LICENSED COMMERCE STACK - CUSTOMER WORKFLOW - API GUIDE

Hybrid Store License Workflow, Delivery Logic, and API Guide

This customer-facing guide explains how the licensed stack works after activation: the storefront, licensing lifecycle, hybrid digital & physical flows, digital delivery, reseller API behavior, webhooks, and the merchant dashboards included in the stack.

Monthly plan - USD 14 / month

Yearly plan - EUR 129 / year

Hybrid digital & physical architecture

WordPress + WooCommerce friendly

Scope note: this document intentionally focuses on how the license and platform behave for the customer after activation. It does not describe the private back-office used to generate or administer license keys.

INSIDE THIS GUIDE

1. Plans and pricing
2. What the customer receives
3. How the license works
4. Storefront workflow
5. Hybrid digital & physical logic
6. Digital delivery workflow

7. API & webhook documentation
8. Dashboards, portals, and FAQ

PLANS AND PRICING

Two billing options, one operational stack

The two plans below describe the same licensed platform behavior while the subscription remains active. In this guide, the difference is the billing cycle and the price shown. The operational workflows, protected runtime, updates, storefront modules, delivery engine, and API layer are described as the licensed stack.

Monthly

Monthly license

$14 /month

- Best for pilots, short launch windows, and staged rollouts.
- Good when you want lower commitment while validating the full workflow.
- Access continues while the subscription stays active and validated.

Billed in USD exactly as shown above.

Yearly

Yearly license

EUR 129 /year

- Best for long-term stores, reseller programs, and stable production deployments.
- Fewer renewal touchpoints across the year.
- Same platform logic, same protected runtime model, longer billing horizon.

Billed in EUR exactly as shown above.

Included while active

Core entitlements

- Domain-aware activation and signed runtime validation.
- Storefront, cart, checkout, and payment routing.
- Digital delivery engine for codes, files, or mixed packs.
- Reseller API, request tracing, HTTPS webhook support, and dashboards.

This guide describes the operational stack, not the private license-generation back office.

Practical reading tip: think of the license as the switch that authorizes the protected runtime, syncs the customer-facing stack, and keeps updates available while the subscription remains valid.

INCLUDED STACK

What the customer receives on the licensed site

The client-side stack shown below is what matters to a licensed merchant operating a live store. The protected licensing service sits behind the scenes, but the merchant mainly interacts with the storefront, the delivery engine, and the reseller API layer.

Hybrid Store Pro Max

A premium storefront layer with product catalog, cart, checkout, payment routing, WooCommerce bridge, wallet support, multi-currency display, and a refined admin/customer experience.

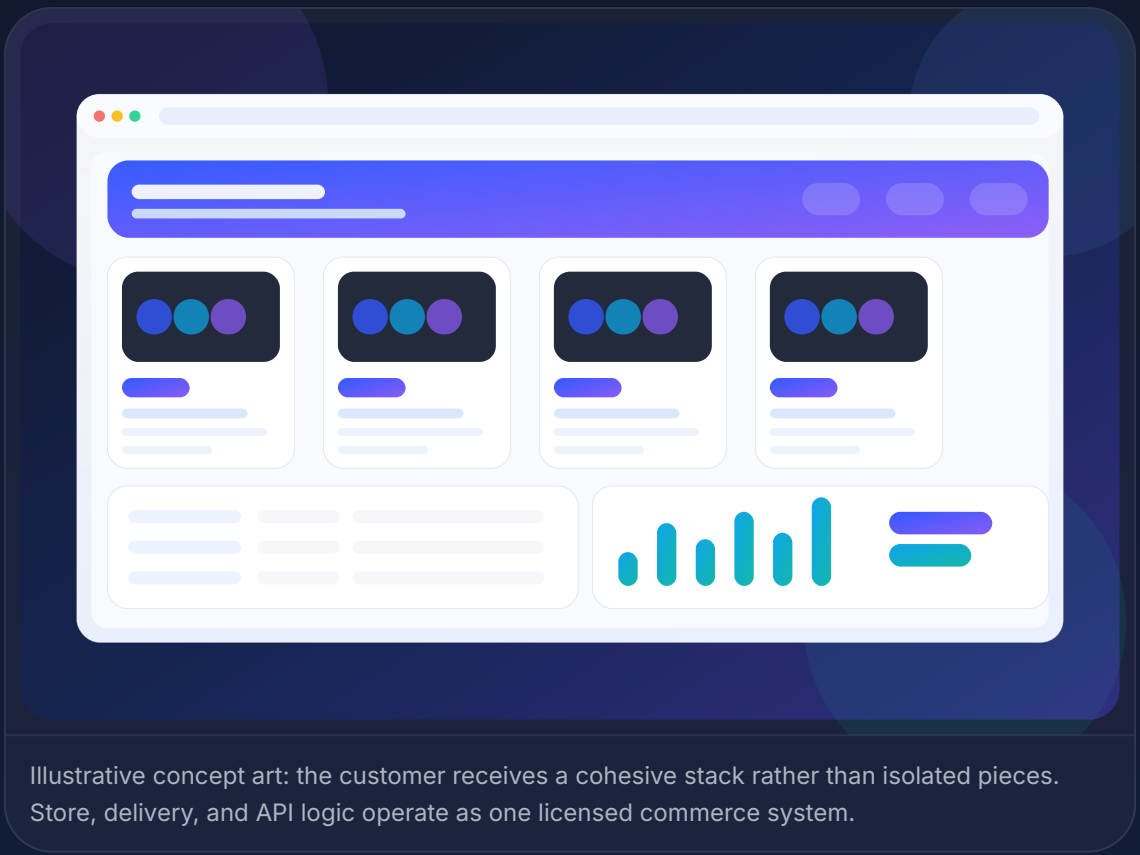
Digital Delivery Complete Immediate Edition

A stock-backed delivery engine that can send text codes, secure file links, or both. It supports immediate attempts, waiting notices when stock is not ready, and automatic retry checks.

SoftMedia Reseller API Pro

An authenticated REST layer for server-to-server ordering, EUR wallet settlement, request_id tracing, external_id idempotency, signed webhook notifications, and merchant dashboards/logs.

- Signed runtime entitlement
- Remote updates while valid
- Modern storefront UI
- Customer digital library
- HTTPS webhook support
- Masked logs and rate limiting



Illustrative concept art: the customer receives a cohesive stack rather than isolated pieces. Store, delivery, and API logic operate as one licensed commerce system.

LICENSE LIFECYCLE

How the license works after a customer buys it

The license is not just a static key. It participates in activation, protected runtime validation, update access, and continued entitlement checks for the customer site.

1

Receive the license key

The customer receives a license key for the purchased billing cycle - monthly or yearly - and prepares the target WordPress domain.

2

Activate on the approved domain

The customer activates the key on the intended site. The domain is normalized and linked to the installation context rather than treated as a loose text field.

3

Validation and signed lease

After successful validation, the site receives a signed runtime entitlement (lease). This authorizes protected runtime behavior and keeps the stack unlocked while valid.

4

Protected modules unlock

The site can synchronize protected runtime assets, customer-facing bundles, and available update data. From the customer perspective, the store becomes fully operational.

5

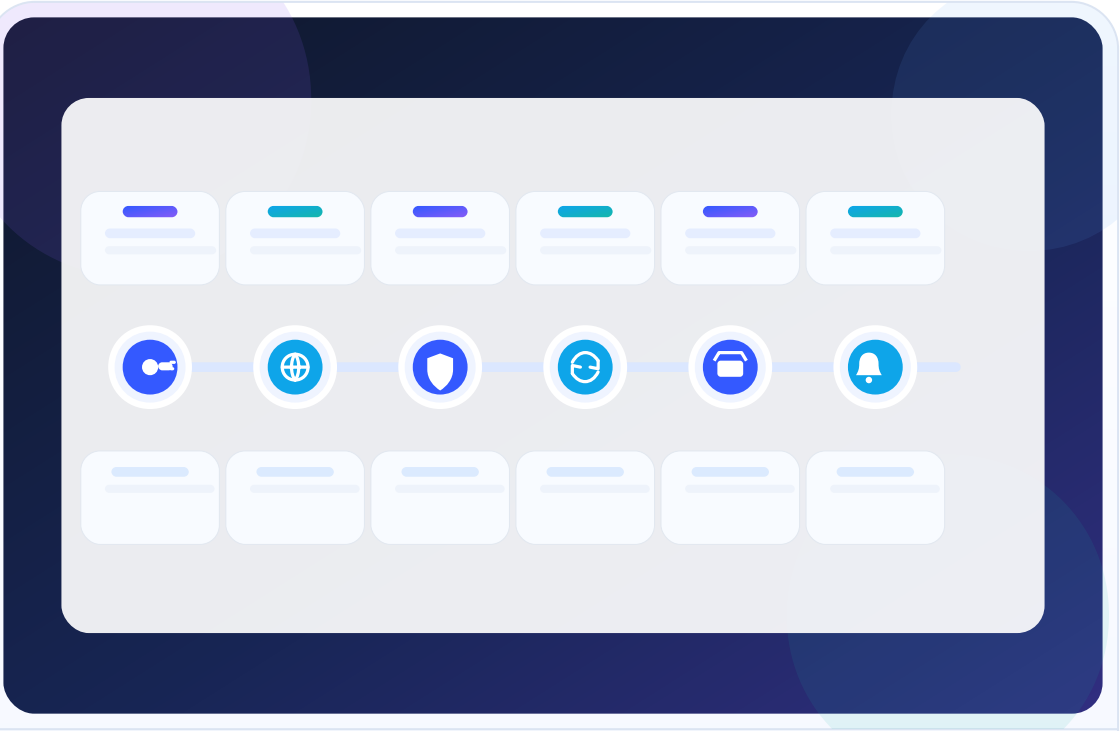
Periodic revalidation

The runtime checks back periodically. A short grace window can absorb a temporary hub outage, but long-term access still depends on an active validated subscription.

6

Renew to keep protected access

If the subscription expires or becomes disabled, protected runtime access and update availability are restricted after the allowed validity and grace behavior end.



Illustrative workflow: key → approved domain → signed validation → sync → live storefront → continued entitlement checks.

What matters to the customer

- The license is tied to the approved site context rather than being a casual copy-paste token.
- Validation produces a signed runtime entitlement used by the customer site.
- Remote updates and protected runtime features depend on an active valid state.
- A short grace mode can protect against brief connectivity problems, not permanent expiry.

Recommended customer language

Use the following promise in sales or onboarding pages: “Your license activates the protected runtime on your approved domain, keeps your storefront stack entitled while active, and allows update access according to the purchased billing cycle.”

STOREFRONT WORKFLOW

How the store behaves for real shoppers on the licensed site

Once the merchant has an active license, the storefront is the public face of the stack. It combines product browsing, product variants, cart management, checkout, payment routing, order creation, and status updates.

1

Discovery and selection

Shoppers browse a premium storefront with categories, featured items, searchable catalog cards, variant selection, and optional payment plan options where configured.

2

Cart and customer details

The cart can collect email plus the customer details needed for order delivery. For physical flows, address-related checkout fields are available in the same checkout experience.

3

Payment routing

The stack can route the shopper through wallet payment, hosted checkout links, WooCommerce order-pay pages, or manual/offline methods. Cash on delivery is appropriate for physical items only.

4

Order creation

When checkout succeeds, a structured order record is created with items, quantities, pricing snapshots, checkout fields, payment status, and API status.

5

Payment status changes

The order moves through pending, paid, failed, or related states depending on the chosen payment path and webhook or WooCommerce sync behavior.

6

Fulfillment begins

After payment is confirmed, eligible digital items can be sent to the delivery engine. Physical items remain local to the store workflow unless the merchant intentionally extends them.

Payment routes available to the licensed site

- Internal wallet for instant paid orders.
- Hosted/redirect checkout for links such as Stripe or PayPal pages.
- WooCommerce bridge for native order-pay gateway flow.
- Manual or offline confirmation for custom local workflows.
- Cash on delivery for physical goods only.

Store display capabilities

- Multi-currency storefront display.
- Product variants and featured badges.
- Single or supported split-plan presentation where configured.
- Optional AI storefront assistant for product Q&A and shopper guidance.

Important distinction: the shopper’s checkout flow and the merchant’s licensed activation flow are related but separate. The license authorizes the runtime; the storefront then handles customer commerce on top of that runtime.

HYBRID LOGIC

Why this is a true hybrid digital & physical system

The store is not limited to one product type. It can present digital products and physical goods in the same branded storefront while allowing each item to follow the correct fulfillment path.

Digital products

Digital items can use automated fulfillment logic. After payment, they may trigger API delivery behavior, secure email delivery, customer-library access, or mixed code+file delivery according to the configured product flow.

Physical products

Physical items stay in the same catalog and checkout, but they do not auto-trigger digital API delivery by default. This keeps local order handling clean and prevents accidental digital fulfillment for physical stock.

Shared storefront

Customers see one branded shop, one cart, one checkout, and one order concept - even when the items inside the order have different downstream fulfillment paths.

Shared payment layer

The same payment routing architecture can support both product types. The difference appears after payment, when the system chooses the correct delivery or local handling path.

Separated fulfillment logic

Digital products can go to email/API/customer portal workflows. Physical items remain in merchant-controlled handling unless extended intentionally.

Illustrative concept art: one storefront can host digital licenses, downloadable packs, and physical goods without collapsing everything into the same fulfillment path.

DIGITAL DELIVERY

How digital items are delivered to the buyer

The delivery engine is stock-aware. It is designed for reliable fulfillment of codes, files, or mixed digital packs while keeping a customer-accessible library of delivered items.

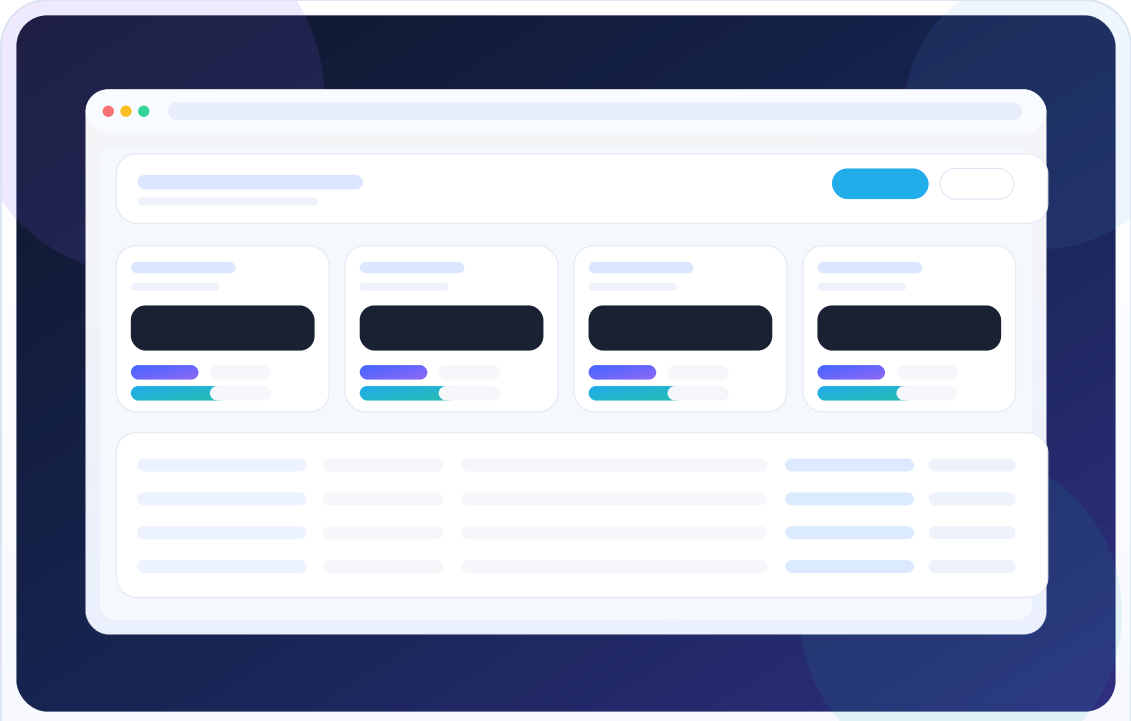
Delivery content types

- **Text code:** a code, serial, token, or plain-text digital asset.
- **File delivery:** a downloadable file exposed through a secure link in the email body.
- **Mixed delivery:** both text content and file delivery in the same item.

For deliverability, file fulfillment is designed around secure links in the email body rather than depending on heavy email attachments.

Reliability behavior

- Reserved stock rows protect the correct quantity before the email is sent.
- If stock is not ready, the buyer can receive a waiting message instead of silent failure.
- A retry worker checks pending deliveries regularly (every 5 minutes in the reviewed stack).
- Delivered rows are marked and connected back to the customer email/order reference.



Illustrative concept art: the delivered-items portal can act as a customer library for codes, download links, timestamps, copy actions, and CSV export.

✉️ Email confirmation

When delivery succeeds, the buyer receives an HTML email containing the digital content or the secure download link.

📚 Customer library

Delivered products can also appear in a “My Digital Products” area where buyers can search, sort, copy codes, print, or export their records.

🧪 Test pathway

The reviewed stack includes a DGTEST-style flow for testing delivery behavior without treating it like a normal paid production item.

API AND WEBHOOKS

Reseller API workflow and documentation

The reseller API is intended for server-to-server ordering. It records an order, authenticates the merchant with an API key, debits the merchant wallet in EUR, assigns a request_id, and then waits for the external delivery layer to mark the item delivered.

1 Send the request

Your server posts to the API endpoint with the API key and the customer order payload.

2 Validate, rate-limit, and trace

The API checks required fields, applies rate limiting, generates a request_id, and rejects duplicate external_id values for the same merchant user.

3 Debit the EUR wallet

The merchant wallet is debited in EUR. If the request currency is EUR, USD, CHF, or CAD, the stack resolves the EUR value before debit.

Field	Required	Description
`api_key`	Yes	Merchant secret used to authenticate the request. Keep it on the server side only.
`product`	Yes	The product label recorded in the order row.
`balance`	Yes for normal orders	Unit price in the submitted currency. DGTEST-style requests can behave differently for testing.
`quantity`	Yes	Quantity to record and price.
`currency`	Yes	Supported API currencies in the reviewed stack: EUR, USD, CHF, CAD.
`client_email`	Yes	The buyer email associated with the order and later delivery confirmation.
`external_id`	Recommended	Merchant-side idempotency key. Duplicate values are rejected for the same merchant user.

```
POST /wp-json/custom-api/v1/add-product/
Content-Type: application/json

{
  "api_key": "YOUR_SERVER_SIDE_KEY",
  "product": "Yearly License Pack",
  "balance": 129,
  "quantity": 1,
  "currency": "EUR",
  "client_email": "buyer@example.com",
  "external_id": "ORD-2026-00041"
}
```

```
{
  "success": true,
  "message": "Produit ajouté avec succès.",
```

```
  "test_mode": false,
  "request_id": "0f6be9b9-2b35-4e1c-8a6a-43a40a39d145",
  "product_id": 1042,
  "debited_eur": 129.0,
  "new_wallet_balance_eur": 871.0,
  "unit_fx": {
    "input": { "amount": 129, "currency": "EUR" },
    "eur_per_unit": 129.0
  },
  "delivery_status": "pending_external_delivery"
}
```


Webhook behavior

The API plugin does **not** mutate delivery state by itself. It waits until the external delivery layer marks the order as delivered and then sends a webhook notification if webhook delivery is enabled for the merchant.

- Webhook destinations are expected to use HTTPS.
- Signed HMAC delivery is supported through `X-UPA-Signature`.
- Retries can be scheduled if the receiving endpoint fails.

Webhook headers and payload

- `X-UPA-Request-Id`
- `X-UPA-Event` = `product.delivered`
- `X-UPA-Timestamp`
- `X-UPA-Signature` when a secret exists


```
{
  "status": "delivered",
  "product": "Yearly License Pack",
  "email": "buyer@example.com",
  "message": "Delivery confirmed.",
  "test": false,
  "request_id": "0f6be9b9-2b35-4e1c-8a6a-43a40a39d145",
  "product_id": 1042,
  "delivered_at": "2026-04-20 15:34:22",
  "delivery_source": "stock_auto"
}
```

Illustrative concept art: the merchant sees request IDs, wallet effects, logs, and webhook behavior without exposing sensitive keys in a public front-end.

Security baseline: keep the API key server-side, use `external_id` for idempotency, prefer HTTPS-only webhook targets, and verify the HMAC signature when you receive webhook calls.

PORTALS, DASHBOARDS, FAQ

What merchants and buyers can actually see and use

The value of the license is not limited to hidden runtime checks. It appears in the interfaces: cleaner order centers, product libraries, API dashboards, and a more coherent customer experience across the site.

STORE

1

Branded storefront, cart, and checkout flow

DELIVERY

3

Code, file, or mixed digital delivery modes

API

1

Authenticated reseller endpoint with tracing

LIFECYCLE

24/7

Ongoing validation while the subscription is active

Merchant order center

Merchants can monitor order creation, payment status, linked WooCommerce flow, and API fulfillment state from a more polished operational surface.

Buyer digital library

Buyers can view delivered products, open file links, copy codes, and export their delivered records without digging through email history.

Merchant API dashboard

API users can review request IDs, wallet impact, logs, webhook configuration, and delivery notifications in a dedicated interface.

What happens if the subscription expires?

After the valid runtime entitlement and any short grace behavior are exhausted, protected runtime access and update availability are restricted until the customer returns to a valid licensed state.

Can digital and physical products live together in one store?

Yes. The reviewed stack is intentionally hybrid. Digital items can use automated delivery paths, while physical items remain in the same storefront and checkout but do not auto-trigger digital delivery by default.

Does the API itself send the digital item?

No. The API records and prices the order, debits the wallet, and exposes traces. Final delivery is confirmed by the dedicated delivery layer, which can then trigger webhook notification.

Why are secure file links used instead of email attachments?

This improves email reliability and reduces the chance that large attachments block or weaken delivery. Buyers still receive the file through the email, but as a controlled download link.

Who should choose monthly vs yearly?

Choose monthly for testing, short campaigns, or staged deployment. Choose yearly when the store is part of a stable production plan and you want fewer renewal interruptions across the year.

Does the guide describe the private license-generation back office?

No. This guide is intentionally customer-facing. It explains operational behavior after purchase and activation - the part the licensed client needs to understand and use.

Bilingual customer guide generated from the reviewed stack: Hybrid Store Pro Max, Digital Delivery Complete Immediate Edition, and SoftMedia Reseller API Pro.