Security Review - Purolator Delivery Pro (Mobile App)

This is a client-side only review of the Delivery Pro Android app. Most behaviors observed are standard for mobile apps. The only item requiring backend confirmation is whether critical validation (e.g., blocked drivers, POD validation, delivery status rules) is enforced server-side. All other notes are low-impact hardening suggestions.

1. Hardcoded APIM Subscription Key (Already Reported)

The APIM subscription key is hardcoded in `util/environments.dart` and also transmitted in the `Ocp-Apim-Subscription-Key` header for all API requests to `https://pl-apim-live.azure-api.net/` .

2. Client-Side Business Logic & Blocked Driver Checks

Severity: Low to Medium (depending on backend)

Location: `network/repositories/authentication_repository.dart` (lines 136-884)

The app performs key checks on the client side only, including:

- `isBlocked` driver verification
- POD field validation (`ui/upload_pod/upload_pod_view_model.dart::isPodAndSignatureRequired()`)
- Address form validation (`ui/upload_pod/update_delivery_address.dart::isValidForm()`)
- Barcode format checks

These validations occur after a successful API response. A modified client could bypass them if backend APIs do not duplicate the same checks.

Code Example:

```
// authentication_repository.dart, lines ~778–790
if (globalUser.isBlocked == true) {
  return Future.error("This number is blocked...");
}
```

Backend Confirmation Requested:

1. Does `/api/drivers/login` block isBlocked users with 401/403?
2. Do POD/delivery/status APIs re-check driver permissions, package ownership, and valid status transitions?

Recommendation:
All critical logic should be enforced on the backend, with the client acting only as a presentation layer.

3. Other Notes (Info)

These items are normal mobile-app patterns and have minimal security impact:

Local Secure Storage:
Uses FlutterSecureStorage + Android Keystore (expected). Storage keys defined in `util/secure_storage/secure_storage.dart` (lines 626-627):

```
static const String userDetails = "userDetails";
static const String savePackagePODImages = "savePackagePODImages";
```

Offline POD Queue:
Typical retry/queue behavior implemented in `ui/package_flow/viewmodels/package_landing_view_model.dart` (line ~1580). Backend should ensure idempotency/deduplication.

Third-Party API Configuration:
HERE Maps integration in `dependency_injection/service_locator.dart` (lines 3533-3540) with no certificate pinning. Standard practice for third-party APIs.

Predictable Secure-Storage Keys:
Normal naming convention; optional obfuscation only.

Session Timeout:
Not visible from client code; backend may clarify token/session policies.

Debug Logs / No Tamper Detection:
Standard for many apps; optional hardening.

The Delivery Pro client follows common mobile patterns. The only meaningful security question is whether the backend enforces the same blocked-driver, POD, and delivery-status rules that the client checks locally. If so, overall risk is low; if not, this is the area that should be addressed.