Apple Pay within Apps

Session 702

Nick Shearer iOS Apps and Frameworks
Rachel Roth User Experience Evangelist

Overview

Overview

What is Apple Pay?
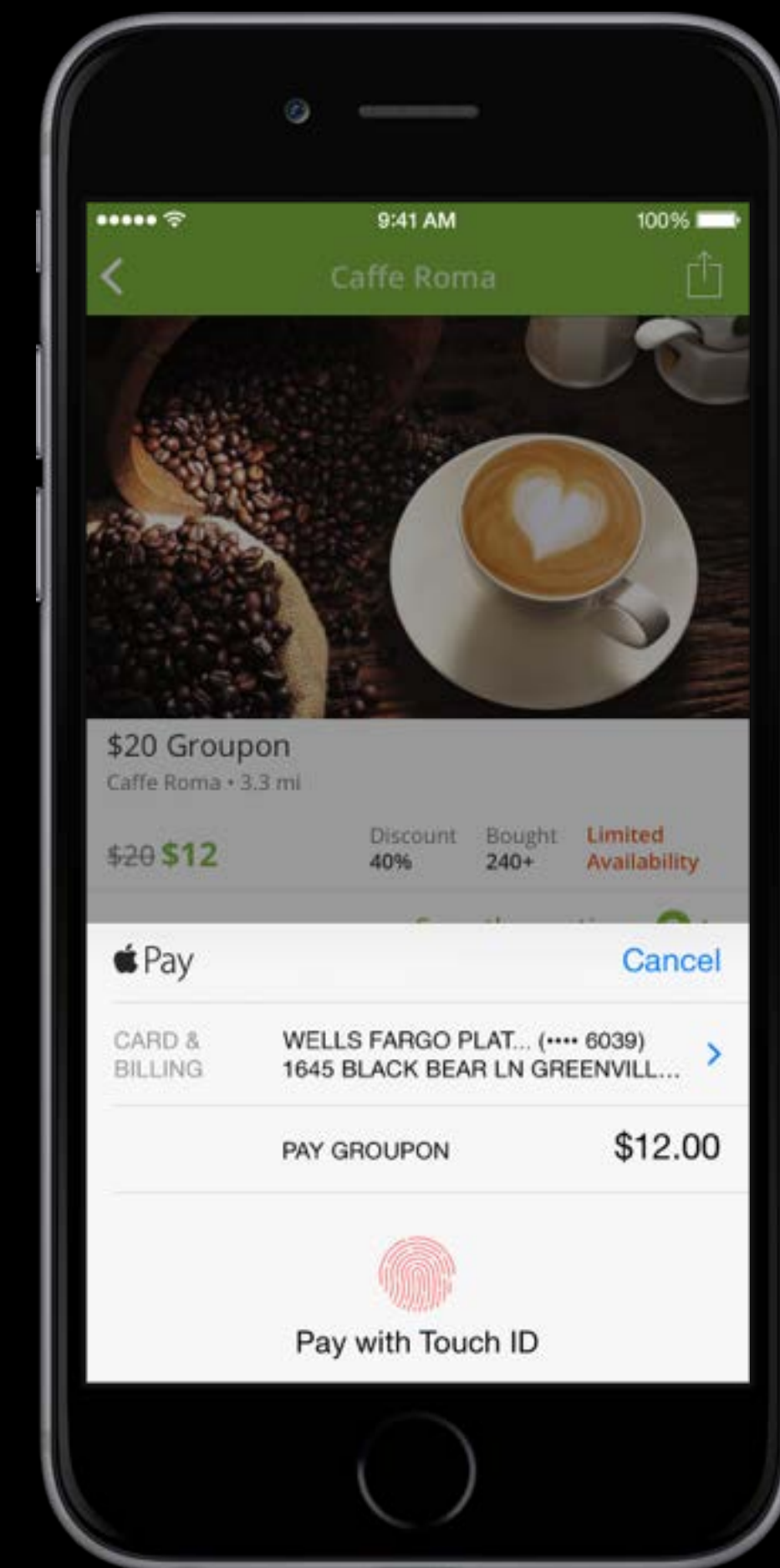

Architecture

Design

Code

What is Apple Pay?

And how do I use it within my app?

An easy, secure, and private way to
pay within app

One-touch payments

Use for physical goods and services

Apple Pay
User benefits

Easy—No need to re-enter payment and contact information

Secure—Pay using Touch ID

Private—Card number not exposed to the merchant

Apple Pay
Developer benefits

Don't need to handle actual card numbers

Higher conversion rates and faster checkouts

No onboarding or existing accounts required

Users love it

Apple Pay users transact more frequently

20%

Apple Pay users transact more frequently

Transaction growth with Apple Pay

OpenTable™

50%

Transaction growth with Apple Pay

 OpenTable™

Increase in overall conversion

STAPLES®

109%

Increase in overall conversion

STAPLES

"Apple Pay is not only driving more purchases but also activating our biggest spenders."

Joe Einhorn, CEO

FANCY

Apple Pay or In-App Purchase?

Apple Pay or In-App Purchase?

	Apple Pay	In-App Purchase
Framework	PassKit	StoreKit
Use For	Physical goods Services	In-App Content Functionality Services Subscriptions
Payment Processor	Developer responsible	Apple
App Store Review Guidelines	Section 29	Section 11

Apple Pay or In-App Purchase?

	Apple Pay	In-App Purchase
Framework	PassKit	StoreKit
Use For	Physical goods Services	In-App Content Functionality Services Subscriptions
Payment Processor	Developer responsible	Apple
App Store Review Guidelines	Section 29	Section 11

Apple Pay or In-App Purchase?

|	Apple Pay	In-App Purchase
Framework	PassKit	StoreKit
Use For	Physical goods Services	In-App Content Functionality Services Subscriptions
Payment Processor	Developer responsible	Apple
App Store Review Guidelines	Section 29	Section 11

Apple Pay or In-App Purchase?

	Apple Pay	In-App Purchase
Framework	PassKit	StoreKit
Use For	Physical goods Services	In-App Content Functionality Services Subscriptions
Payment Processor	Developer responsible	Apple
App Store Review Guidelines	Section 29	Section 11

Supported Devices

iPhone 6

iPhone 6 Plus

iPad Air 2

iPad mini 3
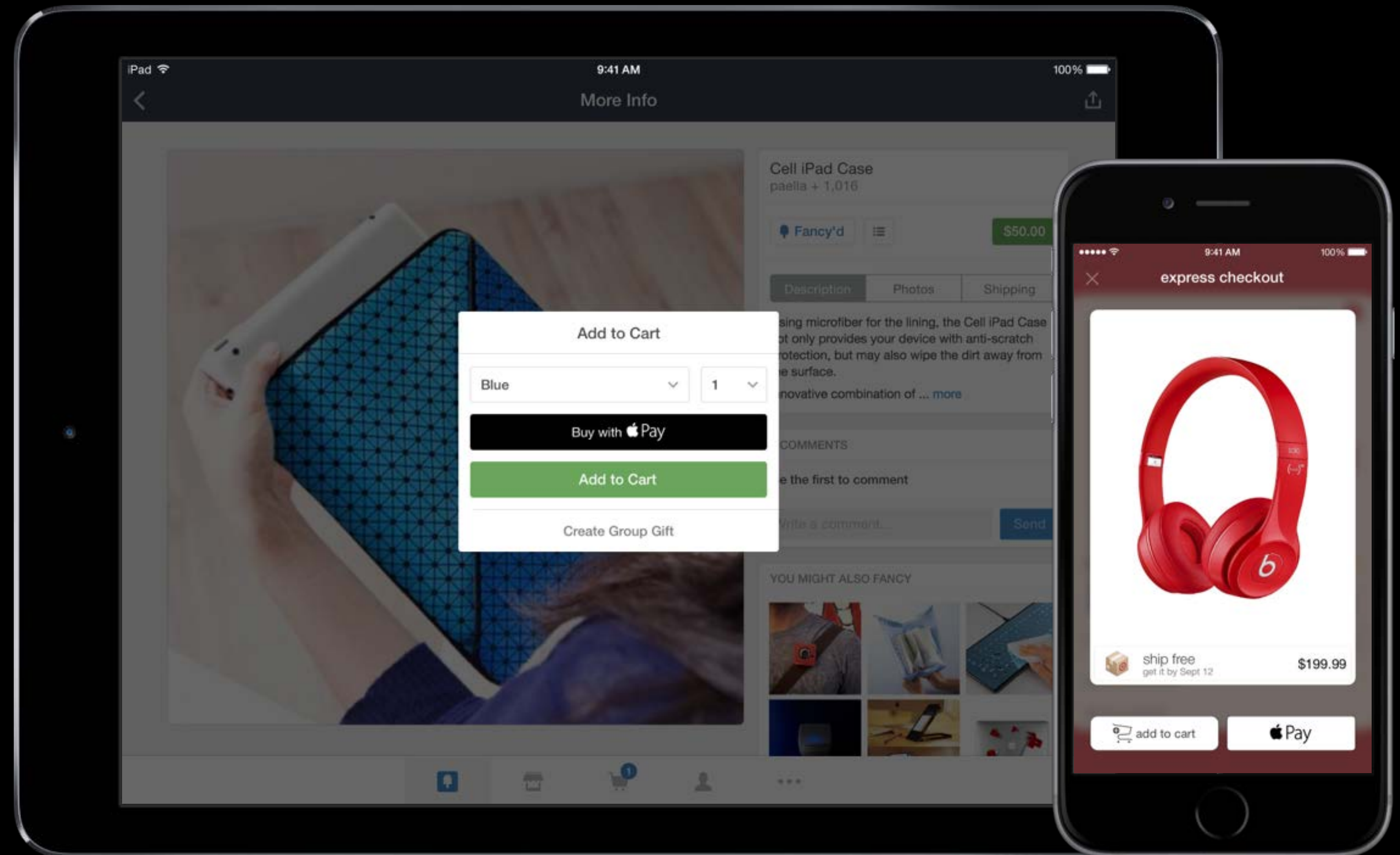
Supported Countries
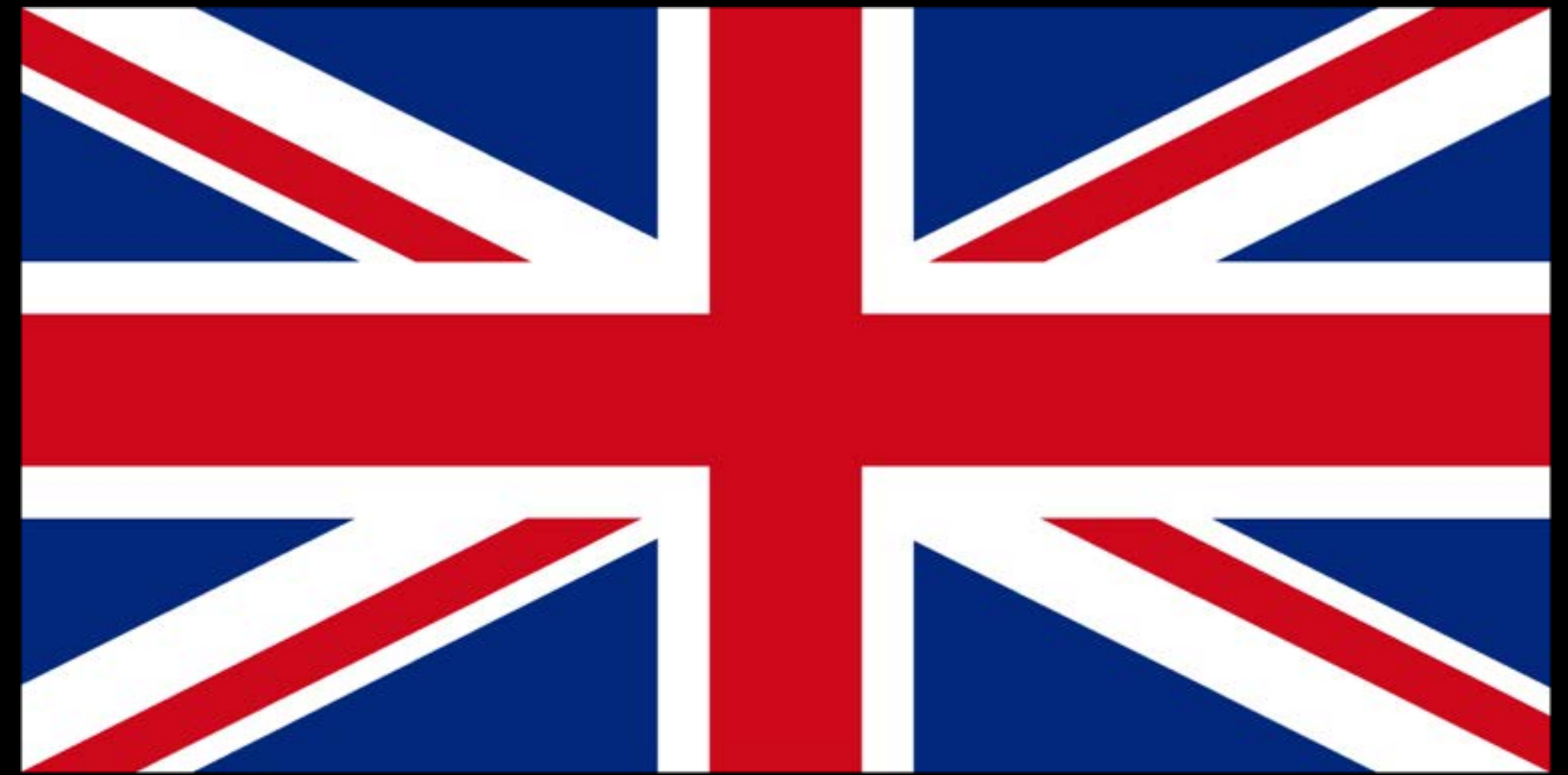
Apple Pay Architecture

How payments are made

First Steps
Merchant Identifier

Apple Pay requires a Merchant Identifier

Create a Merchant Identifier and matching certificate at the Developer Portal

Your identifier and certificate are used to encrypt payment data

Should be a reverse DNS format beginning `merchant.*`

merchant.com.wwdc.emporium

First Steps
Merchant Identifier

Apple Pay requires a Merchant Identifier

Create a Merchant Identifier and matching certificate at the Developer Portal

Your identifier and certificate are used to encrypt payment data

Should be a reverse DNS format beginning `merchant.*`

```
merchant.com.wwdc.emporium
```

Anatomy of a Payment

Anatomy of a Payment

Your app displays the payment sheet

Anatomy of a Payment

Your app displays the payment sheet

Touch ID is used to authorize

Anatomy of a Payment

Your app displays the payment sheet

Touch ID is used to authorize

Your app receives a payment token

Anatomy of a Payment

Your app displays the payment sheet

Touch ID is used to authorize

Your app receives a payment token

This token is encrypted using your Merchant ID's certificate

Anatomy of a Payment

Your app displays the payment sheet

Touch ID is used to authorize

Your app receives a payment token

This token is encrypted using your Merchant ID's certificate

Send this token to your processor

Payment Flow

Checkout

IOS

Apple Pay
Eligible?

Secure
Element
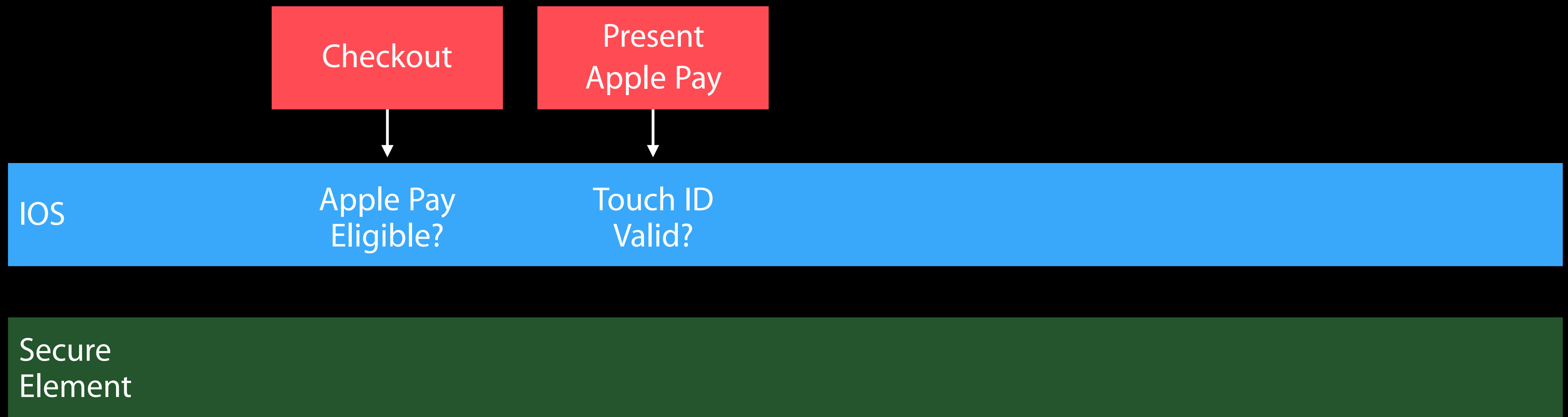
Payment Flow

Payment Flow

```
                    ┌──────────┐      ┌──────────┐
                    │ Checkout │      │  Present │
                    │          │      │ Apple Pay│
                    └────┬─────┘      └────┬─────┘
                         │                 │
                         ▼                 ▼
┌────────────────────────────────────────────────────────────────┐
│ IOS          Apple Pay        Touch ID                           │
│              Eligible?        Valid?                             │
└─────────────────────────────────┬──────────────────────────────┘
                                   │
                                   ▼
┌────────────────────────────────────────────────────────────────┐
│ Secure                   Create Cryptogram                       │
│ Element                  Encrypt Information                     │
└──────────────────────────────┬──────────────────────────────────┘
                                │
                                ▼
                            Apple
                            Server
```

Payment Flow

Checkout

Present Apple Pay

Send for Processing

Merchant Server

IOS

Apple Pay Eligible?

Touch ID Valid?

Secure Element

Create Cryptogram Encrypt Information

Apple Server

Apple Server
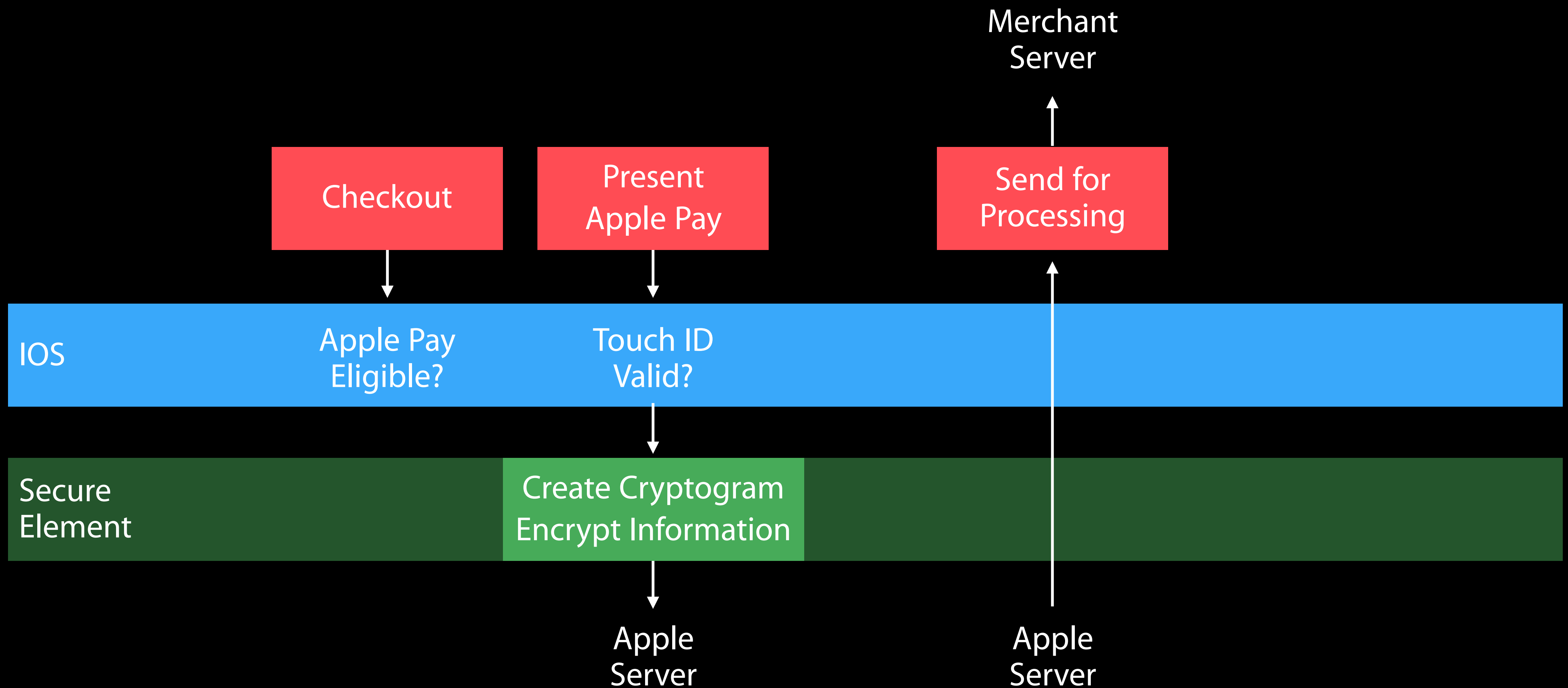
Payment Flow
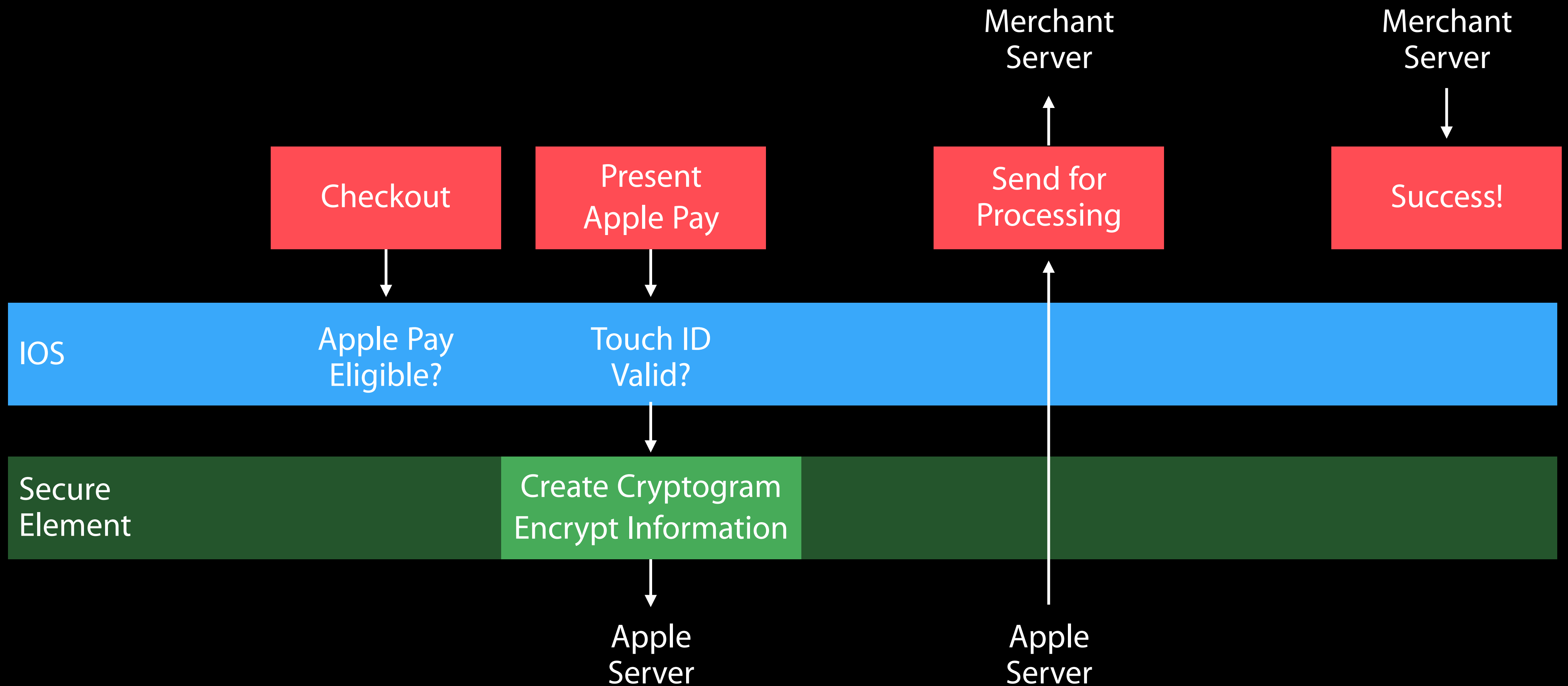
Processing Payments
Payment platforms

Handle decryption on your behalf

Simply send the payment token

Some provide native iOS development kits

The preferred option for most developers

US Payment Processors

US Payment Processors

adyen
wherever people pay

Braintree

First Data
b>yond the transaction

globalpayments

shopify

SIMPLIFY
commerce
by MasterCard

stripe

TSYS

vantiv

worldpay

UK Payment Processors

UK Payment Processors

adyen
wherever people pay

Braintree

First Data
b›yond the transaction

globalpayments

judo

shopify

SIMPLIFY
commerce
by MasterCard

stripe

VALITOR

worldpay

Processing Payments
Processing yourself

Decrypt the payment token yourself server-side

Send the underlying cryptogram and payment information to your acquirer

Only recommended for those with existing payment infrastructure

Further information: Payment Token Reference

Design

Make Shopping Easy

Make Shopping Easy

Account setup

Make Shopping Easy

Account setup

Lengthy forms

Make Shopping Easy

Account setup

Lengthy forms

Outdated billing and shipping information

Without Apple Pay

Without Apple Pay

Sign up now and start discovering the best accessories for your pet.

Your Email

Continue

Without Apple Pay

Cancel **Register**

You must be a member to view our pet accessories.

First Name

Last Name

Password

Confirm Password

Register

By registering, you confirm that you accept our Terms of Use and Privacy Policy.

Without Apple Pay

Without Apple Pay

Cancel	**Billing Address**	Next

Name

Street

Street

City
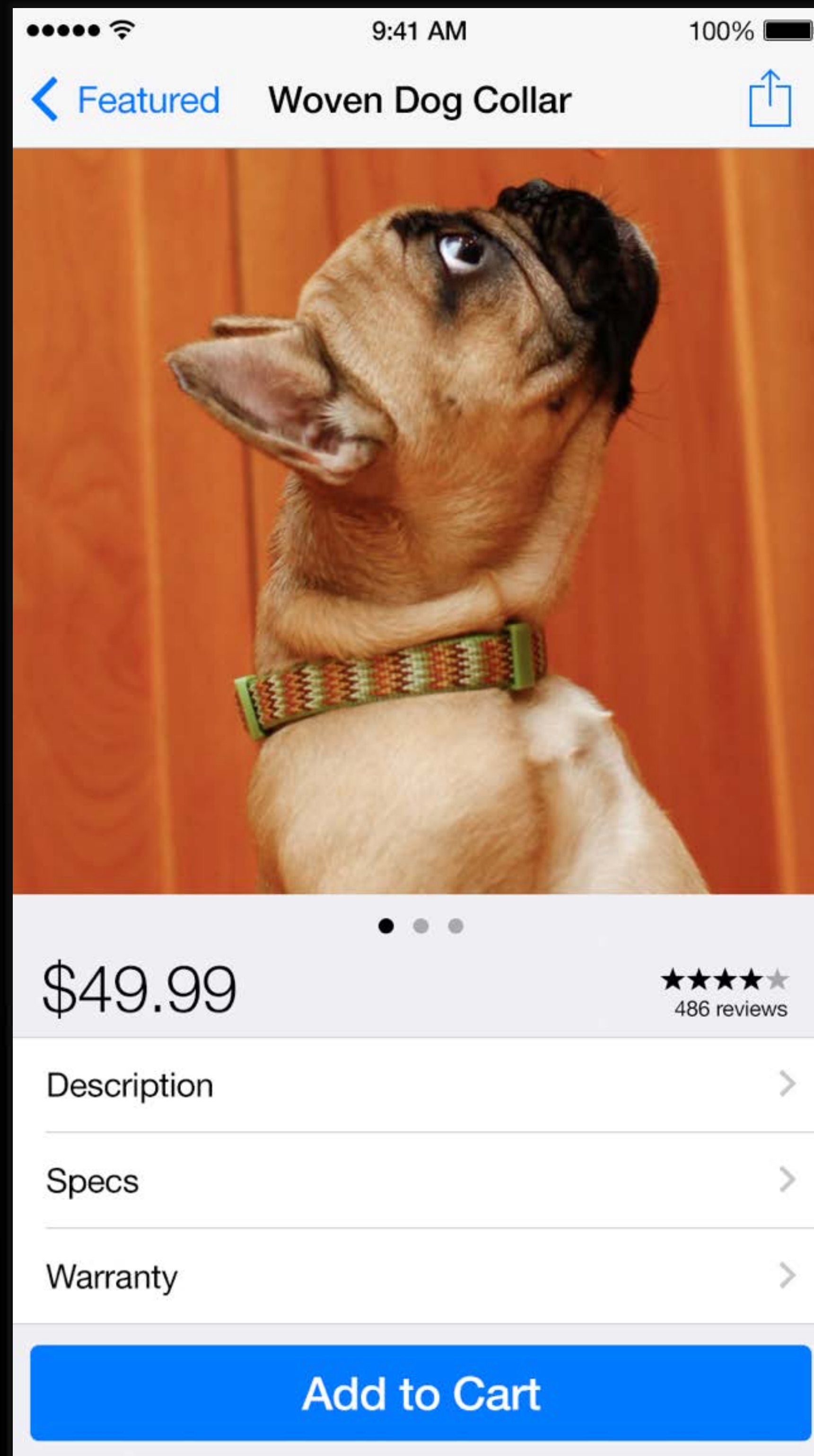
State | ZIP

Country

Without Apple Pay

9:41 AM	100%

Shipping Address

‹ Billing   **Shipping Address**   Next

Same as Billing ⚪

Name

Street

Street

City

State   ZIP

Country

Without Apple Pay

Shipping Options

Billing   Next

Standard
5 to 7 Business Days   Free

Express
2 to 3 Business Days $4.99

Overnight
1 Business Day $14.99

Without Apple Pay

With Apple Pay

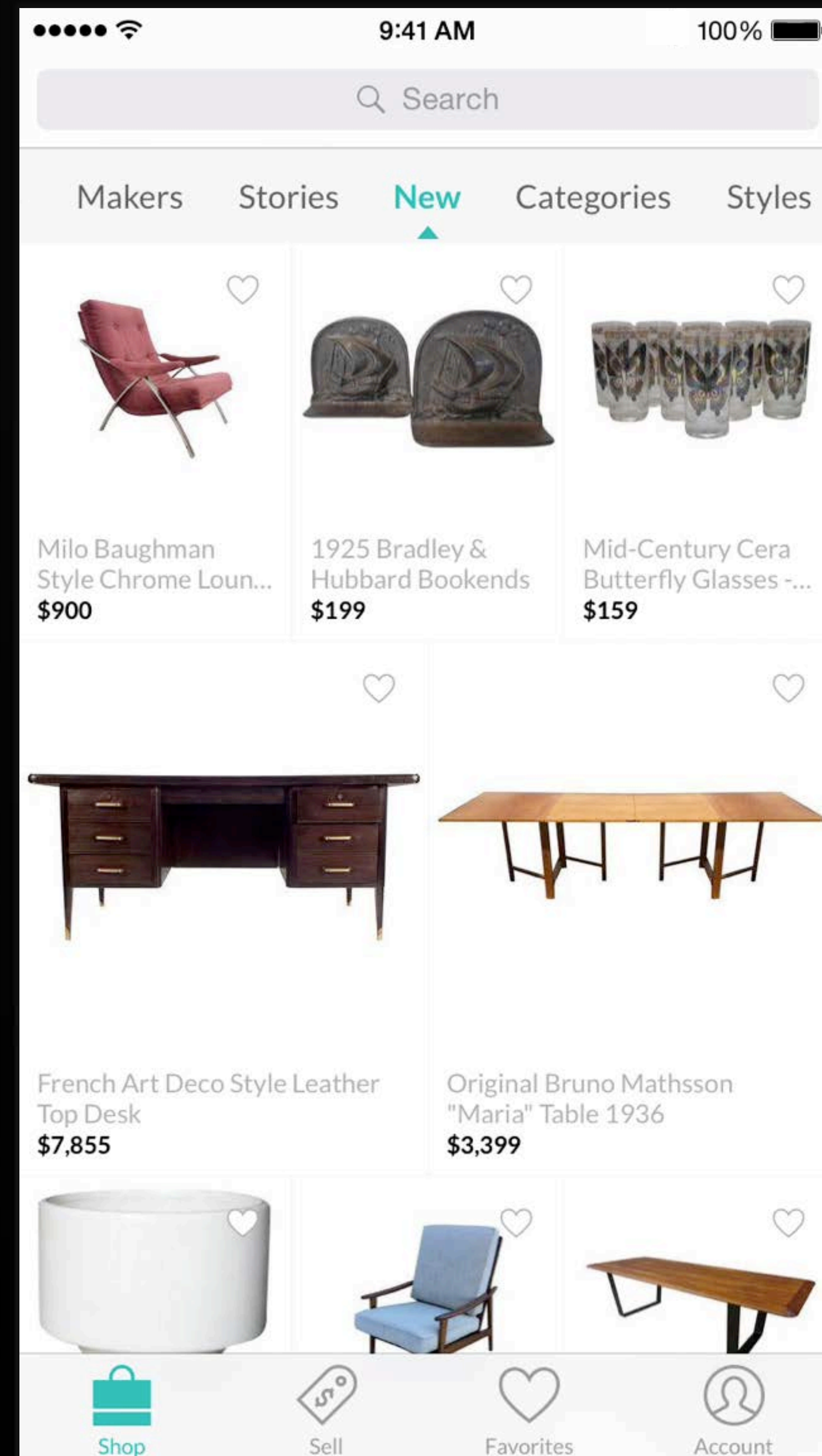# Make Shopping Easy

Account setup

Lengthy forms

Outdated billing and shipping information

Integrating Apple Pay

Integrating Apple Pay
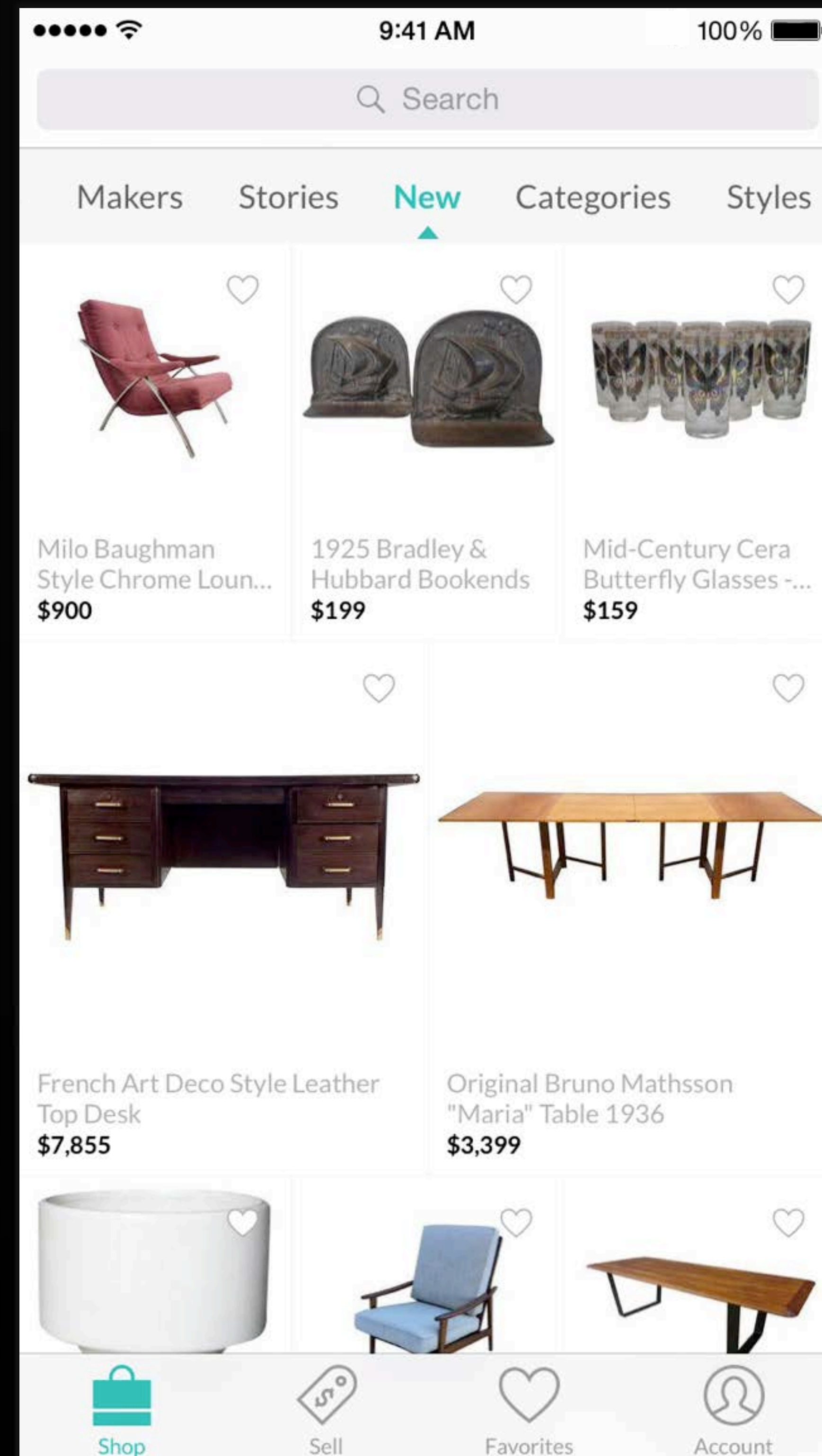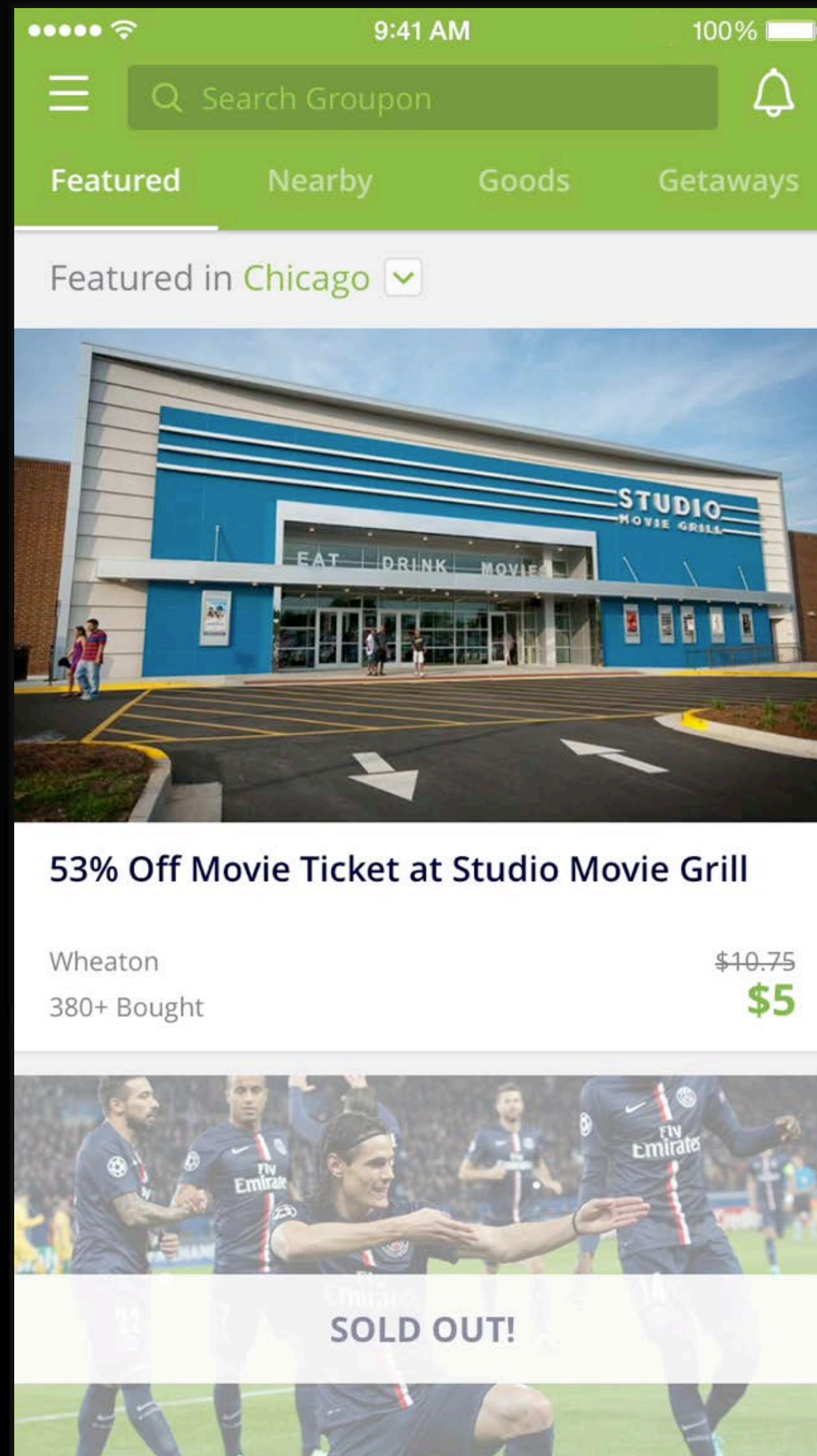
Enable quick checkout

Chairish

Chairish

Groupon

Groupon

Search Groupon

Featured   Nearby   Goods   Getaways

Featured in Chicago

53% Off Movie Ticket at Studio Movie Grill

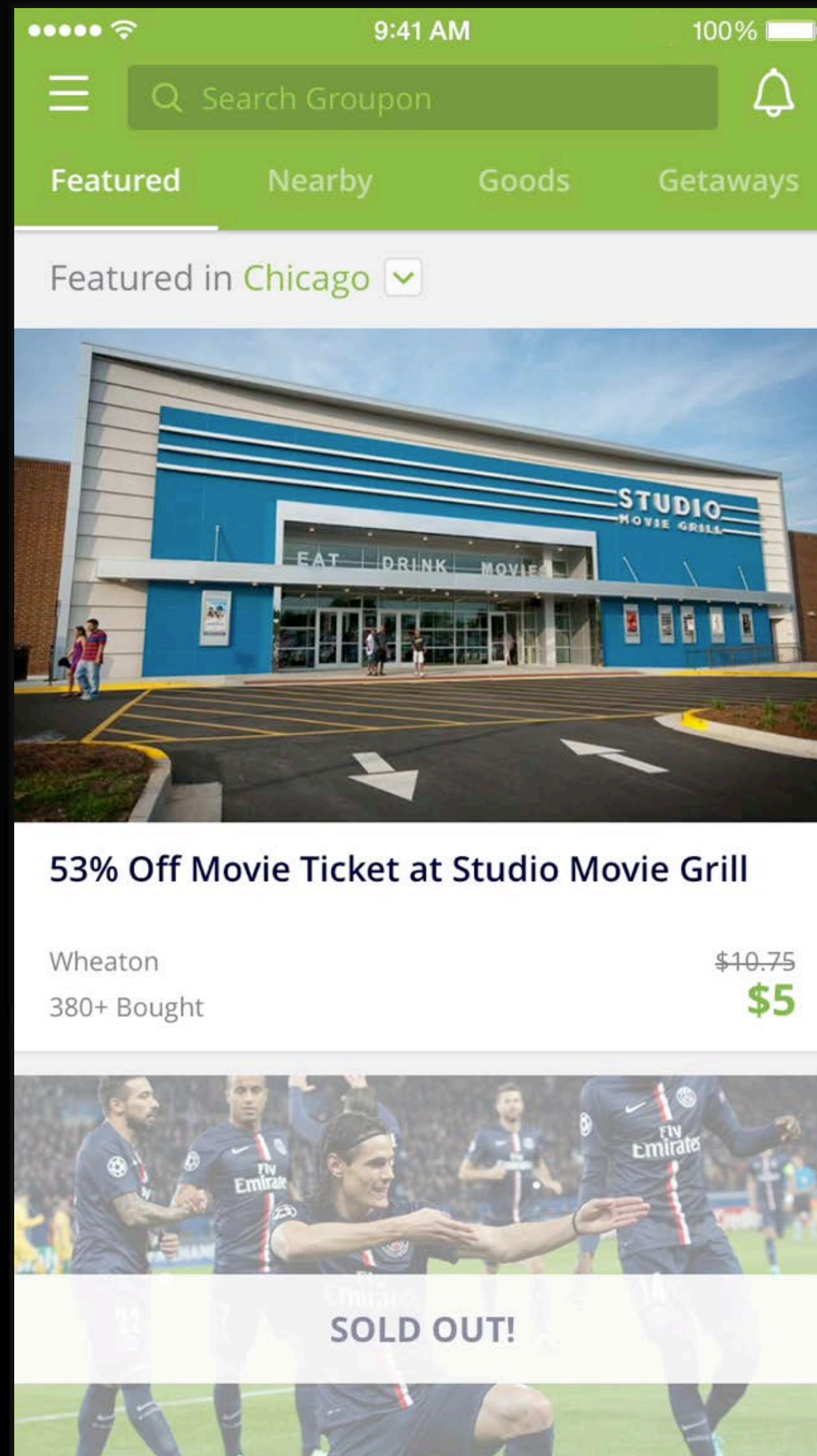Wheaton $10.75
380+ Bought   **$5**

SOLD OUT!

Integrating Apple Pay

Enable quick checkout

Integrating Apple Pay

Enable quick checkout

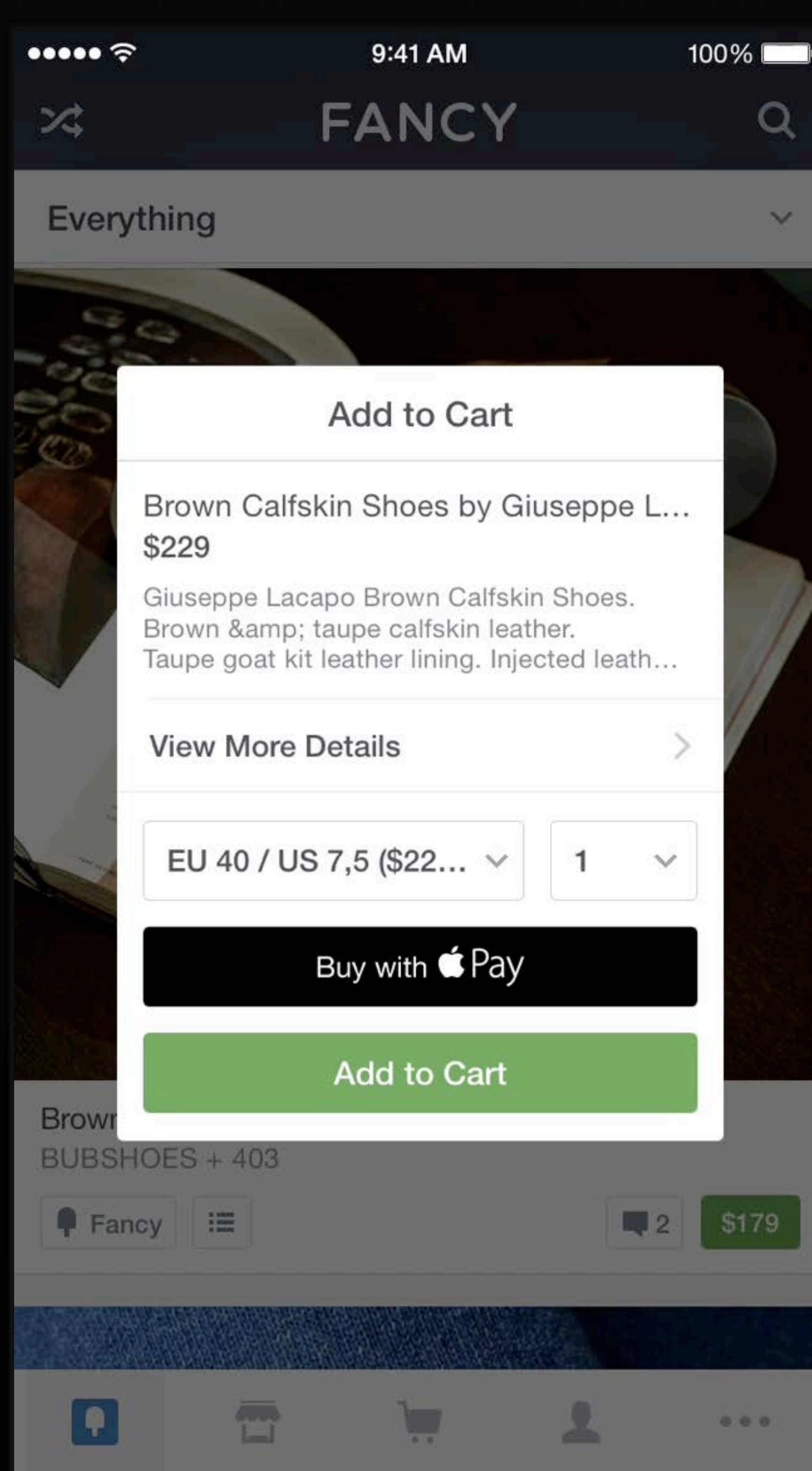Default to Apple Pay for eligible customers

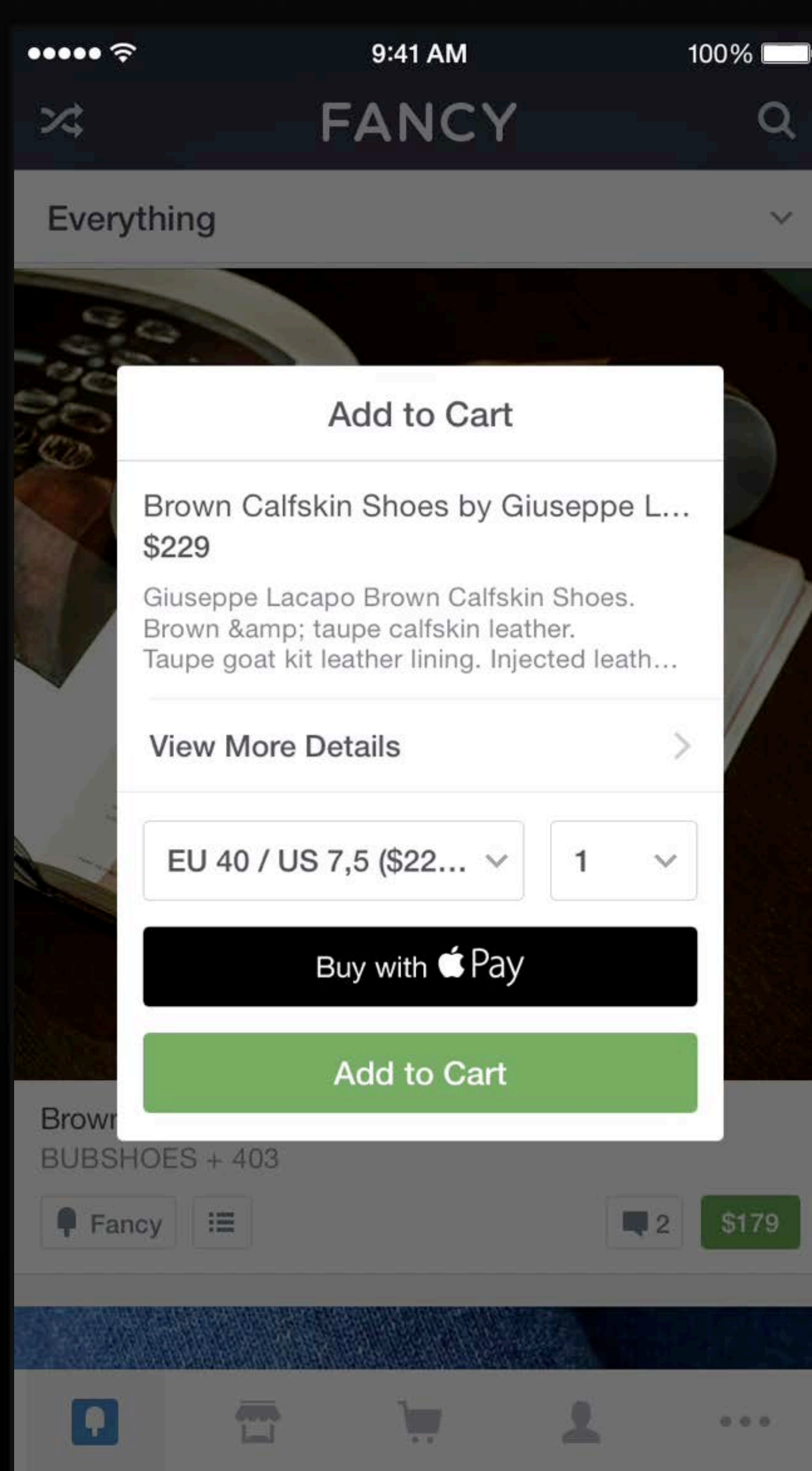# Integrating Apple Pay

Enable quick checkout

Default to Apple Pay for eligible customers
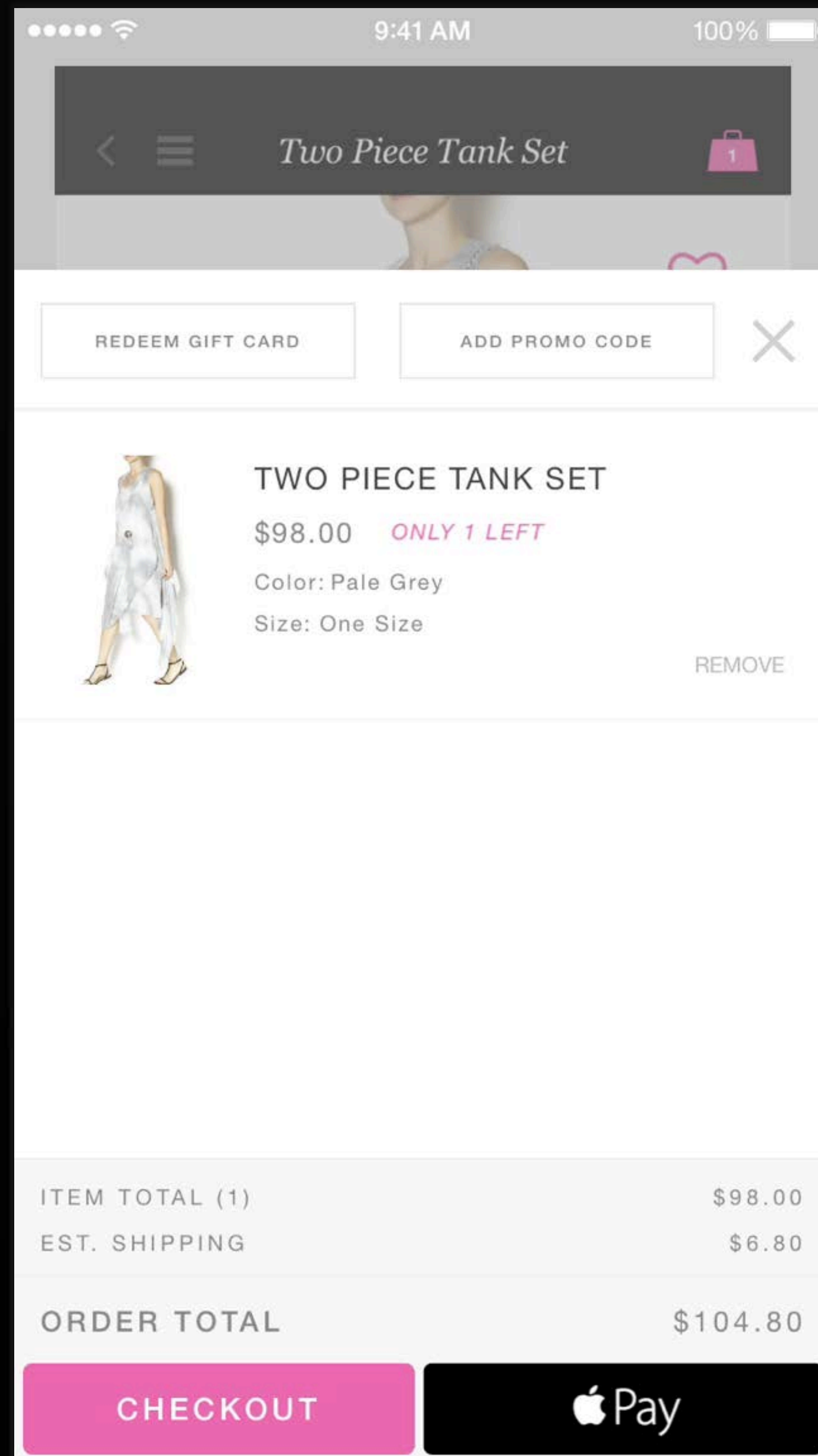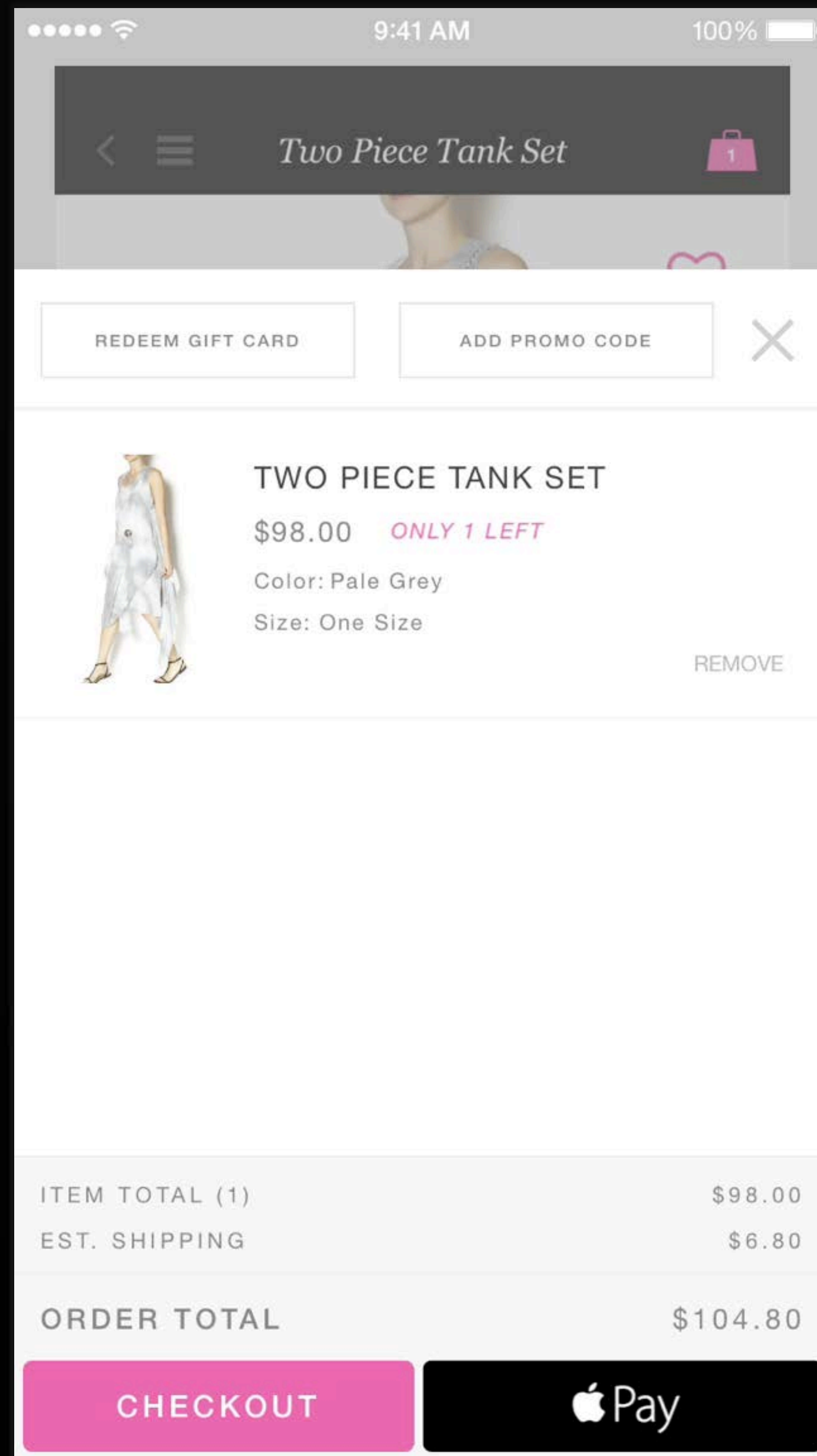
Display Apple Pay buttons prominently

Fancy

Fancy

9:41 AM   100% ▯

FANCY   🔍

Everything   ⌄

Add to Cart

Brown Calfskin Shoes by Giuseppe L...
$229

Giuseppe Lacapo Brown Calfskin Shoes.
Brown & taupe calfskin leather.
Taupe goat kit leather lining. Injected leath...

View More Details   ›

EU 40 / US 7,5 ($22... ⌄   1   ⌄

Buy with Pay

Add to Cart

Brown
BUBSHOES + 403

📍 Fancy   ☰   💬 2 $179

Shoptiques

Two Piece Tank Set

REDEEM GIFT CARD   ADD PROMO CODE   ✕

TWO PIECE TANK SET

$98.00   *ONLY 1 LEFT*

Color: Pale Grey

Size: One Size

REMOVE

ITEM TOTAL (1) $98.00
EST. SHIPPING $6.80

ORDER TOTAL $104.80

CHECKOUT   Pay

Shoptiques

Integrating Apple Pay

Enable quick checkout

Default to Apple Pay for eligible customers

Display Apple Pay buttons prominently

Integrating Apple Pay

Enable quick checkout

Default to Apple Pay for eligible customers

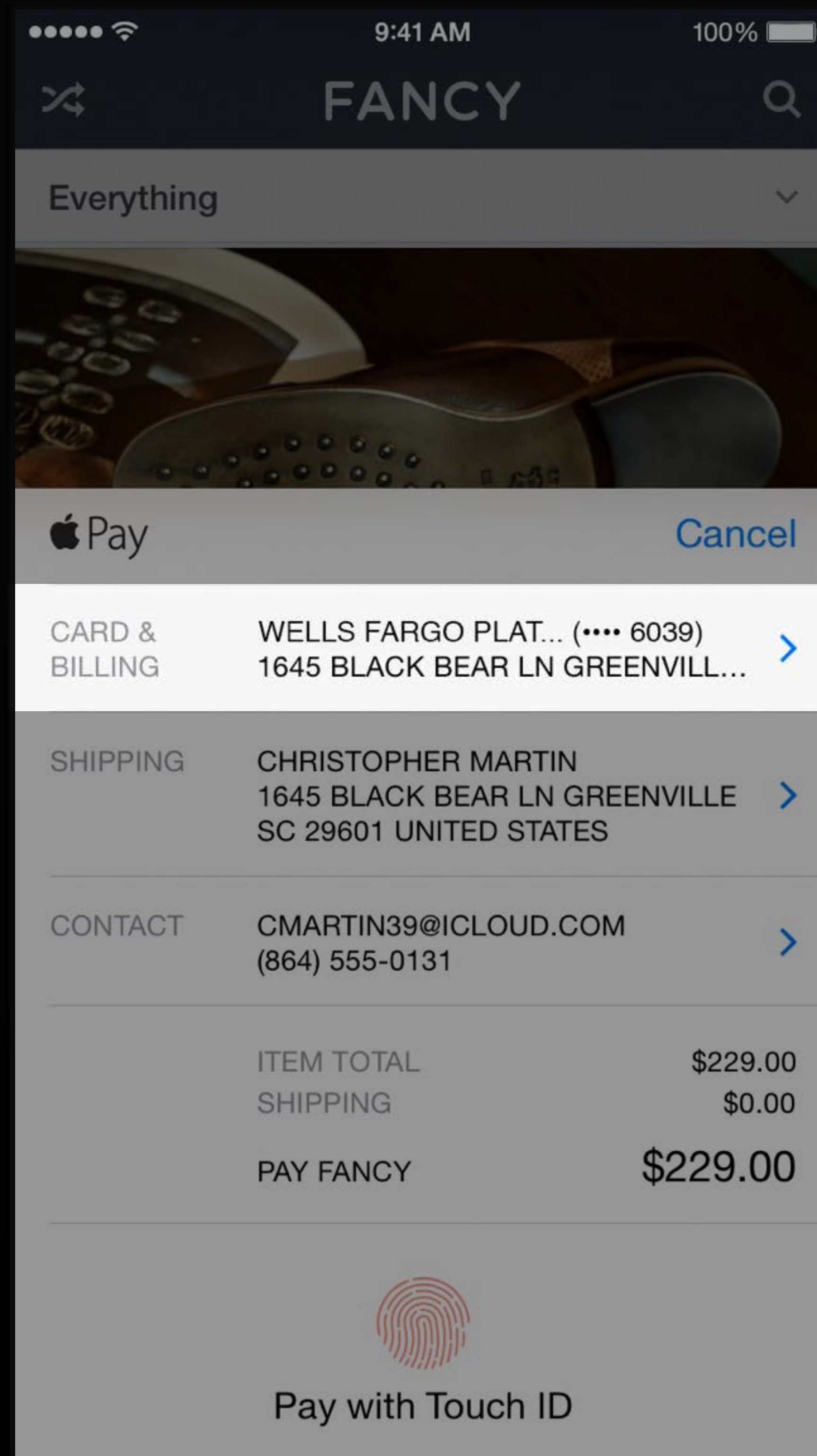Display Apple Pay buttons prominently

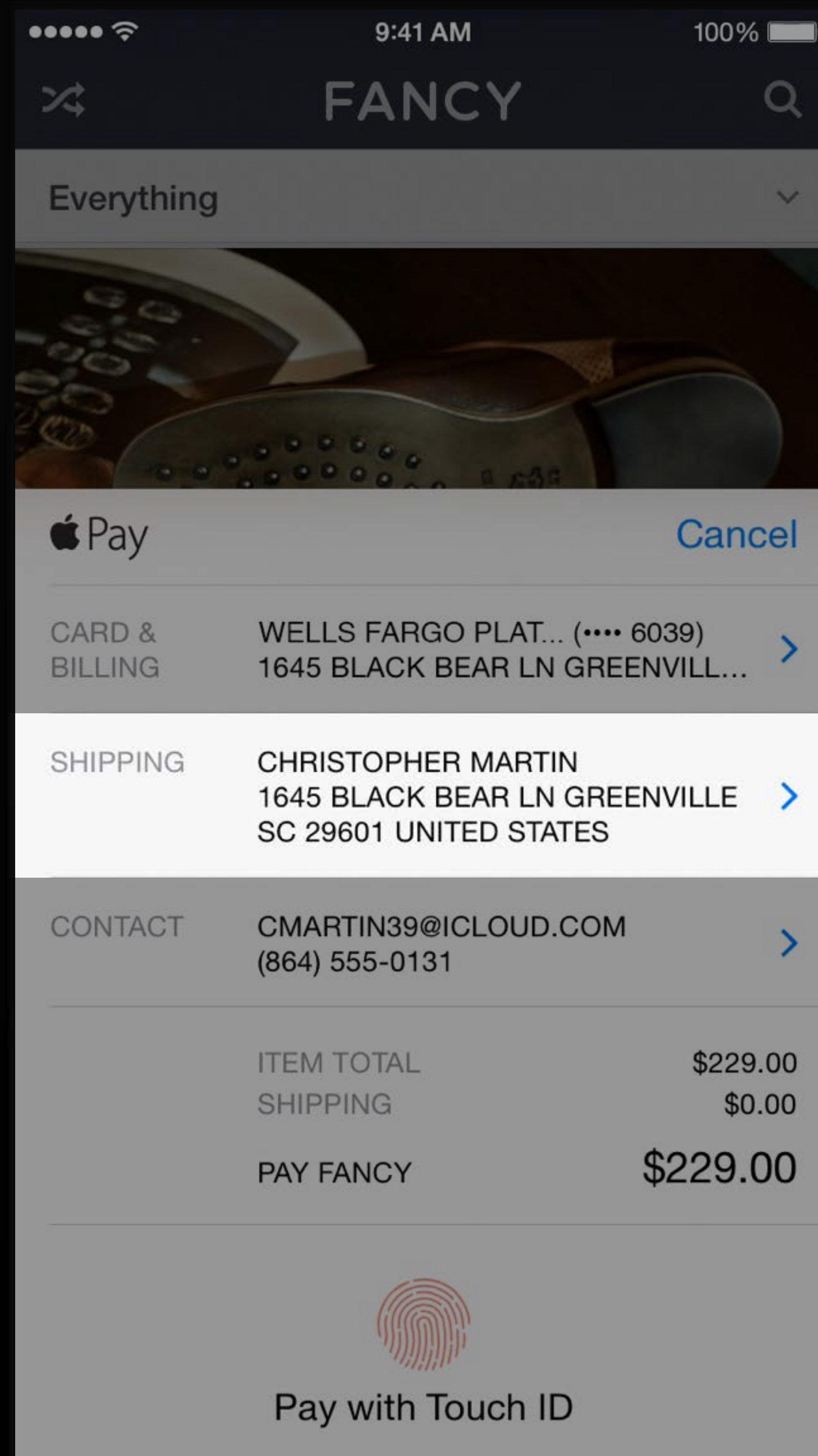Show the payment sheet immediately

Customizing the Payment Sheet

Customizing the Payment Sheet
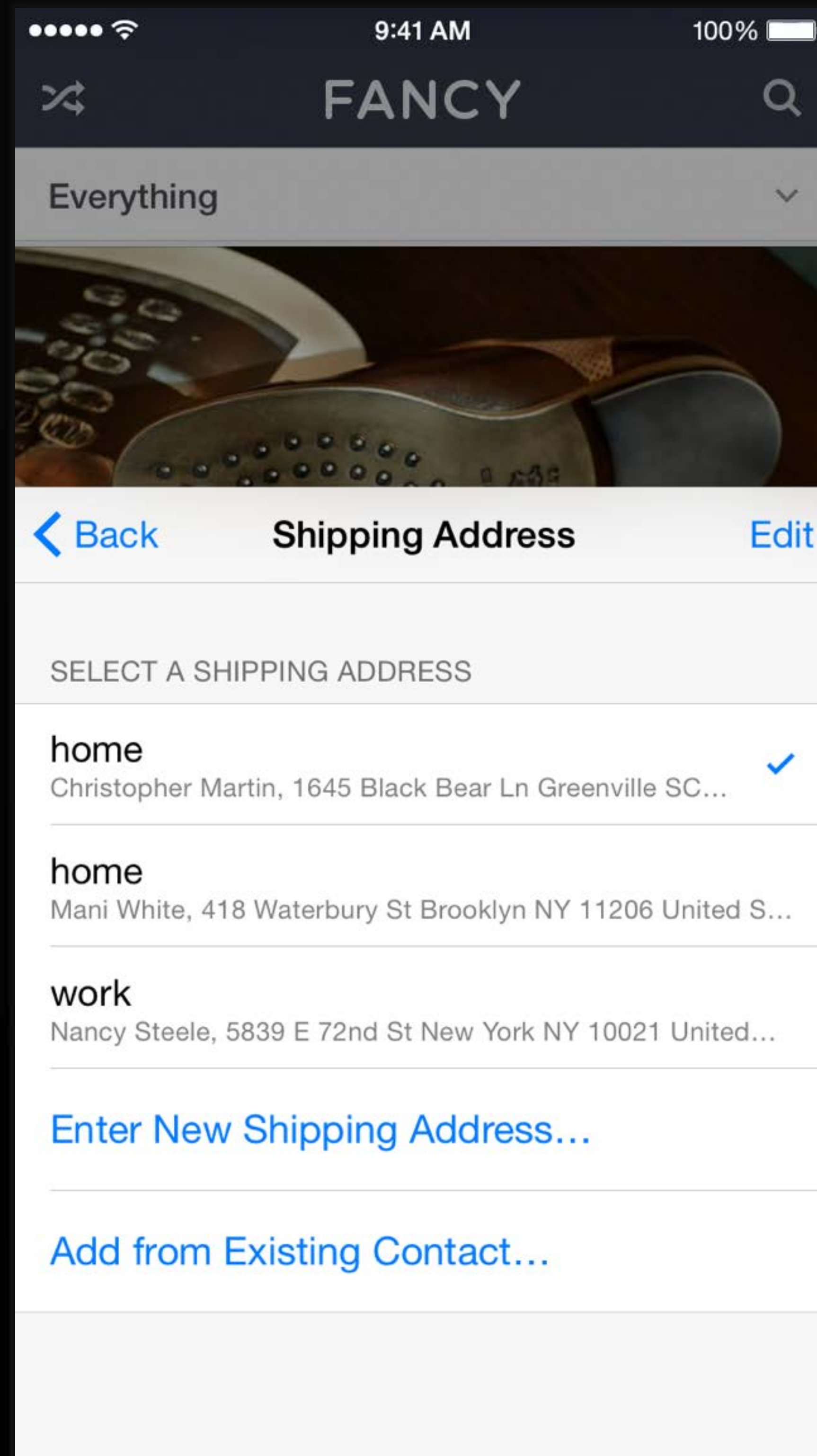
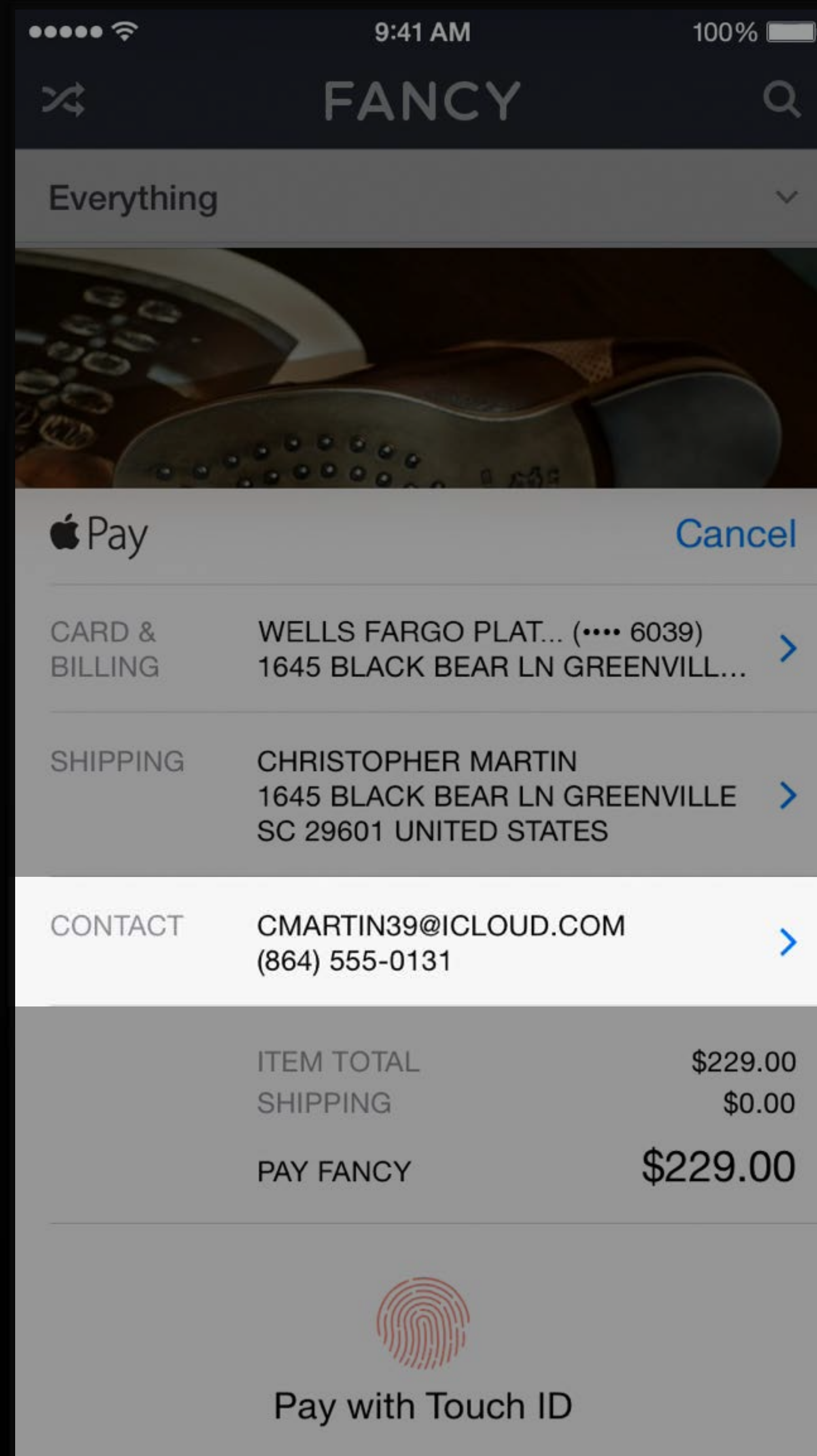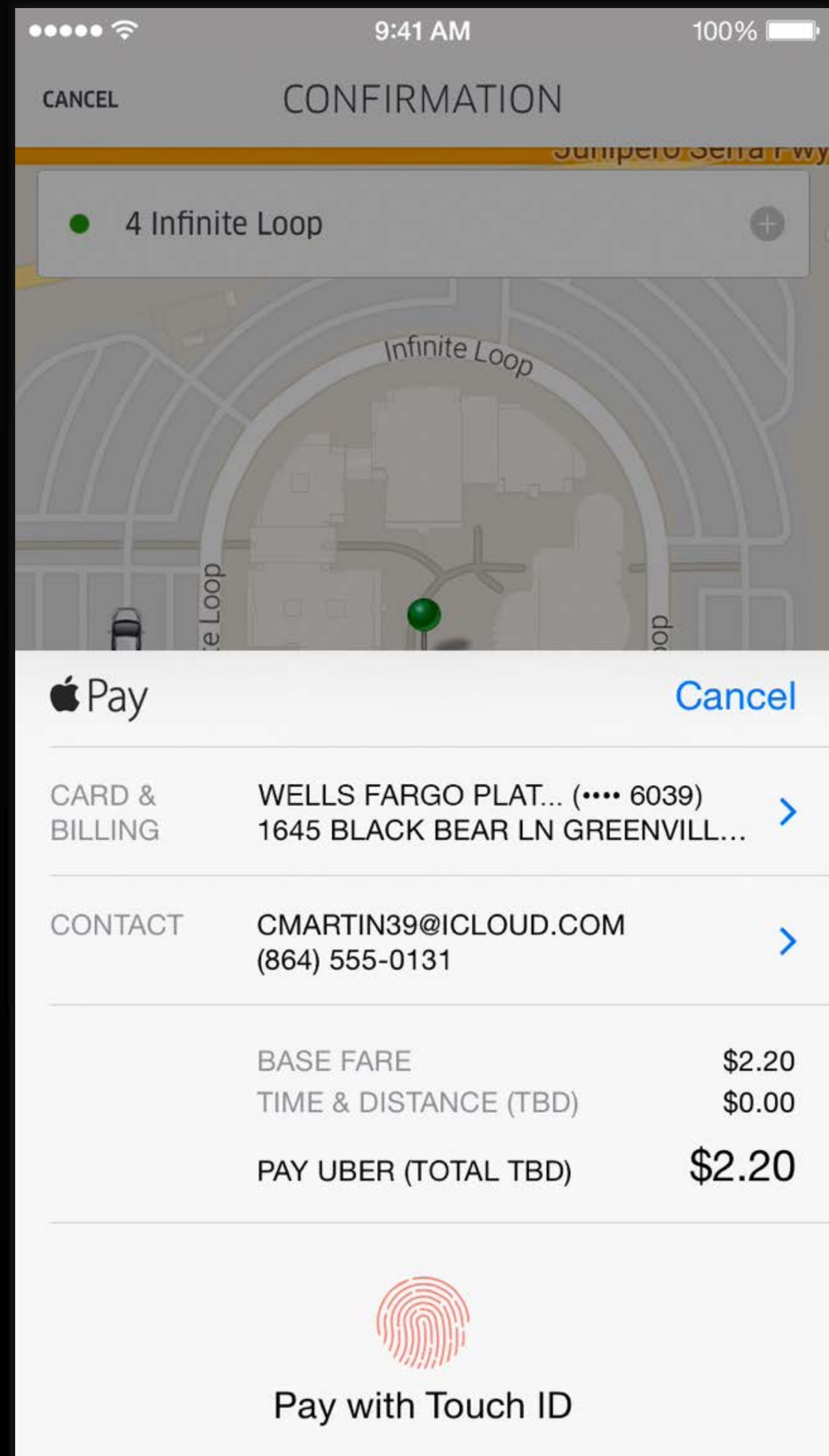Request billing, shipping & contact info if needed

Fancy

FANCY

Everything

Apple Pay   Cancel

CARD &   WELLS FARGO PLAT... (•••• 6039)
BILLING   1645 BLACK BEAR LN GREENVILL...

SHIPPING   CHRISTOPHER MARTIN
                1645 BLACK BEAR LN GREENVILLE
                SC 29601 UNITED STATES

CONTACT   CMARTIN39@ICLOUD.COM
                (864) 555-0131

                ITEM TOTAL $229.00
                SHIPPING $0.00

                PAY FANCY $229.00

                Pay with Touch ID

Fancy

Fancy

Fancy

Uber

CANCEL   CONFIRMATION

Juniperu Serra Pwy

● 4 Infinite Loop   ⊕

Infinite Loop

 Pay   Cancel

CARD &   WELLS FARGO PLAT... (•••• 6039)   ›
BILLING   1645 BLACK BEAR LN GREENVILL...

CONTACT   CMARTIN39@ICLOUD.COM   ›
            (864) 555-0131

            BASE FARE $2.20
            TIME & DISTANCE (TBD) $0.00

            PAY UBER (TOTAL TBD) $2.20

Pay with Touch ID

Customizing the Payment Sheet

Request billing, shipping, and contact info if needed

Customizing the Payment Sheet

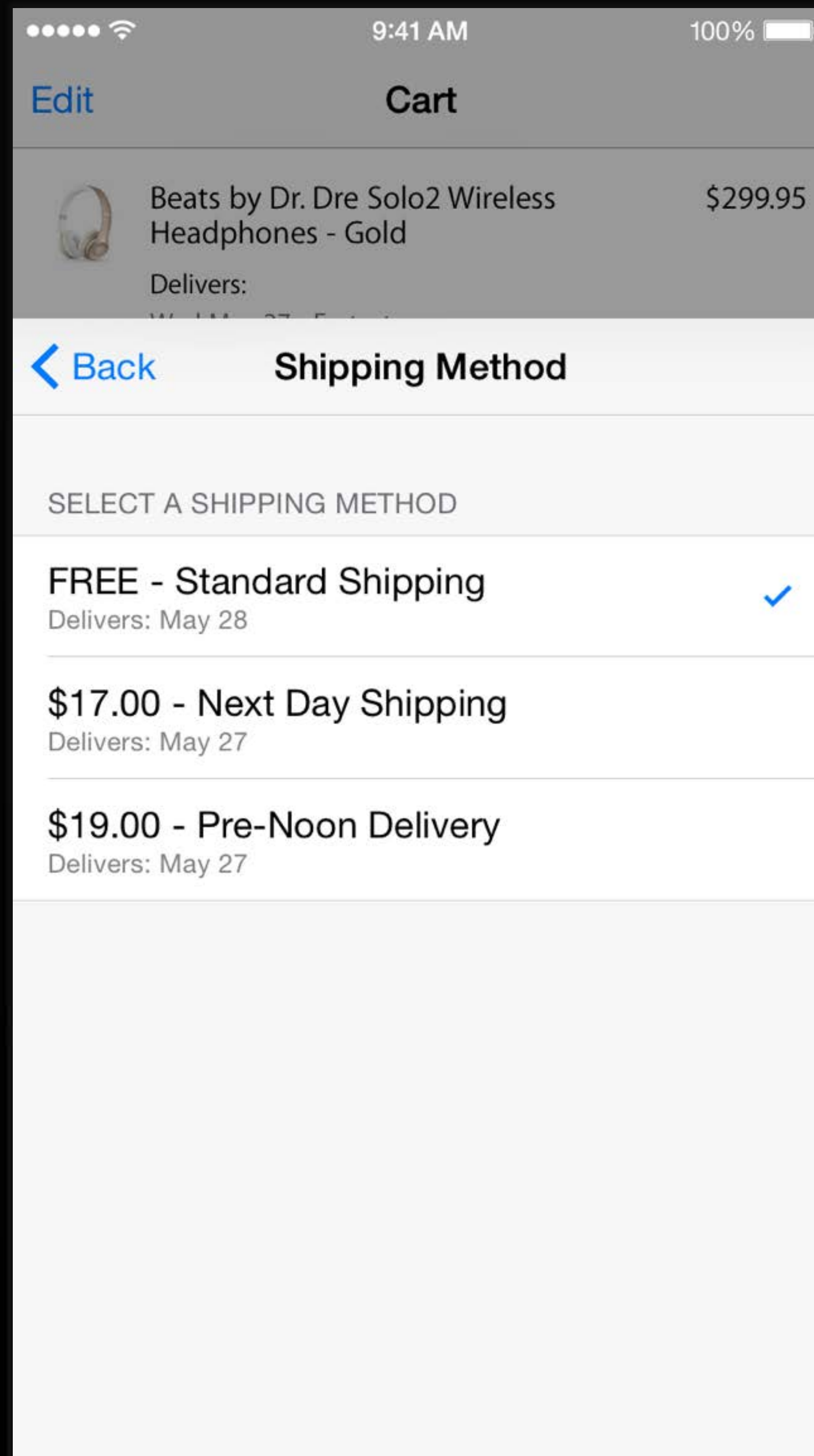Request billing, shipping, and contact info if needed

Specify shipping method or pickup

Apple Store

Apple Store

Customizing the Payment Sheet

Request billing, shipping & contact info if needed

Specify shipping method or pickup

Customizing the Payment Sheet

Request billing, shipping & contact info if needed

Specify shipping method or pickup

List shipping cost, tax, or discounts after the subtotal
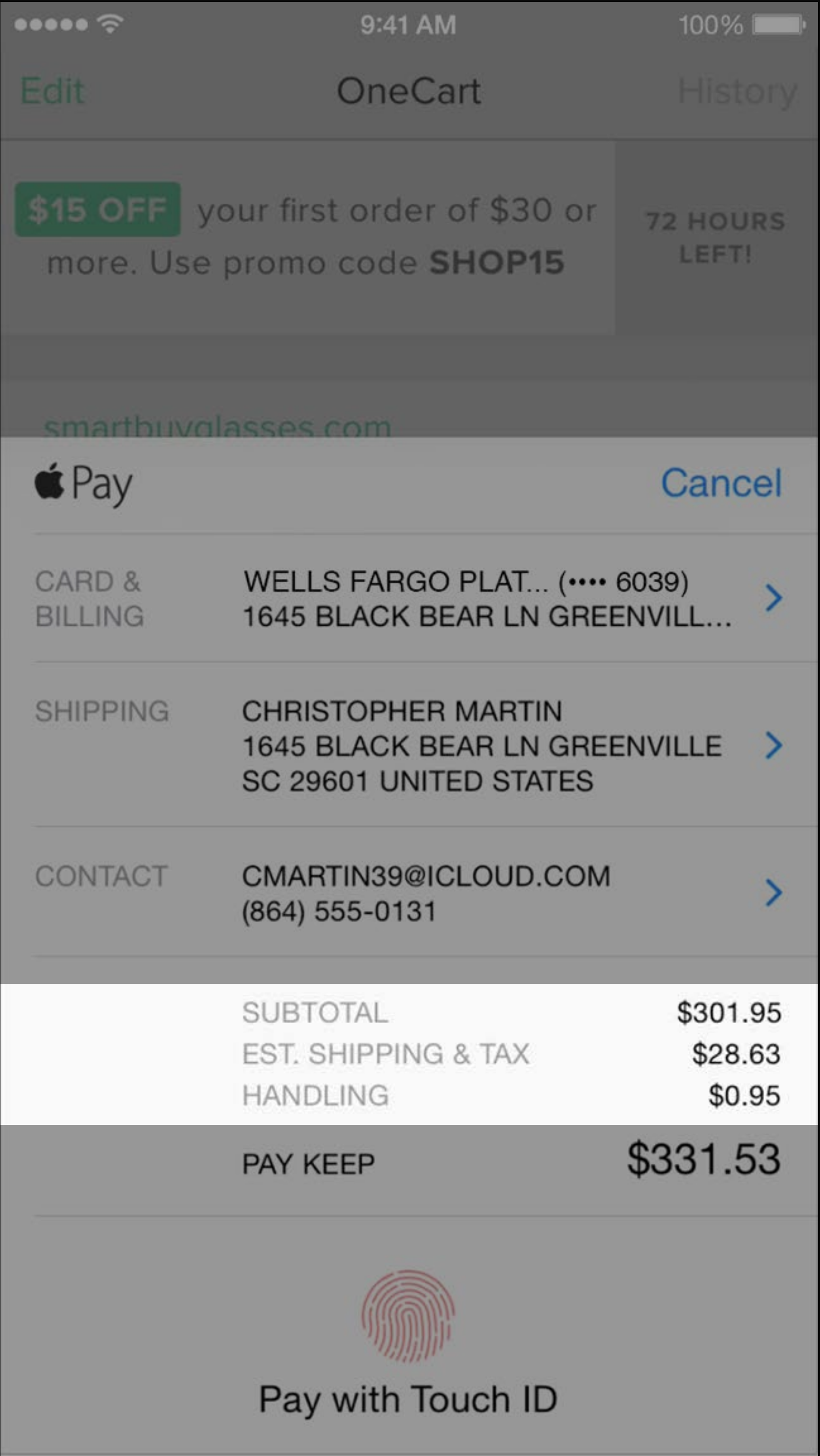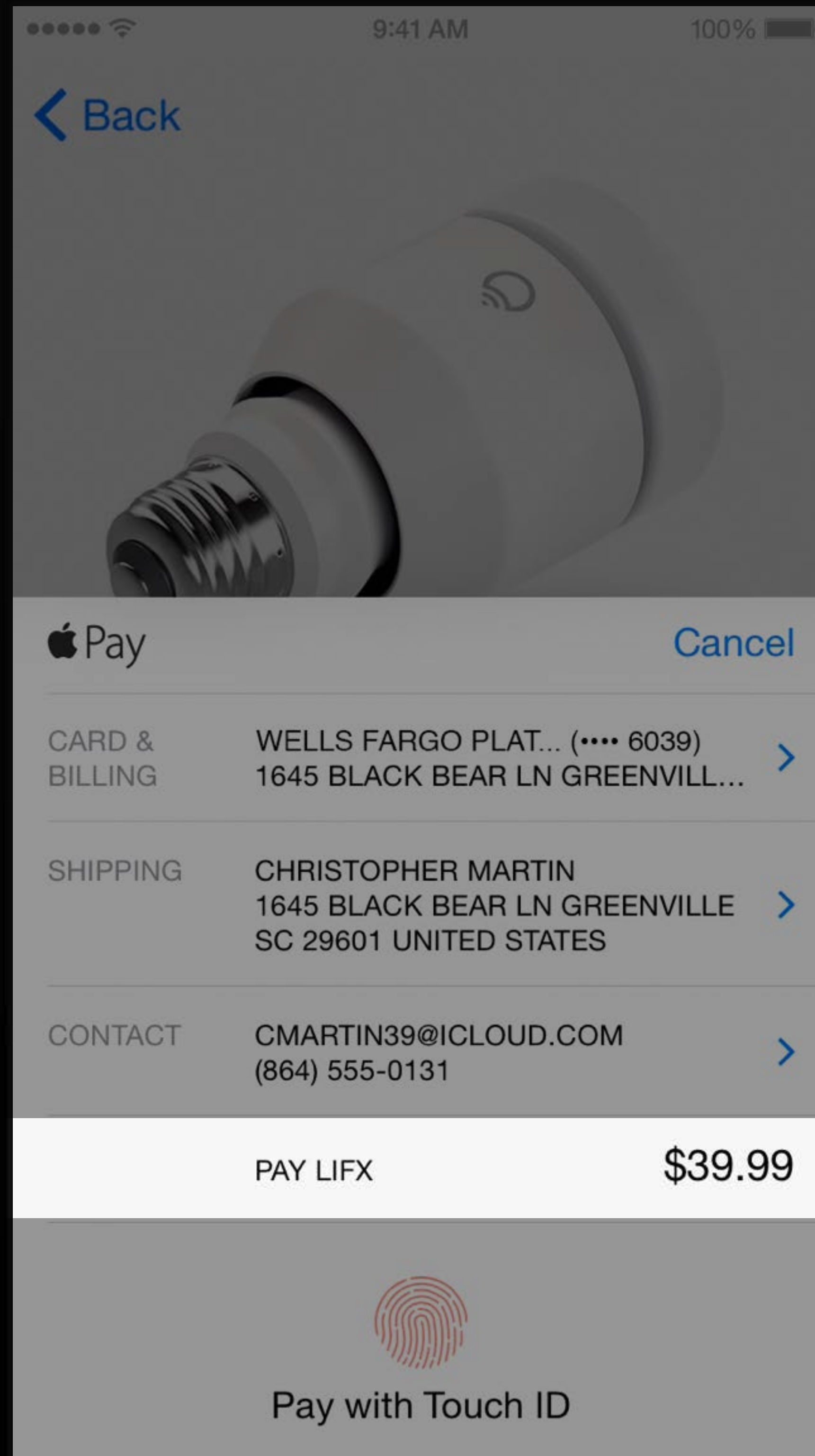
Keep

LIFX

Customizing the Payment Sheet

Request billing, shipping, and contact info if needed

Specify shipping method or pickup

List shipping cost, tax, or discounts after the subtotal

Customizing the Payment Sheet

Request billing, shipping, and contact info if needed

Specify shipping method or pickup

List shipping cost, tax, or discounts after the subtotal

Make estimates clear

Customizing the Payment Sheet

Request billing, shipping, and contact info if needed

Specify shipping method or pickup

List shipping cost, tax, or discounts after the subtotal

Make estimates clear
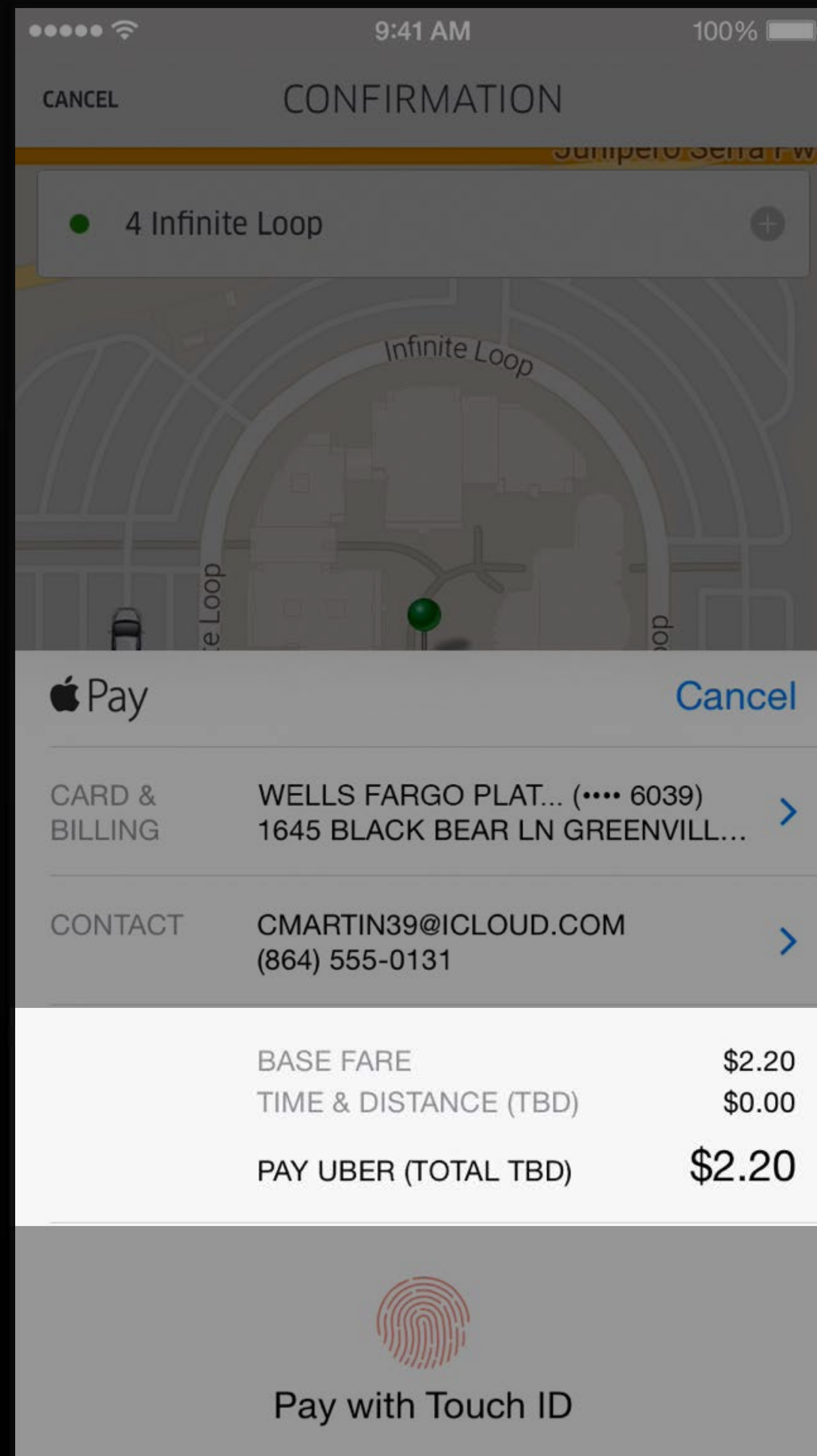
Customizing the Payment Sheet

Request billing, shipping, and contact info if needed

Specify shipping method or pickup

List shipping cost, tax, or discounts after the subtotal

Make estimates clear

Place the business name before the total cost

Fancy

Customizing the Payment Sheet

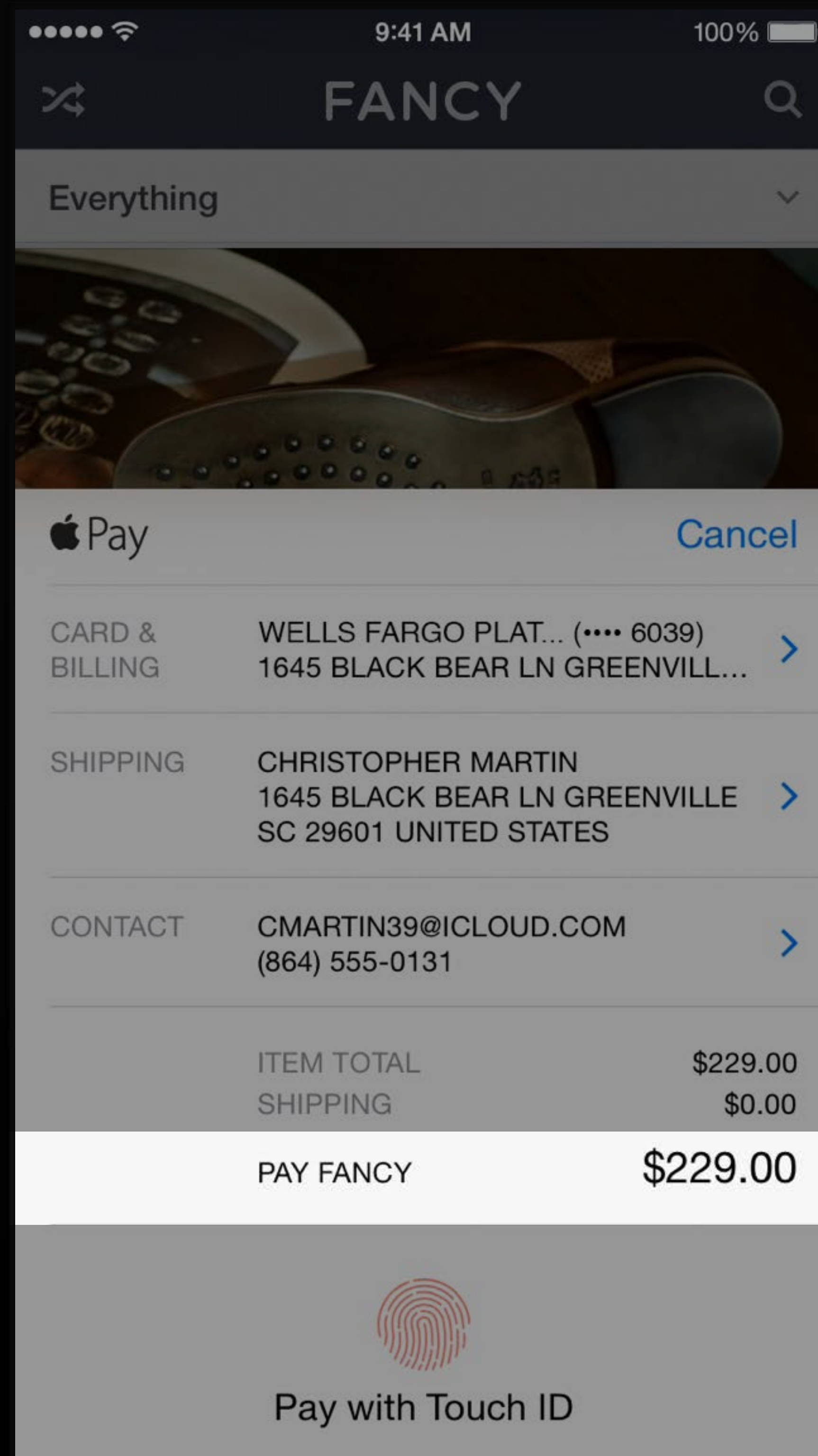Request billing, shipping, and contact info if needed

Specify shipping method or pickup

List shipping cost, tax, or discounts after the subtotal

Make estimates clear

Place the business name before the total cost

Apple Store

Apple Store

Summary

Summary

Remove obstacles to making a purchase

Summary

Remove obstacles to making a purchase

Display the Apple Pay buttons prominently

Summary

Remove obstacles to making a purchase

Display the Apple Pay buttons prominently

Customize the payment sheet

Summary

Remove obstacles to making a purchase

Display the Apple Pay buttons prominently

Customize the payment sheet

Confirm the transaction

Putting It All Together

My First Apple Pay App

Request a payment

Display the payment sheet

Handle the authorization

Let's Talk Classes

Let's Talk Classes

`PKPaymentSummaryItem`

Let's Talk Classes

```
PKPaymentSummaryItem
          |
          |
          ↓
   PKPaymentRequest
```

Let's Talk Classes

PKPaymentSummaryItem

↓

PKPaymentRequest

↓

PKPaymentAuthorizationViewController

Let's Talk Classes

PKPaymentSummaryItem

↓

PKPaymentRequest

↓

PKPaymentAuthorizationViewController

↓
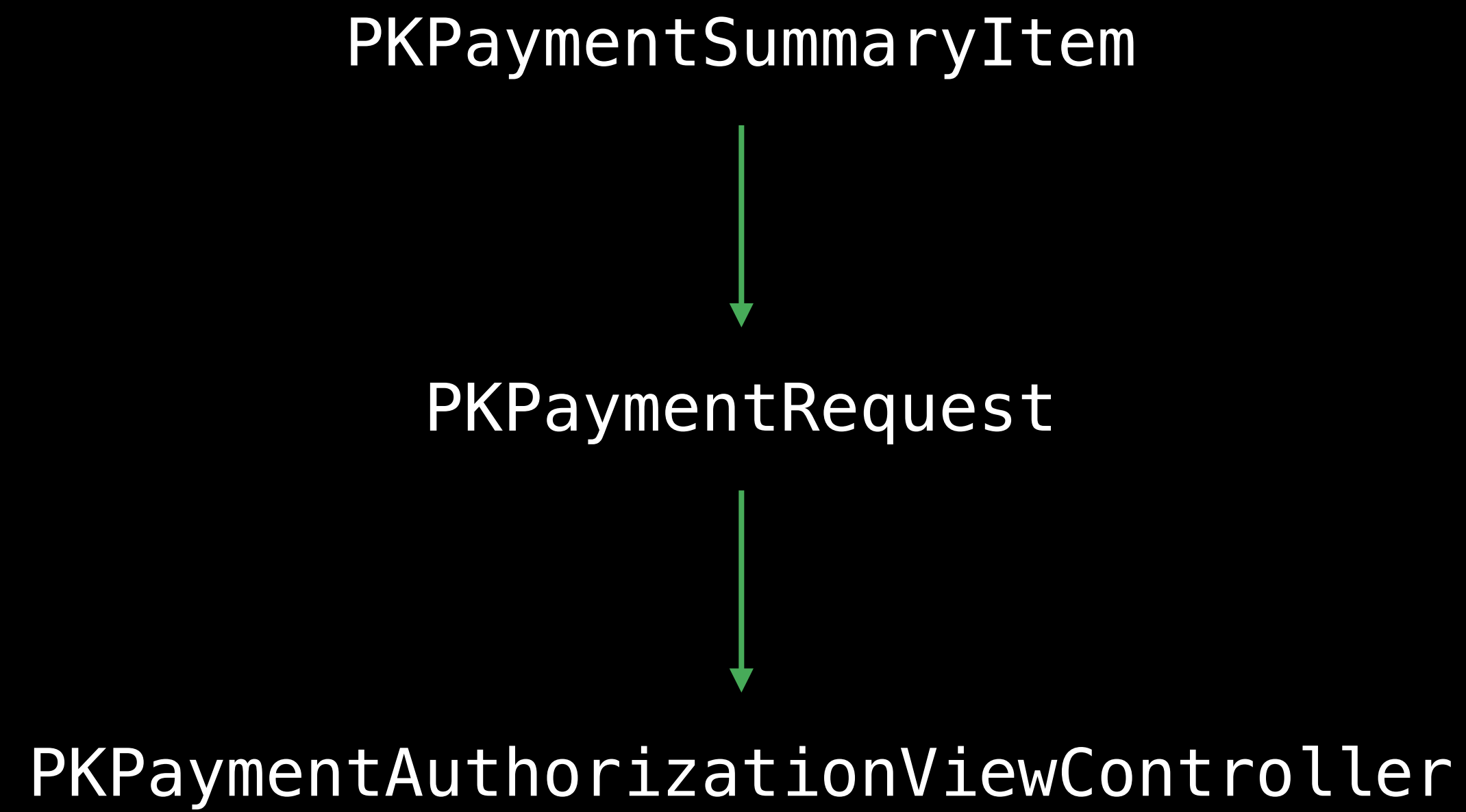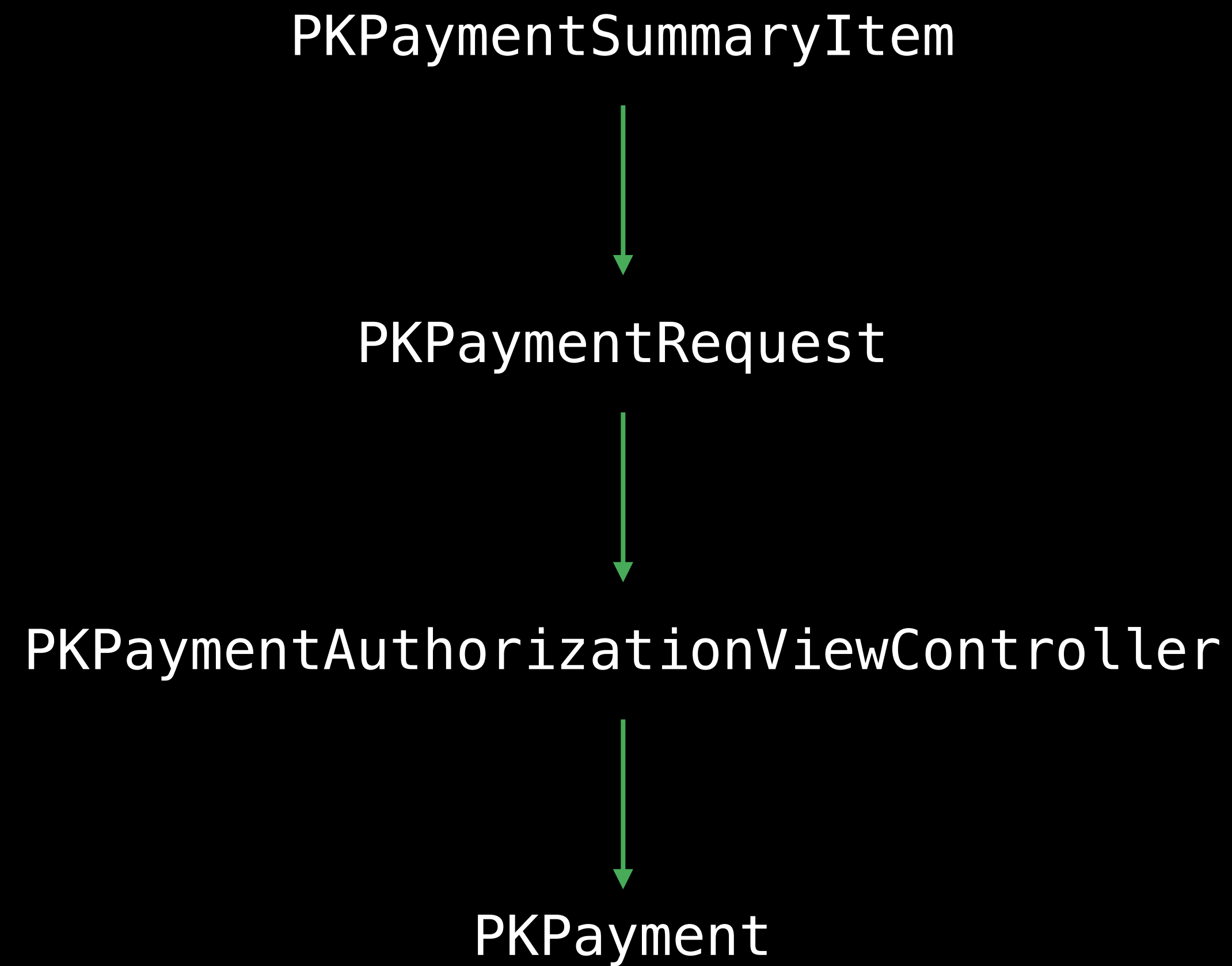
PKPayment

First Steps
Checking for Apple Pay

First Steps
Checking for Apple Pay

Does the user have payment cards available that we can accept?

```swift
let paymentNetworks = [PKPaymentNetworkMasterCard, PKPaymentNetworkVisa]

if PKPaymentAuthorizationViewController.canMakePaymentsUsingNetworks(paymentNetworks)
{
    // Create the payment request
} else {
    // Traditional checkout flow
}
```

First Steps
Checking for Apple Pay

Does the user have payment cards available that we can accept?

```
let paymentNetworks = [PKPaymentNetworkMasterCard, PKPaymentNetworkVisa]
```

```
if PKPaymentAuthorizationViewController.canMakePaymentsUsingNetworks(paymentNetworks)
{
    // Create the payment request
} else {
    // Traditional checkout flow
}
```

First Steps
Checking for Apple Pay

NEW

Check if the hardware supports Apple Pay:

```
PKPaymentAuthorizationViewController.canMakePayments()
```

New in iOS 9 - check for capabilities:

```
PKPaymentAuthorizationViewController.canMakePaymentsUsingNetworks(networks,
                                capabilities: .CapabilityDebit)
```

Creating a Payment
PKPaymentRequest

Summarize the charges you'd like to make and how to pay for them

Request shipping and contact information

```
let request = PKPaymentRequest()

request.merchantIdentifier    = "merchant.com.wwdc.emporium"
request.countryCode           = "US"
request.currencyCode          = "USD"
request.supportedNetworks     = [PKPaymentNetworkAmex, PKPaymentNetworkVisa]
request.merchantCapabilities = .Capability3DS
request.paymentSummaryItems  = […]
```

Creating a Payment
PKPaymentRequest

Summarize the charges you'd like to make and how to pay for them

Request shipping and contact information

```
let request = PKPaymentRequest()

request.merchantIdentifier   = "merchant.com.wwdc.emporium"
request.countryCode          = "US"
request.currencyCode         = "USD"
request.supportedNetworks    = [PKPaymentNetworkAmex, PKPaymentNetworkVisa]
request.merchantCapabilities = .Capability3DS
request.paymentSummaryItems  = […]
```

Creating a Payment
PKPaymentRequest

Summarize the charges you'd like to make and how to pay for them

Request shipping and contact information

```
let request = PKPaymentRequest()

request.merchantIdentifier    = "merchant.com.wwdc.emporium"
request.countryCode           = "US"
request.currencyCode          = "USD"
request.supportedNetworks     = [PKPaymentNetworkAmex, PKPaymentNetworkVisa]
request.merchantCapabilities  = .Capability3DS
request.paymentSummaryItems   = […]
```

Creating a Payment
PKPaymentRequest

Summarize the charges you'd like to make and how to pay for them

Request shipping and contact information

```
let request = PKPaymentRequest()

request.merchantIdentifier   = "merchant.com.wwdc.emporium"
request.countryCode          = "US"
request.currencyCode         = "USD"
request.supportedNetworks    = [PKPaymentNetworkAmex, PKPaymentNetworkVisa]
request.merchantCapabilities = .Capability3DS
request.paymentSummaryItems  = […]
```

Creating a Payment
PKPaymentRequest

Summarize the charges you'd like to make and how to pay for them

Request shipping and contact information

```
let request = PKPaymentRequest()

request.merchantIdentifier    = "merchant.com.wwdc.emporium"
request.countryCode           = "US"
request.currencyCode          = "USD"
request.supportedNetworks     = [PKPaymentNetworkAmex, PKPaymentNetworkVisa]
request.merchantCapabilities = .Capability3DS
request.paymentSummaryItems   = […]
```

Creating a Payment
PKPaymentRequest

Summarize the charges you'd like to make and how to pay for them

Request shipping and contact information

```
let request = PKPaymentRequest()

request.merchantIdentifier   = "merchant.com.wwdc.emporium"
request.countryCode          = "US"
request.currencyCode         = "USD"
request.supportedNetworks    = [PKPaymentNetworkAmex, PKPaymentNetworkVisa]
request.merchantCapabilities = .Capability3DS
request.paymentSummaryItems  = […]
```

Creating a Payment
PKPaymentRequest

Summarize the charges you'd like to make and how to pay for them

Request shipping and contact information

```swift
let request = PKPaymentRequest()

request.merchantIdentifier   = "merchant.com.wwdc.emporium"
request.countryCode          = "US"
request.currencyCode         = "USD"
request.supportedNetworks    = [PKPaymentNetworkAmex, PKPaymentNetworkVisa]
request.merchantCapabilities = .Capability3DS
request.paymentSummaryItems  = […]
```

Creating a Payment
PKPaymentRequest

Summarize the charges you'd like to make and how to pay for them

Request shipping and contact information

```
let request = PKPaymentRequest()

request.merchantIdentifier    = "merchant.com.wwdc.emporium"
request.countryCode           = "US"
request.currencyCode          = "USD"
request.supportedNetworks     = [PKPaymentNetworkAmex, PKPaymentNetworkVisa]
request.merchantCapabilities = .Capability3DS
request.paymentSummaryItems  = [...]
```

Creating a Payment

PKPaymentRequest

NEW

If necessary, only allow certain types of cards to make payments

```
// Only allow users to make payments with a debit card
request.merchantCapabilities = .Capability3DS | .CapabilityDebit
```

Creating a Payment
PKPaymentSummaryItem

Describes what you're paying for and how much it costs

Uses NSDecimalNumber for guaranteed precision

```
NSDecimalNumber
NSDecimalNumber(string)
NSDecimalNumber(decimal)
NSDecimalNumber(double)
```

Creating a Payment
PKPaymentSummaryItem

Describes what you're paying for and how much it costs

Uses NSDecimalNumber for guaranteed precision

```
NSDecimalNumber
NSDecimalNumber(string)
NSDecimalNumber(decimal)
NSDecimalNumber(double)
```

Creating a Payment
PKPaymentSummaryItem

```
let total = PKPaymentSummaryItem(label: "Apple Inc",
                                 amount: NSDecimalNumber(string: "349.99"))

request.paymentSummaryItems = [total]
```

The last item of the summary items array is the total to be charged

Creating a Payment
PKPaymentAuthorizationViewController

Displays the payment information

Modally displayed over your app

On iPad will be a form sheet
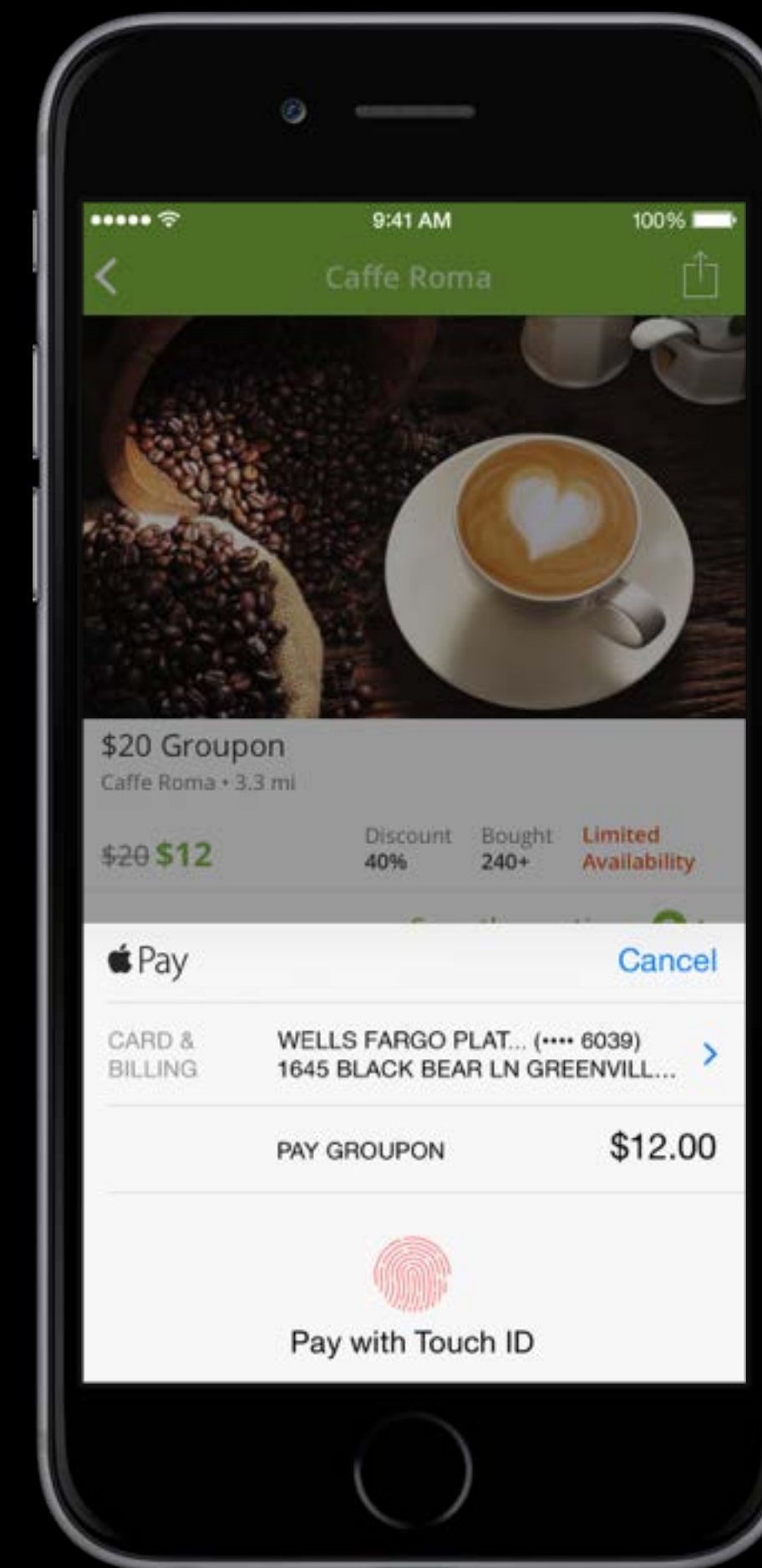
Creating a Payment
PKPaymentAuthorizationViewController

```swift
let vc = PKPaymentAuthorizationViewController(paymentRequest: request)

vc.delegate = self
presentViewController(vc, animated: true, completion: nil)
```

Creating a Payment
PKPaymentButton

Use PKPaymentButton to trigger the sheet

Available in a variety of styles and colors

Fully localized

Available in iOS 8.3

Demo
Making a payment

Accepting a Payment

PKPaymentAuthorizationViewControllerDelegate

Receive callbacks when the payment authorization process has completed

Use the delegate to confirm whether a payment has been made or not

Accepting a Payment
PKPaymentAuthorizationViewControllerDelegate

Send the payment data for processing and return the status

```
func paymentAuthorizationViewController(
        controller: PKPaymentAuthorizationViewController,
        didAuthorizePayment payment: PKPayment,
        completion: (PKPaymentAuthorizationStatus) -> Void)
```

Dismiss the payment view controller

```
func paymentAuthorizationViewControllerDidFinish(
        controller: PKPaymentAuthorizationViewController)
```

Accepting a Payment
PKPayment and PKPaymentToken

Returned by the payment view controller after successful authorization

Contains the encrypted payment data

Any other metadata you requested

Demo

Accepting a payment

Contact Information

Request contact information from users for shipping (email, phone, address)

```
paymentRequest.requiredShippingAddressFields = .PostalAddress | .Email
```

New in iOS 8.3—name only

```
paymentRequest.requiredShippingAddressFields = .Name
```

Contact Information

Optionally, request billing address as well

```
paymentRequest.requiredBillingAddressFields = .PostalAddress
```

Generally, don't request contact information unless you need it

Billing Address

Billing address is not required to process Apple Pay

We recommend against requesting it

Shipping Costs

Update shipping costs when the user selects a new address

For privacy, you'll receive city and ZIP only

Provide updated shippingMethods and summaryItems arrays

```
func paymentAuthorizationViewController(
    controller: PKPaymentAuthorizationViewController,
    didSelectShippingContact contact: CNContact,
    completion: (PKPaymentAuthorizationStatus, [PKShippingMethod],
                [PKPaymentSummaryItem]) -> Void)
```

Shipping Costs

Display shipping costs in-line in the Apple Pay sheet

PKShippingMethod (PKPaymentSummaryItem subclass)

```
let twoDay = PKShippingMethod(label: "Express shipping",
                              amount: NSDecimalNumber(string: "4.99"))
twoDay.detail = "Delivers in two working days"

paymentRequest.shippingMethods = [twoDay]
```

Contact Information

NEW

Un-redacted contact information is available after the user has authorized the payment

```swift
if let shippingContact = payment.shippingContact {
    let name = CNContactFormatter.stringFromContact(shippingContact,
                                      style: .FullName)
    let email = shippingContact.emailAddresses.first
}
```

iOS 8 AddressBook.framework based APIs are deprecated

Demo

Shipping and contacts

Payment Information

PKPaymentMethod

NEW

Find out more about the payment instrument selected

Apply debit / credit card surcharges or discounts

Receive delegate callback when the payment method changes

```swift
if paymentMethod.type == .Debit {
    label = "Thanks for paying with your debit card!"
}
```

Payment Information
PKPaymentMethod

NEW

A minority of older cards may have no known payment type

In this case, you'll receive PKPaymentMethodTypeUnknown

Tips and Tricks

NEW

Pending items

New property on PKPaymentSummaryItem

```
paymentSummaryItem.type = .Pending
```

Use this to indicate charges that aren't final

Additional documentation in a future seed

Tips and Tricks
Simulator support

Display the Apple Pay sheet in the simulator

Always test your apps on real hardware

Tips and Tricks

Apple Pay on Apple Watch

Trigger payments directly from Apple Watch

Handoff directly to an Apple Pay sheet

Sample code available

Summary

Apple Pay: Easy, private, secure

Try out Apple Pay for yourself

Integrate into your apps

Delight your users

More Information

Documentation

Apple Pay for Developers

https://developer.apple.com/apple-pay/

iOS Security White Paper

http://www.apple.com/business/docs/iOS_Security_Guide.pdf

More Information

Technical Support
Apple Developer Forums
http://developer.apple.com/forums

Developer Technical Support
http://developer.apple.com/support/technical

General Inquiries
Paul Danbold, Core OS Evangelist
danbold@apple.com

Rachel Roth, User Experience Evangelist
rroth@apple.com

Related Sessions

Wallet - The Home for Apple Pay and More	Mission	Tuesday 10:00AM
Introducing the Contacts Framework for iOS and OS X	Pacific Heights	Thursday 3:30PM

Related Labs

Wallet and Apple Pay Lab	Frameworks Lab C	Tuesday 1:30PM
Wallet and Apple Pay Lab	Frameworks Lab C	Wednesday 1:30PM