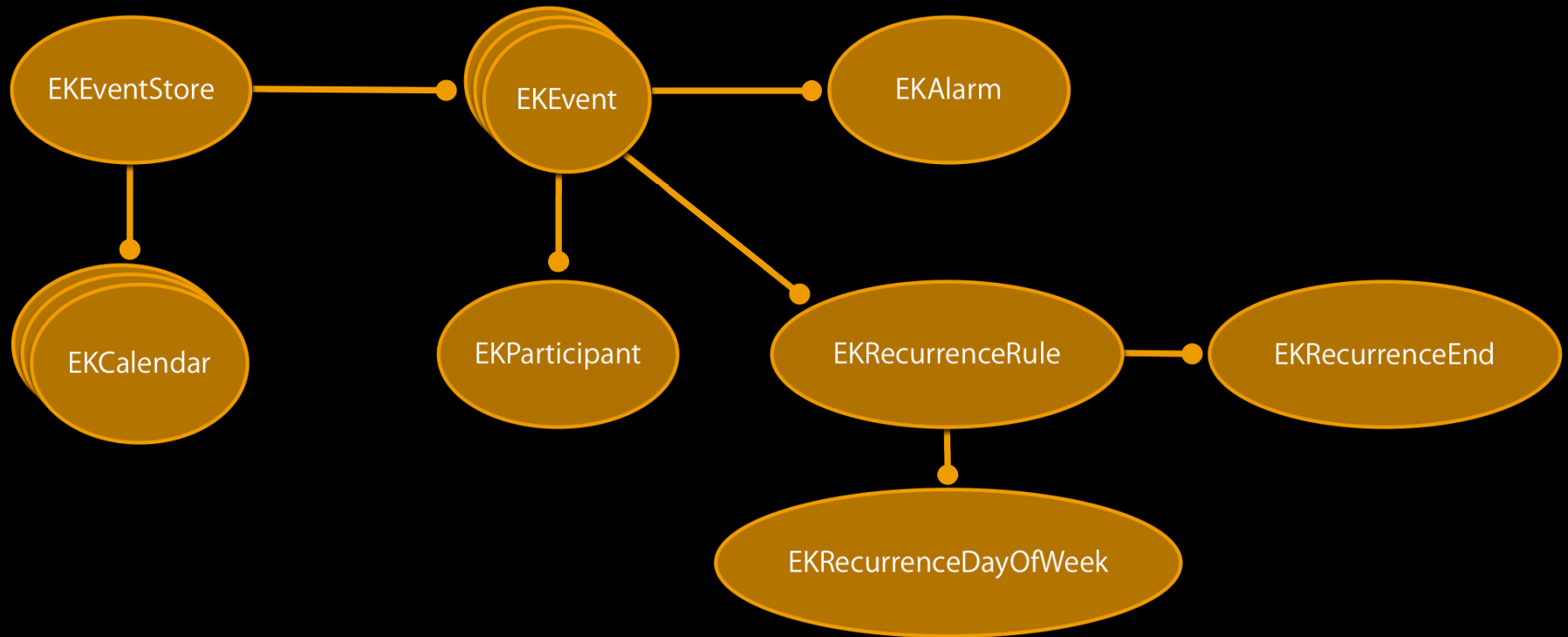


EKEventStore

- Your connection to the database
- Should generally be long-lived
- All objects are bound to the event store they came from
 - Can't cross the streams

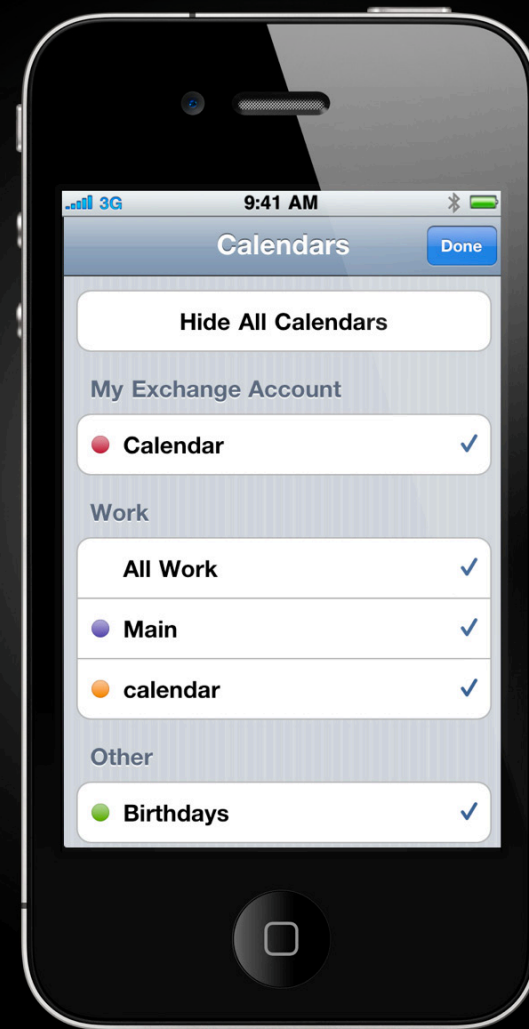
```
EKEventStore *myStore = [[EKEventStore alloc] init];
```

EventKit Classes



EKCalendar

- Supports Exchange, CalDAV, MobileMe, Local
- Sync is automatic
- Read-only vs. Read-write
- Cannot create new calendars in this release



Default Calendar



Accessing Calendars

```
EKEventStore *store = [[EKEventStore alloc] init];
```

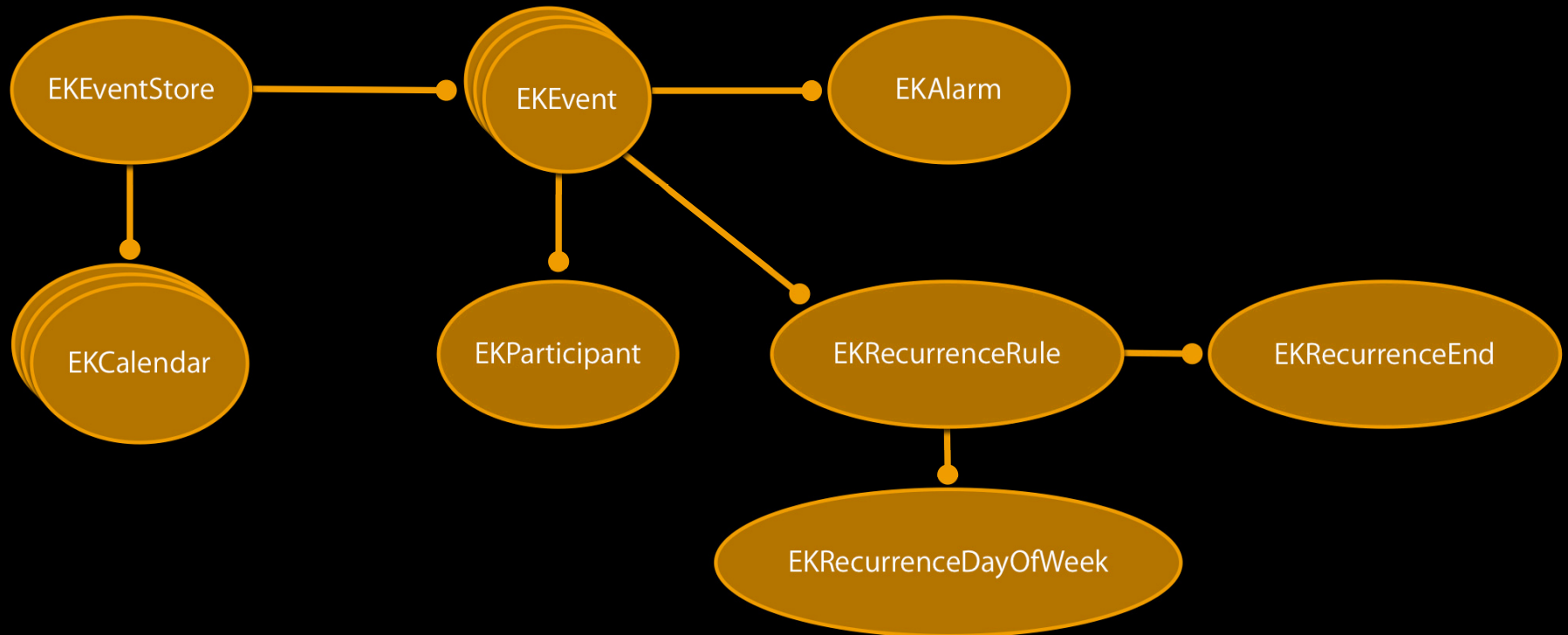
```
NSArray *calendars = store.calendars;
```

```
EKCalendar *myCal = store.defaultCalendarForNewEvents;
```

```
// fetch some properties  
NSTitle *title = myCal.title;  
CGColorRef color = myCal.CGColor;  
EKCalendarType type = myCal.type;
```

```
// Can we add or modify events?  
BOOL writable = myCal.allowsContentModifications;
```

EventKit Classes



EKEvent

- Represents occurrence of an event
- Can get and set most basic properties



Event Properties



Creating Events

```
EKEventStore *store = [[EKEventStore alloc] init];
```

```
EKEvent *event = [EKEvent eventWithEventStore:store];
```

```
event.title = @"My First Event";  
event.startDate = startDate;  
event.endDate = endDate;  
event.calendar = store.defaultCalendarForNewEvents;
```

```
BOOL saved = [store saveEvent:event span:EKSpanThisEvent  
              error:&error];
```

Event Identifiers

- Unique identifier for an event
 - Exist only after an event has been saved
- Can change if...
 - Event is moved to a different calendar/account
 - Event is detached/split
- If ID changes, you won't be able to look it up
 - Save identifying information (title, start date, etc.) and search

```
NSString *eventId = event.eventIdentifier;
```

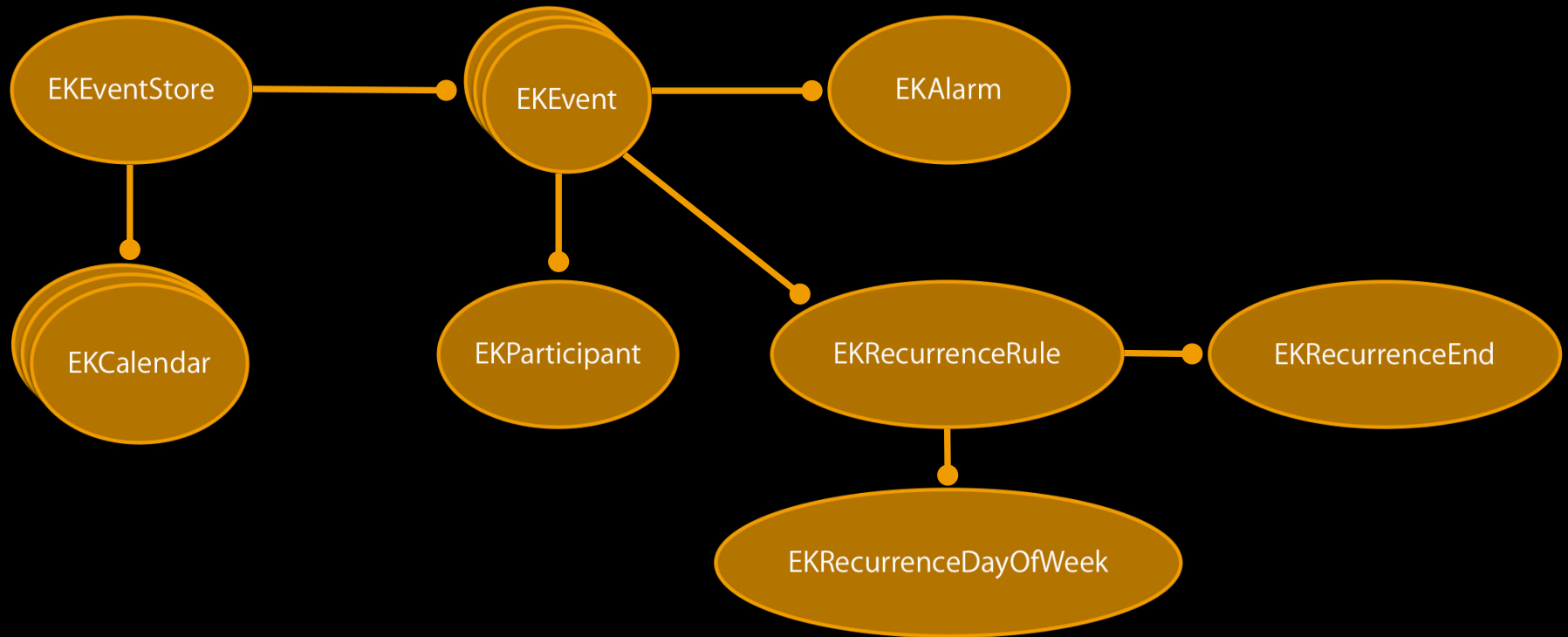
Deleting Events

```
EKEventStore *store = [[EKEventStore alloc] init];
```

```
EKEvent *event = [store eventWithIdentifier:eventId];
```

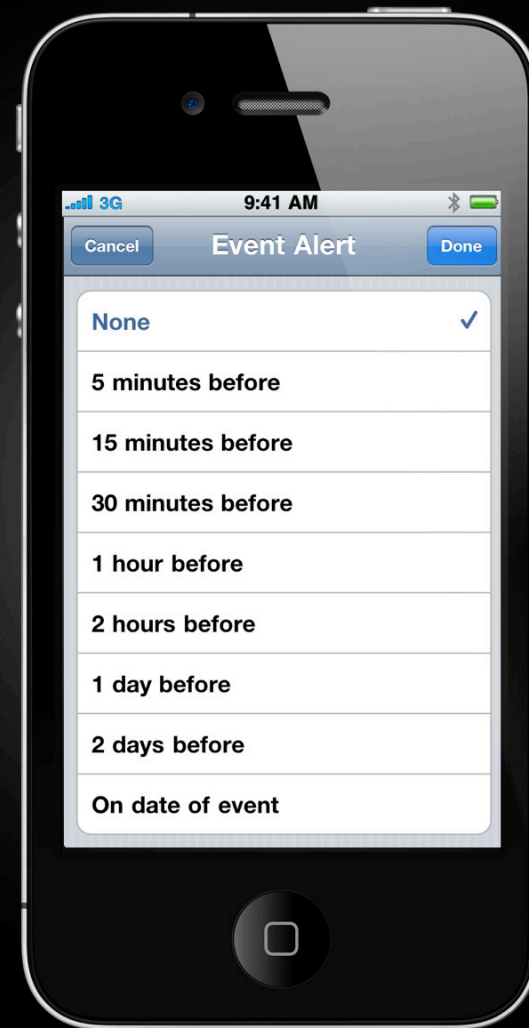
```
BOOL saved = [store removeEvent:event span:EKSpanThisEvent  
              error:&error];
```


EventKit Classes



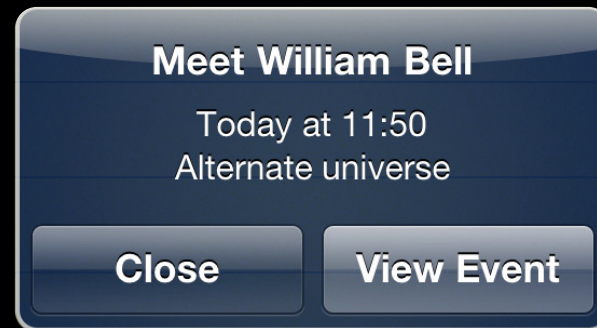
Alarms

- Relative to start date
 - Specified in negative seconds
- Displays standard Calendar alert
- Calendars have different limits
 - Best to stick to just one



Alarms Are Not UILocalNotifications

- Alarms
 - Require you to create an event
 - Visible in Calendar
 - Action button will take user to event
- UILocalNotifications
 - Much more general
 - Don't involve Calendar
 - Action buttons calls your app



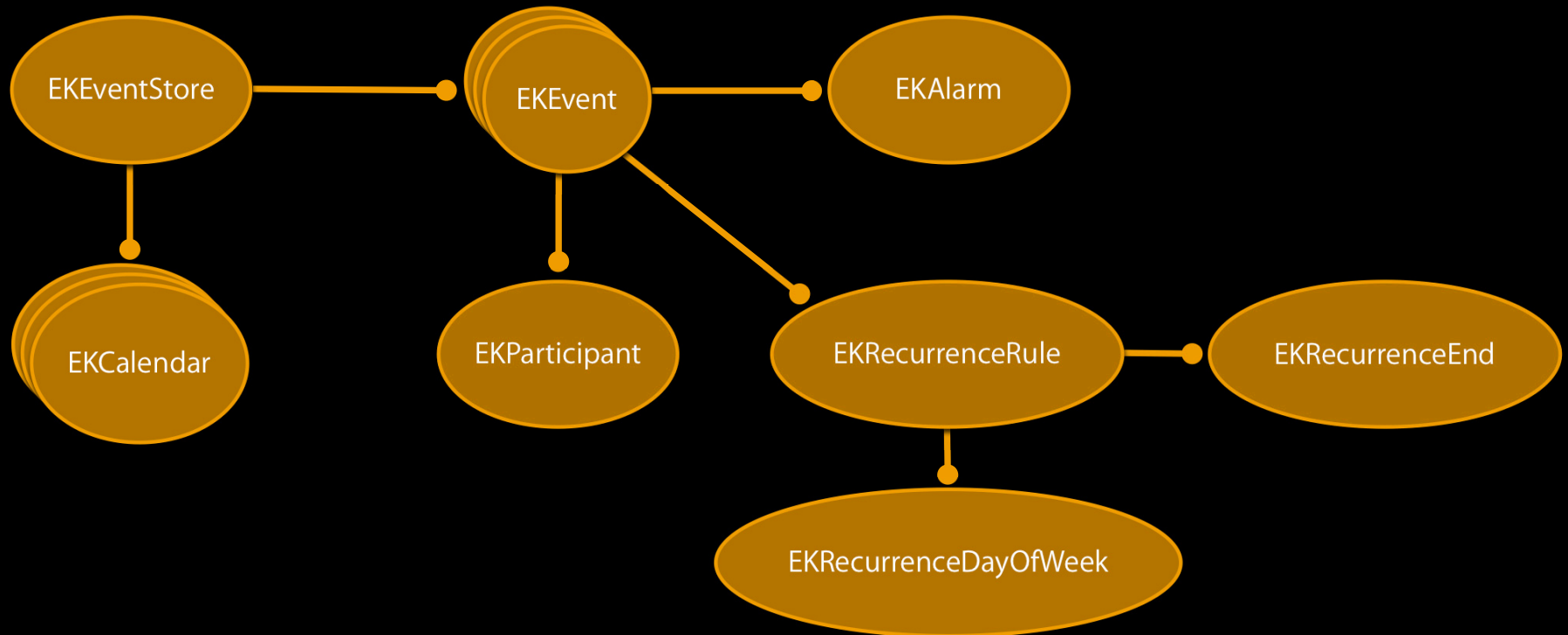
Adding Alarms

```
// specify time in negative seconds from start time  
EKAlarm* alarm = [EKAlarm alarmWithRelativeOffset:-900];
```

```
[event addAlarm:alarm];
```

```
BOOL saved = [store saveEvent:event span:EKSpanThisEvent  
              error:&error];
```

EventKit Classes

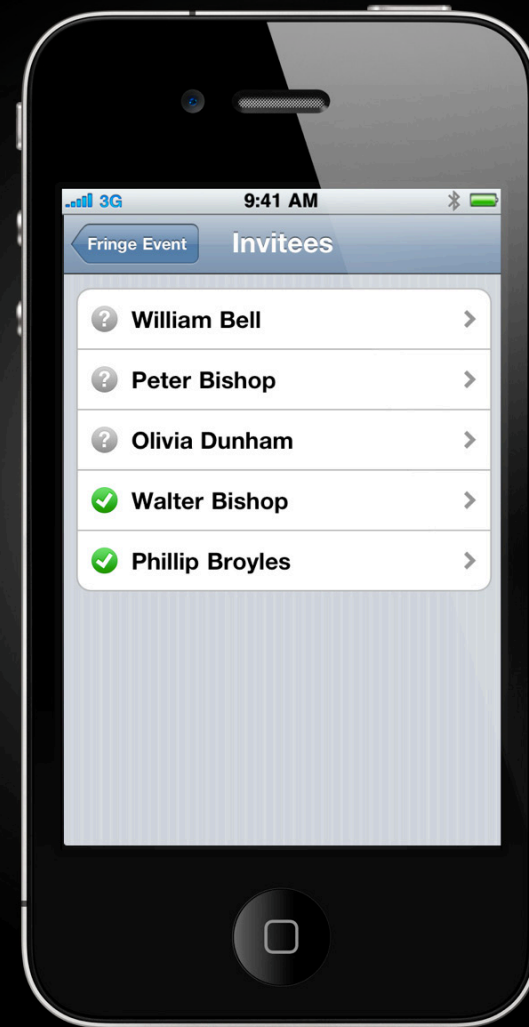


EKParticipant

- Represents organizer or attendee
- Can get attendee status for an invite
 - Might not always be present (primarily with Exchange < 2010)
- Can get Address Book record from a participant
 - Loose coupling—via email
- Read-only in this release

```
EKParticipant *organizer = event.organizer;
```

```
NSArray *attendees = event.attendees;
```



Demo

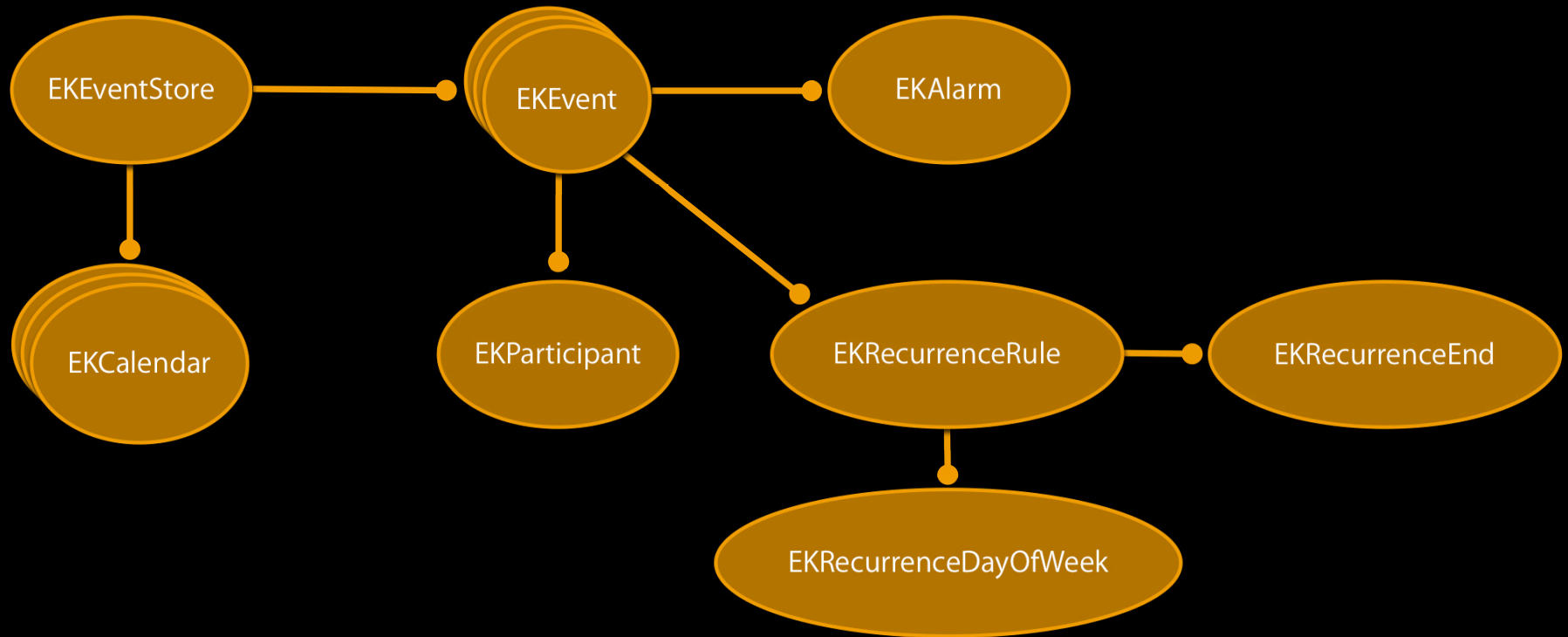
Sharkster

Glen Steele
iOS Software Engineer

Demo Summary

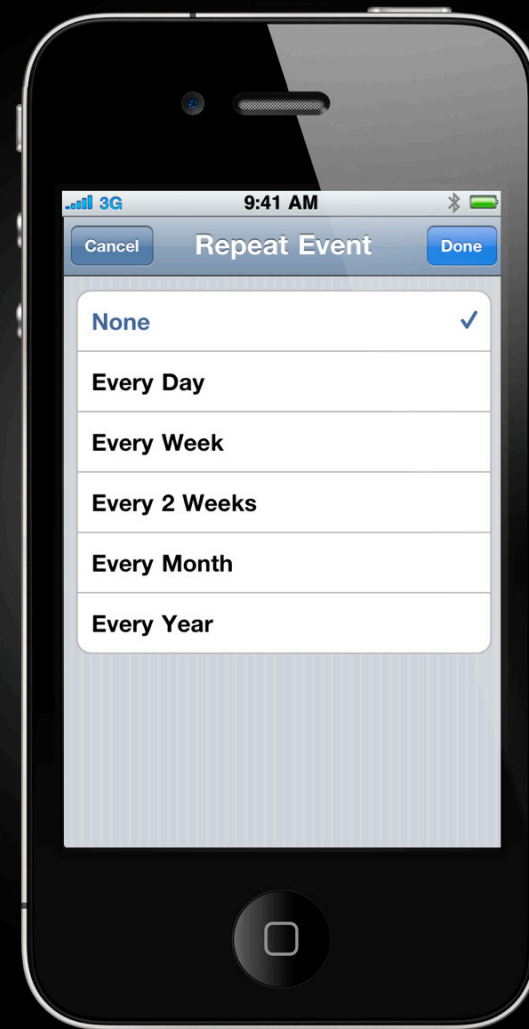
- Inserting Events
 - Create new EKEvent using EKEventStore
 - Set properties
 - Save using `-[EKEvent saveEvent:span:error:]`
- Deleting Events
 - Fetch via `eventIdentifier`
 - Remove using `-[EKEvent removeEvent:span:error:]`

EventKit Classes



Recurrence Rules

- Control when an event repeats
- UI only allows simple rules
- API allows complex rules
 - Third Tuesday of every month
 - Last weekday of the year



Setting Recurrence Rule

```
// Weekly meeting that never ends...  
//           (but then, they all feel like that)
```

```
EKRecurrenceRule *rule = [[EKRecurrenceRule alloc]  
    initWithRecurrenceFrequency:EKRecurrenceFrequencyWeekly  
    interval:1 end:nil];
```

```
event.recurrenceRule = rule;
```

```
[store saveEvent:event span:EKSpanThisEvent error:&error];
```

Recurrence End

```
EKRecurrenceEnd *end =  
    [EKRecurrenceEnd recurrenceEndWithEndDate:date];
```

```
// OR
```

```
EKRecurrenceEnd *end =  
    [EKRecurrenceEnd recurrenceEndWithOccurrenceCount:5];
```

```
EKRecurrenceRule *rule = [[EKRecurrenceRule alloc]  
    initWithRecurrenceFrequency:EKRecurrenceFrequencyWeekly  
    interval:1 end:end];
```

```
event.recurrenceRule = rule;
```

```
[store saveEvent:event span:EKSpanThisEvent error:&error];
```

Fetching Events

Fetching Events

- Search via identifier
 - `-[EKEventStore eventWithIdentifier:]`
 - Yields first event occurrence
- Search via predicate
 - `-[EKEventStore predicateForEventsWithStartDate:endDate:calendars:]`
 - Post-filtering may be necessary

Fetching Events

```
NSPredicate* predicate;  
  
predicate = [store predicateForEventsWithStartDate:startDate  
            endDate:endDate  
            calendars:store.calendars];
```

```
NSArray *events = [store eventsMatchingPredicate:predicate];
```

```
// yields unordered array, so sort it...
```

```
NSMutableArray* mutEvents =  
    [[events mutableCopy] autorelease];  
  
[mutEvents sortUsingSelector:  
    @selector(compareStartDateWithEvent:)];
```

Async Fetching

- Can use `dispatch_async`, `NSOperation`, etc.

```
dispatch_queue_t aQueue = dispatch_get_global_queue(
    DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
```

```
dispatch_async(aQueue, ^{
    NSArray* array = [store eventsMatchingPredicate:predicate];

    dispatch_async(dispatch_get_main_queue(), ^{
        [self setEvents:array];
    });
});
```

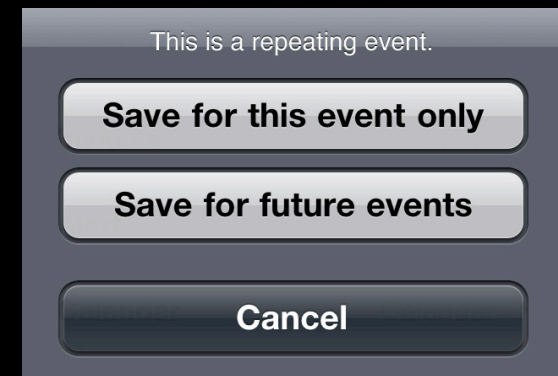

Modifying Events

Modifying Events

- Set the desired properties
- [EKEventStore saveEvent:span:error:]
- Simple events are, well, simple
- Recurring events have different behaviors

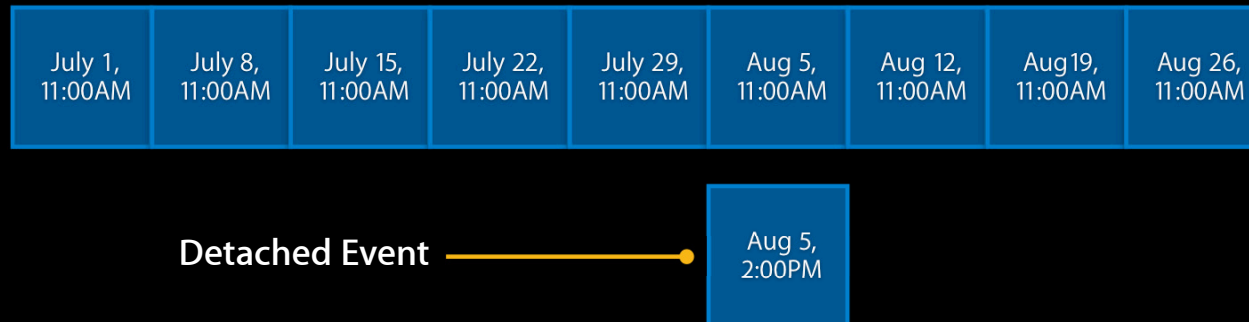
Modifying Recurring Events

- Two ways to modify a recurring event
 - EKSpanThisEvent
 - EKSpanFutureEvents
- Behave very differently from one another
- Both choices are valid when you change any aspect of the event
- Exception: recurrence rule
 - Can only change future events



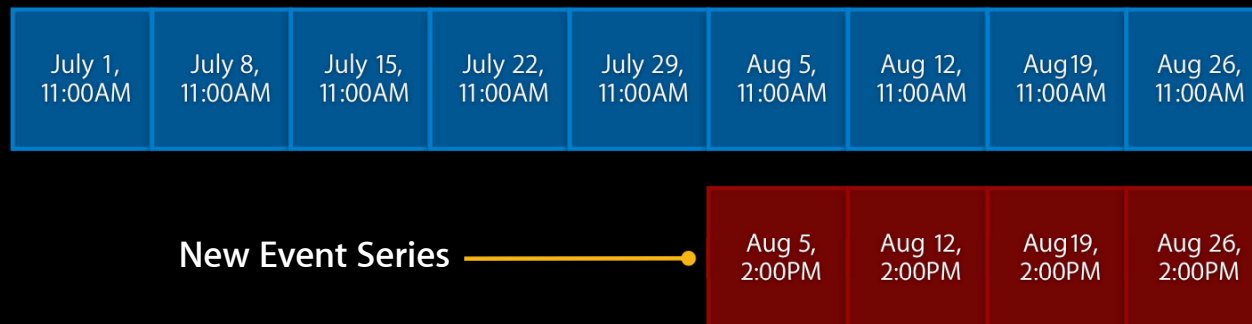
Simple Detach—EKSpanThisEvent

- Creates new event as a child of the series
- Child is still considered part of the series



Future Detach — EKSpanFutureEvents

- Creates new event as separate entity
- There is no relation to the original



Delete 'This Event'

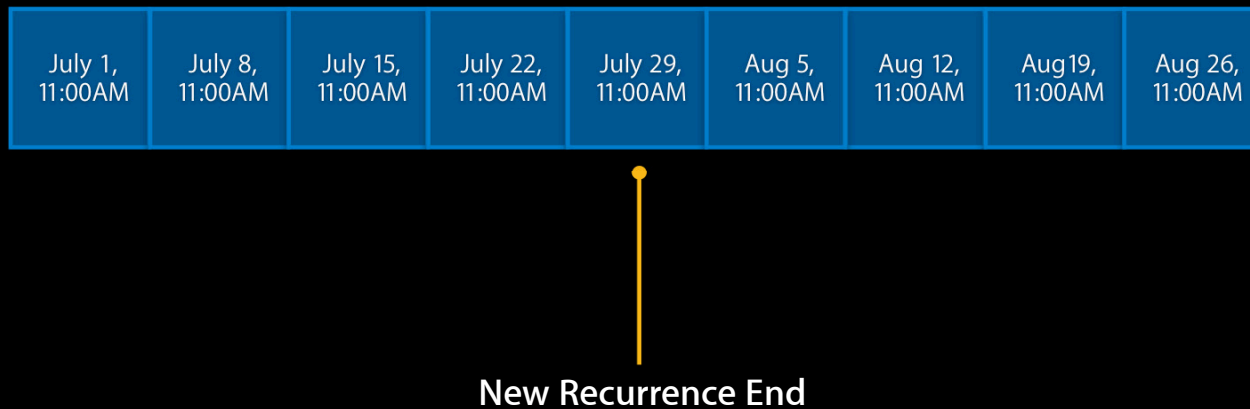
- Simply marks hole with an exception date

July 1, 11:00AM	July 8, 11:00AM	July 15, 11:00AM	July 22, 11:00AM	July 29, 11:00AM	Aug 5, 11:00AM	Aug 12, 11:00AM	Aug19, 11:00AM	Aug 26, 11:00AM
--------------------	--------------------	---------------------	---------------------	---------------------	-------------------	--------------------	-------------------	--------------------

Exception Date
Aug 5, 11AM

Delete Future

- Simply ends recurrence at given point



Demo

Sharkster

Glen Steele
iOS Software Engineer

Demo Summary

- Create the initial event
 - Set properties
 - Set Recurrence Rule
 - Save using `-[EKEvent saveEvent:span:error:]`
- To delete
 - Get it using `eventIdIdentifier` or a predicate
 - Remove using `-[EKEvent removeEvent:span:error:]` with `EKSpanFutureEvents`

Database Changes

- EKEEventStoreChangedNotification
 - Can happen at any time (other applications, sync)
 - Course-grained
 - Coalesced—if suspended, will be received at next resume
- When received, treat EKEEvent and EKCalendar objects as invalid
- [EKEEvent refresh] can be used to test validity, but is not cheap
- Rules of thumb
 - Refresh an event you are actively viewing/editing
 - Refetch all other events

View Controllers

UIKit

Detail View— EKEventViewController

- Same ol' view you know
- Allows user to respond to invites
- Can optionally allow editing
- If current event changes, will refresh
- If current event is deleted, will auto-pop



Displaying Event Details

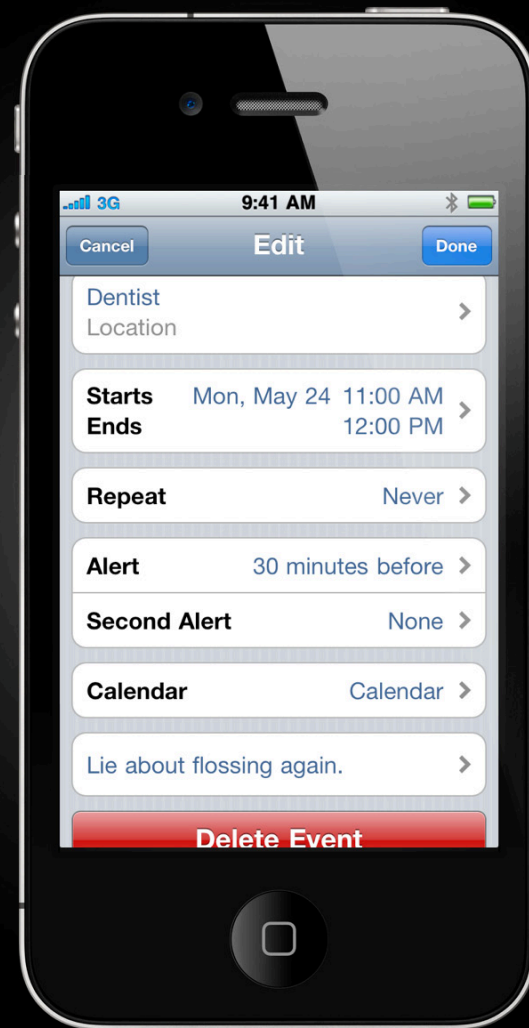
```
EKEventViewController *controller =  
    [[EKEventViewController alloc] init];
```

```
controller.event = myEvent;  
controller.allowsEditing = YES;
```

```
[self.navigationController pushViewController:controller  
    animated:YES];  
[controller release];
```

Edit View — EKEventEditViewController

- Edits existing events
- Create new events
 - Optionally pass in partially constructed event
- Fills in default time/calendar if needed
- Delegate
 - Called when event saved, canceled, or deleted
 - Allows overriding default calendar



Displaying the Editor

```
EKEventEditViewController *controller =  
    [[EKEventEditViewController alloc] init];
```

```
controller.event = myEvent;  
controller.eventStore = myEventStore;  
controller.editViewDelegate = self;
```

```
[self.navigationController  
    presentModalViewController:controller animated:YES];  
[controller release];
```

Demo

Sharkster

Glen Steele
iOS Software Engineer

Demo Summary

- Showing event detail
 - Find event using eventIdentifier or search predicate
 - Create EKEventViewController
 - Set controller properties
 - Push onto navigation stack

Wrap-Up

Working with the Simulator

- No Calendar application in this SDK release
- Can simply add events programmatically to default calendar
- Can use Birthday calendar
 - Populates from Address Book
 - Does not sync
 - Read-only

Wrap-Up

- High-level access to calendar data
- Easy-to-use view controllers for display/edit
- This is 1.0
 - Your requests help us prioritize. File bugs!

Related Labs/Sessions

Calendar Store and Event Kit Lab

Application Frameworks Lab B
Friday, 9:00AM

Introducing Blocks and Grand Central Dispatch on iPhone

Russian Hill
Wednesday, 11:30AM

More Information

Mark Malone

Integration Technologies Evangelist
mgm@apple.com

iPhone Dev Center

<http://developer.apple.com/iphone>

Apple Developer Forums

<http://devforums.apple.com>



