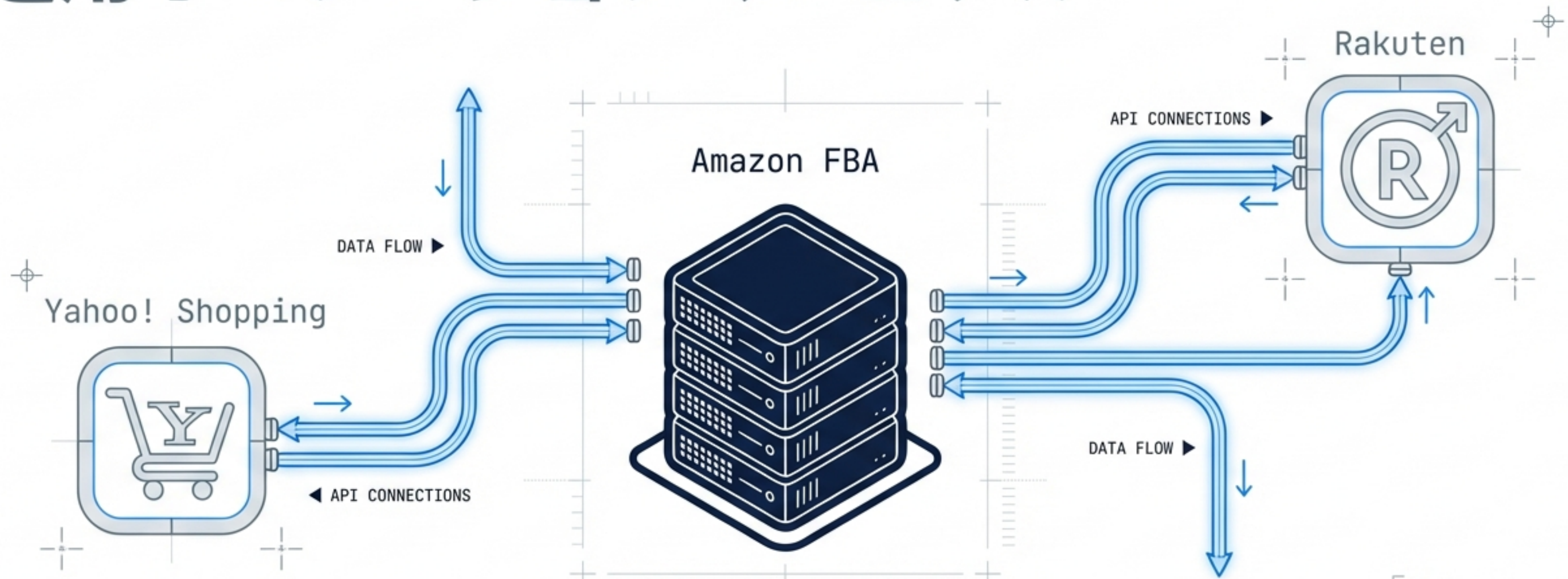


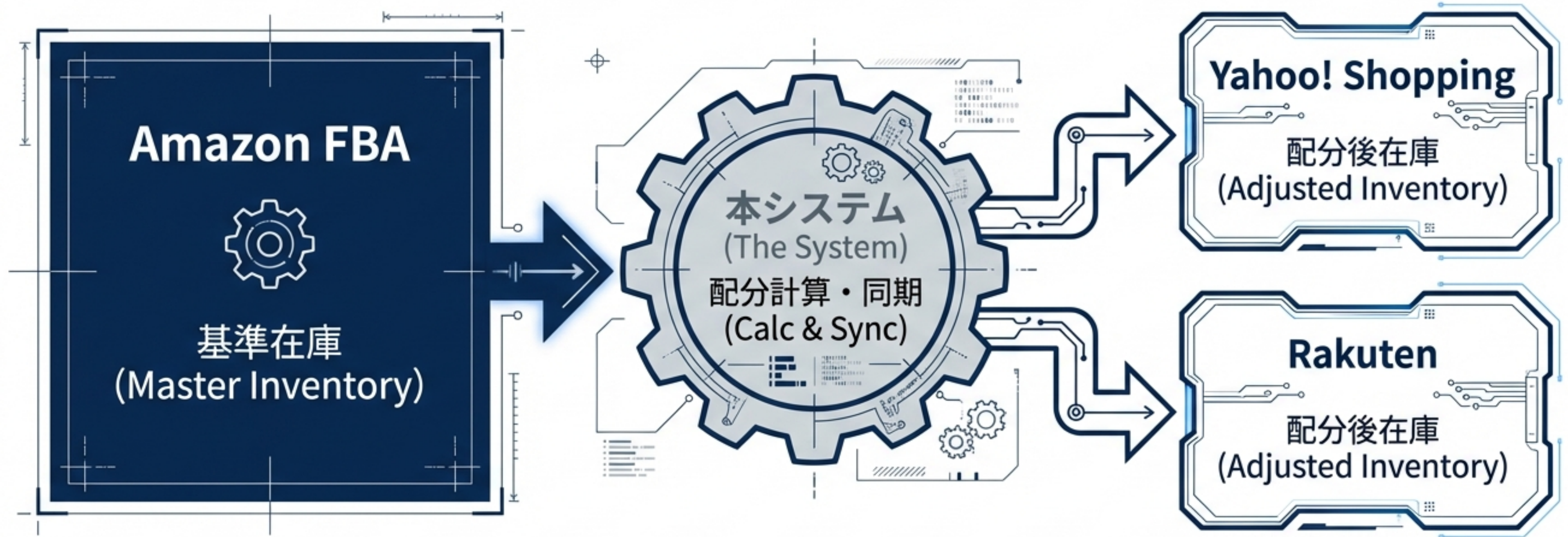
マルチEC在庫管理システム

運用オペレーションマニュアル



VER: 1.0
DATE: 2026-03-02
TARGET: EC OPS / ADMIN

システム概要：Amazon FBAを「正」とする在庫同期



- 本システムは、Amazon FBA在庫数を基準に、他モールへ自動同期します。
- Goal: 売り越し防止 (Prevent Overselling) 🎯

在庫配分ロジック



Amazon在庫	配分率 (Allocation)
20個以上	1/5 (例: 25→5)
10~19個	1/4 (例: 16→4)
6~9個	1/3 (例: 9→3)
1~5個	1/2 (例: 4→2)

※小数点以下は切り捨て

サーバー接続と環境確認

```
login as: user
Authenticating with public key...
(venv) user@server:~$
```

重要：(venv) が表示されていることを確認

Global Options

- -v: Verbose Mode (詳細ログ出力) 🎯
- -c: Config Path (設定ファイル指定) 🎯

安全確認：必ず「ドライラン」から開始する

```
> python main.py sync --dry-run
```

```
[INFO] Starting Dry Run...
```

```
Amazon Inventory: 142 items found
```

```
Mapped SKUs: 138 items
```

```
Updates to be sent: 15 items
```

```
[DRY-RUN] No changes made to API.
```

Amazon Capture:

1 Amazon在庫の取得件数は正しいか？

Mapping:

2 SKUマッピングの照合件数は正しいか？

Calculation:

3 各商品の配分計算結果は適切か？

実際のAPI更新は行われません

同期の実行：手動と自動

Manual Execution (手動)



```
> python main.py sync
```

緊急在庫調整、メンテナンス後

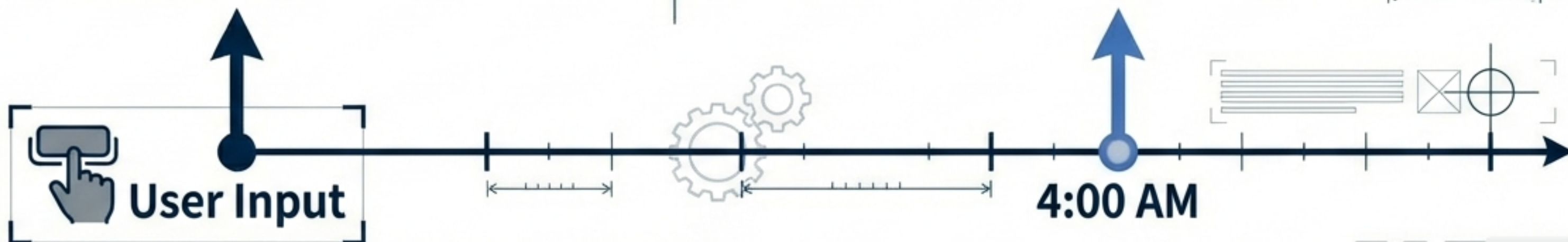
Automated Schedule (自動)



毎日 AM 4:00
(via cron)

```
> tail -f logs/inventory_sync.log
```

自動実行のログ確認



SKUマッピング管理

Path: ../commonsettings/master.xlsx

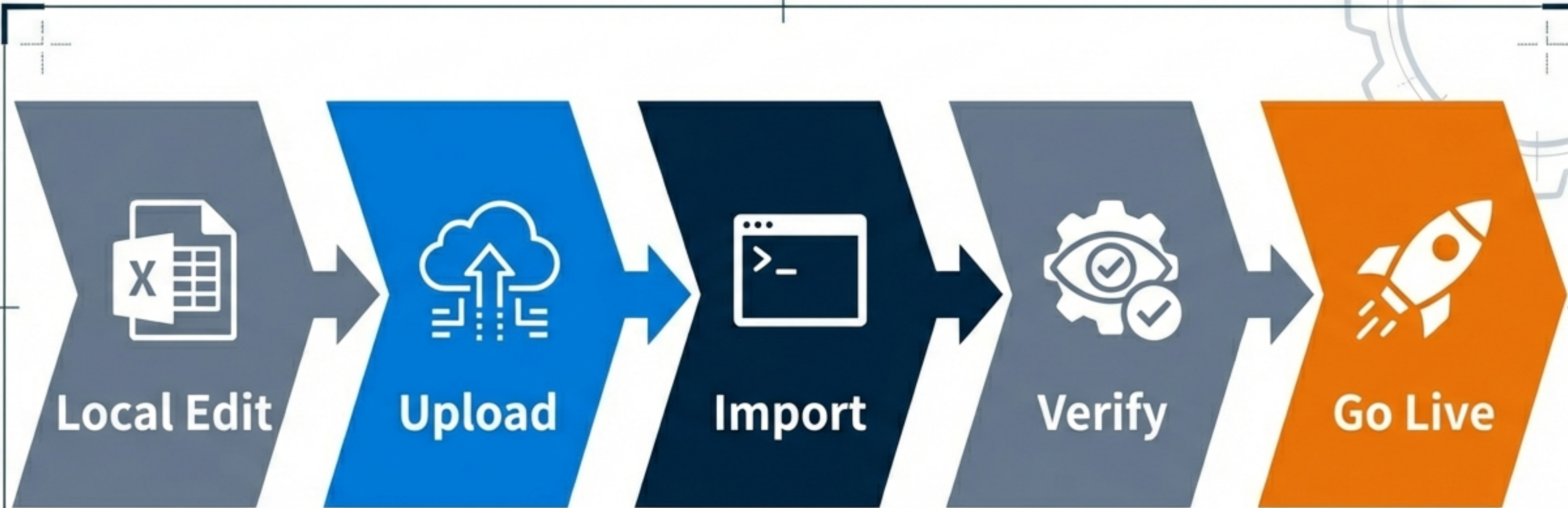
	A	B	C	D
1	amazon_sku	yahoo_item_code	rakuten_item_number	is_active
1	AMZ-P-001	YH0-01234	RKT-98765	TRUE
2	AMZ-P-002	YH0-01235	RKT-98766	FALSE
3	AMZ-P-003	YH0-01236	RKT-98767	TRUE
4	AMZ-P-004	YH0-01237	RKT-98768	TRUE
5	AMZ-P-005	YH0-01238	RKT-98769	FALSE
6				
7				
8				
9				

Unique Key
(重複不可)

Sync ON

Sync OFF
(対象外)

商品情報の更新フロー



Excel編集
(Hiragino Sans)

サーバーへ転送
(Hiragino Sans)

import-sku 実行
(JetBrains Mono)

ドライラン確認
(Hiragino Sans)

本番同期実行
(Hiragino Sans)

マスターデータの取り込み

```
python main.py import-sku -f master.xlsx
```



Option: --replace

既存マッピングを全削除してからインポートします。取扱い注意！

Use Case: 新商品の追加、販売終了商品の除外

実行状況とヘルスの確認

Widget

Status (実行履歴)

```
python main.py status
```



直近の実行ログと成否
Hiragino Sans

Widget

Health (システム状態)

```
python main.py health
```



DB Connection: OK
Hiragino Sans

Widget

Simulate (計算試算)

```
python main.py simulate
```



在庫計算のプレビュー
Hiragino Sans

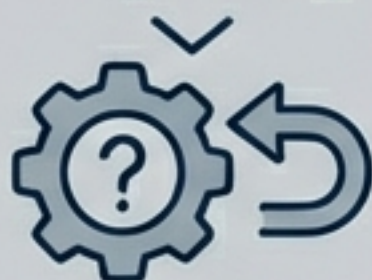
緊急対応：ロールバック

誤った在庫更新を前回（直前）の状態に戻す

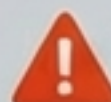
プロセス (The Process)



1. ``status`` (Confirm ID)
確認用IDの取得



2. ``rollback --dry-run`` (Check Impact)
影響範囲の事前確認

 実行結果を確認 (Review Results)



3. ``rollback`` (Execute)
ロールバック実行

制限事項 (Limitations)



24時間以内
(Within 24h)



1回のみ
(One time only)

定期メンテナンスチェックリスト

Daily (日次)

- Check `status` (FAILED?)
- Check Slack/Email Errors

Weekly (週次)

- Review `logs/` trends
- Server Disk Usage

Monthly (月次)

- Token Expiry (Yahoo!)
- DB Size check
- Log rotation

トラブルシューティング：API認証

Yahoo! Refresh Token Limit

28 Days

Solution

有効期限切れ前に更新が必要です

```
python scripts/get_yahoo_token.py
```

Action: Update .env file with new token

Note

Amazon:
Update in
Seller
Central

Rakuten:
Update in
RMS

コマンドリファレンス

Command	説明	Options
sync	在庫同期	--dry-run, --file
import-sku	マスタ取込	--replace
status	履歴確認	--limit
rollback	巻き戻し	--dry-run (24h limit)
health	状態確認	-
cleanup	データ削除	-
File Locations	config/config.yaml ../commonsettings/master.xlsx	

運用体制と問い合わせ



EC Operator (運用担当)

- Master File mgmt
- Daily checks
- Sync execution



System Admin (システム管理)

- Server deploy
- DB maintenance
- Error investigation

Contact: `admin@example.com`

関連文書: Deployment Guide, Spec Sheet

Change Management: **Allocation rules require Manager approval.**