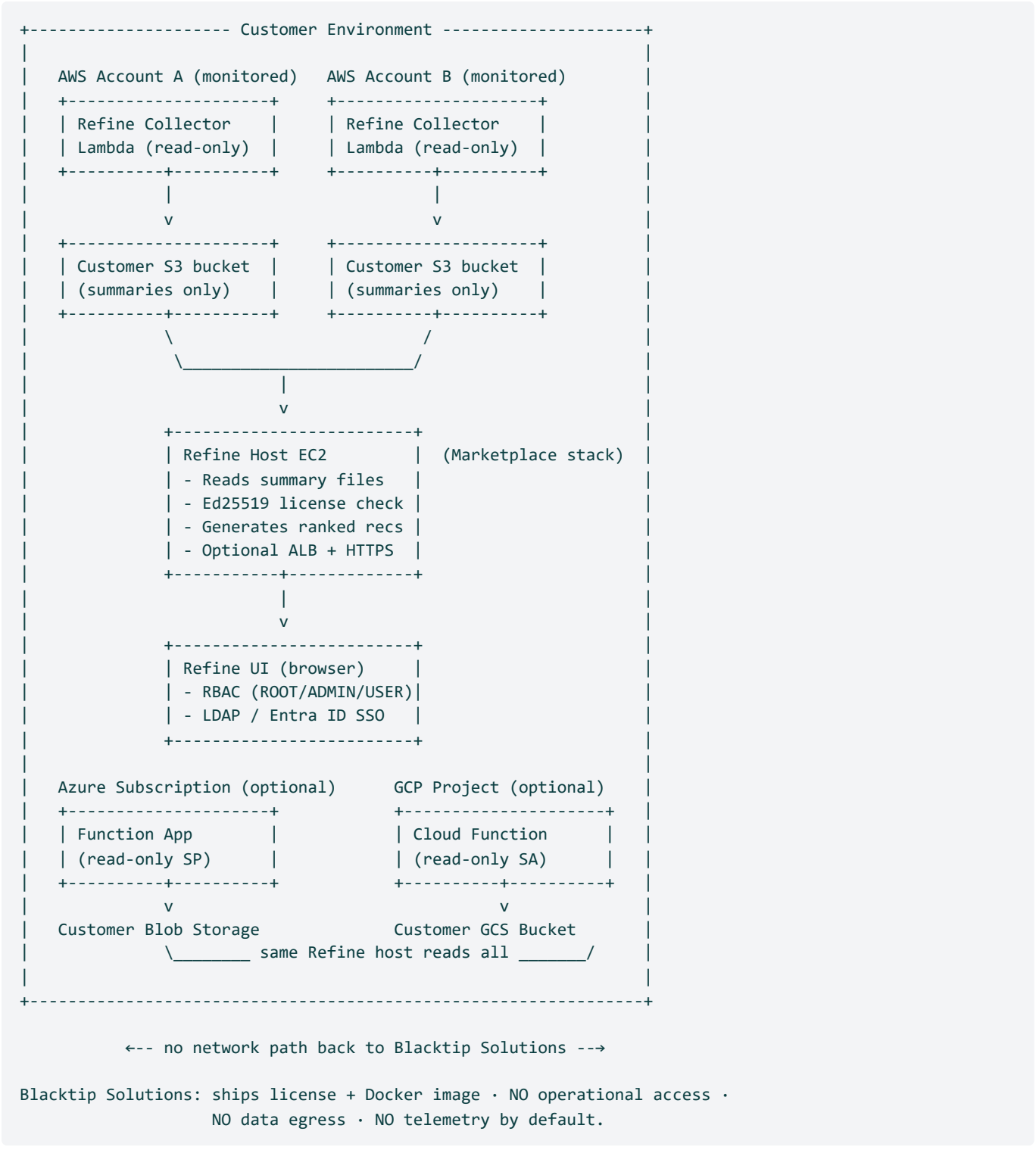
Refine — Architecture (AWS Marketplace Listing)

Purpose: the architecture diagram + one-page narrative that goes on the AWS Marketplace listing's *Architecture* tab. Designed to answer the questions a buyer's security/architecture team asks before approving the subscription.

The diagram (text rendering — PDF + PNG generated from this)



Narrative

Refine is self-hosted in the customer's own AWS account. The AWS Marketplace listing is a **Container product** that ships a CloudFormation template: subscribing puts the template in front of the customer, who launches it in a region

they choose. The stack provisions one EC2 instance (default `t3.medium`), an S3 license-delivery bucket, an IAM role scoped to that bucket + SSM (+ AWS License Manager for the Public/Contract path), and optionally an HTTPS Application Load Balancer.

Refine Docker image pulls directly from AWS Public ECR. On first boot the EC2 pulls `public.ecr.aws/blacktip/refine:latest` (or a pinned tag if the customer override the default) and starts the container immediately in a waiting-for-license loop. There's no waiting on Blacktip to upload an image bundle — the image is publicly hosted, signed, and versioned per the CFN's `RefineImageUri` parameter.

Auto-update is enabled by default via a Watchtower sidecar. A second container, `refine-watchtower` (image: `beatkind/watchtower:latest` — the maintained fork; `containrrr` is unmaintained and segfaults on modern Docker), runs on the EC2 and polls AWS Public ECR on a customer-tunable cron schedule (default: daily at 03:00 UTC). When the digest of the watched tag changes, Watchtower pulls the new image and gracefully restarts the Refine container with it. Updates are label-scoped — Watchtower only touches the container with the `com.centurylinklabs.watchtower.enable=true` label, never itself or any sidecar the customer adds. Customers who want strict version pinning can either set `EnableAutoUpdate=false` (Watchtower not deployed) or set `RefineImageUri` to a specific tag like `:v3.2.1` (digest never changes, so Watchtower finds nothing to update).

Health-check rollback is also enabled by default. The Refine container declares a Docker `HEALTHCHECK` that polls its own `/health` endpoint. A host-side systemd service, `refine-rollback-monitor.service`, polls every 30 seconds and tracks the digest of the last image to successfully pass the health check. If a new image (just pulled by Watchtower) stays unhealthy for more than 5 minutes, the monitor automatically rolls the container back to the last-known-good digest, pins the docker-compose file to a stable rollback tag (so Watchtower stops attempting updates until the operator investigates), and enters a 1-hour cooldown. All events are logged to `/var/log/refine-rollback-monitor.log` and `journalctl -fu refine-rollback-monitor.service`. The `EnableRollbackOnHealthFailure` CFN parameter (default `true`) toggles the monitor. With rollback enabled, Watchtower's `--cleanup` is disabled so the previous image stays on disk for instant rollback.

License is delivered by email; customer uploads it to their delivery bucket. After the stack reaches `CREATE_COMPLETE`, the customer submits a one-page BYOL license-request form at https://blacktip-ops.com/marketplace-license, providing their Marketplace customer ID, their `DeliveryBucketName` from the CFN Outputs, and the cloud accounts Refine will monitor. Blacktip Solutions generates an Ed25519-signed `refine.license` file (~500 bytes) and emails it to the customer with a one-line `aws s3 cp` command. The customer drops it into their own `DeliveryBucketName` from any shell that can reach their AWS account. A license-watcher daemon running on the EC2 polls the bucket every 60s, atomic-swaps the file into place when it lands, and restarts the container — which validates the license offline against an Ed25519 public key baked into the image. Typical time from `CREATE_COMPLETE` to first dashboard view: **10–30 minutes**, dominated by the human latency at Blacktip between form submission and license email.

> ***Why customer-upload instead of Blacktip-push?*** *Cross-partition IAM doesn't exist in AWS — a commercial Blacktip seller account cannot write to a customer's GovCloud S3 bucket. Granting Blacktip cross-account write access to commercial buckets would also require a bucket policy in the CFN that adds blast radius. Email + customer-upload works identically across both Marketplace channels (Commercial and GovCloud), preserves least-privilege on the customer bucket, and adds only ~5 seconds of customer-side effort.*

> ***GovCloud variant.*** *For air-gapped GovCloud customers who can't pull from public ECR, the GovCloud Marketplace CFN keeps the legacy "delivery ZIP" model — the customer receives a single ZIP containing the image tarball + license + day-2 scripts and uploads it to their bucket. Same security properties, same UX, just a heavier delivery payload that subsumes the license-only flow above.*

Cloud-account data collection is per-account, read-only, customer-owned. Once Refine is running, the customer adds each cloud account they want to monitor from the UI. For each:

- **AWS:** a per-account CloudFormation stack creates a Lambda (read-only across EC2/RDS/S3/etc.) and an S3 bucket the customer owns. The Lambda runs nightly via EventBridge and writes per-resource summary JSON to that bucket.
- **Azure:** a `curl | bash` script run in Azure Cloud Shell creates an Entra ID App Registration with the Reader role, a Storage Account + Blob container, and a daily Cost Management Export.
- **GCP:** a Terraform module creates a Service Account with `roles/viewer` + `roles/bigquery.dataViewer` and a BigQuery billing export.

Refine reads pre-summarized JSON from each customer-owned bucket. It does not call cloud-provider APIs from the host EC2 (with the narrow exception of cross-account `sts:AssumeRole` to verify connectivity at account-link time). All recommendations + commitment analysis + cost reconciliation are computed locally inside the customer's EC2.

Customer data never leaves the customer environment. Blacktip Solutions has no network path to the customer's AWS account, Azure tenant, or GCP project. The license file is the only artifact that flows from Blacktip → customer, and it carries no customer data — just a signed payload of `{customer name, expiry, allowed account list, installation ID}`.

Air-gap compatibility. GovCloud customers can deploy Refine in a fully air-gapped environment: the EC2 doesn't need outbound internet (the image is delivered via the S3 bucket; the license is validated offline against an Ed25519 public key baked into the binary). The Marketplace listing covers both Commercial and GovCloud channels.

Resources created (by stack)

Stack	Resources	Created when
Marketplace Host Stack (`refine-marketplace-cfn.yaml`)	1 EC2 + 1 EBS volume + 1 IAM role + 1 IAM instance profile + 1 Security Group + 1 S3 bucket + optional Elastic IP + optional ALB / ACM listener / target group / ALB SG	Customer launches from Marketplace
Per-Account Onboarding Stack (`customer-onboarding-v2.yaml`)	1 IAM role + 1 IAM managed policy + 1 Lambda + 1 Lambda execution role + 1 EventBridge schedule + 1 S3 bucket (or reuses customer-supplied)	Customer launches once per AWS account from inside Refine UI
GovCloud Onboarding Stack (`customer-onboarding-govcloud.yaml`)	Same as above, GovCloud-partition ARNs, Cost Explorer permissions dropped	Customer launches once per GovCloud account

See IAM_PERMISSIONS.md for the exact policy text on each role.

Data flow summary

1. **Collector Lambda** runs nightly in each monitored AWS account → writes summary JSON to the customer-owned S3 bucket.
2. **Refine Host EC2** polls each linked bucket every 6 hours (configurable) → ingests new summaries into its local SQLite/PostgreSQL store.

3. **Refine UI** (served by the EC2) renders dashboards, recommendations, and CLI scripts the customer's admins review and execute.
4. **No outbound traffic** from the host EC2 to Blacktip Solutions. The only outbound is the customer's own cloud SDKs talking to their own accounts.

Security & compliance posture

- **Encryption at rest:** all customer-owned buckets default to AES-256 (SSE-S3) or customer-managed KMS (configurable).
- **Encryption in transit:** Cloud SDK calls use TLS; the optional HTTPS ALB terminates TLS with the customer's ACM cert.
- **Auth:** HttpOnly JWT cookie auth into the UI; LDAP / Active Directory / Entra ID supported for SSO; RBAC with ROOT / ADMIN / USER roles + per-account-group scoping.
- **Audit trail:** every recommendation lifecycle event (Suggested → Accepted → Implemented → Verified) is logged with the user who took the action.
- **Tenant isolation:** Refine runs as a single-tenant container in each customer's environment — there is no multi-tenant Blacktip-side store.
- **Compliance inheritance:** Refine inherits the customer's account-level boundary (FedRAMP via GovCloud, HIPAA via AWS BAA, ISO 27001, SOC 2 via AWS, etc.). Blacktip does not hold standalone authorizations because there is no Blacktip-side data path to authorize.
