When Your App Needs A New Business Model

StoreKit Migration without breaking everything

ASTRID LIN

Astrid Lin

iOS Developer at KKBOX

Feels great when people enjoy what I built

Love to travel

Tonku


try! Swift
TOKYO
try! Swift
TOKYO

Agenda

- StoreKit APIs Overview
- KKBOX Context & Strategy
- Technical Challenges
- Tools for Testing
- App Store Review Reality
- Takeaways

Know which In-App Purchase framework your company is using

Hands up

Started using StoreKit 2

Hands up

Original API for In-App Purchase

StoreKit 1

StoreKit 1

Making Purchases

```
func makePurchase(product: SKProduct) {


}
```

StoreKit 1

Making Purchases

```
func makePurchase(product: SKProduct) {
    let payment = SKPayment(product: product)

}
```

StoreKit 1

Making Purchases

```
func makePurchase(product: SKProduct) {
    let payment = SKPayment(product: product)
    SKPaymentQueue.default().add(payment)
}
```

StoreKit 1

Making Purchases

```
// Obs
class
    fu
}
```

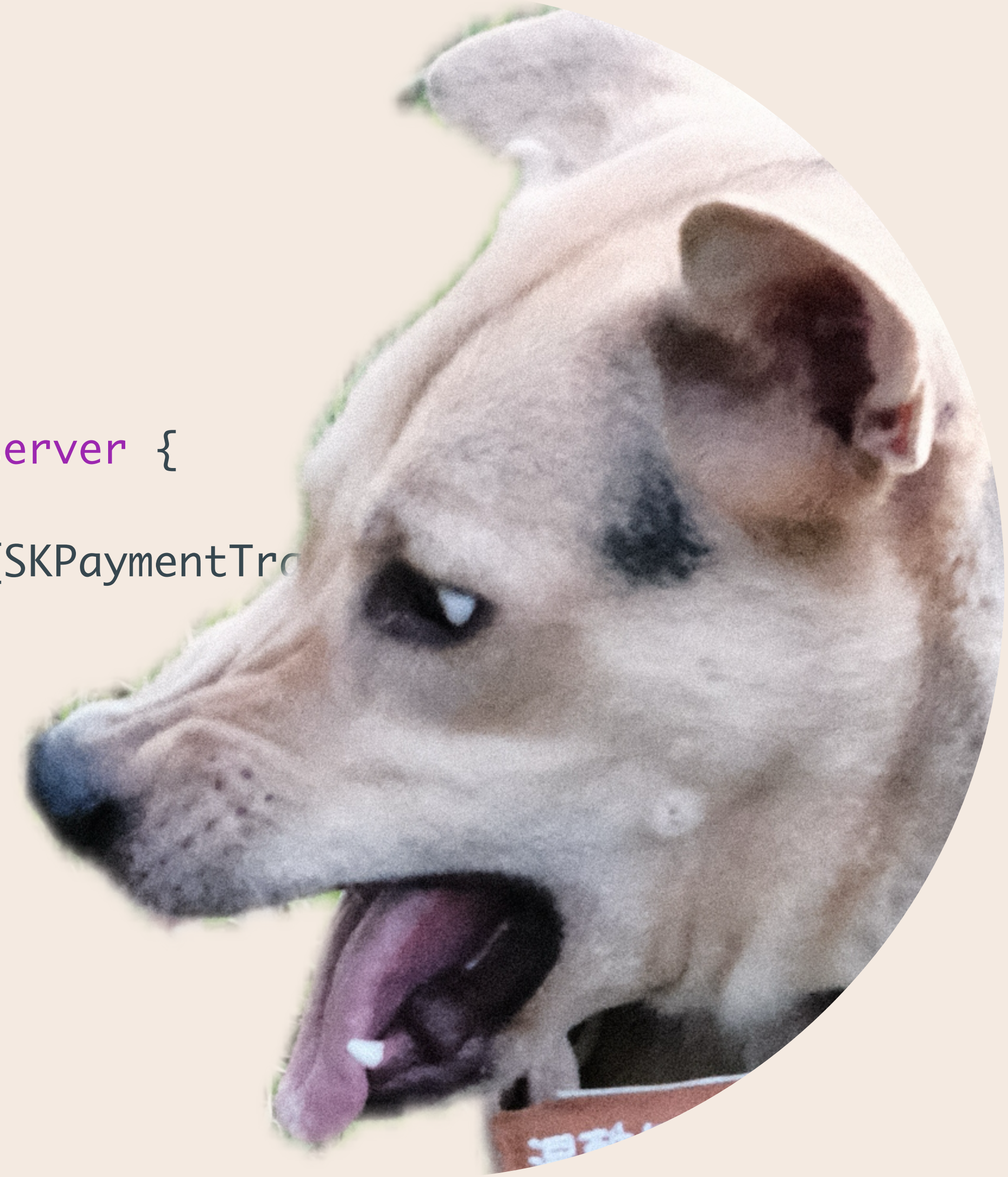
```
            }).done(function(response) {
                for (var i = 0; i < response.length; i++) {
                    var layer = L.marker(
                        [response[i].latitude, response[i].longitude]
                        // ,{icon: myIcon}
                    );
                    layer.addTo(group);

                    layer.bindPopup(
                        "<p>" + "Species: " + response[i].species + "<
                        "<p>" + "Description: " + response[i].descript
                        "<p>" + "Seen at: " + response[i].latitude + "
                        "<p>" + "On: " + response[i].sighted_at + "</
                }

                $('select').change(function() {
                    species = this.value;
                });
            });
        }
    $.ajax({
                url: queryURL,
                method: "GET"
            }).done(function(response) {
                for (var i = 0; i < response.length; i++) {
                    var layer = L.marker(
                        [response[i].latitude, response[i].longitude]
                        // ,{icon: myIcon}
                    );
                    layer.addTo(group);
```

```
Observer {

[SKPaymentTra
```



StoreKit 2

StoreKit 2

Making Purchases

```
// Simple async purchase
func makePurchase(product: Product) async throws -> Transaction? {




}
```

StoreKit 2

Making Purchases

```
// Simple async purchase
func makePurchase(product: Product) async throws -> Transaction? {

    let result = try await product.purchase()
    switch result {
                    ...
    }
}
```

StoreKit 2

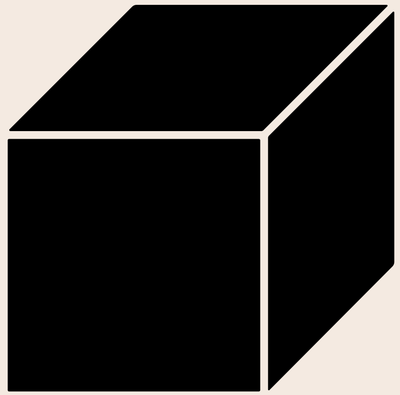
Making Purchases

```
// Simple async purchase
func makePurchase(product: Product) async throws -> Transaction? {

    let result = try await product.purchase()
    switch result {
                    ...
    }
}
```

StoreKit 1

Unified Receipt



StoreKit 2

Transaction Object

```
struct Transaction {

}
```

StoreKit 2

Transaction Object

```
struct Transaction {
    let productID: String

}
```

StoreKit 2

Transaction Object

```
struct Transaction {
    let productID: String
    let purchaseDate: Date



}
```

StoreKit 2

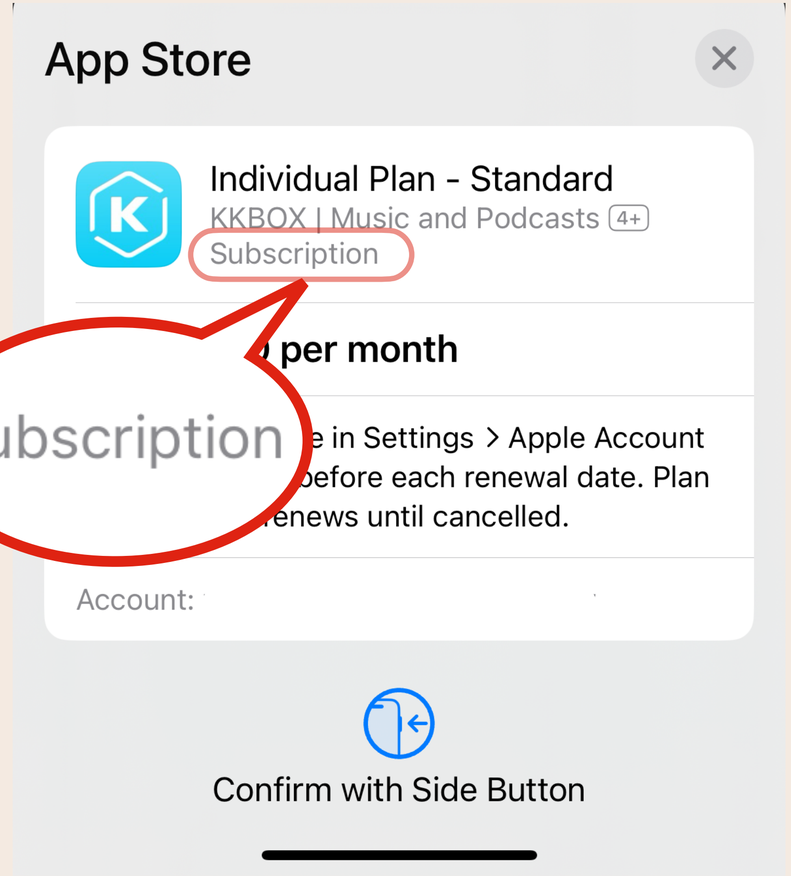
Transaction Object

```
struct Transaction {
    let productID: String
    let purchaseDate: Date
    let environment: AppStore.Environment
    let storefront: String
    let originalID: UInt64
    let originalPurchaseDate: Date
    let id: UInt64
    let appBundleID: String
    let productType: Product.ProductType
    let expirationDate: Date
    …
}
```

KKBOX Implementation Context

Original Situation

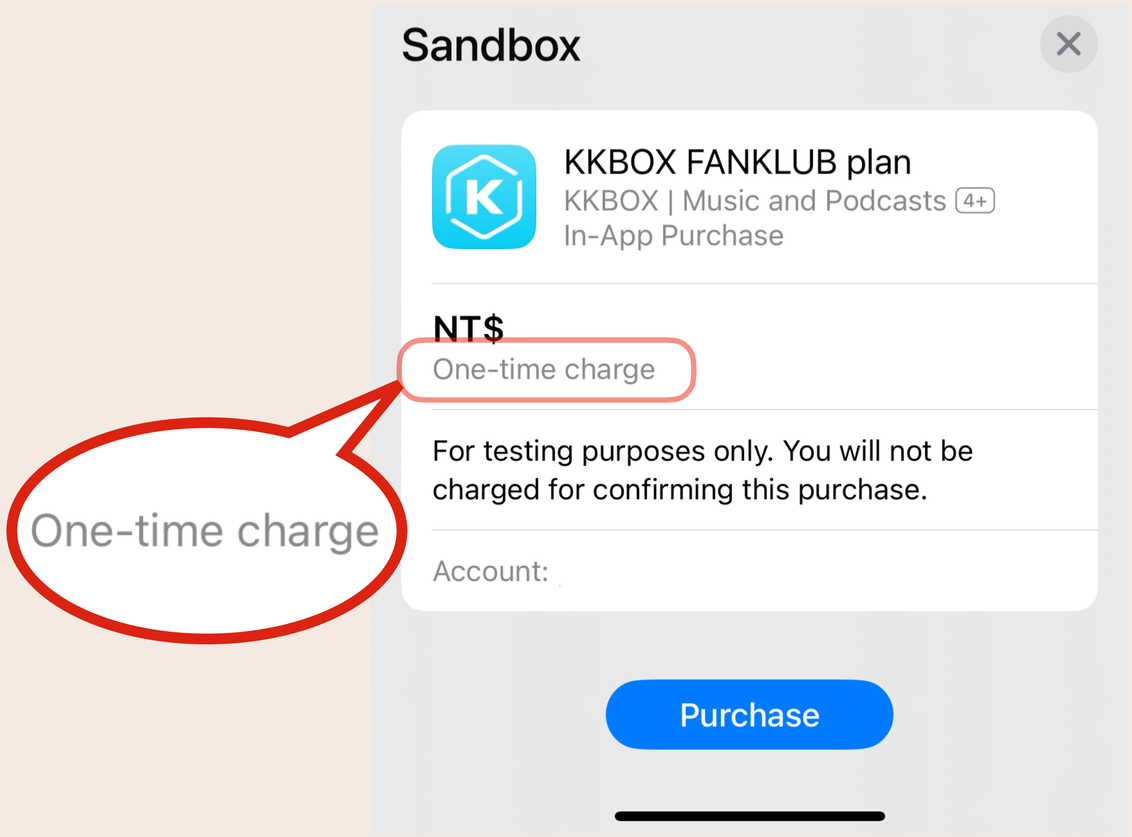
- StoreKit 1, verifyReceipt endpoint
- auto-renewable, non-renewable
- Legacy billing system
- Works well in production



KKBOX Implementation Context

NEW FANKLUB

- Consumable product
- Swift Concurrency
- FanKlub billing system



Strategic Options

Option 1: Fully move to StoreKit 2

- Changes core parts
- Cross-team work
- Delay FanKlub launch

Strategic Options

Option 2: Stay with StoreKit 1

- StoreKit 1, verifyReceipt API deprecated

Strategic Options

Option 3: Hybrid Approach

- StoreKit 1 for music streaming
- StoreKit 2 for FanKlub
- iOS 15+ OK

Recap

- StoreKit 2, modern and easier
- Hybrid approach
- Adds complexity

Technical Implementation Challenges

Transaction history

“

All transactions are available in all StoreKit API

Purchases made using original StoreKit APIs are available in StoreKit 2 APIs

Purchases made using StoreKit 2 APIs are available inside the unified receipt

”

- WWDC21 Meet StoreKit 2

Overlapping transaction handling

- StoreKit 1's `updatedTransactions` triggered by StoreKit 2 purchases
- StoreKit 2's `Transaction.updates` includes StoreKit 1 purchases

Overlapping transaction handling - Solution

All FanKlub product IDs contain “fanKlub”

- StoreKit 1's `updatedTransactions` method

```
guard !transaction.payment
        .productIdentifier.contains("fanKlub") else {
        continue // Skip FanKlub products
    }
```

Overlapping transaction handling - Solution

All FanKlub product IDs contain “fanKlub”

- StoreKit 2's `Transaction.updates`

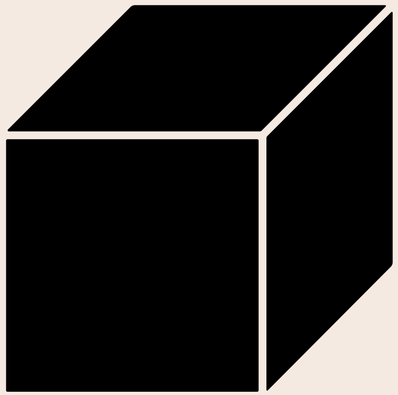
```
guard transaction.productID
        .contains("fanKlub") else { return }
```

Mixed Receipt Validation

StoreKit 1 uploads a unified receipt

Including:

- Music streaming ✅
- FanKlub purchases ⚠️



Legacy billing system 😵

Mixed Receipt Validation - Solution

StoreKit 1 uploads a unified receipt

Ignores FanKlub transactions in legacy billing system

Testing at all stages of development

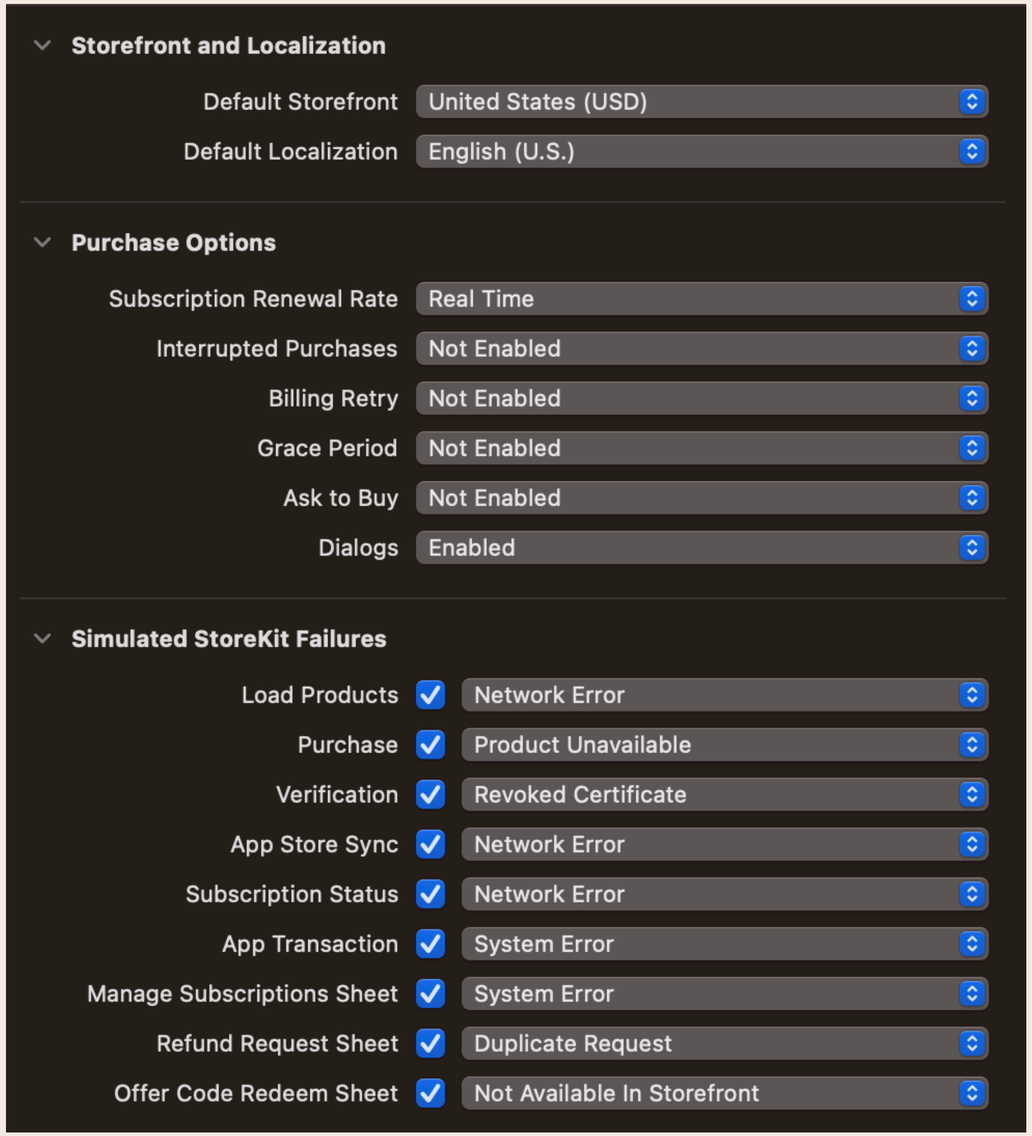
StoreKit Testing

Sandbox

TestFlight

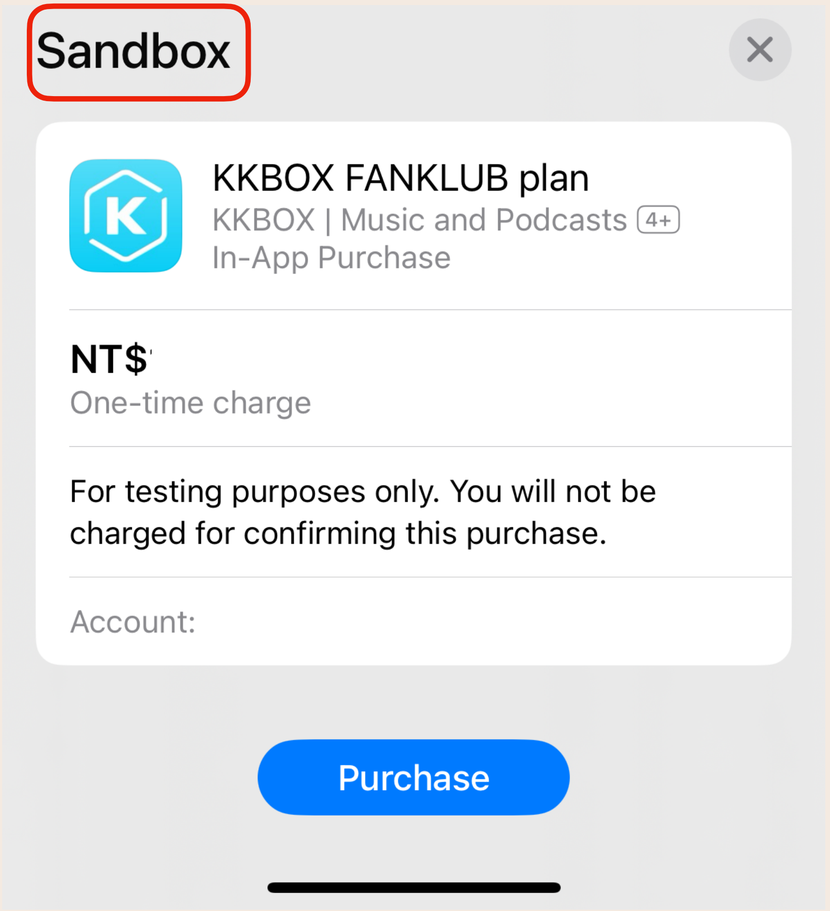
StoreKit Testing in Xcode

- Doesn’t need internet
- Offline playground
- Handle possible scenarios



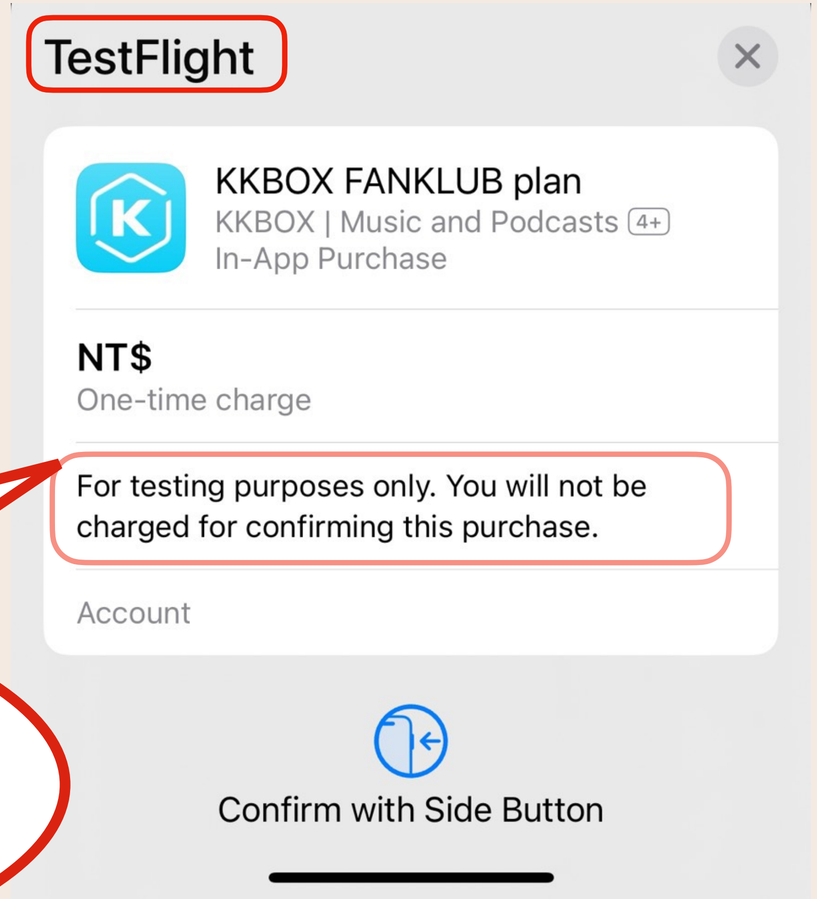
App Store Sandbox

- Test both flows
- Send sandbox receipts to server
- Solve issues with the app server



TestFlight

- For internal testers
- Improve the user experience



For testing purposes only. You will not be charged for confirming this purchase.

App 內購買項目已退回。請修正標示項目並重新提交。若想瞭解更多資訊，請參閱 App 審查備註。

需要開發者進行處理 (10)

參照名稱	產品 ID	類型	
		消耗性項目	需要開發者進行處理
		消耗性項目	需要開發者進行處理
		消耗性項目	需要開發者進行處理
		消耗性項目	需要開發者進行處理
		消耗性項目	需要開發者進行處理
		消耗性項目	需要開發者進行處理
		消耗性項目	需要開發者進行處理
		消耗性項目	需要開發者進行處理
		消耗性項目	需要開發者進行處理
		消耗性項目	需要開發者進行處理

App Review Notes

Guideline 3.1.1 – Business – Payments – In-App Purchase

“ We have begun the review of your in-app purchases but aren't able to continue because your submitted in-app purchases indicate **a change of business model** for your app.

Specifically, your existing Auto-Renewable Subscription and Non-Renewing Subscription business model has **changed to include a consumable in-app purchase** business model type.

App Review Notes

Guideline 3.1.1 – Business – Payments – In-App Purchase

“ We have begun the review of your in-app purchases but aren't able to continue because your submitted in-app purchases indicate **a change of business model** for your app.

Specifically, your existing Auto-Renewable Subscription and Non-Renewing Subscription business model has **changed to include a consumable in-app purchase** business model type.

App Review Notes

Guideline 3.1.1 – Business – Payments – In-App Purchase

Next Steps

In order to approve your new in-app purchase business model, we need to verify that the items being sold are **purchasable.** Please **upload a new binary** and make sure that your new in-app purchase products **are available for purchase at the time of review.**”

What We Did Before Resubmission

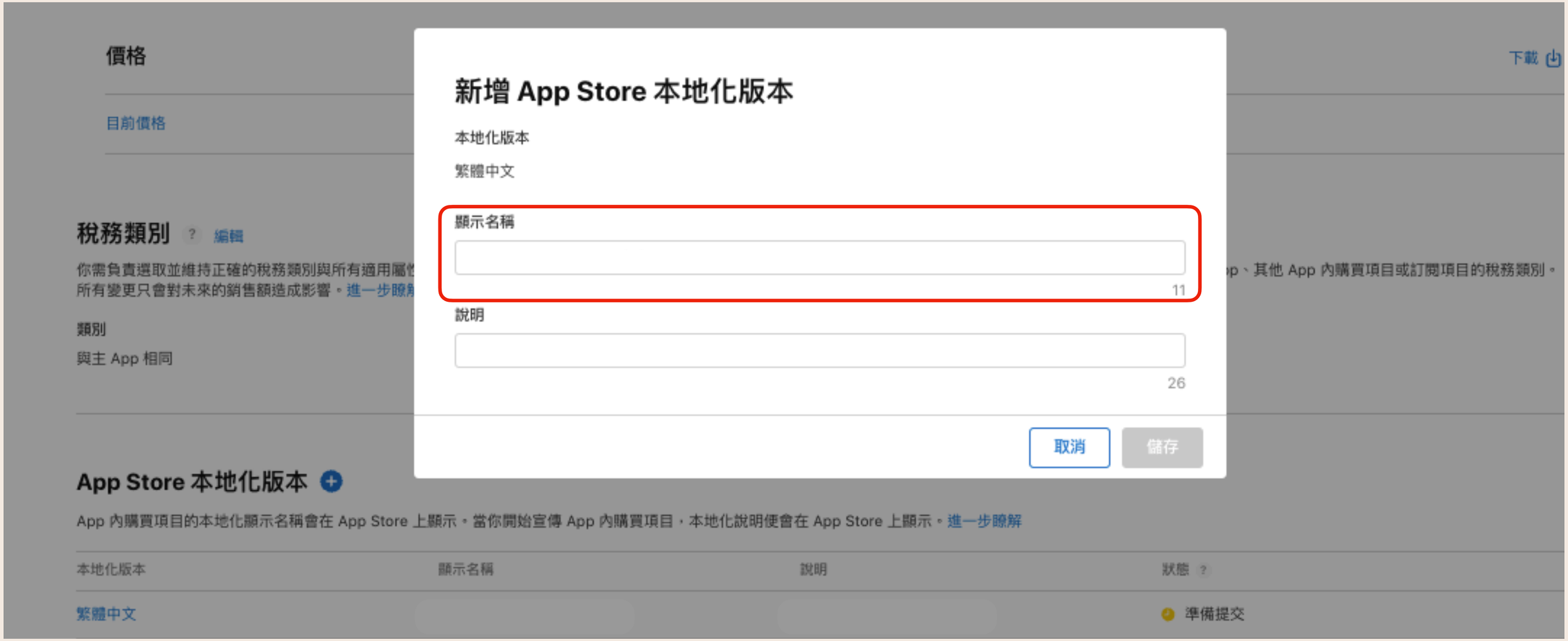
- Backend production setup
- Enabled sandbox transactions
- Recorded demo video
- Renamed localized product names

Persistent “Product ID error”

Block our resubmission

Persistent “Product ID error” - solution

Trick: Edit product localized name



通過核准 (11)

類型	
消耗性項目	通過核准
消耗性項目	通過核准
消耗性項目	通過核准
消耗性項目	通過核准
消耗性項目	通過核准
消耗性項目	通過核准
消耗性項目	通過核准
消耗性項目	通過核准
消耗性項目	通過核准
消耗性項目	通過核准

Submit Early

Global Trend: Loosening IAP Requirements

Region	External Payment Allowed?
U.S.	Partially (external links allowed)
EU	Yes (DMA - external payment with entitlement)
Korea	Yes (national law)
Taiwan	No (must use Apple IAP)

Practical Implementation Checklist

For anyone planning a similar hybrid approach

- ☑ Coordinate with backend about mixed receipts
- ☑ Filter purchases by product ID
- ☑ Test both flows using testing tools
- ☑ Leave buffer time for App Review (New business model)
- ☑ Ensure purchasability during review (New business model)

MUSIC ARTIST

FAN LUB Beta

FANS KKBOX

近 近 的 喜 歡 你

Slide

Tonku

FAN LUB