StoreFlow — A1.1 Architecture & System Design (Part 2)

6. Deployment Models

StoreFlow supports multiple deployment architectures:

VM Deployment:

- Ubuntu LTS recommended
- Nginx + PHP-FPM + MySQL + Redis
- Supervisor for queue workers
- Certbot for TLS
- Suitable for small–medium scale

Docker Deployment:

- Containers: app, queue, scheduler, websocket, mysql, redis
- Networking via Docker Compose
- Horizontal scaling by adding more app containers

Kubernetes Deployment:

- Pods: app, queue, scheduler, websockets
- StatefulSets: MySQL, Redis
- Ingress controller for routing
- ConfigMaps for environment configuration
- Secrets for sensitive data
- Horizontal Pod Autoscaler for scaling

7. Environment Configuration

Three primary environments:

Local (XAMPP friendly):

- MySQL running in XAMPP
- Laravel artisan serve
- Websockets using local loopback
- Queues using database driver

Staging:

- Separate MySQL
- Redis for queue
- Websockets via laravel-websockets
- Staging domains with permissive CORS
- Stripe test keys

Production:

- Strict environment isolation
- Redis as cache + queue
- MySQL 8 with backup rotation
- SSL mandatory
- Stripe live keys (only after merchant onboarding)

8. Logging & Monitoring

LOGGING:

• Laravel Monolog channels:

- daily logs

- slack notifications for critical errors

- JSON structured logs for ingestion

MONITORING:

• Health endpoint for uptime monitors

• Heartbeat for scheduler jobs

• Redis metrics (queue length, failures)

• Websocket uptime reporting

• Optional Sentry integration

9. WebSocket Architecture

WebSockets are critical for live order updates.

Components:

- Laravel Websockets package
- Echo server client
- Broadcast events mapped to channels:

store.{store_id}.orders

merchant.{merchant_id}.notifications

Event Types:

- OrderCreated
- OrderUpdated
- ShippingUpdated
- LoyaltyUpdated (reserved)

Latency Considerations:

- Keep event payload minimal
- Offload heavy tasks to queues

10. Stripe Connect Readiness

Stripe Connect (Standard accounts) is planned.

Current placeholders:

- stripe_account_id on merchants table
- onboarding_url generation path reserved
- payment_intent_id stored per order

Constraints:

- Until payments go live, orders treated as paid
- Future batching of Stripe usage for StoreFlow billing

11. Domain Service Boundaries

Domain services encapsulate business logic:

OrderService:

- Create order
- Validate fulfilment type
- Trigger events

ShippingService:

- Run shipping calculations
- Resolve zones and rates

ProductService:

- Sync product data
- Enforce store linkage

LoyaltyService:

- Calculate points
- Apply redemptions

AuditService:

- Append audit logs across all domains

12. Queue Architecture

Queues enable async operations:

• Email sending (future)

• Audit aggregation

• Order event broadcasting

• Shipping recalculation for bulk updates

Recommended:

• Redis queue

• Supervisor with:

- horizon or standard worker

13. API Strategy

StoreFlow dashboard uses Inertia, minimizing API needs.

Public API Endpoints:

- /shipping/quote
- /checkout
- /track/{public_id}

Internal API Endpoints:

- /orders/update
- /products/create
- /shipping/config

Versioning:

- v1 prepended for public API
- Dashboard endpoints remain server-driven

14. Caching & Performance Layers

Caching:

- products list cached per store
- shipping zones cached per store
- loyalty config cached per merchant

Performance Notes:

- Pagination for heavy tables
- Lazy loading Vue components
- Avoid N+1 with eager loading

15. Media Storage Architecture

Options:

Local Disk:

- XAMPP-friendly
- Development only

S3 Bucket:

- Recommended for production
- Private ACL with signed URLs

File Types:

- product images
- store logos

16. Backup & Disaster Recovery

MySQL backups:

- Daily full dump
- 7-day retention
- Offsite backup optional

Redis persistence:

- RDB snapshots every 15 minutes

Disaster Recovery:

- Deploy duplicate environment
- Restore DB snapshot
- Rehydrate cached store settings

17. Security & Rate Limiting

Rate Limiting Policies:

- Login: 5 attempts per minute
- Checkout: 30 requests per IP per hour
- Shipping quote: 20/min per IP
- Dashboard: global throttle per user

Security Enhancements:

- CORS locked to storefront domains
- Force HTTPS
- SameSite cookies
- Session regeneration
- No cross-store resource access