

ASTERIA 2.0 Integration Playbook – System-at-a-Glance

Phase 0 – Environment Bootstrap

- Verify Node 18+, Firebase CLI, `gcloud`, Git hooks
- Create **tag-inner-circle-v01** Firebase project & enable Firestore
- Enable Google Cloud Secret Manager API
- *Sources:*

Phase 1 – Secret Manager Migration

- Inventory all API keys (OpenAI, Tavily, Slack, Twilio, Stripe, ElevenLabs, Amadeus, G-Cal) in `.env.local`
- Run `migrate-secrets-to-gcp.js` → 16 secrets + 1 JSON added
- Grant `secretAccessor` IAM to Firebase & Cloud Run service accounts
- Replace env look-ups with `get<Provider>Key()` helpers & caching
- *Sources:*

Phase 2 – Secret-Aware API Refactor

- Update all routes (`/chat` , `/search` , `/tts` , `/voice` , `/webhooks/*`) to call Secret Manager helpers
- Provide env-var fallback for local dev; validate with `test-secret-manager.js`
- *Sources:*

Phase 3 – Workflow Engine Core (5.2)

- Define `WorkflowState` , `WorkflowStep` , `ApprovalRequest` types
- Build Firestore-backed `WorkflowStateManager` with real-time listeners
- Implement `WorkflowEngine` (parallel exec, dependency graph, retries, timeouts)
- Provide `StepExecutor` types (`api_call`, `notification`, `payment`, `booking`, `approval`, `custom`)

- *Sources:*

Phase 4 – Workflow Templates (5.3)

- Create default templates:
 - *Payment* → Stripe intent → notify
 - *Booking* → Google Calendar free/busy → event create
 - *Travel* → Amadeus search → options ↔ member
- Add tier-based variants & fallback logic
- *Sources:*

Phase 5 – External Service Adapters

- `services/stripe.ts` – intents, customers, webhooks
- `services/calendar.ts` – availability, events, sync
- `services/travel.ts` – flights, hotels, OAuth2 caching
- *Sources:*

Phase 6 – Agent ↔ Workflow Bridge (5.4)

- `AgentWorkflowBridge` maps intent + member tier → template
- Inject workflow metadata into agent responses
- Fallback to classic tools if workflow creation fails
- *Sources:*

Phase 7 – Knowledge-Base RAG Layer

- Create Postgres + pgvector tables `knowledge_documents`, `knowledge_chunks`
- `ingest()` → split docs → embed → batch UPSERT
- `retrieve()` → top-k chunks (cos sim > 0.7)
- Expose `searchKnowledgeBase` tool (intent-aware query expansion)
- *Sources:*

Phase 8 – Notification Channels

- Slack webhook & Twilio SMS helpers pull creds from Secret Manager

- Implement `getActiveChannels()` per member tier / concierge status
- *Sources:*

Phase 9 – Security & Observability

- No hard-coded secrets; audit via GCP logs
- Ensure bundle ≤ 304 kB, zero TypeScript errors, build ≤ 7 s
- Stream workflow & agent logs to Firestore in real time
- *Sources:*

Phase 10 – Production Readiness Gate

- 100 % Secret Manager coverage
 - Workflow engine passes unit + integration tests
 - End-to-end demo: "Book jet" → Stripe payment workflow → Slack/Twilio notifications
 - *Sources:*
-

Heads-up on file expiry

A few of the older uploads referenced in earlier turns have expired from the session. If you need them re-indexed for future breakdowns, please re-upload and I'll fold them in.

ASTERIA 2.0 Integration Playbook – System-at-a-Glance

Use this as a pared-down knowledge-base index. Each heading is a milestone; bullets are the concrete, code-free tasks or checks that must be satisfied before moving on.

#	Phase	Key Tasks / Checks
0	Environment Bootstrap	<ul style="list-style-type: none"> • Verify Node 18 +, Firebase CLI, <code>gcloud</code>, and Git hooks • Create tag-inner-circle-v01 Firebase project & enable Firestore • Enable Google Cloud Secret Manager API
1	Secret Manager Migration	<ul style="list-style-type: none"> • Inventory all API keys (OpenAI, Tavily, Slack, Twilio, Stripe, ElevenLabs, Amadeus, G-Cal) in <code>.env.local</code> • Run <code>migrate-secrets-to-gcp.js</code> → 16 secrets + 1 JSON added • Grant <code>secretAccessor</code> IAM to Firebase & Cloud Run svc accts • Replace env look-ups with <code>get<Provider>Key()</code> helpers & caching
2	Secret-Aware API Refactor	<ul style="list-style-type: none"> • Update all routes (<code>/chat</code> , <code>/search</code> , <code>/tts</code> , <code>/voice</code> , <code>/webhooks/*</code>) to call Secret Manager helpers • Fallback to env vars for local dev; verify with <code>test-secret-manager.js</code>
3	Workflow Engine Core (Phase 5.2)	<ul style="list-style-type: none"> • Create <code>WorkflowState</code> , <code>WorkflowStep</code> , <code>ApprovalRequest</code> types • Build Firestore-backed <code>WorkflowStateManager</code> with real-time listeners • Implement <code>WorkflowEngine</code> (parallel exec, dependency graph, retries, timeouts) • Provide <code>StepExecutor</code> (api_call, notification, payment, booking, approval, custom)
4	Workflow Templates (Phase 5.3)	<ul style="list-style-type: none"> • Define three default templates: – <i>Payment</i> → Stripe intent → notify – <i>Booking</i> → Google Calendar free/busy → event create – <i>Travel</i> → Amadeus search → options ↔ member • Add tier-based variants & fallback logic
5	External Service Adapters	<ul style="list-style-type: none"> • <code>services/stripe.ts</code> – intents, customers, webhooks • <code>services/calendar.ts</code> – availability, events, sync • <code>services/travel.ts</code> – flights, hotels, OAuth2 caching
6	Agent ↔ Workflow Bridge (Phase 5.4)	<ul style="list-style-type: none"> • <code>AgentWorkflowBridge</code> analyses intent + member tier → selects template • Injects workflow metadata into agent response • Executor falls back to classic tools if workflow creation fails

7	Knowledge-Base RAG Layer	<ul style="list-style-type: none"> • Postgres + pgvector tables <code>knowledge_documents</code> , <code>knowledge_chunks</code> • <code>ingest()</code> splits docs → embeds → batch UPSERT • <code>retrieve()</code> returns top <i>k</i> chunks by cosine sim (> 0.7) • Expose <code>searchKnowledgeBase</code> tool (intent-aware query expansion)
8	Notification Channels	<ul style="list-style-type: none"> • Slack webhook & Twilio SMS helpers use Secret Manager creds • <code>getActiveChannels()</code> selects channels per member tier / concierge status
9	Security & Observability	<ul style="list-style-type: none"> • Zero hard-coded secrets; all access audited via GCP logs • Bundle size ≤ 304 kB; TypeScript errors 0; Build ≤ 7 s • Real-time workflow & agent logs streamed to Firestore
10	Production Readiness Gate	<p>✓ 100 % Secret Manager coverage ✓ Workflow engine passes unit & integration tests ✓ End-to-end demo: <i>Member</i> → “Book jet” → <i>Stripe payment workflow</i> completes with Slack/Twilio notifications</p>

How to Use This Sheet

- **Bookmark:** treat each numbered phase as an anchor in your knowledge base.
- **Drill-down:** link the bullet items to code snippets or full SOPs as they evolve.
- **Traceability:** match “Phase x.y Complete” commits in your CHANGELOG back to the checklist items above for quick progress audits.

This condensed map lets you track Asteria’s full 2.0 integration path—no scrolling through 80 + pages required.