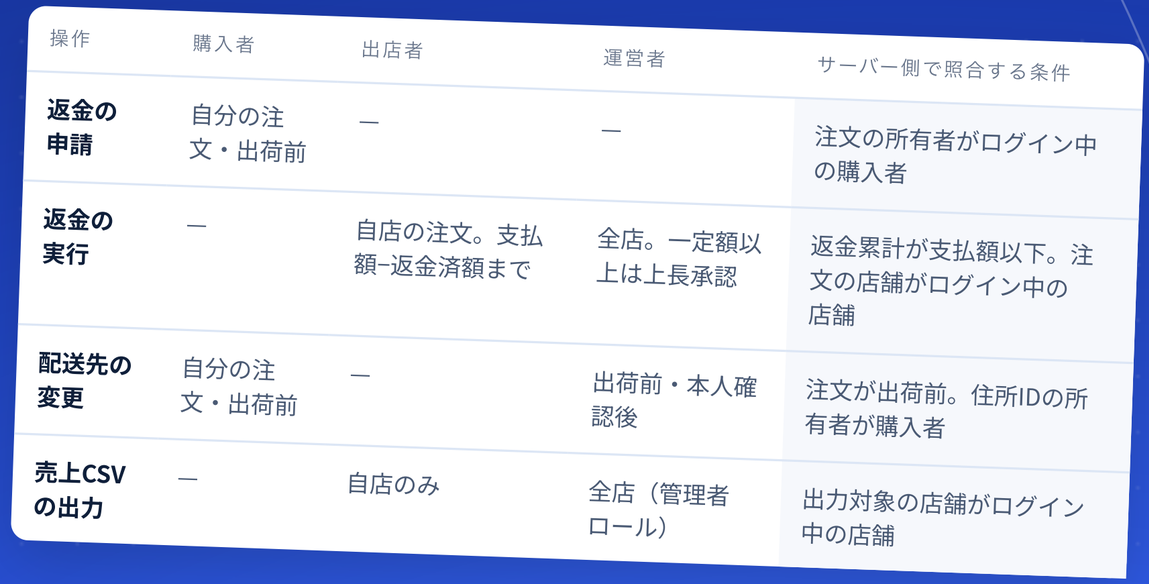
03 全8ページ

EC・マーケットプレイス向け

返金・注文・出店者権限の不正操作を防ぐ

設計の考え方

照合の場所・権限の書き方・直した後の確認を
1冊にまとめる



操作	購入者	出店者	運営者	サーバー側で照合する条件
返金の申請	自分の注文・出荷前	—	—	注文の所有者がログイン中の購入者
返金の実行	—	自店の注文。支払額−返金済額まで	全店。一定額以上は上長承認	返金累計が支払額以下。注文の店舗がログイン中の店舗
配送先の変更	自分の注文・出荷前	—	出荷前・本人確認後	注文が出荷前。住所IDの所有者が購入者
売上CSVの出力	—	自店のみ	全店（管理者ロール）	出力対象の店舗がログイン中の店舗

Securiy セキュリー

WHEN YOU NEED IT

機能追加のたびに、権限確認のない経路が増える

見つけて直すだけでは、次の機能追加で同じ穴が開きます。この冊子は、不正操作が成立しない設計の考え方と、確認の仕方をまとめたものです。

PULL REQUEST ／ 機能追加のレビュー（想定）　app/refund.py

```
1    @app.post('/refund')
2 +  def handler(req):
3 +      order = Order.get(req.order_id)
4 +      refund(order, req.amount)   # ← 注文額・所有者と照合していない
5 +      return ok()
```

レビュー　権限の確認がありません。返金は、注文の金額・対象・状態と一致する要求だけが通る、という条件をサーバー側のどこで照合しますか？

設計を見直す場面

返金・注文変更の機能追加

CS向けの便利機能が、権限確認のない経路になりやすい

出店者向け管理画面の拡張

店舗IDの照合が1カ所抜けると、他店のデータに届く

外部連携・AIチャットの導入

画面と同じ権限確認が、新しい経路にも適用されているかを確かめる必要がある

PERMISSION TABLE

権限は「誰が・何を・どの条件で」の表にする

購入者／出店者／運営者について、操作ごとの許可条件と、サーバー側で照合する条件を1枚にします。「—」のマスが実際に拒否されるかが、検証の対象です。

操作	購入者	出店者	運営者	サーバー側で照合する条件
返金の申請	自分の注文・出荷前	—	—	注文の所有者がログイン中の購入者
返金の実行	—	自店の注文。支払額−返金済額まで	全店。一定額以上は上長承認	返金累計が支払額以下。注文の店舗がログイン中の店舗
配送先の変更	自分の注文・出荷前	—	出荷前・本人確認後	注文が出荷前。住所IDの所有者が購入者
売上CSVの出力	—	自店のみ	全店（管理者ロール）	出力対象の店舗がログイン中の店舗

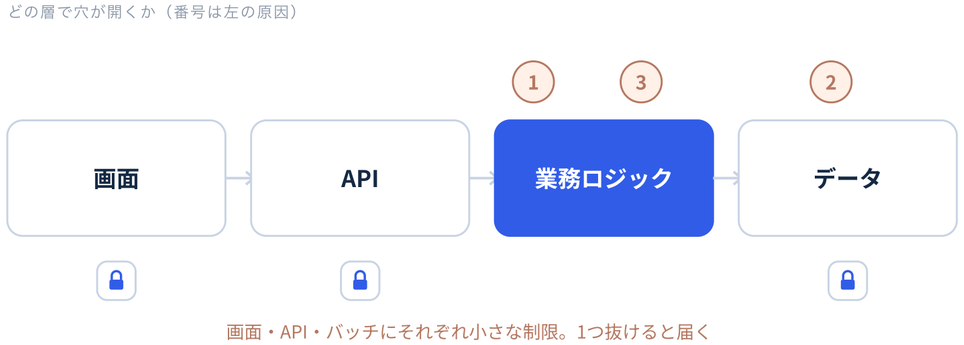
使い方　新しい機能を足すときは、先にこの表に行を足す。右列が書けない操作は、照合の場所が決まっていない。

WHERE TO CHECK

照合はサーバー側の1カ所に置く

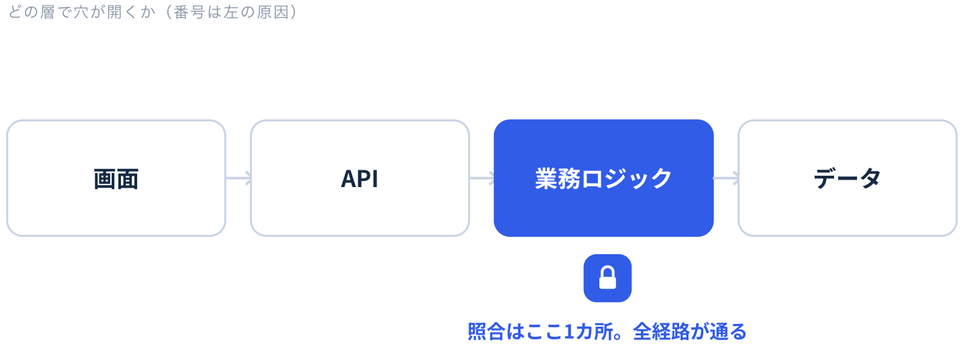
画面・API・バッチ・AI と経路が増えても、照合の場所を増やさない。データ取得・更新の共通層に1カ所置き、どの経路もそこを通す。

Before ／ 経路ごとに散らばる



返金額と対象を、画面の値のまま信じている

After ／ 共通層に1カ所



返金はサーバー側で注文額・支払額・返金累計・対象の所有を照合し、累計が支払額を超える要求は拒否する。一定額以上は承認必須にする。

DESIGN PRINCIPLES

不正操作が成立しない設計の3原則

直す場所を増やすより、照合の場所を1カ所にまとめる方が、機能追加に強くなります。

1

照合の場所を1カ所にまとめる

返金額は注文額と、対象は注文の所有者と、サーバー側で必ず照合します。画面側の制限には頼りません。

2

権限は「誰が・何を・どの条件で」で書く

購入者／出店者／運営者の3者について、操作ごとの許可条件を表にします（P.03）。

3

直した後に、通常の返金が通ることを確かめる

拒否されることと、正常な返金・注文変更が止まらないことを、同じ表で確認します（P.07）。

DESIGN REVIEW

設計レビューの10問（機能追加のたびに）

「いいえ」「不明」が1つでもあれば、その操作が検証の候補です。共同検証の初回2操作もここから選べます。

01	テナント・店舗・会員のIDは、リクエストの値ではなく、ログイン中の所属と照合してから使っているか	☐ はい ☐ いいえ ☐ 不明	
02	照合はデータ取得・更新の共通層に1カ所あるか。画面側の制限だけに頼っていないか	☐ はい ☐ いいえ ☐ 不明	
03	一覧・検索・CSV出力・帳票など、後から足した経路も同じ共通層を通るか	☐ はい ☐ いいえ ☐ 不明	
04	金額・数量・ポイントは、画面から来た値ではなく、サーバー側の実績（注文額・残高）と照合しているか	☐ はい ☐ いいえ ☐ 不明	
05	累計（返金累計・付与累計）で上限を判定しているか。1回ごとの判定だけになっていないか	☐ はい ☐ いいえ ☐ 不明	
06	並行リクエストで残高・回数の確認が競合しないか（条件つき更新・ロック）	☐ はい ☐ いいえ ☐ 不明	
07	確定後（出荷後・確定後）の変更は、状態を検査し、承認付きの経路に限定しているか	☐ はい ☐ いいえ ☐ 不明	
08	運営者・本部の機能は別ロール・別経路か。URLを知っているだけで到達しないか	☐ はい ☐ いいえ ☐ 不明	
09	共有URL・招待リンク・クーポンコードは推測不能で、失効・1回限りの制約があるか	☐ はい ☐ いいえ ☐ 不明	
10	AI チャット・外部 API からの要求にも、画面と同じ照合が適用されているか	☐ はい ☐ いいえ ☐ 不明	

RETEST

直した後に、通常操作が通ることを同じ表で確かめる

修正で不正操作が拒否されるようになったことと、通常の操作がそのまま通ることを、経路×シーンの同じ表で確認してからリリースします。



拒否されること

他店の注文への返金は拒否。別の手順・モデルでも同じ結果

通常操作が通ること

通常の返金は継続。正規の担当者の操作が止まらない

未確認

初回に入れなかったシーンと経路は、理由と次回の予定を記録

記載の検証シーンは共同検証のテーマ例で、業界別の導入実績・認証取得を示すものではありません。対象は、検証と修正を行える自社アプリ・独自APIです。

権限表と照合の場所を、共同検証で一緒に決める。

初回スコープ（1アプリ・2操作・10営業日）で、権限表の「—」のマスが実際に拒否されるかを試し、修正案・再検証チェック・再実行用テストをお渡しします。本番の認証情報や実データは不要です。

Securiy は Tempura technologies株式会社のセキュリティ事業です。

FREE CONSULTATION

30分無料相談を予約する

権限表の下書きをお持ちいただければ、どの操作から検証するかを一緒に決めます。

予約ページへ →

SET

01 検証シーン集 ／ 02 プログラムのご案内

EC・マーケットプレイスの3シーンは 01、対象・期間・費用・体制は 02 へ。