Integrating with Facebook, Twitter, and Sina Weibo

Social Engineering

Session 306

Lestat Ali
iOS Software Engineer

Julien Robert
OS X Software Engineer

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Social Networks—iOS 6

New

Social Networks—iOS 6

New

Twitter

Social Networks—iOS 6

New

Twitter

Facebook

Social Networks—iOS 6

New

Twitter

Facebook

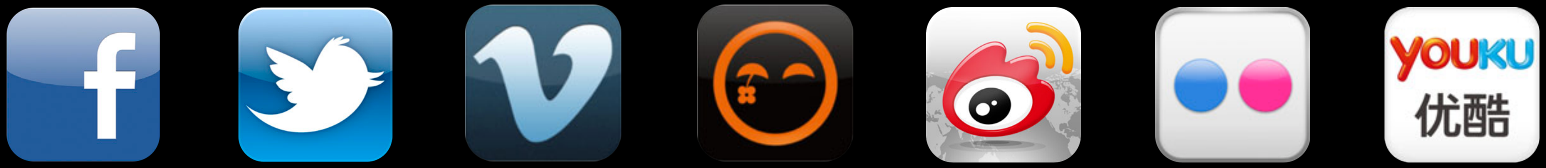
Sina Weibo

Social Networks—Mountain Lion

New

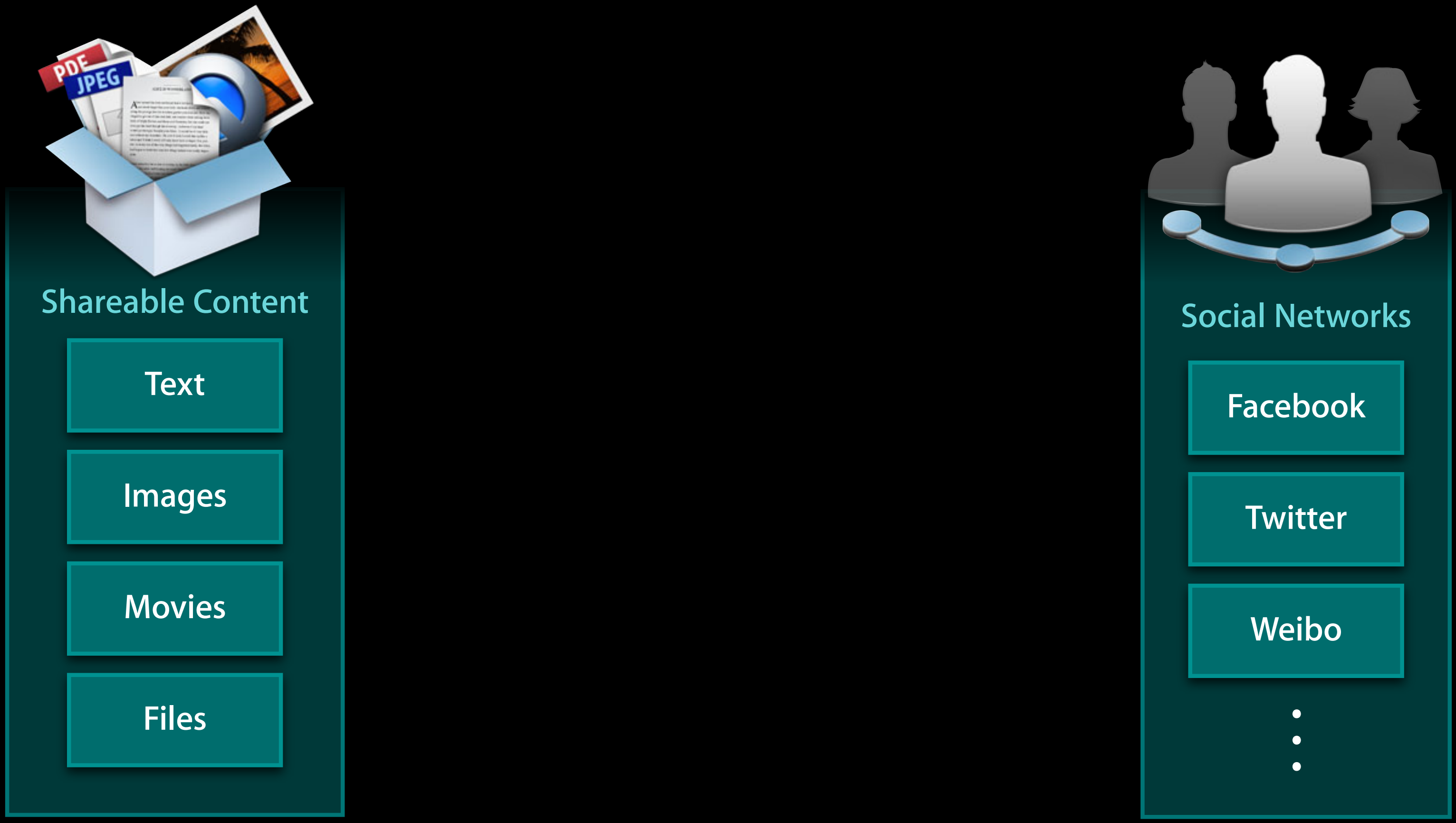
Social Networks—Mountain Lion

New



Shareable Content

- Text
- Images
- Movies
- Files

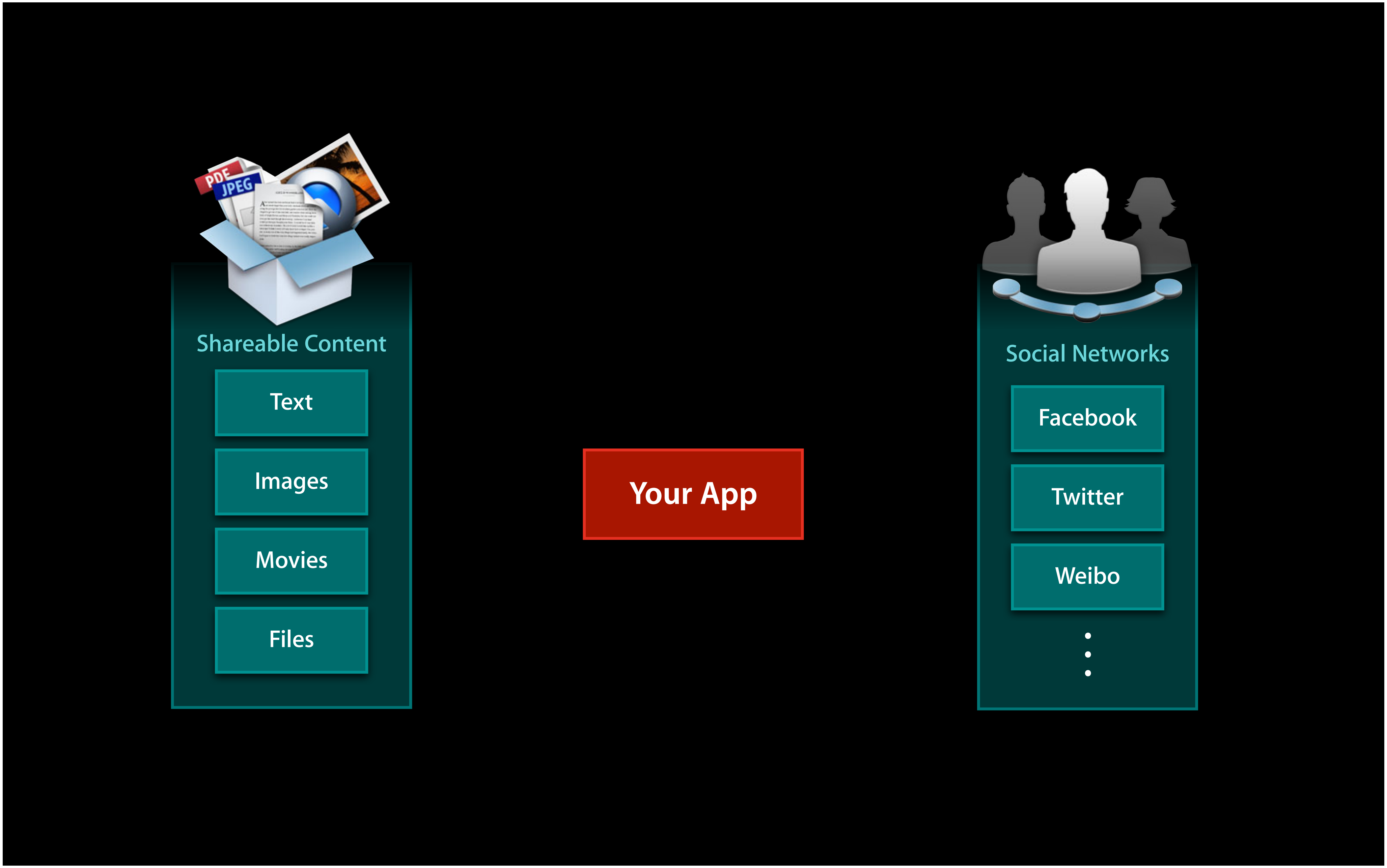


Shareable Content

- Text
- Images
- Movies
- Files

Social Networks

- Facebook
- Twitter
- Weibo
- ⋮



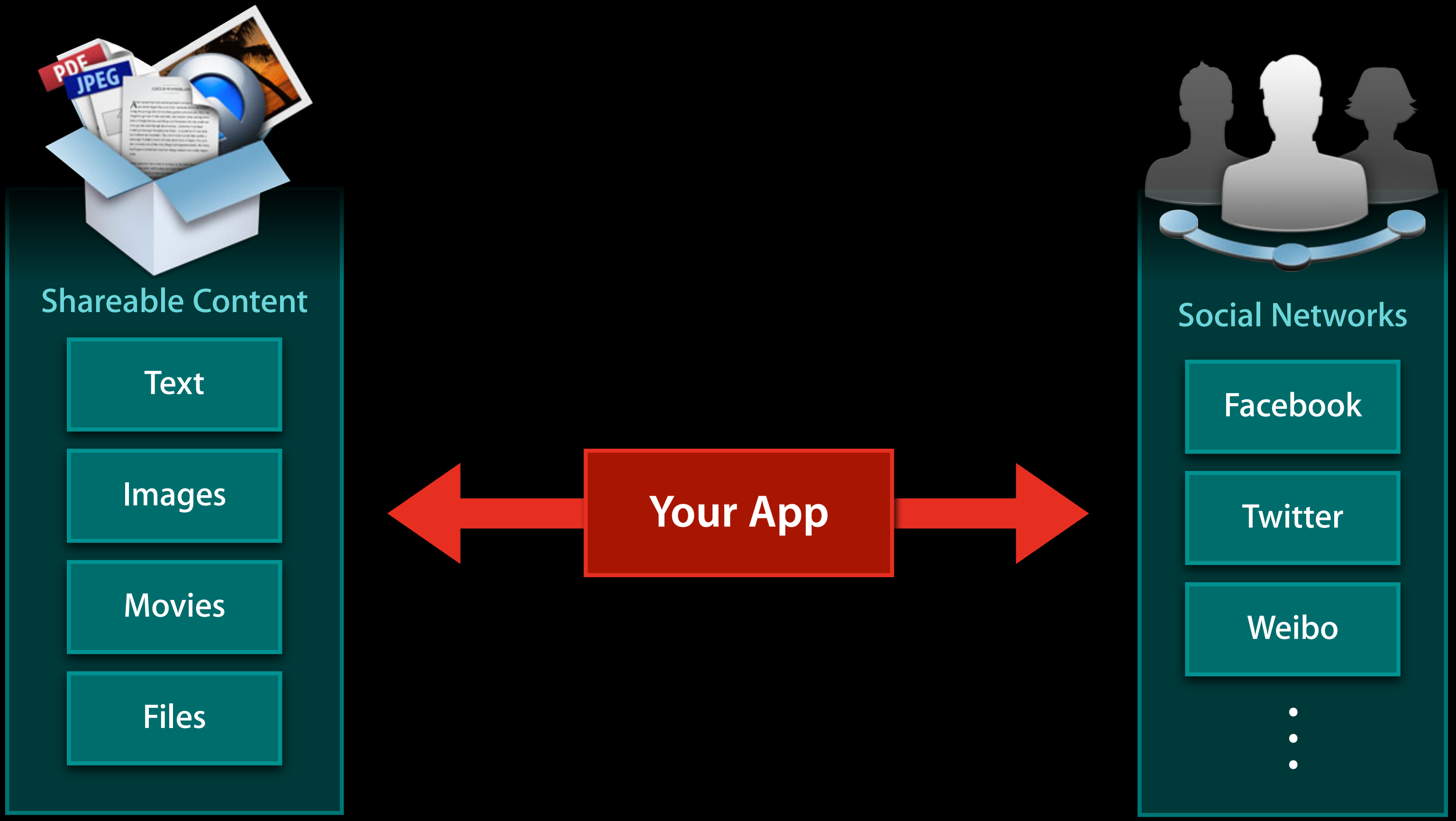
Shareable Content

- Text
- Images
- Movies
- Files

Your App

Social Networks

- Facebook
- Twitter
- Weibo
- ⋮



Shareable Content

Text

Images

Movies

Files

Your App

Social Networks

Facebook

Twitter

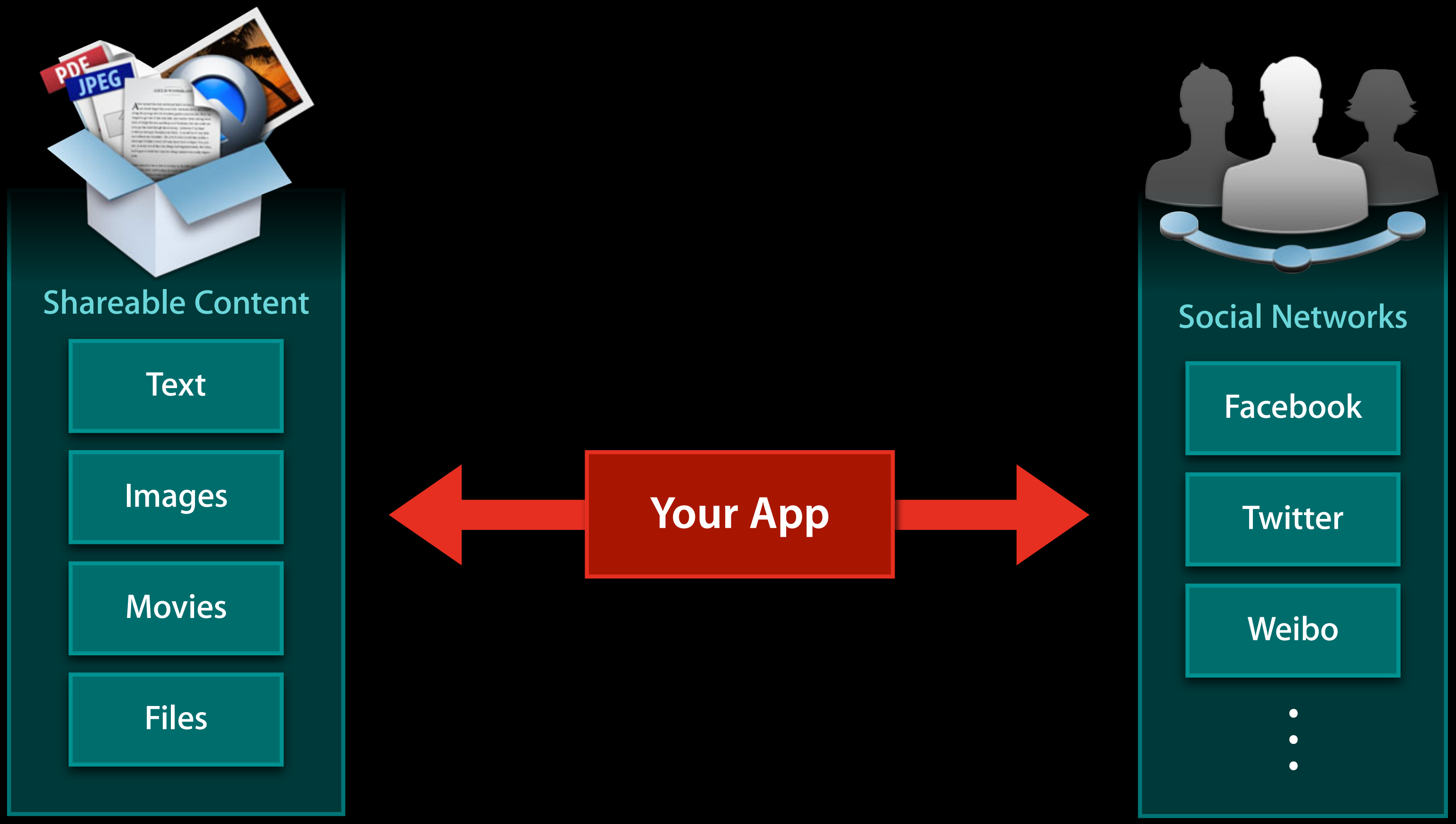
Weibo

⋮

What You Will Learn

- UI integration on iOS
- UI integration on OS X
- Advanced integration

UI Integration



Shareable Content

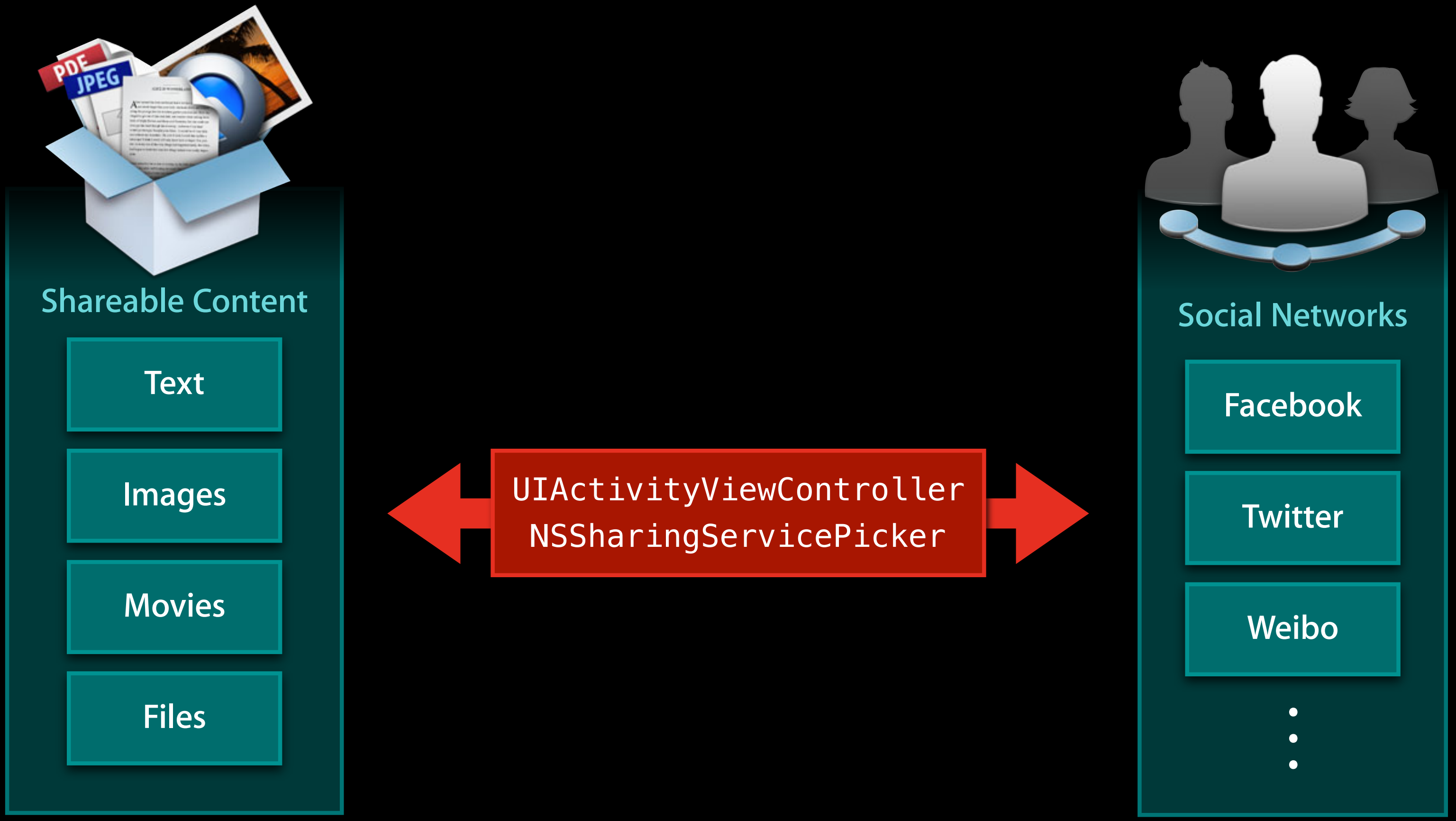
- Text
- Images
- Movies
- Files

Your App

Social Networks

- Facebook
- Twitter
- Weibo

⋮



Shareable Content

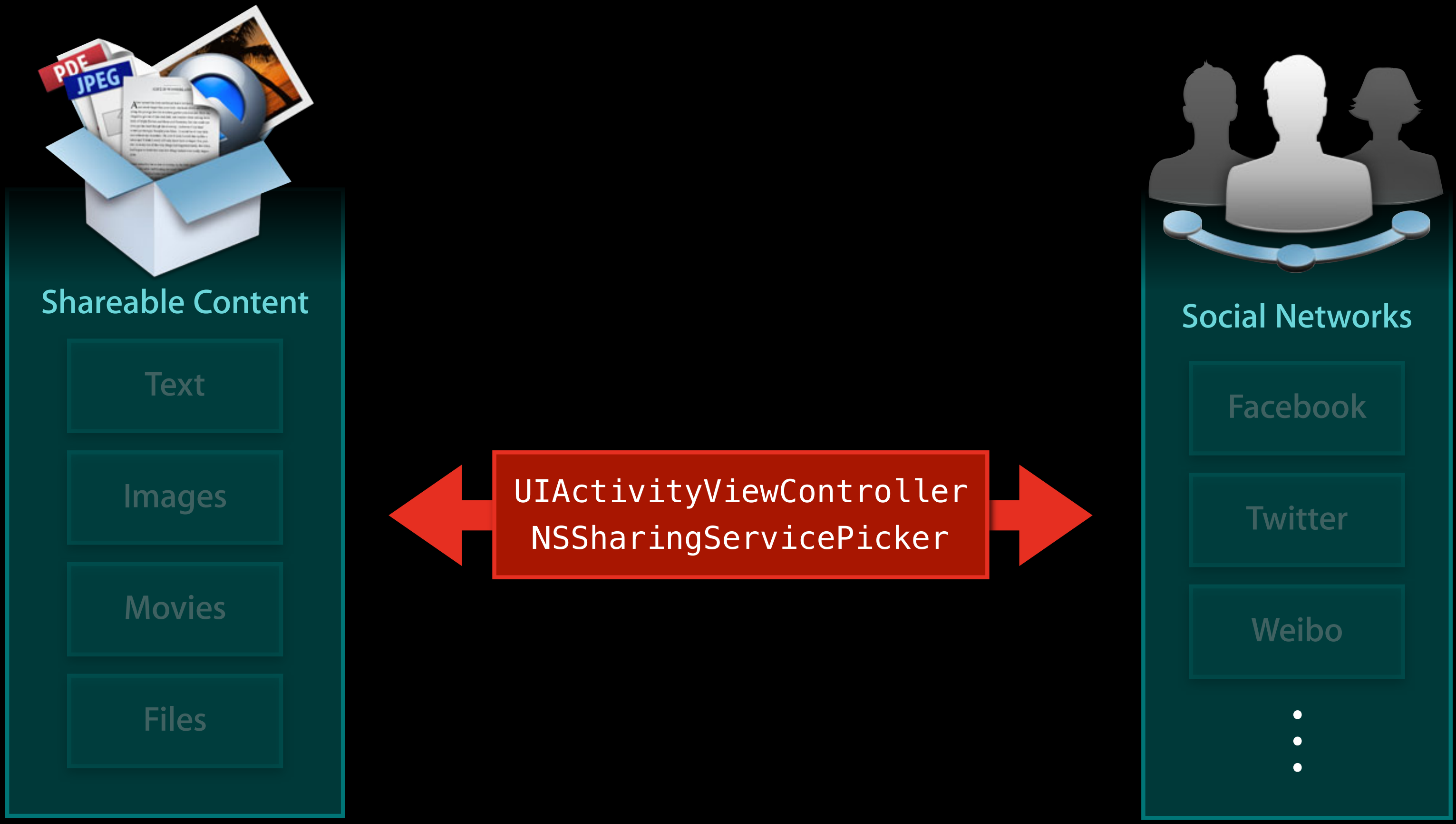
- Text
- Images
- Movies
- Files

UIActivityViewController
NSSharingServicePicker

Social Networks

- Facebook
- Twitter
- Weibo

⋮



Shareable Content

Text

Images

Movies

Files

UIActivityViewController
NSSharingServicePicker

Social Networks

Facebook

Twitter

Weibo

⋮

Benefits of Using Built-in UI

Benefits of Using Built-in UI

- Creates a consistent user experience

Benefits of Using Built-in UI

- Creates a consistent user experience
- Uses single sign-on

Benefits of Using Built-in UI

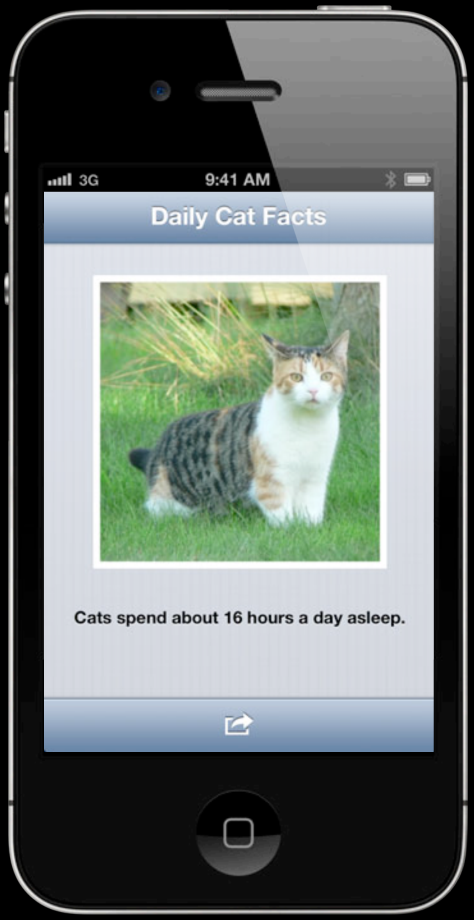
- Creates a consistent user experience
- Uses single sign-on
- Easy to integrate

Benefits of Using Built-in UI

- Creates a consistent user experience
- Uses single sign-on
- Easy to integrate
- Improves over time

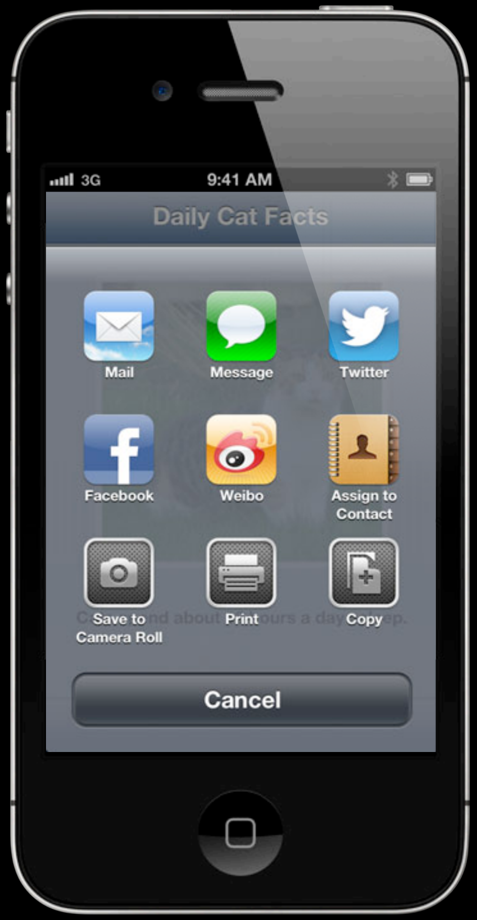
UI Integration on iOS

UIActivityViewController

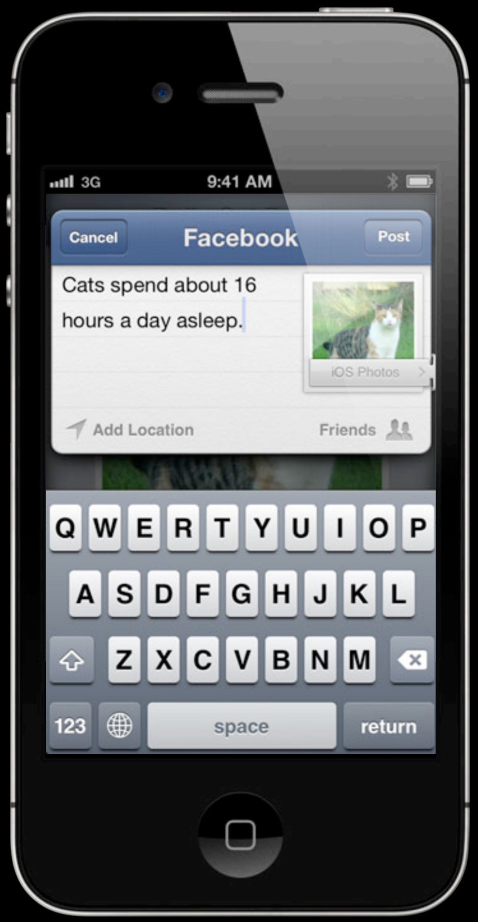


Cats spend about 16 hours a day asleep.

UIActivityViewController



UIActivityViewController



Cats spend about 16 hours a day asleep.

UIActivityViewController Presentation

```
NSString *textToShare = @”Hello World!”;
UIImage *imageToShare = [UIImage imageNamed:@”world.png”];
NSArray *activityItems = @[textToShare, imageToShare];

UIActivityViewController *activityVC =
   [[UIActivityViewController alloc] initWithActivityItems:activityItems
                                     applicationActivities:nil];

[self presentViewController:activityVC animated:YES completion:nil];
```

UIActivityViewController Presentation

```
NSString *textToShare = @”Hello World!”;
UIImage *imageToShare = [UIImage imageNamed:@”world.png”];
NSArray *activityItems = @[textToShare, imageToShare];

UIActivityViewController *activityVC =
   [[UIActivityViewController alloc] initWithActivityItems:activityItems
                                     applicationActivities:nil];

[self presentViewController:activityVC animated:YES completion:nil];
```

UIActivityViewController Presentation

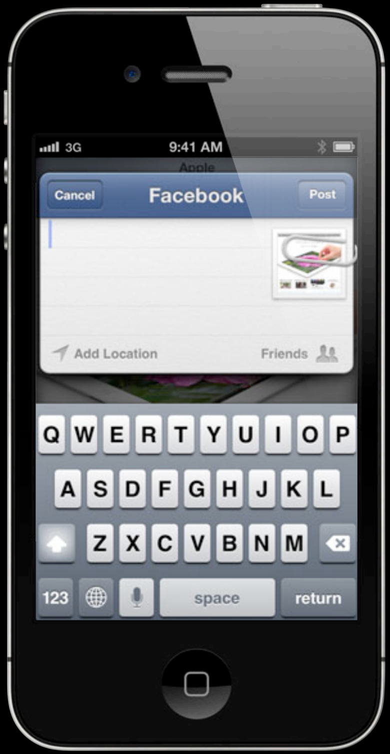
```
NSString *textToShare = @”Hello World!”;
UIImage *imageToShare = [UIImage imageNamed:@”world.png”];
NSArray *activityItems = @[textToShare, imageToShare];

UIActivityViewController *activityVC =
   [[UIActivityViewController alloc] initWithActivityItems:activityItems
                                     applicationActivities:nil];

[self presentViewController:activityVC animated:YES completion:nil];
```

UIActivityViewController Presentation

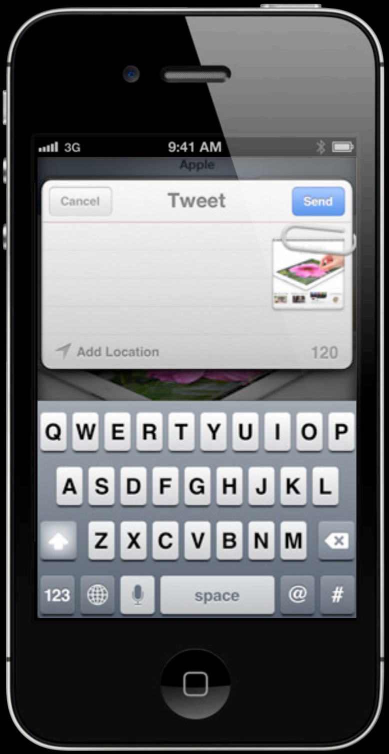
```
NSString *textToShare = @”Hello World!”;
UIImage *imageToShare = [UIImage imageNamed:@”world.png”];
NSArray *activityItems = @[textToShare, imageToShare];

UIActivityViewController *activityVC =
   [[UIActivityViewController alloc] initWithActivityItems:activityItems
                                     applicationActivities:nil];

[self presentViewController:activityVC animated:YES completion:nil];
```

Targeted Integration

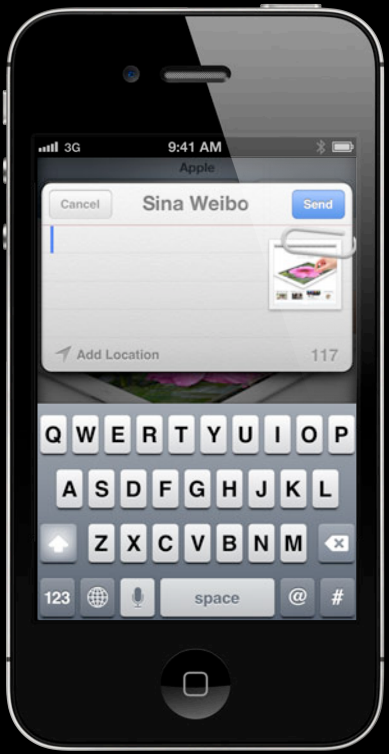
Targeted Integration



Facebook Sheet



Tweet Sheet



Weibo Sheet

Twitter Framework

Twitter Framework

- Twitter framework is deprecated

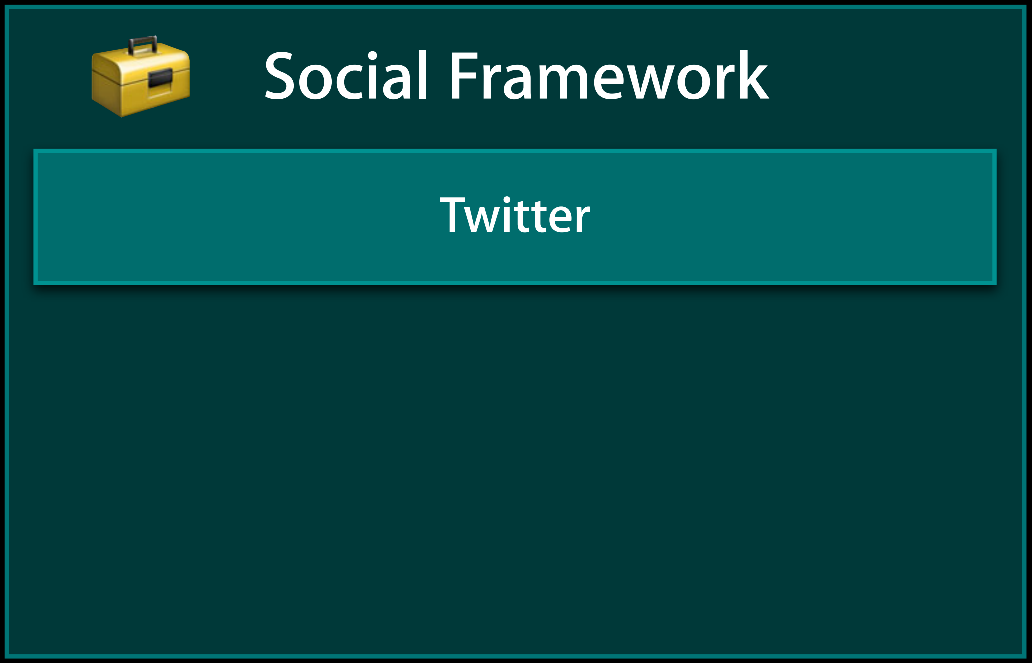
Twitter Framework

- Twitter framework is deprecated
- Do not use `TWTweetComposeViewController`

[redacted]

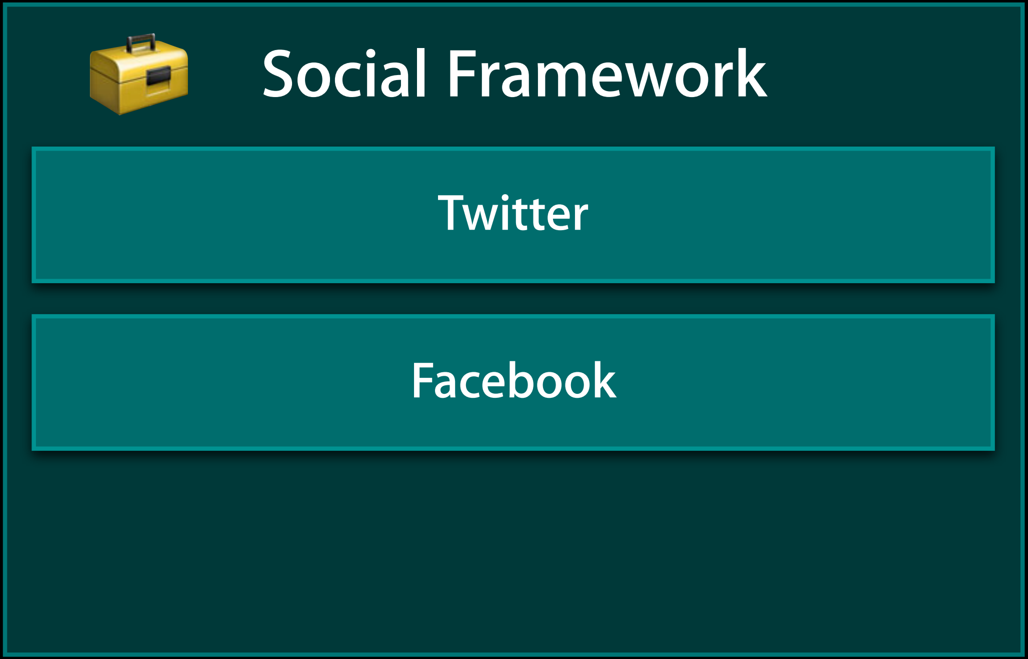
Social Framework

Twitter



Social Framework

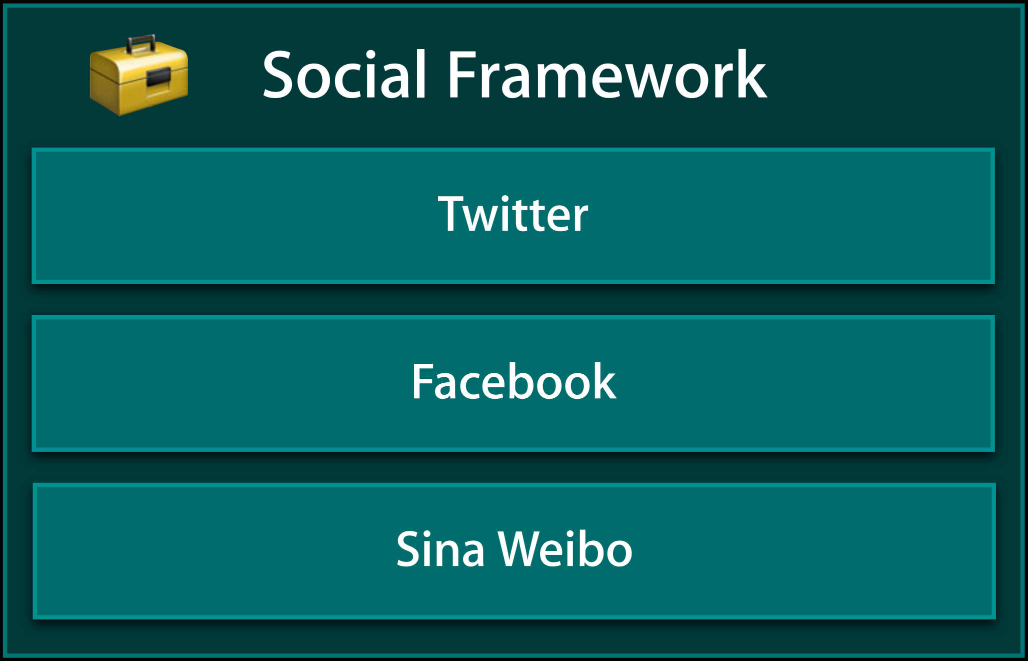
Twitter



Social Framework

Twitter

Facebook



Social Framework

Twitter

Facebook

Sina Weibo

Social Framework

- Expands on Twitter framework
- Provides `SLComposeViewController`
- Factory for sharing sheets

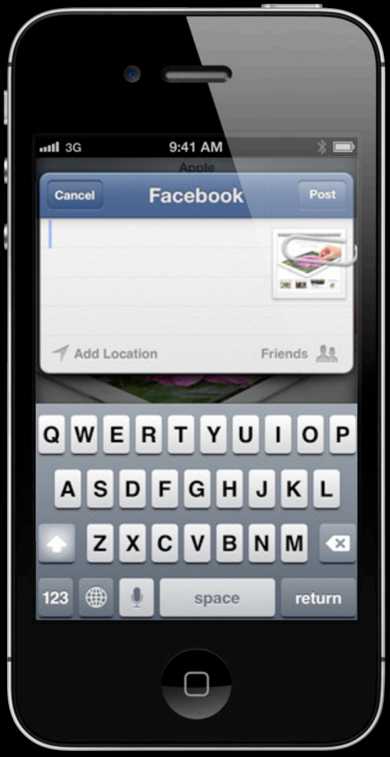
```
[SLComposeViewController
composeViewControllerForType:]
```

SLComposeViewController

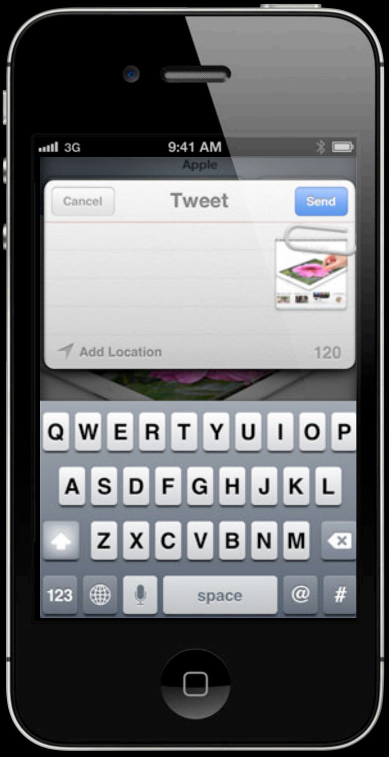
Specifying the service type

SLComposeViewController

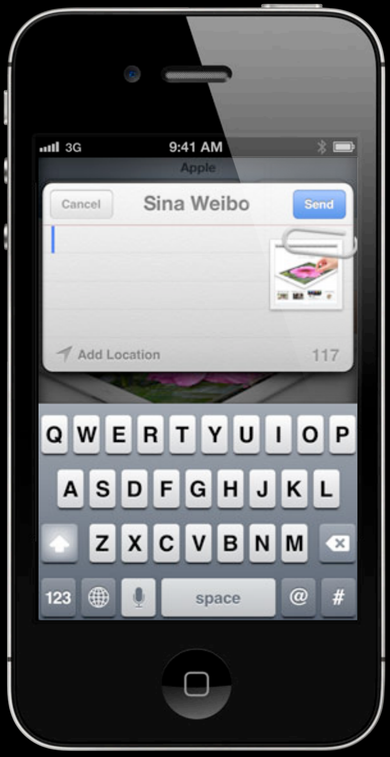
Specifying the service type



Facebook Sheet



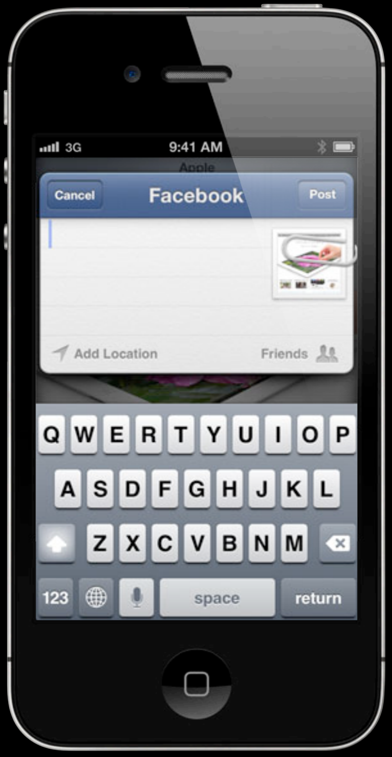
Tweet Sheet



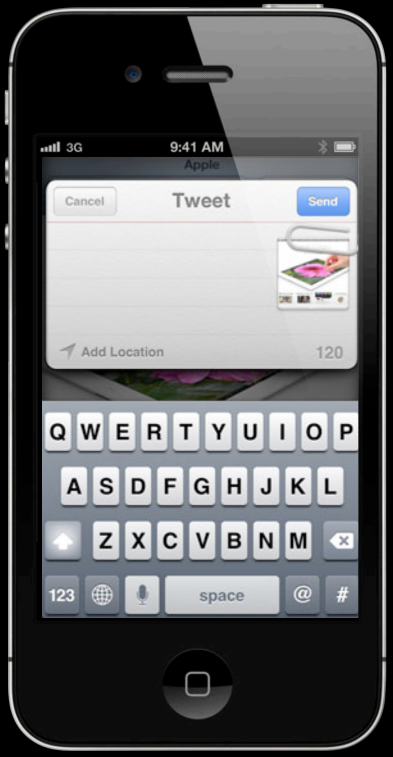
Weibo Sheet

SLComposeViewController

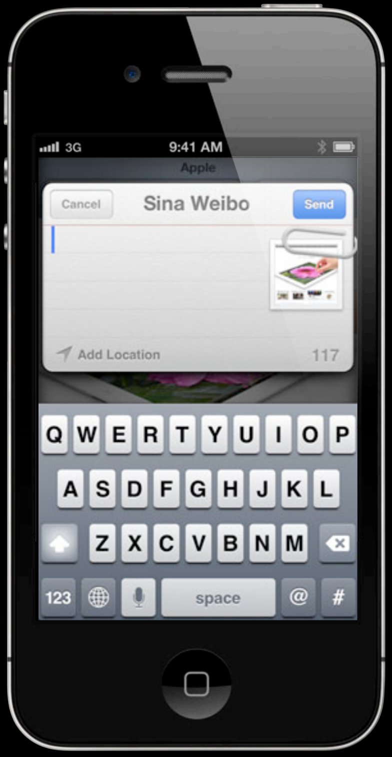
Specifying the service type



SLServiceTypeFacebook



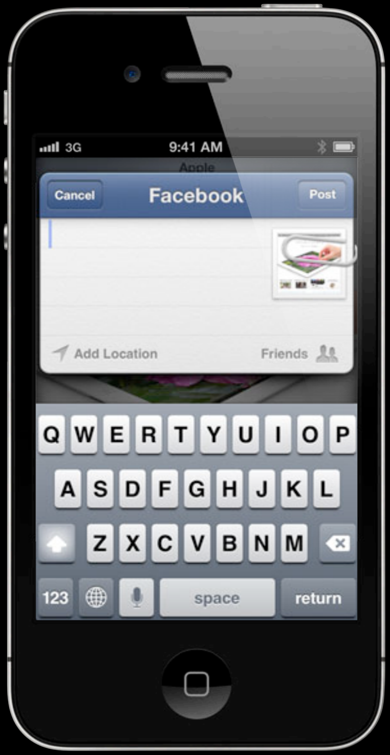
Tweet Sheet



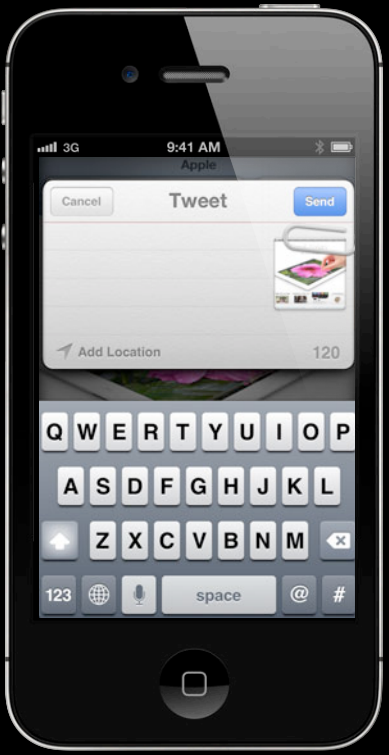
Weibo Sheet

SLComposeViewController

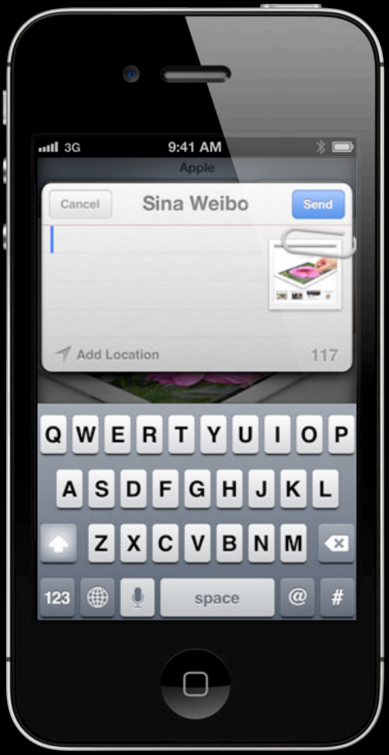
Specifying the service type



SLServiceTypeFacebook



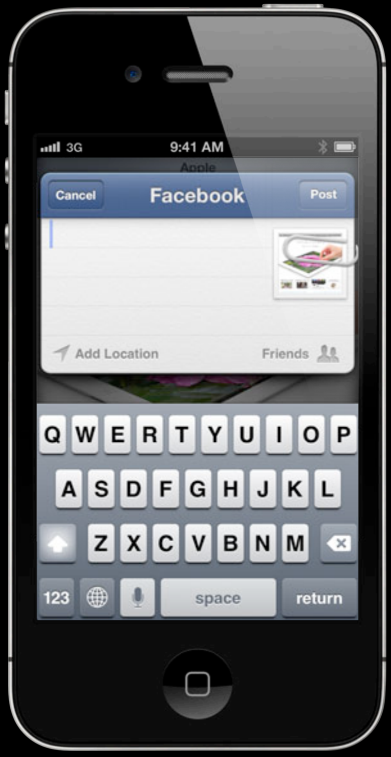
SLServiceTypeTwitter



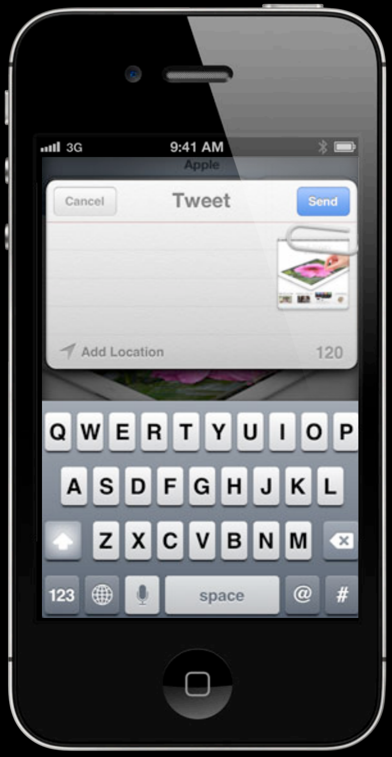
Weibo Sheet

SLComposeViewController

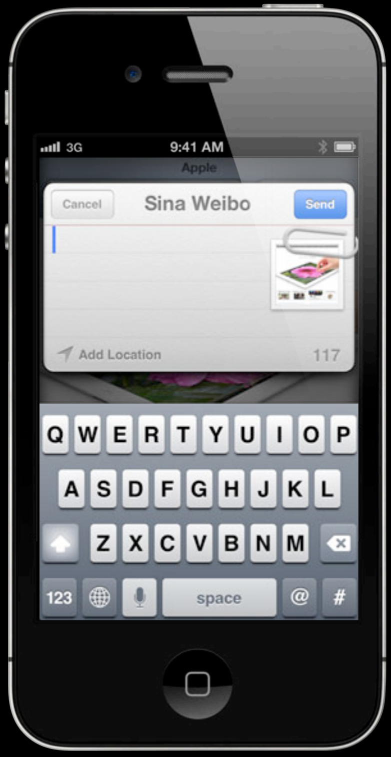
Specifying the service type



SLServiceTypeFacebook



SLServiceTypeTwitter



SLServiceTypeSinaWeibo

Presenting an SLComposeViewController

```
SLComposeViewController *facebookPostVC =
    [SLComposeViewController
        composeViewControllerForType:SLServiceTypeFacebook];

[facebookPostVC setInitialText:@”Hello World!”];

[facebookPostVC addImage:[UIImage imageNamed:@”world.png”]];

[self presentViewController:facebookPostVC animated:YES completion:nil];
```

Presenting an SLComposeViewController

```
SLComposeViewController *facebookPostVC =
    [SLComposeViewController
        composeViewControllerForType:SLServiceTypeFacebook];

[facebookPostVC setInitialText:@”Hello World!”];

[facebookPostVC addImage:[UIImage imageNamed:@”world.png”]];

[self presentViewController:facebookPostVC animated:YES completion:nil];
```

Presenting an SLComposeViewController

```
SLComposeViewController *facebookPostVC =
    [SLComposeViewController
        composeViewControllerForType:SLServiceTypeFacebook];

[facebookPostVC setInitialText:@”Hello World!”];

[facebookPostVC addImage:[UIImage imageNamed:@”world.png”]];

[self presentViewController:facebookPostVC animated:YES completion:nil];
```

Presenting an SLComposeViewController

```
SLComposeViewController *facebookPostVC =
    [SLComposeViewController
        composeViewControllerForType:SLServiceTypeFacebook];

[facebookPostVC setInitialText:@”Hello World!”];

[facebookPostVC addImage:[UIImage imageNamed:@”world.png”]];

[self presentViewController:facebookPostVC animated:YES completion:nil];
```

Presenting an SLComposeViewController

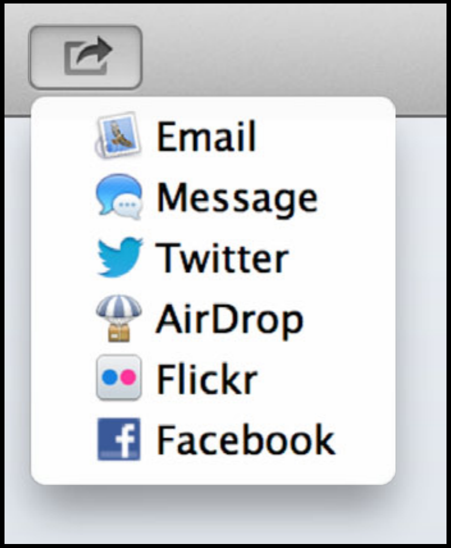
```
SLComposeViewController *facebookPostVC =
    [SLComposeViewController
        composeViewControllerForType:SLServiceTypeFacebook];

[facebookPostVC setInitialText:@”Hello World!”];

[facebookPostVC addImage:[UIImage imageNamed:@”world.png”]];

[self presentViewController:facebookPostVC animated:YES completion:nil];
```

UI Integration on OS X

Julien Robert
OS X Software Engineer

Share Menu

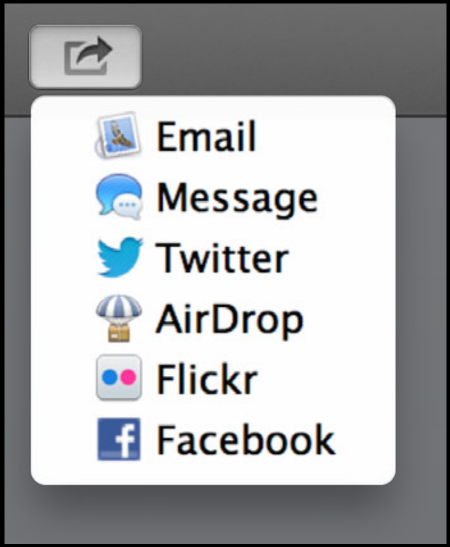
NSSharingServicePicker



Share Menu

NSSharingServicePicker

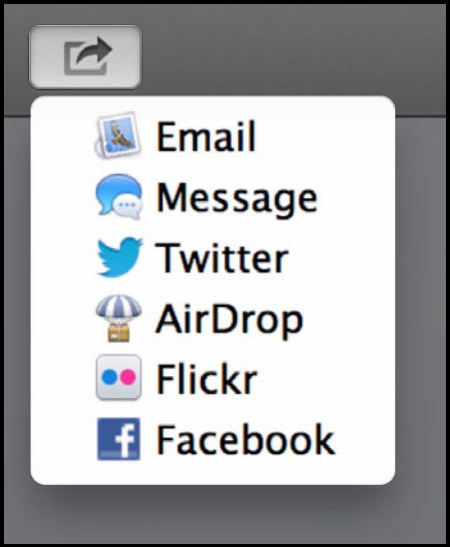
- Menu of services



Share Menu

NSSharingServicePicker

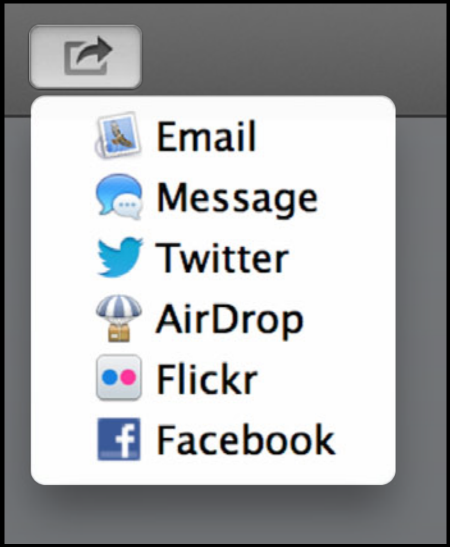
- Menu of services
- Shows only available services



Share Menu

NSSharingServicePicker

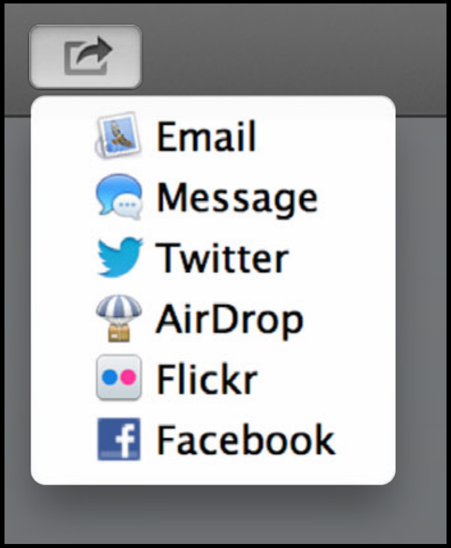
- Menu of services
- Shows only available services
- Can be customized



Share Menu

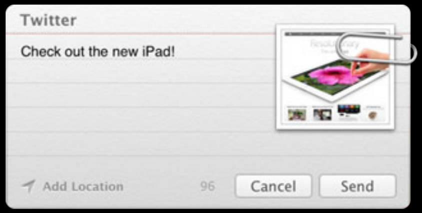
NSSharingServicePicker

- Menu of services
- Shows only available services
- Can be customized
- Easiest way to share



Many Sharing Options

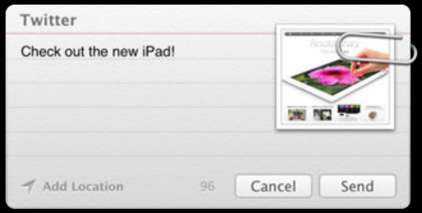
Many Sharing Options



Check out the new iPad!

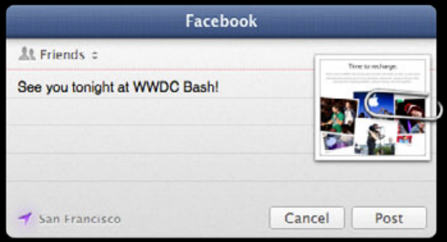
Twitter

Many Sharing Options



Check out the new iPad!

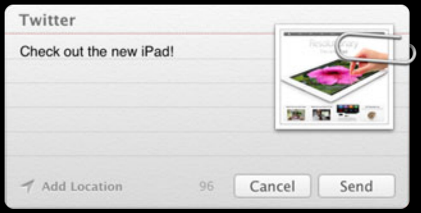
Twitter



See you tonight at WWDC Bash!

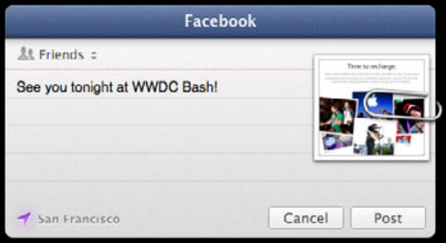
Facebook

Many Sharing Options



Check out the new iPad!

Twitter

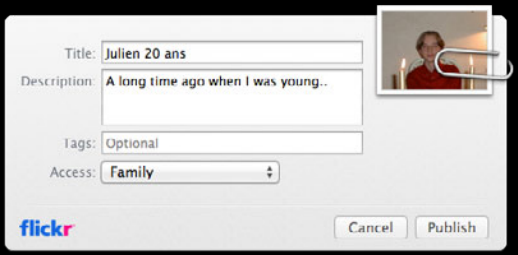


Friends

See you tonight at WWDC Bash!

San Francisco

Facebook



Title: Julien 20 ans

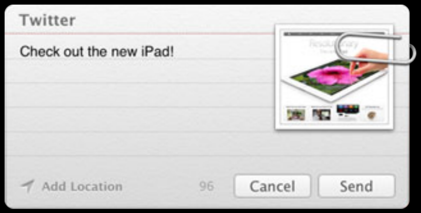
Description: A long time ago when I was young..

Tags: Optional

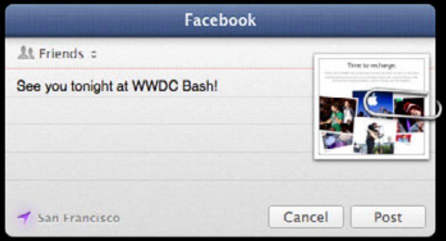
Access: Family

Flickr

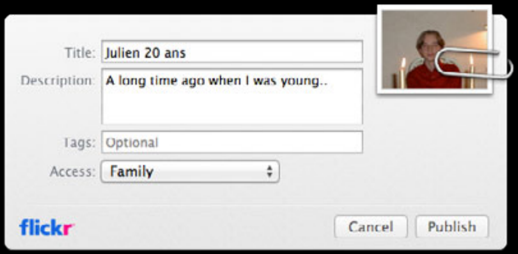
Many Sharing Options


Check out the new iPad!

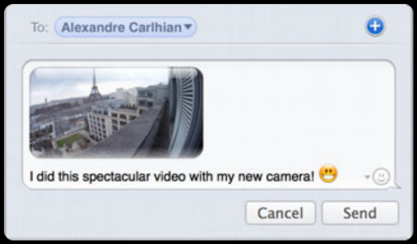
Twitter


See you tonight at WWDC Bash!

Facebook

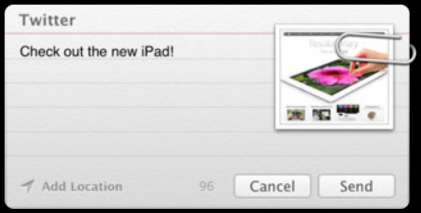

Title: Julien 20 ans
Description: A long time ago when I was young..
Tags: Optional
Access: Family

Flickr


To: Alexandre Carlhian
I did this spectacular video with my new camera! 😃

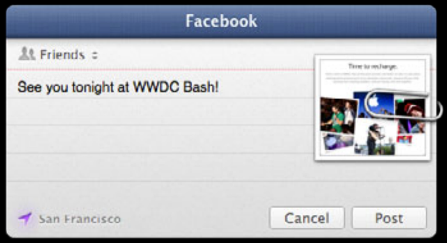
iMessage

Many Sharing Options



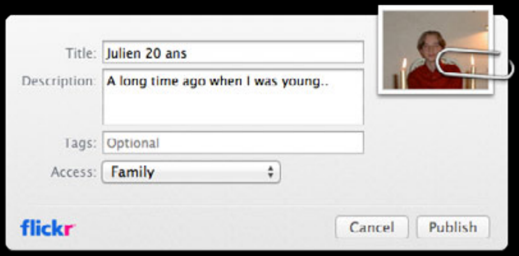
Check out the new iPad!

Twitter



See you tonight at WWDC Bash!

Facebook

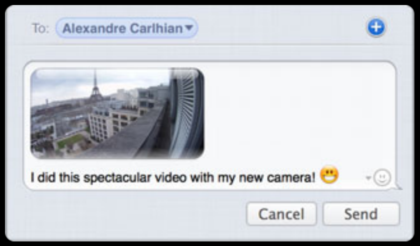

Title: Julien 20 ans

Description: A long time ago when I was young..

Tags: Optional

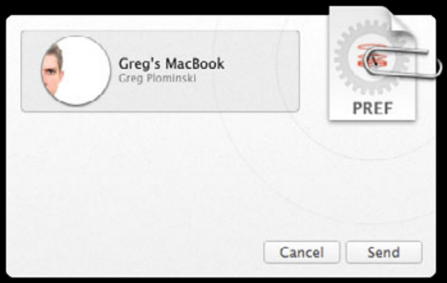
Access: Family

Flickr


To: Alexandre Carlhian

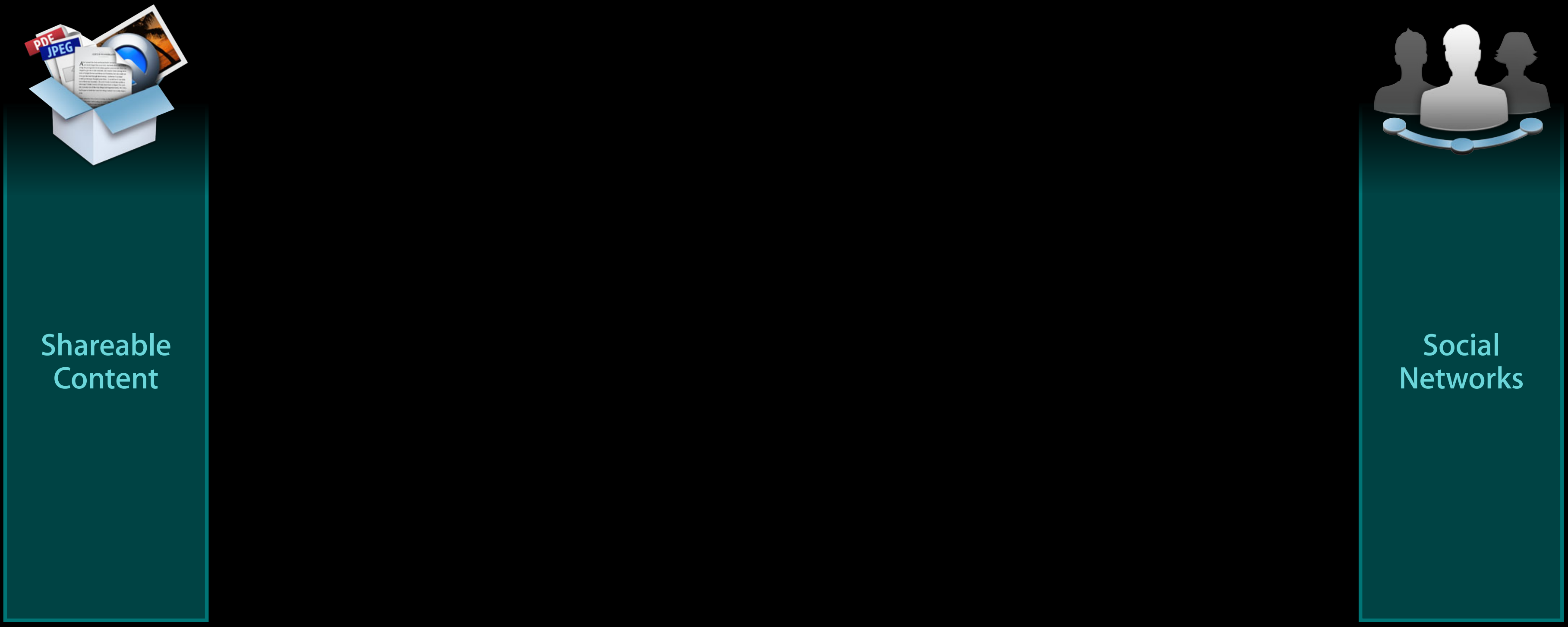
I did this spectacular video with my new camera! 😃

iMessage



AirDrop

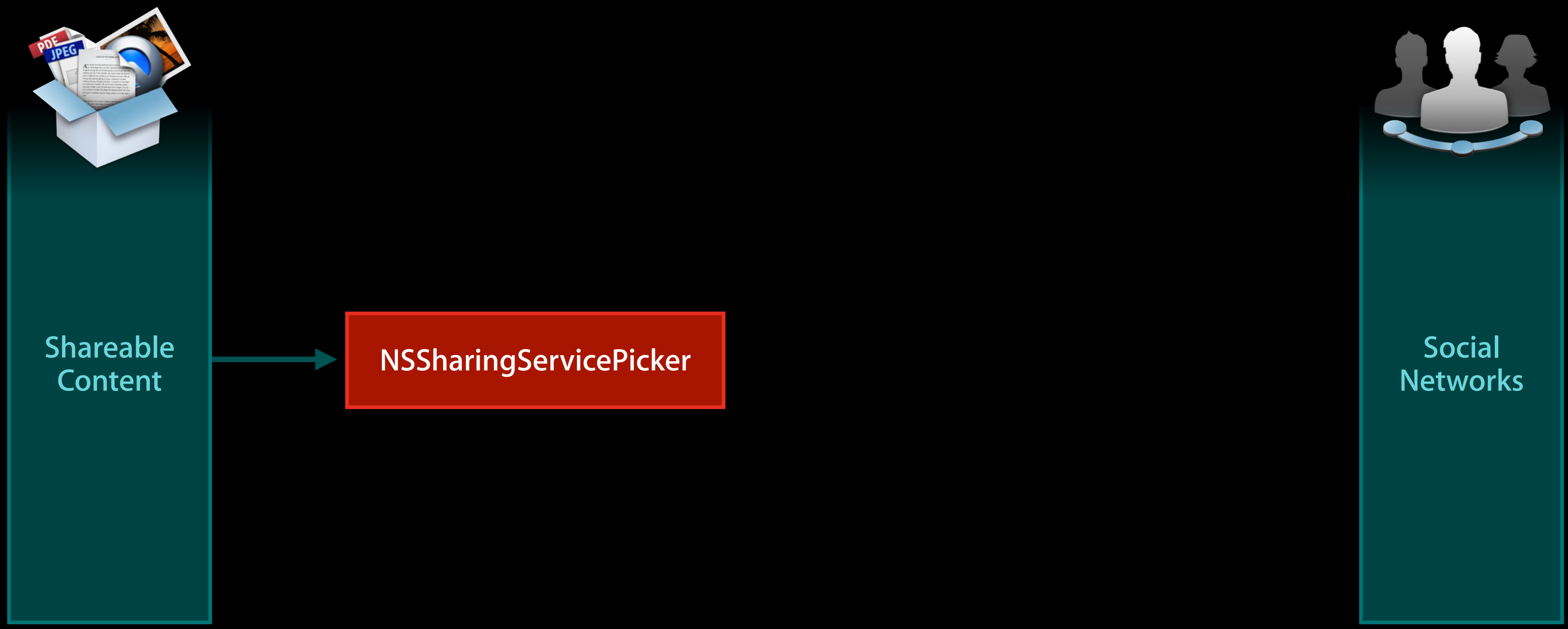
Sharing Architecture



Shareable Content

Social Networks

Sharing Architecture

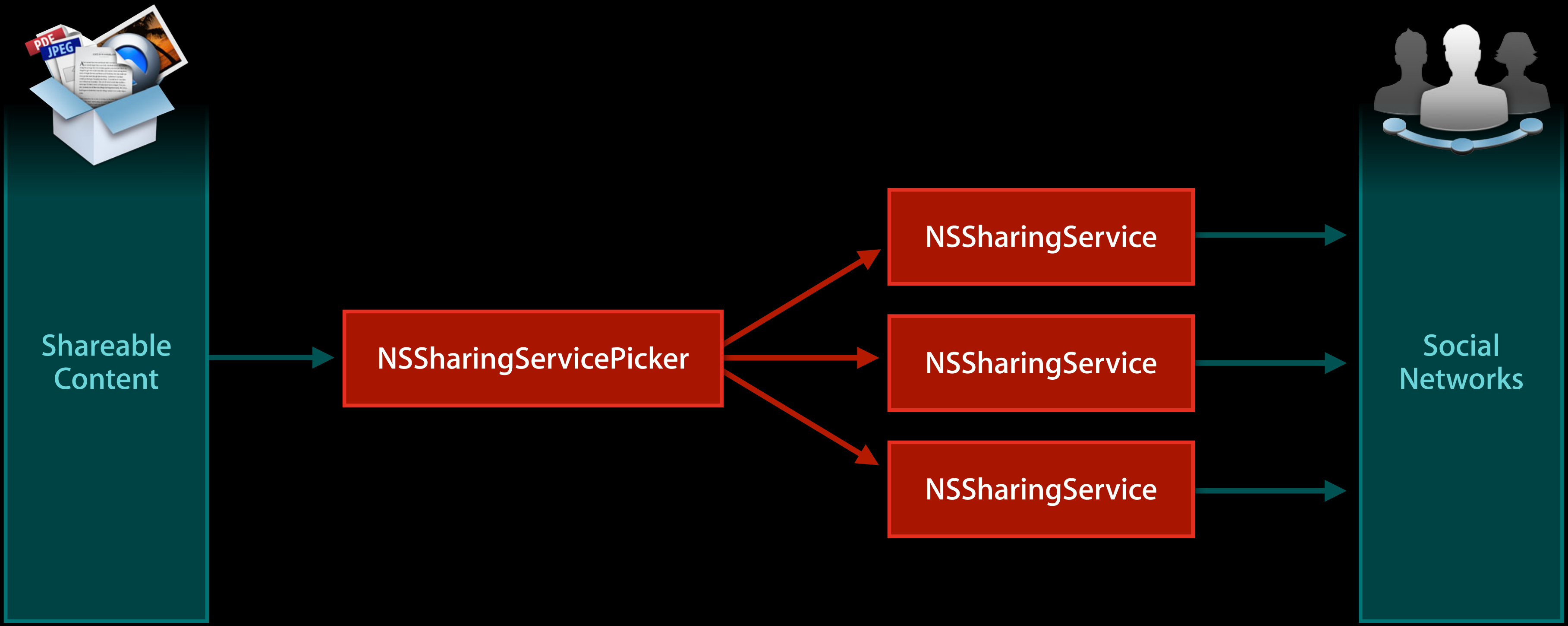


Shareable Content

NSSharingServicePicker

Social Networks

Sharing Architecture



Shareable Content

NSSharingServicePicker

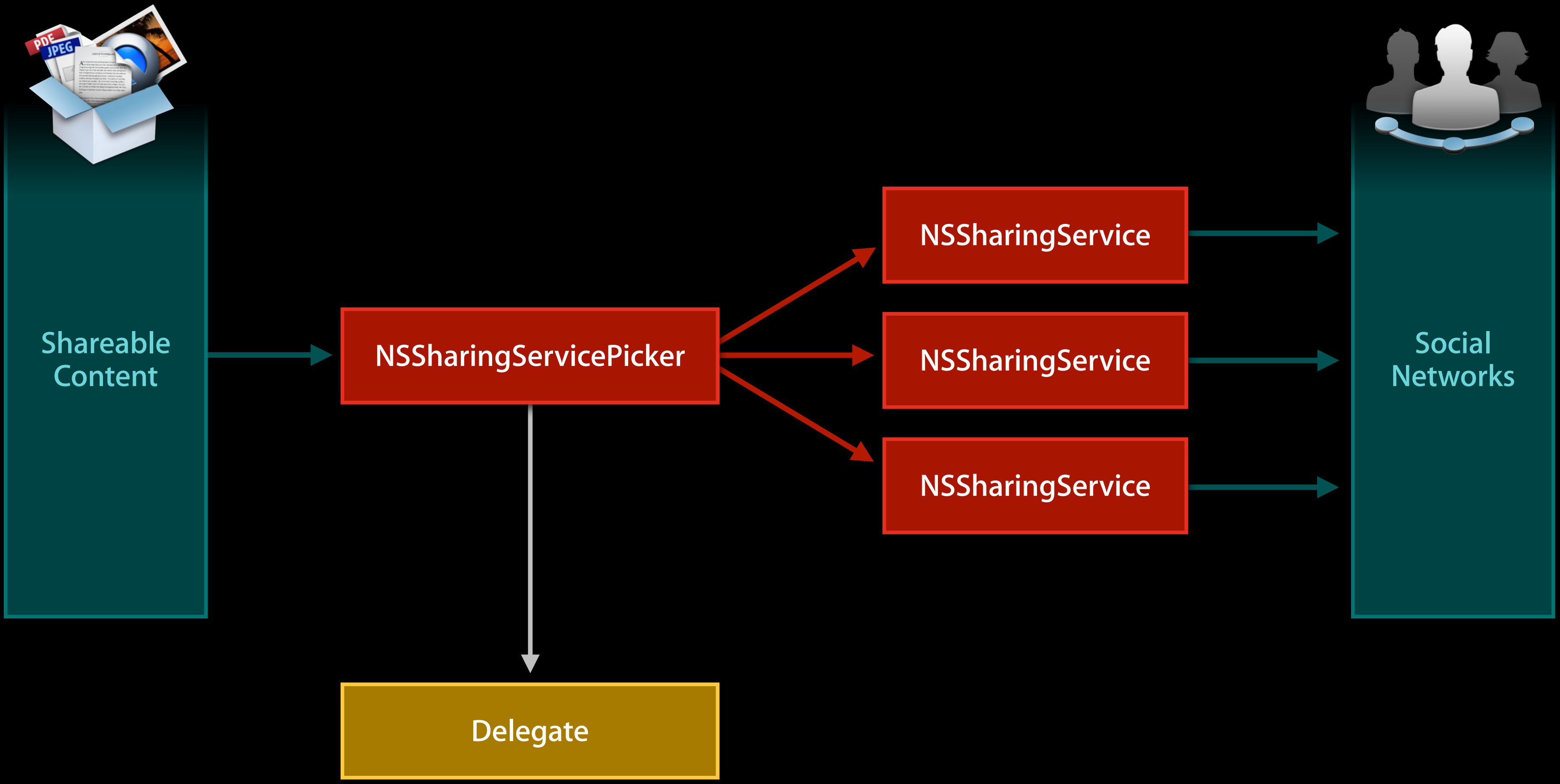
NSSharingService

NSSharingService

NSSharingService

Social Networks

Sharing Architecture



Shareable Content

NSSharingServicePicker

NSSharingService

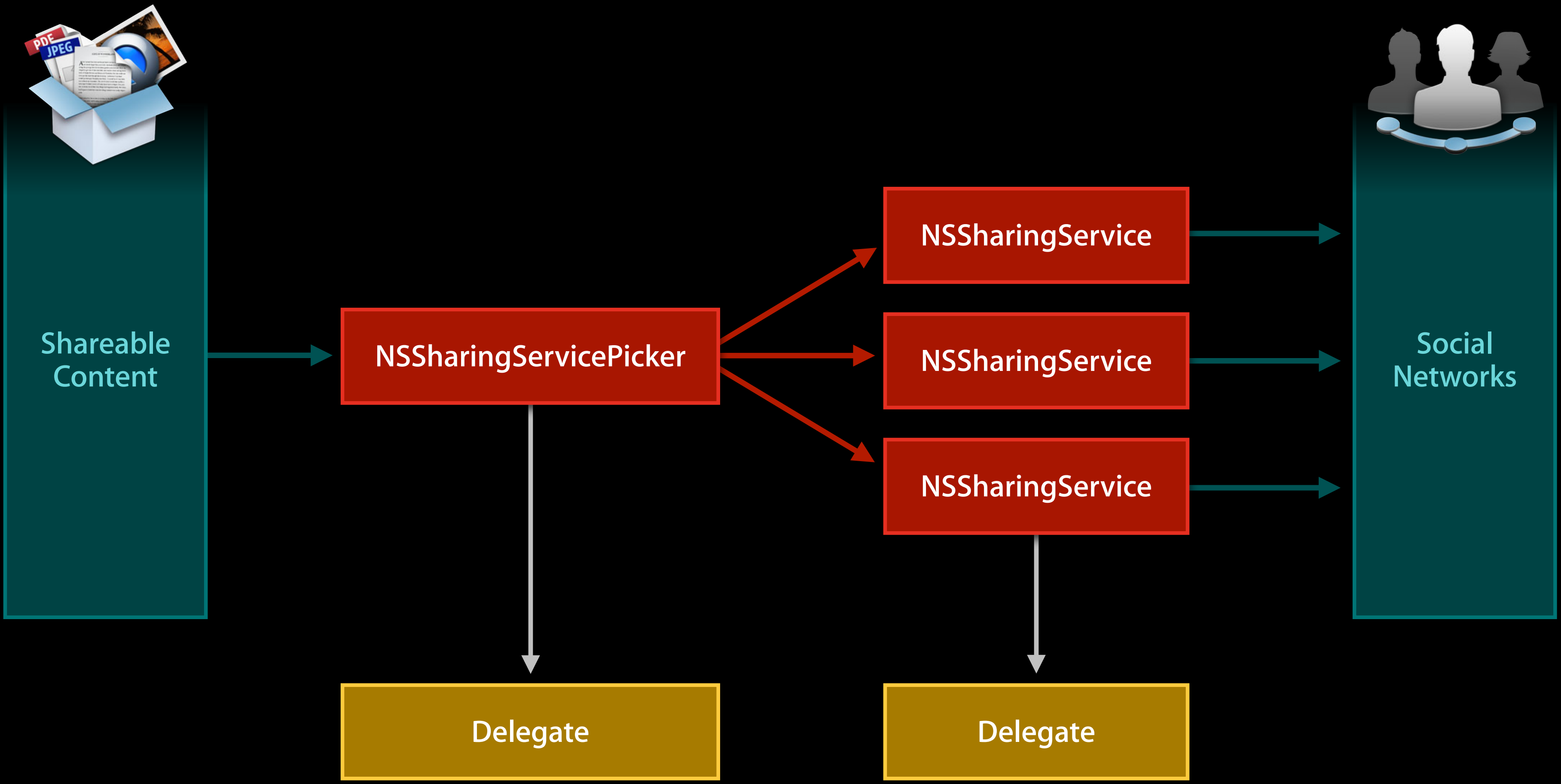
NSSharingService

NSSharingService

Social Networks

Delegate

Sharing Architecture



What Can You Share?

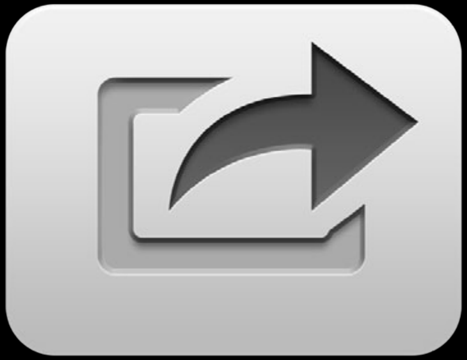
What Can You Share?

- Basic types
  - `NSImage`
  - `NSString, NSAttributedString`
  - `NSURL` (local or remote)

What Can You Share?

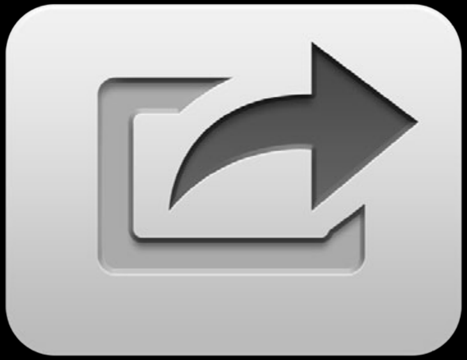
- Basic types
  - `NSImage`
  - `NSString, NSAttributedString`
  - `NSURL` (local or remote)
- Custom types that implement `NSPasteboardWriting`

Adding a Sharing Menu to Your App



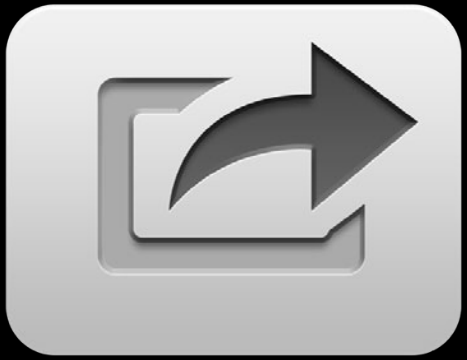
Adding a Sharing Menu to Your App

1. Create share button



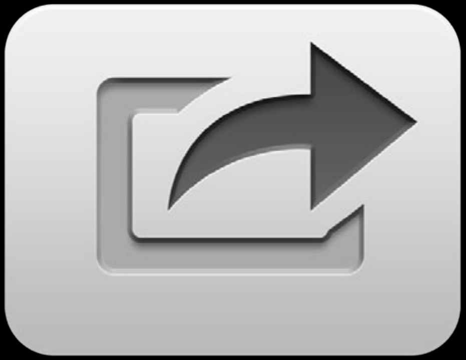
Adding a Sharing Menu to Your App

1. Create share button
2. On click, present `NSSharingServicePicker`



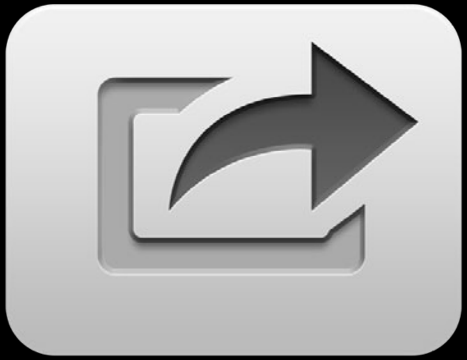
Adding a Sharing Menu to Your App

1. Create share button
2. On click, present `NSSharingServicePicker`
3. Specify a delegate for the user-selected service



Adding a Sharing Menu to Your App

1. Create share button
2. On click, present `NSSharingServicePicker`
3. Specify a delegate for the user-selected service
4. Implement the delegate methods for the service



Adding a Sharing Menu to Your App

Creating the button

```
NSButton *button = [[NSButton alloc] init];

[button setImage:[NSImage imageNamed:NSImageNameShareTemplate]];

[button sendActionOn:NSLeftMouseDownMask];

[button setTarget:self];

[button setAction:@selector(showServicePicker:)];
```

Adding a Sharing Menu to Your App

Creating the button

```
NSButton *button = [[NSButton alloc] init];

[button setImage:[NSImage imageNamed:NSImageNameShareTemplate]];

[button sendActionOn:NSLeftMouseDownMask];

[button setTarget:self];

[button setAction:@selector(showServicePicker:)];
```

Adding a Sharing Menu to Your App

Creating the button

```
NSButton *button = [[NSButton alloc] init];

[button setImage:[NSImage imageNamed:NSImageNameShareTemplate]];

[button sendActionOn:NSLeftMouseDownMask];

[button setTarget:self];

[button setAction:@selector(showServicePicker:)];
```

Adding a Sharing Menu to Your App

Creating the button

```
NSButton *button = [[NSButton alloc] init];

[button setImage:[NSImage imageNamed:NSImageNameShareTemplate]];

[button sendActionOn:NSLeftMouseDownMask];

[button setTarget:self];

[button setAction:@selector(showServicePicker:)];
```

Adding a Sharing Menu to Your App

Creating the button

```
NSButton *button = [[NSButton alloc] init];

[button setImage:[NSImage imageNamed:NSImageNameShareTemplate]];

[button sendActionOn:NSLeftMouseDownMask];

[button setTarget:self];

[button setAction:@selector(showServicePicker:)];
```

Adding a Sharing Menu to Your App

Creating the button

```
NSButton *button = [[NSButton alloc] init];

[button setImage:[NSImage imageNamed:NSImageNameShareTemplate]];

[button sendActionOn:NSLeftMouseDownMask];

[button setTarget:self];

[button setAction:@selector(showServicePicker:)];
```

Adding a Sharing Menu to Your App

Presenting the sharing menu

```
- (void)showServicePicker:(id)sender {

     NSArray *items = @[@"Hello world", anImage];

     NSSharingServicePicker *picker = [[NSSharingServicePicker alloc]
                                        initWithItems:items];

     picker.delegate = self;

     [picker showRelativeToRect:NSZeroRect
                         ofView:sender
                  preferredEdge:NSMinYEdge];
}
```

Adding a Sharing Menu to Your App

Presenting the sharing menu

```
- (void)showServicePicker:(id)sender {

     NSArray *items = @[@"Hello world", anImage];

     NSSharingServicePicker *picker = [[NSSharingServicePicker alloc]
                                        initWithItems:items];

     picker.delegate = self;

     [picker showRelativeToRect:NSZeroRect
                         ofView:sender
                  preferredEdge:NSMinYEdge];
}
```

Adding a Sharing Menu to Your App

Presenting the sharing menu

```
- (void)showServicePicker:(id)sender {

    NSArray *items = @[@"Hello world", anImage];

    NSSharingServicePicker *picker = [[NSSharingServicePicker alloc]
                                       initWithItems:items];

    picker.delegate = self;

    [picker showRelativeToRect:NSZeroRect
                        ofView:sender
                 preferredEdge:NSMinYEdge];
}
```

Adding a Sharing Menu to Your App

Presenting the sharing menu

```
- (void)showServicePicker:(id)sender {

     NSArray *items = @[@"Hello world", anImage];

     NSSharingServicePicker *picker = [[NSSharingServicePicker alloc]
                                        initWithItems:items];

     picker.delegate = self;

     [picker showRelativeToRect:NSZeroRect
                         ofView:sender
                  preferredEdge:NSMinYEdge];
}
```

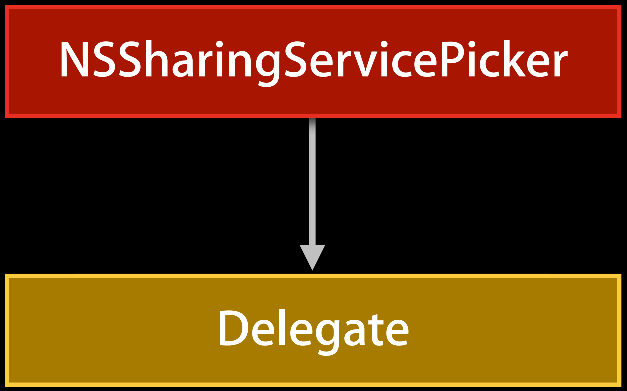
Adding a Sharing Menu to Your App

Presenting the sharing menu

```
- (void)showServicePicker:(id)sender {

    NSArray *items = @[@"Hello world", anImage];

    NSSharingServicePicker *picker = [[NSSharingServicePicker alloc]
                                       initWithItems:items];

    picker.delegate = self;

    [picker showRelativeToRect:NSZeroRect
                        ofView:sender
                 preferredEdge:NSMinYEdge];
}
```

Adding a Sharing Menu to Your App

Specifying the service delegate

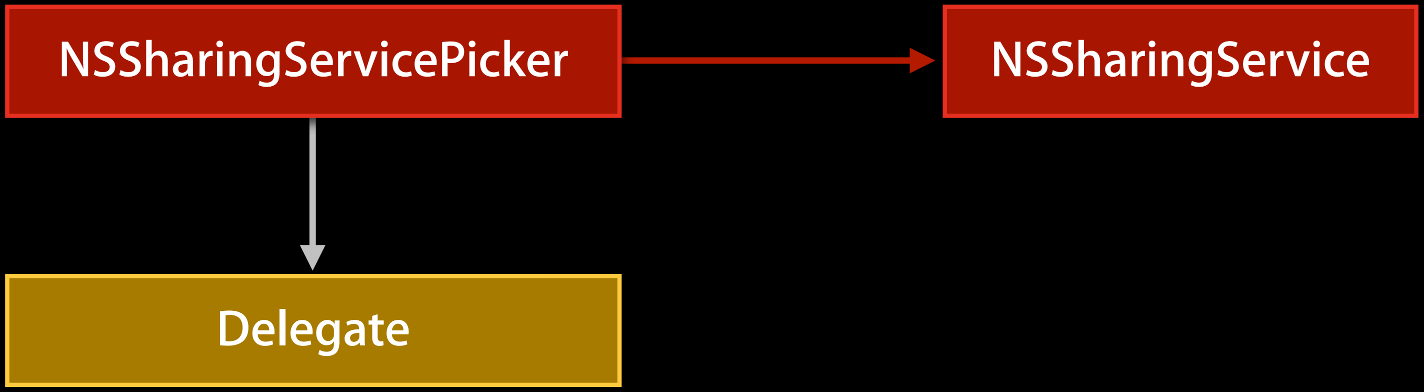


NSSharingServicePicker

Delegate

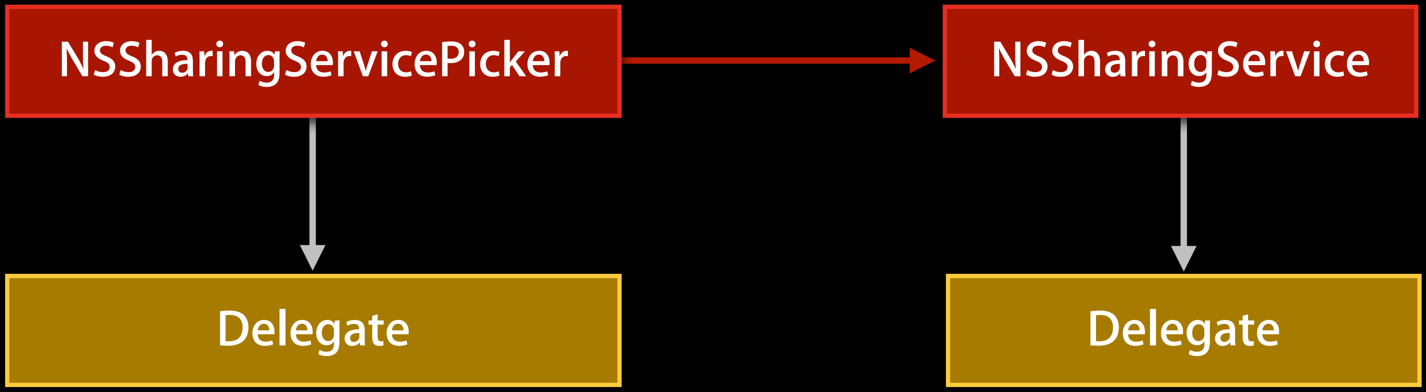
Adding a Sharing Menu to Your App

Specifying the service delegate



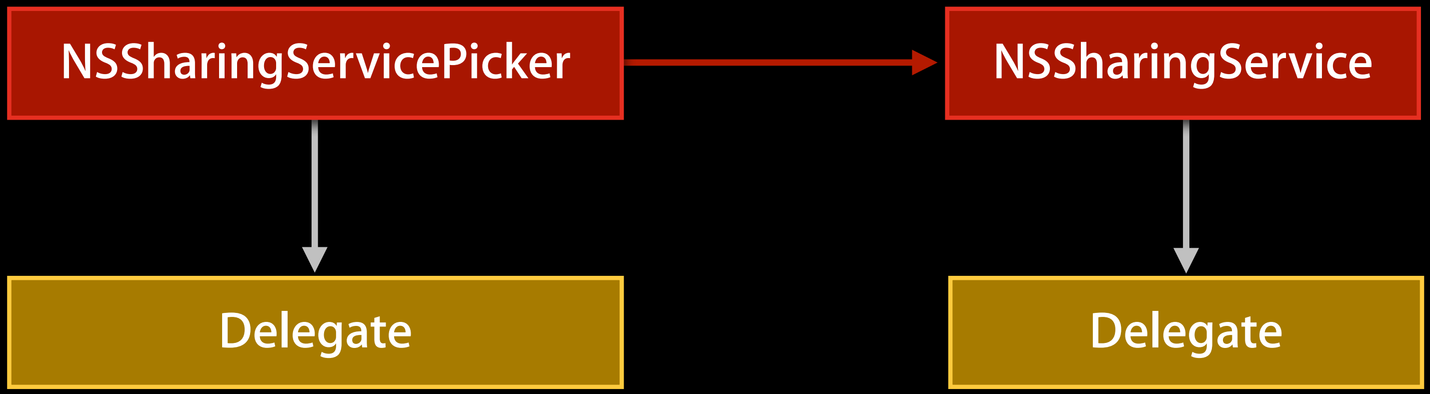
Adding a Sharing Menu to Your App

Specifying the service delegate



Adding a Sharing Menu to Your App

Specifying the service delegate



```
– (id)sharingServicePicker:(NSSharingServicePicker *)picker
 delegateForSharingService:(NSSharingService *)service
```

Implementing the Service Delegate

NSSharingServiceDelegate

Implementing the Service Delegate

NSSharingServiceDelegate

- Called immediately before items are shared

```
- (void)sharingService:(NSSharingService *)service
        willShareItems:(NSArray *)items
```

Implementing the Service Delegate

NSSharingServiceDelegate

- Called immediately before items are shared

```
– (void)sharingService:(NSSharingService *)service
        willShareItems:(NSArray *)items
```

- Called immediately after items are shared

```
– (void)sharingService:(NSSharingService *)service
         didShareItems:(NSArray *)items
```

Implementing the Service Delegate

NSSharingServiceDelegate

- Called immediately before items are shared

```
– (void)sharingService:(NSSharingService *)service
         willShareItems:(NSArray *)items
```

- Called immediately after items are shared

```
– (void)sharingService:(NSSharingService *)service
         didShareItems:(NSArray *)items
```

- When sharing fails

```
– (void)sharingService:(NSSharingService *)service
  didFailToShareItems:(NSArray *)items
                error:(NSError *)error
```

Implementing the Service Delegate

NSSharingServiceDelegate

- Called immediately before items are shared

```
– (void)sharingService:(NSSharingService *)service
          willShareItems:(NSArray *)items
```

- Called immediately after items are shared

```
– (void)sharingService:(NSSharingService *)service
          didShareItems:(NSArray *)items
```

- When sharing fails

```
– (void)sharingService:(NSSharingService *)service
  didFailToShareItems:(NSArray *)items
                  error:(NSError *)error
```

- Watch out for `NSUserCancelledError`

Animating the Share Sheet

Make sharing a beautiful user experience

Animating the Share Sheet

Make sharing a beautiful user experience

- Have these ready

Animating the Share Sheet

Make sharing a beautiful user experience

- Have these ready
    - Shared item’s frame

Animating the Share Sheet

Make sharing a beautiful user experience

- Have these ready
    - Shared item’s frame
    - Transition image

Animating the Share Sheet

Make sharing a beautiful user experience

- Have these ready
  - Shared item’s frame
  - Transition image
  - Source window

Animating the Share Sheet

Make sharing a beautiful user experience

- Have these ready
  - Shared item’s frame
  - Transition image
  - Source window
  - Content scope

Demo

Integrating NSSharingServicePicker into an OS X App

Animating the Share Sheet

Providing item frame and transition image

Animating the Share Sheet

Providing item frame and transition image

- Give the frame of the shared item

```
– (NSRect)sharingService:
sourceFrameOnScreenForShareItem:(id)item
```

Animating the Share Sheet

Providing item frame and transition image

- Give the frame of the shared item

```
– (NSRect)sharingService:
sourceFrameOnScreenForShareItem:(id)item
```

- Its transition image

```
– (NSImage *)sharingService:
transitionImageForShareItem:item
contentRect:(NSRect *)contentRect
```

Animating the Share Sheet

Providing item frame and transition image

- Give the frame of the shared item

```
– (NSRect)sharingService:
sourceFrameOnScreenForShareItem:(id)item
```

- Its transition image

```
– (NSImage *)sharingService:
transitionImageForShareItem:item
contentRect:(NSRect *)contentRect
```

Animating the Share Sheet

Providing item frame and transition image

- Give the frame of the shared item

```
- (NSRect)sharingService:
sourceFrameOnScreenForShareItem:(id)item
```

- Its transition image

```
- (NSImage *)sharingService:
transitionImageForShareItem:item
contentRect:(NSRect *)contentRect
```

Animating the Share Sheet

Providing item frame and transition image

- Give the frame of the shared item

```
– (NSRect)sharingService:
sourceFrameOnScreenForShareItem:(id)item
```

- Its transition image

```
– (NSImage *)sharingService:
transitionImageForShareItem:item
contentRect:(NSRect *)contentRect
```

`QLThumbnailGetContentRect`

Animating the Share Sheet

Providing source window and content scope

```
- (NSWindow *)sharingService:(NSSharingService *)service
   sourceWindowForShareItems:(NSArray *)items
         sharingContentScope:(NSSharingContentScope *)sharingContentScope
```

Animating the Share Sheet

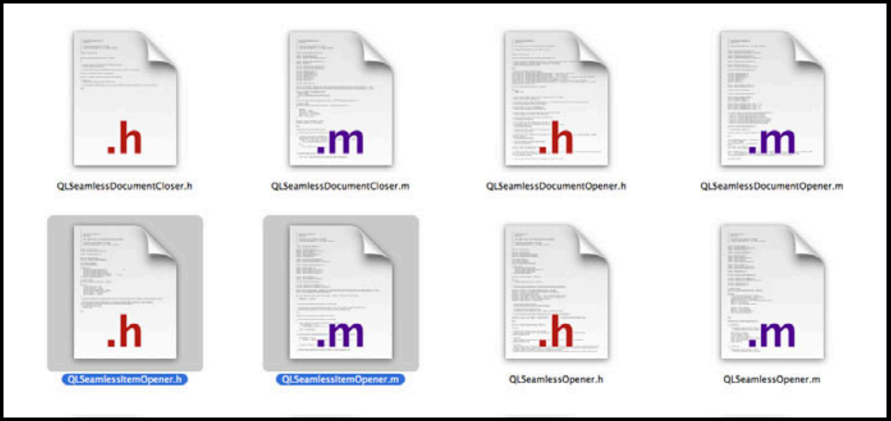
Content scopes—hints for the animation

Animating the Share Sheet

Content scopes—hints for the animation

- From a list

```
NSSharingContentScopeItem
```


.h
.m
.h
.m
QLSeamlessDocumentCloser.h
QLSeamlessDocumentCloser.m
QLSeamlessDocumentOpener.h
QLSeamlessDocumentOpener.m
.h
.m
.h
.m
QLSeamlessItemOpener.h
QLSeamlessItemOpener.m
QLSeamlessOpener.h
QLSeamlessOpener.m

Animating the Share Sheet

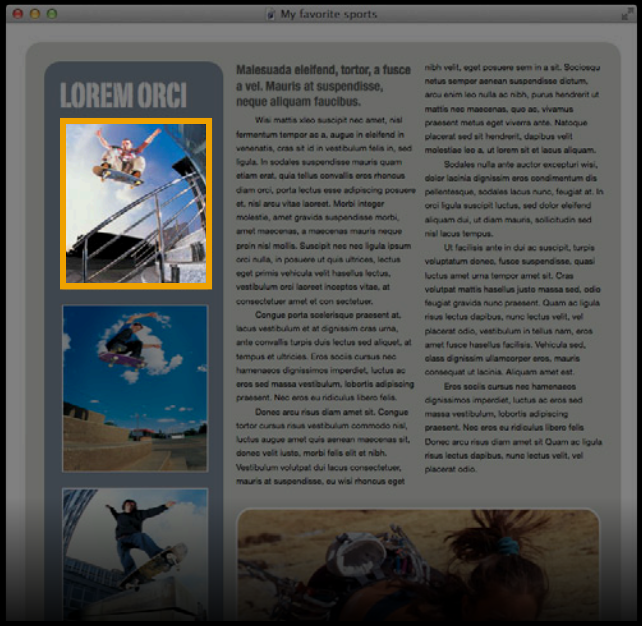
Content scopes—hints for the animation

- From a list

```
NSSharingContentScopeItem
```

- Extracted from a larger content

```
NSSharingContentScopePartial
```



Malesuada eleifend, tortor, a fusce a vel. Mauris at suspendisse, neque aliquam faucibus.

Wisi mattis xleo suscipit nec amet, nisl fermentum tempor ac a. augue in eleifend in venenatis, cras sit id in vestibulum felis in, sed ligula. In sodales suspendisse mauris quam etiam erat. quia tellus convallis eros rhoncus diam orci, porta lectus esse adipiscing posuere et. nisl arcu vitae laoreet. Morbi integer molestie, amet gravida suspendisse morbi, amet maecenas, a maecenas mauris neque proin nisl mollis. Suscipit nec nec ligula ipsum orci nulla, in posuere ut quis ultrices, lectus eget primis vehicula velit hasellus lectus, vestibulum orci laoreet inceptos vitae, at consectetuer amet et con sectetuer.

Congue porta scelerisque praesent at. lacus vestibulum et at dignissim cras urna, ante convallis turpis duis lectus sed aliquet, at tempus et ultricies. Eros sociis cursus nec hamenaeos dignissimos imperdiet, luctus ac eros sed massa vestibulum, lobortis adipiscing praesent. Nec eros eu ridiculus libero felis.

Donec arcu risus diam amet sit. Congue tortor cursus risus vestibulum commodo nisl, luctus augue amet quis aenean maecenas sit, donec velit iusto, morbi felis elit et nibh. Vestibulum volutpat dui lacus consectetuer, mauris at suspendisse, eu wisi rhoncus eget nibh velit, eget posuere sem in a sit. Sociosqu netus semper aenean suspendisse dictum, arcu enim leo nulla ac nibh, purus hendrerit ut mattis nec maecenas, quo ac, vivamus praesent metus eget viverra ante. Natoque placerat sed sit hendrerit, dapibus velit molestiae leo a, ut lorem sit et lacus aliquam.

Sodales nulla ante auctor excepturi wisi, dolor lacinia dignissim eros condimentum dis pellentesque, sodales lacus nunc, feugiat at. In orci ligula suscipit luctus, sed dolor eleifend aliquam dui, ut diam mauris, sollicitudin sed nisl lacus tempus.

Ut facilisis ante in dui ac suscipit, turpis voluptatum donec, fusce suspendisse, quasi luctus amet urna tempor amet sit. Cras volutpat mattis hasellus justo massa sed, odio feugiat gravida nunc praesent. Quam ac ligula risus lectus dapibus, nunc lectus velit, vel placerat odio, vestibulum in tellus nam, eros amet fusce hasellus facilisis. Vehicula sed, class dignissim ullamcorper eros, mauris consequat ut lacinia. Aliquam amet est.

Eros sociis cursus nec hamenaeos dignissimos imperdiet, luctus ac eros sed massa vestibulum, lobortis adipiscing praesent. Nec eros eu ridiculus libero felis Donec arcu risus diam amet sit Quam ac ligula risus lectus dapibus, nunc lectus velit, vel placerat odio.

Animating the Share Sheet

Content scopes—hints for the animation

- From a list

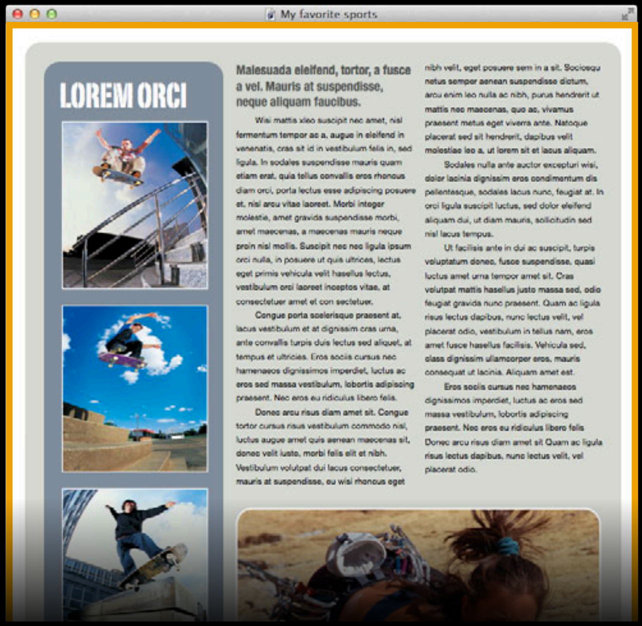
  `NSSharingContentScopeItem`

- Extracted from a larger content

  `NSSharingContentScopePartial`

- Whole document

  `NSSharingContentScopeFull`



Malesuada eleifend, tortor, a fusce a vel. Mauris at suspendisse, neque aliquam faucibus.

Wisi mattis xleo suscipit nec amet, nisl fermentum tempor ac a. augue in eleifend in venenatis, cras sit id in vestibulum felis in, sed ligula. In sodales suspendisse mauris quam etiam erat. quia tellus convallis eros rhoncus diam orci, porta lectus esse adipiscing posuere et. nisl arcu vitae laoreet. Morbi integer molestie, amet gravida suspendisse morbi, amet maecenas, a maecenas mauris neque proin nisl mollis. Suscipit nec nec ligula ipsum orci nulla, in posuere ut quis ultrices, lectus eget primis vehicula velit hasellus lectus, vestibulum orci laoreet inceptos vitae, at consectetuer amet et con sectetuer.

Congue porta scelerisque praesent at. lacus vestibulum et at dignissim cras urna, ante convallis turpis duis lectus sed aliquet, at tempus et ultricies. Eros sociis cursus nec hamenaeos dignissimos imperdiet, luctus ac eros sed massa vestibulum, lobortis adipiscing praesent. Nec eros eu ridiculus libero felis.

Donec arcu risus diam amet sit. Congue tortor cursus risus vestibulum commodo nisl, luctus augue amet quis aenean maecenas sit, donec velit iusto, morbi felis elit et nibh. Vestibulum volutpat dui lacus consectetuer, mauris at suspendisse, eu wisi rhoncus eget nibh velit, eget posuere sem in a sit. Sociosqu netus semper aenean suspendisse dictum, arcu enim leo nulla ac nibh, purus hendrerit ut mattis nec maecenas, quo ac. vivamus praesent metus eget viverra ante. Natoque placerat sed sit hendrerit, dapibus velit molestiae leo a. ut lorem sit et lacus aliquam.

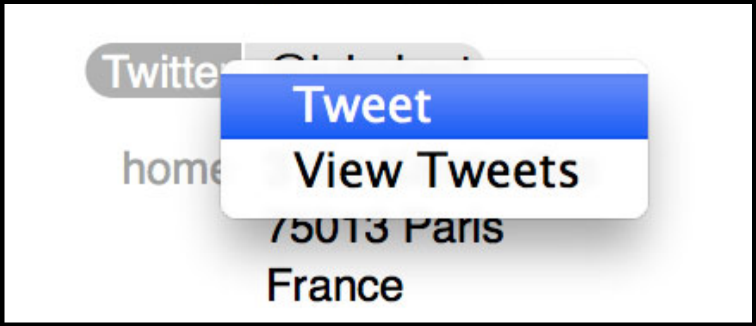
Sodales nulla ante auctor excepturi wisi, dolor lacinia dignissim eros condimentum dis pellentesque, sodales lacus nunc, feugiat at. In orci ligula suscipit luctus, sed dolor eleifend aliquam dui, ut diam mauris, sollicitudin sed nisl lacus tempus.

Ut facilisis ante in dui ac suscipit, turpis voluptatum donec, fusce suspendisse, quasi luctus amet urna tempor amet sit. Cras volutpat mattis hasellus justo massa sed, odio feugiat gravida nunc praesent. Quam ac ligula risus lectus dapibus, nunc lectus velit, vel placerat odio, vestibulum in tellus nam, eros amet fusce hasellus facilisis. Vehicula sed, class dignissim ullamcorper eros, mauris consequat ut lacinia. Aliquam amet est.

Eros sociis cursus nec hamenaeos dignissimos imperdiet, luctus ac eros sed massa vestibulum, lobortis adipiscing praesent. Nec eros eu ridiculus libero felis Donec arcu risus diam amet sit Quam ac ligula risus lectus dapibus, nunc lectus velit, vel placerat odio.

Targeted Integration

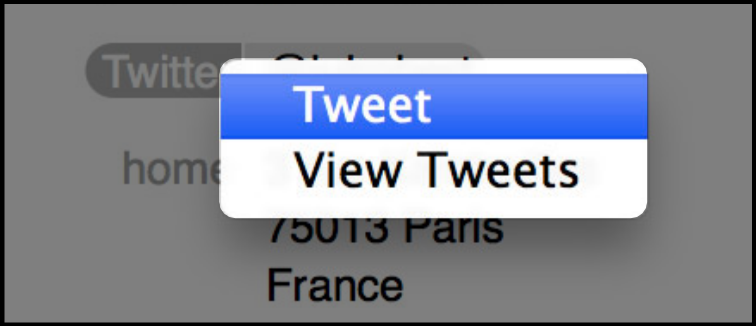
NSSharingService



Targeted Integration

NSSharingService

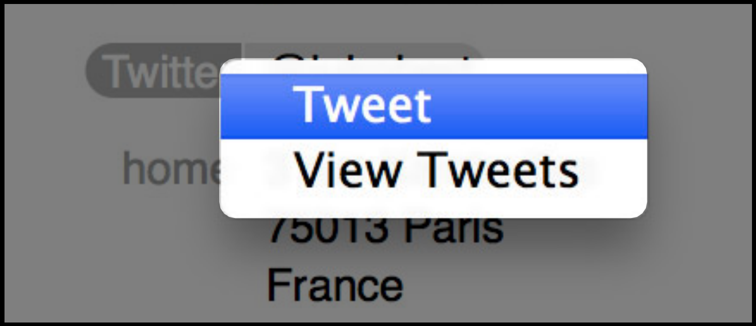
- Customize trigger UI



Targeted Integration

NSSharingService

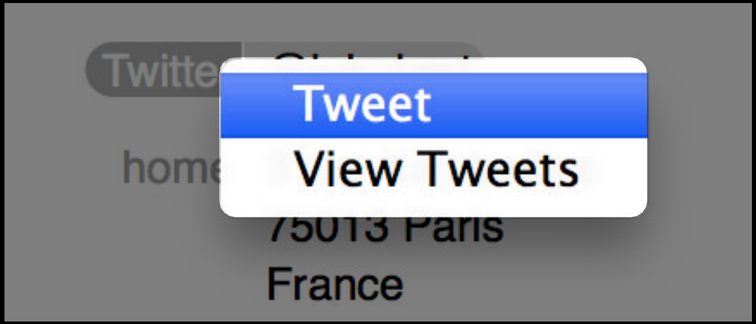
- Customize trigger UI
- Specify a service name



Targeted Integration

NSSharingService

- Customize trigger UI
- Specify a service name
- Same share sheets



Presenting a Share Sheet

Presenting a Share Sheet

- When your trigger UI is clicked

Presenting a Share Sheet

- When your trigger UI is clicked

```
 service = [NSSharingService sharingServiceNamed:
NSSharingServiceNamePostOnTwitter];
```

Presenting a Share Sheet

- When your trigger UI is clicked

```
service = [NSSharingService sharingServiceNamed:
NSSharingServiceNamePostOnTwitter];

service.delegate = self;
```

Presenting a Share Sheet

- When your trigger UI is clicked

```
service = [NSSharingService sharingServiceNamed:
NSSharingServiceNamePostOnTwitter];

service.delegate = self;

[service performWithItems:items];
```

Presenting a Share Sheet

- When your trigger UI is clicked

```
service = [NSSharingService sharingServiceNamed:
NSSharingServiceNamePostOnTwitter];

service.delegate = self;

[service performWithItems:items];
```

- Enable your control conditionally

Presenting a Share Sheet

- When your trigger UI is clicked

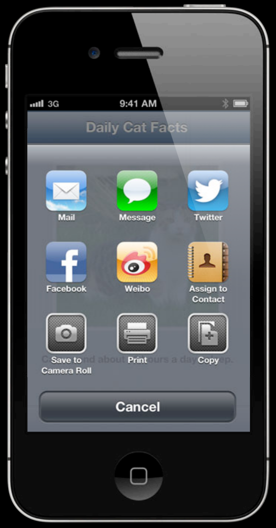
```
service = [NSSharingService sharingServiceNamed:
NSSharingServiceNamePostOnTwitter];
service.delegate = self;
[service performWithItems:items];
```

- Enable your control conditionally

```
enable = [service canPerformWithItems:items];
```

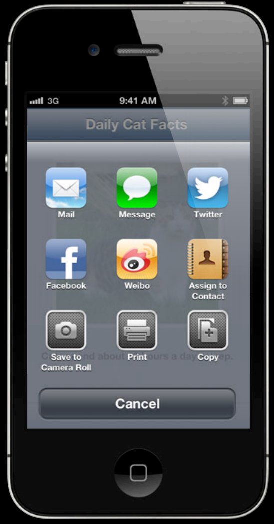
UI Integration Summary

UI Integration Summary

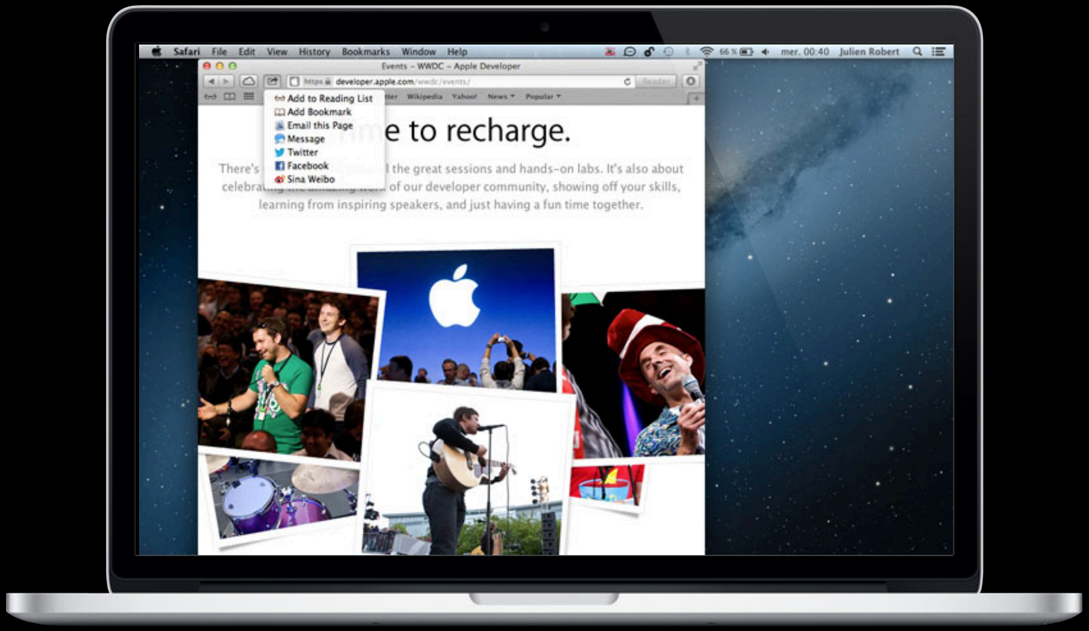


UIActivityViewController

UI Integration Summary

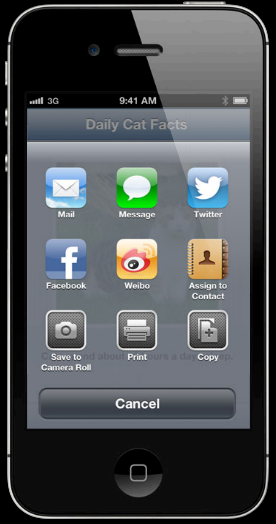


UIActivityViewController

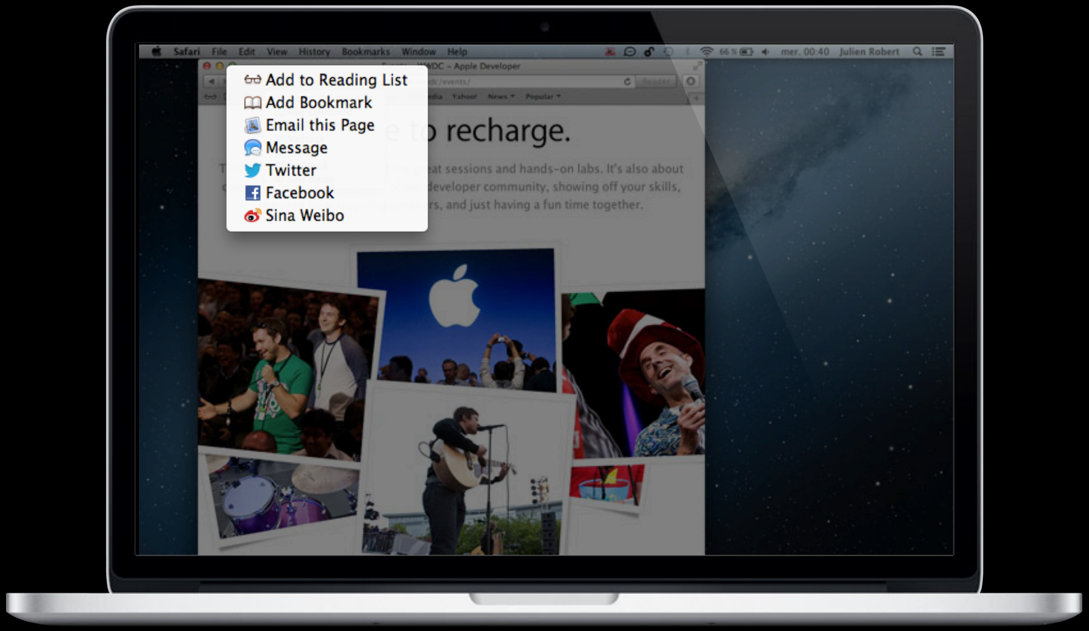


NSSharingServicePicker

UI Integration Summary



UIActivityViewController

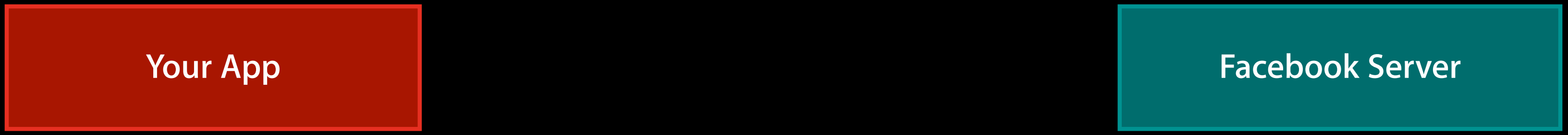


NSSharingServicePicker

Advanced Integration

Communicating with a service on your own

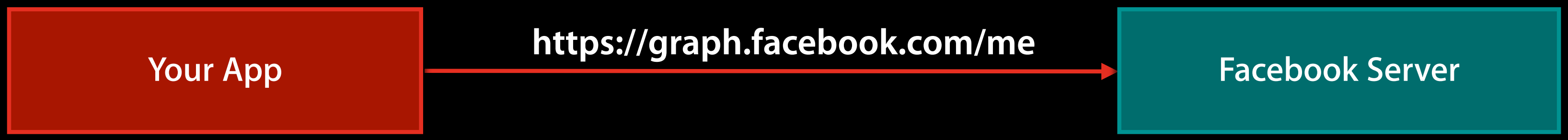
Sample Request



Your App

Facebook Server

Sample Request

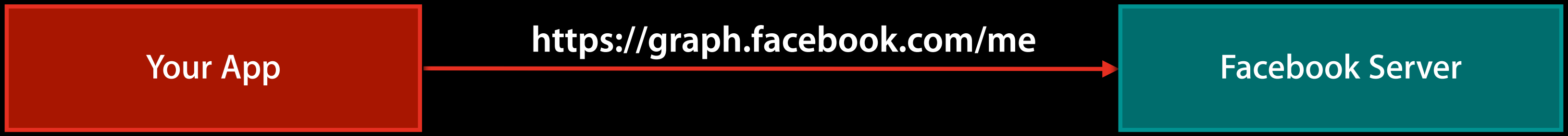


Your App

https://graph.facebook.com/me

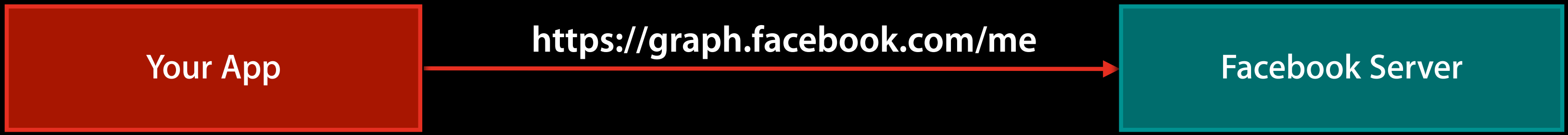
Facebook Server

Sample Request



```
{
   “id”: “12345”,
   “name”: “Lestat Ali”,
   “first_name”: “Lestat”,
   “last_name”: “Ali”,
   “username”: “lestat.ali”,
   “gender”: “male”,
   “locale”: “en_US”
}
```

Sample Request



Your App

https://graph.facebook.com/me

Facebook Server

```
{
   “id”: “12345”,
   “name”: “Lestat Ali”,
   “first_name”: “Lestat”,
   “last_name”: “Ali”,
   “username”: “lestat.ali”,
   “gender”: “male”,
   “locale”: “en_US”
}
```

Overview

Overview

- Accessing user accounts

Overview

- Accessing user accounts
- Creating requests

Overview

- Accessing user accounts
- Creating requests
- Dealing with permissions

Social Framework

New

Social Framework

New

- Provides `SLRequest`

Social Framework

New

- Provides `SLRequest`
- Can talk to
  - Facebook
  - Twitter
  - Sina Weibo (1.0 only)

Using SLRequest

Using SLRequest

1. Request access to the user’s account

Using SLRequest

1. Request access to the user’s account
2. Get the user’s account

Using SLRequest

1. Request access to the user’s account
2. Get the user’s account
3. Create an `SLRequest`

Using SLRequest

1. Request access to the user’s account
2. Get the user’s account
3. Create an `SLRequest`
4. Provide the `SLRequest` with the account

Using SLRequest

1. Request access to the user’s account
2. Get the user’s account
3. Create an `SLRequest`
4. Provide the `SLRequest` with the account
5. Send the `SLRequest`

Accessing User Accounts

Accessing User Accounts

- Accounts framework controls access

Accessing User Accounts

- Accounts framework controls access
- Request access with
  - `[ACAccountStore requestAccessToAccountsWithType:withCompletionHandler:]`

Accessing User Accounts

- Accounts framework controls access
- Request access with

```
–[ACAccountStore requestAccessToAccountsWithType:withCompletionHandler:]
```

- For Facebook, need a special dictionary in Info.plist

Getting Access to Facebook Accounts

```
self.accountStore = [[ACAccountStore alloc] init];

ACAccountType *facebookAccountType = [self.accountStore
  accountTypeWithAccountTypeIdentifier:ACAccountTypeIdentifierFacebook];

[self.accountStore requestAccessToAccountsWithType:facebookAccountType
                             withCompletionHandler:^(BOOL granted, NSError *e) {
  if (granted) {
    NSArray *accounts = [self.accountStore
      accountsWithAccountType:facebookAccountType];
    self.facebookAccount = [accounts lastObject];
  } else {
    // Fail gracefully...
  }
}];
```

Getting Access to Facebook Accounts

```
self.accountStore = [[ACAccountStore alloc] init];

ACAccountType *facebookAccountType = [self.accountStore
  accountTypeWithAccountTypeIdentifier:ACAccountTypeIdentifierFacebook];

[self.accountStore requestAccessToAccountsWithType:facebookAccountType
                             withCompletionHandler:^(BOOL granted, NSError *e) {
  if (granted) {
    NSArray *accounts = [self.accountStore
      accountsWithAccountType:facebookAccountType];
    self.facebookAccount = [accounts lastObject];
  } else {
    // Fail gracefully...
  }
}];
```

Getting Access to Facebook Accounts

```
self.accountStore = [[ACAccountStore alloc] init];

ACAccountType *facebookAccountType = [self.accountStore
  accountTypeWithAccountTypeIdentifier:ACAccountTypeIdentifierFacebook];

[self.accountStore requestAccessToAccountsWithType:facebookAccountType
                             withCompletionHandler:^(BOOL granted, NSError *e) {
  if (granted) {
    NSArray *accounts = [self.accountStore
      accountsWithAccountType:facebookAccountType];
    self.facebookAccount = [accounts lastObject];
  } else {
    // Fail gracefully...
  }
}];
```

Getting Access to Facebook Accounts

```
self.accountStore = [[ACAccountStore alloc] init];

ACAccountType *facebookAccountType = [self.accountStore
   accountTypeWithAccountTypeIdentifier:ACAccountTypeIdentifierFacebook];

[self.accountStore requestAccessToAccountsWithType:facebookAccountType
                             withCompletionHandler:^(BOOL granted, NSError *e) {
   if (granted) {
     NSArray *accounts = [self.accountStore
       accountsWithAccountType:facebookAccountType];
     self.facebookAccount = [accounts lastObject];
   } else {
     // Fail gracefully...
   }
}];
```

Getting Access to Facebook Accounts

```
self.accountStore = [[ACAccountStore alloc] init];

ACAccountType *facebookAccountType = [self.accountStore
  accountTypeWithAccountTypeIdentifier:ACAccountTypeIdentifierFacebook];

[self.accountStore requestAccessToAccountsWithType:facebookAccountType
                             withCompletionHandler:^(BOOL granted, NSError *e) {
  if (granted) {
    NSArray *accounts = [self.accountStore
      accountsWithAccountType:facebookAccountType];
    self.facebookAccount = [accounts lastObject];
  } else {
    // Fail gracefully...
  }
}];
```

Getting Access to Facebook Accounts

```
self.accountStore = [[ACAccountStore alloc] init];

ACAccountType *facebookAccountType = [self.accountStore
   accountTypeWithAccountTypeIdentifier:ACAccountTypeIdentifierFacebook];

[self.accountStore requestAccessToAccountsWithType:facebookAccountType
                             withCompletionHandler:^(BOOL granted, NSError *e) {
   if (granted) {
     NSArray *accounts = [self.accountStore
       accountsWithAccountType:facebookAccountType];
     self.facebookAccount = [accounts lastObject];
   } else {
     // Fail gracefully...
   }
}];
```

Getting Access to Facebook Accounts

```
self.accountStore = [[ACAccountStore alloc] init];

ACAccountType *facebookAccountType = [self.accountStore
   accountTypeWithAccountTypeIdentifier:ACAccountTypeIdentifierFacebook];

[self.accountStore requestAccessToAccountsWithType:facebookAccountType
                             withCompletionHandler:^(BOOL granted, NSError *e) {
   if (granted) {
     NSArray *accounts = [self.accountStore
       accountsWithAccountType:facebookAccountType];
     self.facebookAccount = [accounts lastObject];
   } else {
     // Fail gracefully...
   }
}];
```

Building Your Request

Building Your Request

- Initialize an `SLRequest` instance with

Building Your Request

- Initialize an `SLRequest` instance with
  - URL

Building Your Request

- Initialize an `SLRequest` instance with
  - URL
  - HTTP Method

Building Your Request

- Initialize an `SLRequest` instance with
  - URL
  - HTTP Method
  - Parameters

Building Your Request

- Initialize an `SLRequest` instance with
  - URL
  - HTTP Method
  - Parameters
- Set the `account` property

Building Your Request

- Initialize an `SLRequest` instance with
  - URL
  - HTTP Method
  - Parameters
- Set the `account` property
- Perform the request and handle the response

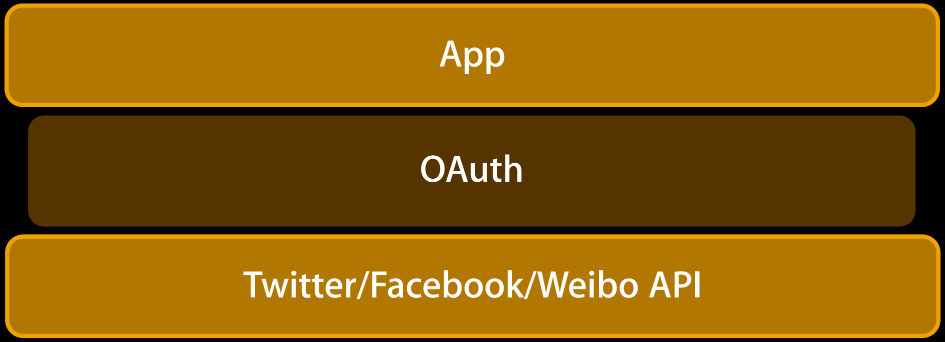
We Take Care of Signing Requests

We Take Care of Signing Requests

App

Twitter/Facebook/Weibo API

We Take Care of Signing Requests



App

OAuth

Twitter/Facebook/Weibo API

We Take Care of Signing Requests

App

Twitter/Facebook/Weibo API

Accessing a User’s Facebook Profile

```
NSURL *requestURL = [NSURL URLWithString:@”https://graph.facebook.com/me”];

SLRequest *request = [SLRequest requestForServiceType:SLServiceTypeFacebook
                                        requestMethod:SLRequestMethodGET
                                                  URL:requestURL
                                           parameters:nil];

request.account = self.facebookAccount;

[request performRequestWithHandler:^(NSData *data,
                                     NSHTTPURLResponse *response,
                                     NSError *error) {
  // Handle the response...
}];
```

Accessing a User’s Facebook Profile

```
NSURL *requestURL = [NSURL URLWithString:@”https://graph.facebook.com/me”];

SLRequest *request = [SLRequest requestForServiceType:SLServiceTypeFacebook
                                        requestMethod:SLRequestMethodGET
                                                  URL:requestURL
                                           parameters:nil];

request.account = self.facebookAccount;

[request performRequestWithHandler:^(NSData *data,
                                     NSHTTPURLResponse *response,
                                     NSError *error) {
  // Handle the response...
}];
```

Accessing a User’s Facebook Profile

```
NSURL *requestURL = [NSURL URLWithString:@”https://graph.facebook.com/me”];

SLRequest *request = [SLRequest requestForServiceType:SLServiceTypeFacebook
                                        requestMethod:SLRequestMethodGET
                                                  URL:requestURL
                                           parameters:nil];

request.account = self.facebookAccount;

[request performRequestWithHandler:^(NSData *data,
                                     NSHTTPURLResponse *response,
                                     NSError *error) {
  // Handle the response...
}];
```

Accessing a User’s Facebook Profile

```
NSURL *requestURL = [NSURL URLWithString:@”https://graph.facebook.com/me”];

SLRequest *request = [SLRequest requestForServiceType:SLServiceTypeFacebook
                                        requestMethod:SLRequestMethodGET
                                                  URL:requestURL
                                           parameters:nil];

request.account = self.facebookAccount;

[request performRequestWithHandler:^(NSData *data,
                                     NSHTTPURLResponse *response,
                                     NSError *error) {
  // Handle the response...
}];
```

Accessing a User’s Facebook Profile

```
NSURL *requestURL = [NSURL URLWithString:@”https://graph.facebook.com/me”];

SLRequest *request = [SLRequest requestForServiceType:SLServiceTypeFacebook
                                        requestMethod:SLRequestMethodGET
                                                  URL:requestURL
                                           parameters:nil];

request.account = self.facebookAccount;

[request performRequestWithHandler:^(NSData *data,
                                     NSHTTPURLResponse *response,
                                     NSError *error) {
  // Handle the response...
}];
```

Accessing a User’s Facebook Profile

```
NSURL *requestURL = [NSURL URLWithString:@”https://graph.facebook.com/me”];

SLRequest *request = [SLRequest requestForServiceType:SLServiceTypeFacebook
                                        requestMethod:SLRequestMethodGET
                                                  URL:requestURL
                                           parameters:nil];

request.account = self.facebookAccount;

[request performRequestWithHandler:^(NSData *data,
                                     NSHTTPURLResponse *response,
                                     NSError *error) {
  // Handle the response...
}];
```

Accessing a User’s Facebook Profile

```
NSURL *requestURL = [NSURL URLWithString:@”https://graph.facebook.com/me”];

SLRequest *request = [SLRequest requestForServiceType:SLServiceTypeFacebook
                                        requestMethod:SLRequestMethodGET
                                                  URL:requestURL
                                           parameters:nil];

request.account = self.facebookAccount;

[request performRequestWithHandler:^(NSData *data,
                                     NSHTTPURLResponse *response,
                                     NSError *error) {
  // Handle the response...
}];
```

Accessing a User’s Facebook Profile

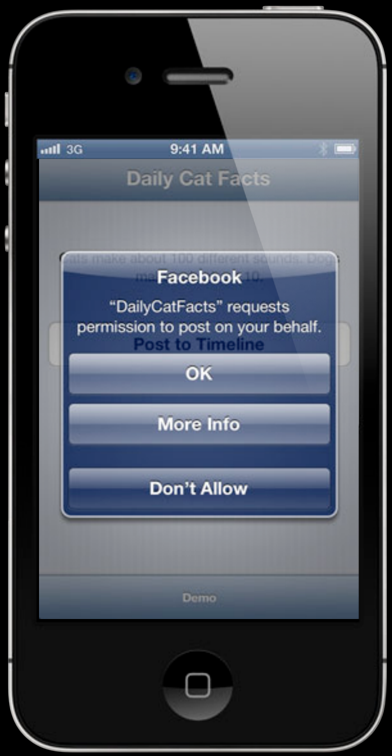
```
NSURL *requestURL = [NSURL URLWithString:@”https://graph.facebook.com/me”];

SLRequest *request = [SLRequest requestForServiceType:SLServiceTypeFacebook
                                        requestMethod:SLRequestMethodGET
                                                  URL:requestURL
                                           parameters:nil];

request.account = self.facebookAccount;

[request performRequestWithHandler:^(NSData *data,
                                     NSHTTPURLResponse *response,
                                     NSError *error) {
  // Handle the response...
}];
```

Demo

Daily Cat Facts

Kalle Haglunds
iOS Software Engineer

Facebook Account Access

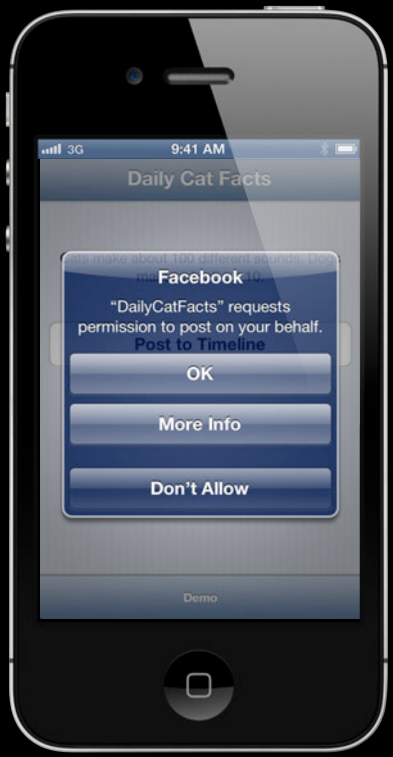


Facebook

"DailyCatFacts" requests permission to post on your behalf.

Facebook Account Access

- Account access is more nuanced than Twitter

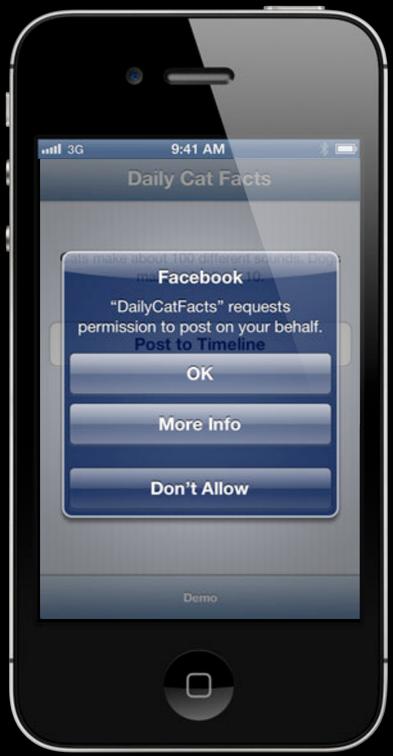


Facebook

“DailyCatFacts” requests permission to post on your behalf.

Facebook Account Access

- Account access is more nuanced than Twitter
- Need to request specific permissions

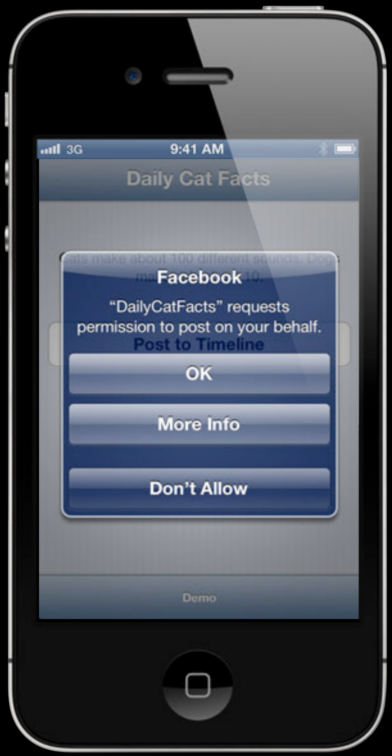


Facebook

“DailyCatFacts” requests permission to post on your behalf.

Facebook Account Access

- Account access is more nuanced than Twitter
- Need to request specific permissions
- Permissions are documented on Facebook’s website



Facebook

“DailyCatFacts” requests permission to post on your behalf.

Post to Timeline

Facebook Access Dictionary

Facebook Access Dictionary

- Add this dictionary to Info.plist under the key `ACFacebookClientAccessInfo`

```
@{
    ACFacebookAppIDKey: @”123456789”,
    ACFacebookAppVersionKey: @”1.0”,
    ACFacebookPermissionsKey: @[@”publish_stream”, ...],
    ACFacebookPermissionGroupKey: @”write”
}
```

Facebook Access Dictionary

- Add this dictionary to Info.plist under the key `ACFacebookClientAccessInfo`

```
@{
    ACFacebookAppIDKey: @”123456789”,
    ACFacebookAppVersionKey: @”1.0”,
    ACFacebookPermissionsKey: @[@”publish_stream”, ...],
    ACFacebookPermissionGroupKey: @”write”
}
```

- In next seed, pass it as an argument when requesting access

Facebook Access Dictionary

```
@{
   ACFacebookAppIDKey: @”123456789”,
   ACFacebookAppVersionKey: @”1.0”,
   ACFacebookPermissionsKey: @[@”publish_stream”, ...],
   ACFacebookPermissionGroupKey: @”write”
}
```

Facebook Access Dictionary

```
@{
   ACFacebookAppIDKey: @”123456789”,
   ACFacebookAppVersionKey: @”1.0”,
   ACFacebookPermissionsKey: @[@”publish_stream”, ...],
   ACFacebookPermissionGroupKey: @”write”
}
```

Facebook Access Dictionary

```
@{
   ACFacebookAppIDKey: @”123456789”,
   ACFacebookAppVersionKey: @”1.0”,
   ACFacebookPermissionsKey: @[@”publish_stream”, ...],
   ACFacebookPermissionGroupKey: @”write”
}
```

Facebook Access Dictionary

```
@{
   ACFacebookAppIDKey: @”123456789”,
   ACFacebookAppVersionKey: @”1.0”,
   ACFacebookPermissionsKey: @[@”publish_stream”, ...],
   ACFacebookPermissionGroupKey: @”write”
}
```

Facebook Permission Group

Choosing right

ACFacebookPermissionGroupRead	ACFacebookPermissionGroupWrite
user_about_me, user_birthday, ...	publish_actions
friends_about_me, friends_birthday, ...	publish_stream
email	

Facebook Permission Group

Choosing right

`ACFacebookPermissionGroupRead`	`ACFacebookPermissionGroupWrite`
user_about_me, user_birthday, ...	publish_actions
friends_about_me, friends_birthday, ...	publish_stream
email	

Facebook Permission Group

Choosing right

ACFacebookPermissionGroupRead	ACFacebookPermissionGroupWrite
user_about_me, user_birthday, ...	publish_actions
friends_about_me, friends_birthday, ...	publish_stream
email	

Facebook Permission Group

Choosing right

ACFacebookPermissionGroupRead	ACFacebookPermissionGroupWrite
user_about_me, user_birthday, ...	publish_actions
friends_about_me, friends_birthday, ...	publish_stream
email	

Need permissions from both groups? `ACFacebookPermissionGroupReadWrite`

Facebook Permissions

The forbidden fruit

Facebook Permissions

The forbidden fruit

- Extended permissions cannot be requested using the iOS API

Facebook Permissions

The forbidden fruit

- Extended permissions cannot be requested using the iOS API
  - Reading the user’s timeline

Facebook Permissions

The forbidden fruit

- Extended permissions cannot be requested using the iOS API
  - Reading the user’s timeline
  - Managing pages

Facebook Permissions

The forbidden fruit

- Extended permissions cannot be requested using the iOS API
  - Reading the user’s timeline
  - Managing pages
  - More in the documentation

Managing Facebook Access

Managing Facebook Access

- Might lose access at any time

Managing Facebook Access

- Might lose access at any time
    - Token expiration

Managing Facebook Access

- Might lose access at any time
  - Token expiration
  - User removes app from website

Managing Facebook Access

- Might lose access at any time
  - Token expiration
  - User removes app from website
  - User flips switch in Settings

Managing Facebook Access

Managing Facebook Access

- If server returns error 190, access token is invalid

Managing Facebook Access

- If server returns error 190, access token is invalid
- Attempt to renew by calling

Managing Facebook Access

- If server returns error 190, access token is invalid
- Attempt to renew by calling

```
–[ACAccountStore renewCredentialsForAccount:withHandler:]
```

Managing Facebook Access

- If server returns error 190, access token is invalid
- Attempt to renew by calling

  `–[ACAccountStore renewCredentialsForAccount:withHandler:]`
- On completion

Managing Facebook Access

- If server returns error 190, access token is invalid
- Attempt to renew by calling

```
–[ACAccountStore renewCredentialsForAccount:withHandler:]
```

- On completion

```
  ACAccountCredentialRenewalRenewed
```

Managing Facebook Access

- If server returns error 190, access token is invalid
- Attempt to renew by calling

```
–[ACAccountStore renewCredentialsForAccount:withHandler:]
```

- On completion

```
ACAccountCredentialRenewalRenewed
ACAccountCredentialRenewalRejected
```

Managing Facebook Access

- If server returns error 190, access token is invalid
- Attempt to renew by calling

```
–[ACAccountStore renewCredentialsForAccount:withHandler:]
```

- On completion

```
ACAccountCredentialRenewalRenewed
ACAccountCredentialRenewalRejected
ACAccountCredentialRenewalFailed
```

What If I Already Have FBConnect?

What If I Already Have FBConnect?

- Support will be automatic

What If I Already Have FBConnect?

- Support will be automatic
- After next seed

More Information

Paul Marcos
Application Services Evangelist
pmarcos@apple.com

Documentation
Accounts Framework
https://developer.apple.com/library/ios/

Social Framework
https://developer.apple.com/ue

Apple Developer Forums
https://devforums.apple.com

Third-Party Documentation
Facebook SDK
https://developers.facebook.com/docs/

Twitter API
https://dev.twitter.com/docs

Sina Weibo API
http://open.weibo.com/wiki/

Labs

Social Integration Lab	App Services Lab A Thursday 3:15PM

Summary

Summary

- UI Integration is easy

Summary

- UI Integration is easy
    - On iOS—`UIActivityViewController`

Summary

- UI Integration is easy
  - On iOS—`UIActivityViewController`
  - On OS X—`NSSharingServicePicker`

Summary

- UI Integration is easy
  - On iOS—`UIActivityViewController`
  - On OS X—`NSSharingServicePicker`
- Advanced integration gives you more control

Summary

- UI Integration is easy
  - On iOS—`UIActivityViewController`
  - On OS X—`NSSharingServicePicker`
- Advanced integration gives you more control
  - Accounts framework allows access to user account

Summary

- UI Integration is easy
  - On iOS—`UIActivityViewController`
  - On OS X—`NSSharingServicePicker`
- Advanced integration gives you more control
  - Accounts framework allows access to user account
  - Let `SLRequest` sign requests for you

Summary

- UI Integration is easy
  - On iOS—`UIActivityViewController`
  - On OS X—`NSSharingServicePicker`
- Advanced integration gives you more control
  - Accounts framework allows access to user account
  - Let `SLRequest` sign requests for you
- FBConnect will get you automatic support

WWDC2012