Escher - Multi-Cloud Operations Management Platform

Product Vision & Architecture Goals

Last Updated: October 2025 **Status**: Active Discussion - Defining Complete Scope

---

TABLE OF CONTENTS

QUICK START (Read These First)

1. Critical Architecture Principle - Our #1 non-negotiable
2. Product Overview - What is Escher?
3. Where Does Escher Run? - Choose your setup

CORE ARCHITECTURE

4. Where Does Escher Run? - Two private options
    - Option 1: Run on Your Laptop
    - Option 2: Extend to Your Cloud
5. Alert & Event System - Real-time + Scheduled
    - Real-Time Operational Alerts
    - Scheduled Scan Alerts (Morning Report)
6. Escher AI Server - Stateless brain architecture
7. RAG Architecture - Client-side + Server-side knowledge

BUSINESS FEATURES

8. User Personas - Manager vs Executor
9. Cloud Operations - What you can do
    - Resource Operations | Cost Management | Reports & Analytics
    - Automation & Scheduling | Security & Compliance
    - Multi-Account Management | Collaboration & Workflows
    - AI-Powered Operations | Alerts & Monitoring

IMPLEMENTATION DETAILS

10. Architecture Questions - Resolved vs Open
11. “Extend Me” Pattern - Pre-approved operations
12. Next Steps - What’s remaining
13. Alignment Check - Status summary

Critical Architecture Principle

> **ESCHER AI SERVER IS 100% STATELESS - REGARDLESS OF WHERE ESCHER RUNS**

Whether you **Run on Your Laptop** or **Extend to Your Cloud**:

What Escher AI Server Does	What It Does NOT Do
Receives requests	Stores user data
Processes with RAG	Stores cloud estate
Returns responses	Stores credentials
Forgets everything after	Stores chat history

Privacy Guarantee: User's cloud estate and credentials NEVER leave user's control.

↑ Back to Top

Product Overview

Escher is a Multi-Cloud Operations AI Platform that enables users to manage cloud operations across **AWS, Azure, and GCP** through a unified conversational interface.

Core Philosophy

```
Multi-Cloud Support    Single platform for AWS/Azure/GCP
Conversational         Natural language for all ops
Unified Experience     Consistent across clouds
User-Controlled State  Your data stays with you
AI-Powered             Smart recommendations & automation
Flexible Deployment    Local-only or cloud-extended
```

↑ Back to Top

Where Does Escher Run?

Escher offers **two ways to run** - both are **100% private** with your data in YOUR control:

Privacy Parity - Both Options Are Equally Private

```
        BOTH OPTIONS: YOUR DATA STAYS WITH YOU


  Option 1: Run on Your Laptop
   Primary Storage: Your Laptop (Vector Store)
   Backup: YOUR S3/Blob/GCS (disaster recovery)
   Data Owner: YOU

  Option 2: Extend to Your Cloud
   Primary Storage: YOUR S3/Blob/GCS (Vector Store)
   Also Acts As: Backup (cloud-native durability)
   Data Owner: YOU (not Escher!)

  IN BOTH CASES:
   Escher AI Server stores NOTHING
   Your credentials stay with YOU
   Zero trust architecture
```

Quick Comparison

Feature	Run on Your Laptop	Extend to Your Cloud
Privacy	100% Private	100% Private (YOUR cloud)
Primary Storage	Laptop Vector Store	YOUR S3/Blob/GCS Vector Store
S3/Blob/GCS Role	Backup only	Primary storage + backup
Target Users	Individuals, simple ops	Teams, 24/7 requirements
Laptop Requirement	Must stay online	Can be offline
Real-Time Alerts	Requires always-on laptop	Works 24/7
Scheduled Jobs	Laptop must be online	Runs in cloud reliably
Cloud Costs	$0 compute (only backup storage)	EventBridge + Fargate + S3
Setup Complexity	Simple	Moderate
Best For	Exploration, dev work	Production, automation

Option 1: Run on Your Laptop (Beta / Lightweight Users)

Architecture Diagram

```
Physical Laptop (Tauri App)
  React Frontend
  Rust Backend
  Local RAG (Vector Store)
  Local Credentials
  Periodic Backup → S3/Blob/GCS (hourly)
```

Key Points

- **Zero cloud compute costs** (only storage for backups)
- **Complete local control** of all data
- **Simple setup** - install and go
- **Laptop must stay online** for scheduled jobs and alerts
- **No cross-device access** to state

Data Flow (Local Only)

```
User Query → Physical Laptop searches RAG → Sends query + context to AI Server
→ AI Server processes → Returns response
→ Physical Laptop executes locally → Stores results in local RAG
→ Periodic backup to S3/Blob/GCS (hourly)
```

↑ Back to Top

Option 2: Extend to Your Cloud (Main Release / Power Users)

Architecture Diagram

```
Physical Laptop (Tauri App) ←→ Escher AI Server (Stateless Brain)
        ↓↑                              ↑
Extend My Laptop (User's Cloud)          |
        ↓                                |
Cloud Schedulers + Execution + State ←--
```

Components in User's Cloud

Cloud Provider	Scheduler	Execution	State Storage	Credentials
AWS	EventBridge	Fargate	S3	SSM Parameter Store

Cloud Provider	Scheduler	Execution	State Storage	Credentials
Azure	Logic Apps	Container Instances	Blob Storage	Key Vault
GCP	Cloud Scheduler	Cloud Run	Cloud Storage	Secret Manager

Setup Process (5 Steps)

1. User chooses "Extend to Cloud" from physical laptop
2. Select cloud provider (AWS, Azure, or GCP)
3. Physical laptop provisions infrastructure in **user's account**:
   - Deploys Escher-provided container image
   - Creates scheduler
   - Creates state storage
   - Creates credential storage
4. User installs cloud credentials (same as local laptop)
5. Physical laptop becomes thin client

Execution Model

Component	Handles
Physical Laptop	Interactive ops, ad-hoc queries, real-time tasks
Extend My Laptop	Scheduled ops, long-running tasks, automation
Event-Based Lifecycle	Starts on-demand, stops when idle (cost optimization)

Data Flows (Extend My Laptop)

Interactive Query:

```
User Query → Physical Laptop searches local RAG → Sends to AI Server
→ AI Server processes → Returns response
→ Physical Laptop → Extend My Laptop executes
→ Results stored in S3/Blob/GCS RAG
→ Physical Laptop syncs latest state
```

Scheduled Execution:

```
Scheduler triggers → Extend My Laptop starts
→ Loads RAG from S3/Blob/GCS
→ Sends query + context to AI Server
→ AI Server returns execution plan
→ Extends My Laptop executes → Cloud APIs
→ Stores results in RAG → Uploads to S3/Blob/GCS
→ Extend My Laptop shuts down
```

Multi-Cloud Management

- Extend My Laptop (e.g., on AWS) manages **all clouds** (AWS + Azure + GCP)
- User installs credentials for all clouds in credential store
- Example: AWS Fargate with AWS + Azure + GCP credentials in SSM

Key Benefits

- **24/7 operations** without laptop online
- **Scheduled jobs** run reliably
- **Long-running operations** don't block laptop
- **Cross-device access** to state
- **Event-based compute** = lower costs than always-on

↑ Back to Top

User Choice

Users can switch between models: - Start with **Local Only** for simplicity - Upgrade to **Extend My Laptop** when they need scheduling/automation - Downgrade back to **Local Only** anytime (Extend My Laptop infrastructure can be destroyed)

↑ Back to Top

Alert & Event Handling Architecture

Escher provides **two types of alert systems** - "Sensors" that continuously monitor the cloud environment and alert the "Brain" (AI Server) when action is needed.

Quick Comparison

Type	Real-Time Operational	Scheduled Scan
Trigger	Critical events happen	Daily at 2am
Purpose	Immediate action	Proactive insights
Examples	DB down, S3 public	Cost trends, idle VMs
Delivery	Push notifications	Morning report banner
Response Time	Seconds	Next day
Auto-Fix	Yes (pre-approved)	Yes (1-click buttons)

---

1. Real-Time Operational Alerts (Can't Wait - Immediate Action Required)

Purpose Immediate notification and action for critical events that require urgent attention.

Target Events by Severity

Severity	Examples	Response Time
CRITICAL	DB down, S3 public (PII), budget exceeded 200%	Immediate
HIGH	Performance degradation, cost spike, compliance violation	< 5 minutes
MEDIUM	Resource warnings, capacity approaching limits	< 15 minutes
INFO	Informational events	Aggregated in morning report

Setup Process (During Extend My Laptop Installation) **Step 1: Add Escher Listener to Source of Truth** User grants permission to add event listeners: - **AWS**: CloudWatch Alarms → EventBridge → Extend My Laptop - **Azure**: Azure Monitor Alerts → Event Grid → Extend My Laptop - **GCP**: Cloud Monitoring → Pub/Sub → Extend My Laptop

Step 2: Pre-Approve Auto-Remediation (Setup Wizard)

```
 Automatically make public S3 buckets private
 Automatically stop idle instances after 2 hours
 Automatically enable encryption on unencrypted volumes
 Automatically scale up resources approaching capacity
 Automatically restart failed services
 Automatically rollback failed deployments
```

- User can modify these settings anytime
- Each action logs to immutable audit trail

Real-Time Alert Flow

```
Critical Event Occurs (e.g., S3 bucket made public)
↓
Cloud-Native Alert detects at source of truth
```

```
↓
Event published to EventBridge/Event Grid/Pub Sub
↓
Extend My Laptop wakes up (Fargate/Container Instance/Cloud Run)
↓
Loads RAG from S3/Blob/GCS:
  Estate: Which bucket? Production or dev? Contains PII?
  Alert Rules: User's configured severity thresholds
  Previous Incidents: Similar alerts? How resolved?
  Auto-Remediation Settings: Is "make bucket private" pre-approved?
↓
Normalize event to unified schema
↓
Send unified event + context to AI Server (Escher Brain)
↓
AI Server analyzes:
  Severity Assessment: CRITICAL (PII exposed publicly)
  Root Cause: Security group rule changed by john@company.com
  First Aid Recommendation: Make bucket private immediately
  Impact Assessment: ~1.2M customer records exposed
  Playbook: Step-by-step remediation
  Risk: GDPR violation if not fixed within 1 hour
↓
Decision Point - Is auto-remediation pre-approved?
  YES → Execute immediately
      Run: aws s3api put-bucket-acl --bucket my-data --acl private
      Verify: Bucket now private
      Store result in RAG
      Prepare notification: "CRITICAL alert auto-resolved"
  NO → Request approval
      Store alert in RAG
      Prepare notification: "CRITICAL alert requires approval"
↓
Notification via cloud-native services:
   CRITICAL: Email + SMS + Slack/PagerDuty
   HIGH: Email + Slack only
   MEDIUM: In-app notification banner
↓
Store complete alert record in RAG (immutable)
↓
Extend My Laptop shuts down
```

User Experience - Alert Banners **CRITICAL Alert Banner** (Red background, top of screen):

```
 CRITICAL ALERT - AUTO-RESOLVED 2 minutes ago

S3 bucket 'my-data' made public at 2:34 AM
 Escher automatically made bucket private

Details:
• 1.2M customer records were exposed for 2 minutes
• Bucket made public by user john@company.com
• Auto-remediation executed: aws s3api put-bucket-acl
• Verification: Bucket now private

 Recommended Next Steps:
1. Review bucket policy to prevent future occurrences
2. Notify security team about exposure
3. Check CloudTrail for access during exposure window

[View Full Timeline] [Create Prevention Playbook]
[Notify Security Team] [Acknowledge]
```

HIGH Alert Banner (Orange background):

```
 HIGH ALERT - Requires Your Approval

5 EC2 instances idle for 7 days (CPU < 5%)
Potential savings: $203/month

Instances:
• i-abc123 (dev) - t3.large - $52/month
• i-def456 (dev) - t3.large - $52/month
• i-ghi789 (dev) - t3.medium - $35/month

 Ask: "Is it safe to stop these instances?"
 Ask: "What are they used for?"

[Stop All] [Stop Dev Only] [Remind Me Tomorrow]
[View Details] [Dismiss]
```

Conversational Follow-Up User can ask questions about any alert:

User: "Show me the full timeline of the S3 bucket exposure"

AI Response:

```
Here's the complete timeline:
```

```
2:32:15 AM - Bucket policy changed by john@company.com
2:32:15 AM - CloudWatch detected public access enabled
2:32:20 AM - EventBridge published event
2:32:22 AM - Extend My Laptop started
2:32:25 AM - RAG loaded: Bucket contains PII
2:32:26 AM - Event sent to AI Server for analysis
2:32:28 AM - AI Server: Severity = CRITICAL (PII exposed)
2:32:29 AM - Auto-remediation check: APPROVED
2:32:30 AM - Executed: aws s3api put-bucket-acl
2:32:32 AM - Verification: Bucket now private
2:32:33 AM - Notifications sent: Email + SMS + Slack
2:34:45 AM - Extend My Laptop shutdown

**Total exposure time: 2 minutes 17 seconds**

CloudTrail Analysis:
• No external access detected during exposure window
• Only john@company.com accessed bucket (authorized user)
• Risk: LOW (exposure was brief, no unauthorized access)

Would you like me to:
1. Create a prevention playbook?
2. Send security incident report?
3. Review IAM policies for john@company.com?
```

Unified Event Schema (Cross-Cloud Normalization)

```typescript
interface UnifiedEvent {
  event_id: string;
  event_type: string; // "s3_bucket_public", "vm_stopped", "cost_spike"
  severity: "CRITICAL" | "HIGH" | "MEDIUM" | "INFO";
  cloud_provider: "aws" | "azure" | "gcp";
  account_id: string;
  region: string;
  resource: {
    type: string; // "s3_bucket", "ec2_instance", etc.
    id: string;
    name: string;
    tags: Record<string, string>;
    metadata: Record<string, any>;
  };
  context: {
    environment?: "production" | "staging" | "dev";
    data_classification?: "PII" | "confidential" | "public";
    cost_impact?: number;
    affected_users?: number;
  };
  timestamp: string; // ISO-8601
  raw_event: any; // Original cloud-specific event
}
```

Local Only Limitation

- Real-time alerts require **Extend My Laptop** for 24/7 monitoring
- OR laptop must remain **always-on** for Local Only mode
- Local Only users with always-on can receive alerts via polling (every 1 minute for CRITICAL)

↑ Back to Top

---

2. Scheduled Scan Alerts (Can Wait - Interactive Morning Report)

Purpose Proactive monitoring, optimization suggestions, and aggregated insights delivered daily.

Scan Schedule Daily at 2am (same as cost/audit sync), user-configurable.

Scans Performed

Category	What It Checks
Cost Analysis	Spending trends, budget tracking, anomaly detection, waste
Security Posture	Compliance (CIS, SOC2), policy violations, encryption
Resource Optimization	Idle resources, rightsizing, over-provisioned VMs
Operational Health	Backup status, snapshot age, service availability
Performance	Resource utilization, bottlenecks, capacity planning

Scheduled Scan Flow

```
2am: Scheduler triggers daily scan
↓
Extend My Laptop wakes up (or Physical Laptop if online)
↓
Execute scans across all clouds in parallel:
  AWS Cost Explorer API (yesterday's costs)
  Azure Cost Management API (spending trends)
  GCP Billing API (cost breakdown)
  Security scans (public resources, encryption)
  Performance metrics (CloudWatch/Azure Monitor/GCP Monitoring)
  Resource inventory (idle instances, old snapshots)
↓
Load RAG from S3/Blob/GCS:
  Estate: Current inventory for comparison
  Previous Scans: Yesterday's results for deltas
  Alert Rules: User's customized thresholds
  Immutable Reports: Historical data for trends
  Report Templates: User's customized preferences
↓
Send scan results + context to AI Server
↓
AI Server analyzes:
  Aggregate findings (group similar issues)
  Calculate deltas (what changed?)
  Prioritize by severity and cost impact
  Generate actionable recommendations (1-click fixes)
  Create interactive morning report
  Format for conversational Q&A
↓
Store report in Immutable Reports (permanent storage)
↓
When user opens laptop:
Display interactive morning report banner
```

Interactive Morning Report (Better Than Email)

```
 Good Morning Report - March 15, 2025
Generated at 2:00 AM | Data current as of 11:59 PM yesterday



 CRITICAL ALERTS (Last 24h):
[Red background, requires immediate attention]
```

- `Production RDS exceeded 90% storage capacity`
  `Auto-scaled from 100GB → 150GB  (+$7.50/month)`
  `Root cause: Log retention increased from 7 to 30 days`

- `Security group sg-abc123 opened port 22 to 0.0.0.0/0`
  `Auto-remediated: Restricted to company IP`
  `Alert sent to: security@company.com`

```
  COST SUMMARY:
Yesterday: $1,247 | This Month (MTD): $18,705 | Budget: $25,000

+$186 (+17.5%) vs previous day   Above your threshold ($100)
+$2,450 (+15%) vs last month   Trending higher

Top Cost Drivers (Yesterday):
1. EC2 Instances: $567 (+$144 from 3 new m5.2xlarge in production)
2. RDS: $289 (+$25 from storage auto-scaling)
3. S3 Storage: $156 (+$12 from new backups)

  Potential Savings Identified: $412/month



  TOP CHANGES (Requires Your Attention):

1.   3 new EC2 instances launched in production
   • Instance Type: m5.2xlarge (8 vCPU, 32GB RAM)
   • Cost Impact: +$144/day ($4,320/month)
   • Launched by: john@company.com at 10:34 AM
   • Purpose (from tags): "web-tier-scaling"

     Ask: "Why were these instances created?"
     Ask: "Are these still needed?"
     Ask: "Can we use spot instances instead?"

2.   RDS snapshot storage increased 25GB
   • New Size: 125GB (+25GB from yesterday)
   • Cost Impact: +$2.50/day
   • Reason: Daily snapshots accumulating

     Ask: "Can we reduce snapshot retention to 7 days?"
     1-Click: Reduce retention to 7 days (saves $18/month)



  SECURITY & COMPLIANCE:

  GOOD NEWS:
• No public S3 buckets detected
• All production RDS instances encrypted
• IAM password policy compliant

  ATTENTION REQUIRED:
```

```
• 2 unencrypted EBS volumes detected
    Environment: dev-environment
    Volumes: vol-abc123 (50GB), vol-def456 (100GB)
    Risk: Medium (dev data, may contain test PII)

   Ask: "Show me these volumes"
   Ask: "What data is on them?"
   1-Click: Enable encryption



 OPTIMIZATION OPPORTUNITIES:

 5 idle EC2 instances detected (Saves: $203/month)
• Criteria: CPU < 5% for 7 consecutive days
• Instances: i-abc123, i-def456, i-ghi789, i-jkl012, i-mno345
• Environment: dev (3), staging (2)

 Ask: "Which instances are idle?"
 Ask: "What are they used for?"
 Ask: "Is it safe to stop them?"
 Stop All |  Stop Dev Only |  Remind Me Tomorrow

 3 over-provisioned VMs (Saves: $142/month)
• Criteria: Average utilization < 30% over 30 days
• Recommendations:
  - vm-web-01: m5.2xlarge → m5.xlarge (save $72/month)
  - vm-api-02: m5.2xlarge → m5.xlarge (save $72/month)
  - vm-db-staging: db.m5.large → db.t3.medium (save $28/month)

 Ask: "Show me utilization graphs"
 Ask: "Will downsizing impact performance?"
 View Rightsizing Recommendations

 67GB of old EBS snapshots (Saves: $6.70/month)
• Criteria: Snapshots older than 90 days, source volume deleted
• Count: 13 snapshots

 Ask: "Are these snapshots safe to delete?"
 Delete Old Snapshots



 OPERATIONAL HEALTH:

 ALL SYSTEMS OPERATIONAL:
```

```
• All backups successful (42 RDS snapshots, 18 EC2 AMIs)
• No failed deployments
• Service availability: 100% uptime



  ASK ME ANYTHING ABOUT THIS REPORT:

Examples:
• "Why did spending increase 17%?"
• "Show me the 5 idle instances"
• "Which optimizations should I do first?"
• "Create a playbook to fix all security issues"
• "Compare costs with last week"
• "Show me cost breakdown by environment"

Type your question below ↓
```

Interactive Q&A on Report User can ask questions conversationally. The AI Server has complete report context:

User: "Why did spending increase 17%?"

AI Response:

```
Spending increased by $186 (17.5%) yesterday due to three factors:

1. **3 new EC2 instances in production** (+$144/day = 77% of increase)
   - Instance IDs: i-0a1b2c3, i-4d5e6f7, i-8g9h0i1
   - Type: m5.2xlarge (8 vCPU, 32GB RAM) - $0.384/hour each
   - Launched: Yesterday 10:34 AM by john@company.com
   - Purpose (from tags): "web-tier-scaling" for Black Friday prep
   - Running time: 13.5 hours = $46.80 total (daily cost will be $144)

2. **RDS storage auto-scaling** (+$25/day = 13% of increase)
   - Database: prod-db-01 scaled from 100GB → 150GB
   - Trigger: 90% capacity threshold reached
   - Root cause: Log retention changed from 7 → 30 days
   - Cost: $0.115/GB-month = $2.50/day additional

3. **Increased S3 storage** (+$12/day = 6% of increase)
   - New backup policy storing application logs to S3
   - Data ingested: 960GB over last 24 hours
   - Cost: $0.023/GB-month = $12/day

  **To reduce costs**:
- If Black Friday prep complete, stop 3 EC2 instances (saves $144/day)
- Reduce RDS log retention to 14 days (saves $10/day)
- Enable S3 Intelligent-Tiering for logs (saves ~30% = $3.60/day)

**Total potential daily savings: $157.60 ($4,728/month)**

Would you like me to:
1. Check if EC2 instances are still needed?
2. Create a playbook to optimize these costs?
3. Schedule automatic shutdown after business hours?
```

Report Customization (Template Models) **Default Template** (Escher provides): - Comprehensive morning report (shown above) - Includes: Critical alerts, cost, security, optimization, operational health - Format: Concise, scannable, actionable - Q&A enabled by default

User Customization Options:

```
Report Settings:
  Sections to Include:
     Critical Alerts (always included)
     Cost Summary
     Top Changes
     Security & Compliance
     Optimization Opportunities
     Operational Health
     Performance Metrics (optional, adds graphs)

  Thresholds:
   • Cost increase alert: >$100 or >10% (customizable)
   • Idle instance: <5% CPU for 7 days (customizable)
   • Old snapshot: >90 days (customizable)

  Focus Areas:
     Balanced (default - equal weight to all areas)
     Cost-Focused (emphasize savings)
     Security-Focused (emphasize compliance)
     Operational-Focused (emphasize uptime)

  Format:
     Detailed (default - ~50 lines)
     Compact (summary only, ~20 lines)
     Executive (high-level + top 3 issues, ~15 lines)

  Severity Customization:
    Define what's "CRITICAL" for your organization:
      Any public S3 bucket
      Budget overrun >$1000
      Any unencrypted volume (default: only production)
      Production database >85% capacity
```

Template Storage: Stored in RAG (Alerts & Events collection), synced across devices.

↑ Back to Top

Escher AI Server Architecture

Stateless Processing Engine

The Escher AI Server is a **pure stateless processing engine** - it receives requests, processes them using RAG, and returns responses without storing any user data.

Server Capabilities

```
                ESCHER AI SERVER (STATELESS)

   Built-in RAG Knowledge Base:
   • Playbook Library (AWS, Azure, GCP operations)
   • CLI Command Database (complete reference)
   • Best Practices (architecture, security, cost)
   • Multi-Cloud Operations (equivalents, migrations)

   AI Processing:
   • Natural language understanding
   • Context-aware response generation
   • Operation planning and sequencing
   • Playbook generation and customization
   • Anomaly detection and recommendations
```

Data Flow Details

1. Interactive Query Flow

```
User: "Show me all running EC2 instances in us-east-1"

Physical/Extend My Laptop (Client-Side):
  Search local RAG:
      Cloud Estate Inventory: Check for EC2 in us-east-1
      Chat History: Previous conversation context
      Executed Operations: Recent EC2-related operations
  Prepare context from RAG results

→ Send to Escher AI Server:
    Query: "Show me all running EC2 instances in us-east-1"
    Context: Last 5 messages + relevant estate info

AI Server Processing:
  Parse intent: List resources
  Identify scope: EC2, us-east-1, running state
  RAG lookup: EC2 list commands/APIs
  Analyze context: Recent activities
  Generate response type: Information query (not execution)

→ Return to Physical/Extend My Laptop:
    Response Type: "information"
    Operation: { type: "list_ec2", filters: {...} }
    Suggested Display: Table format

Physical/Extend My Laptop:
  Query cloud APIs locally
  Display results
  Store interaction in RAG (chat history)
```

Key Points: - **Client searches RAG first**: Estate, chat history, operations - **Context sent to AI Server**: Previous chat + relevant RAG results (not full snapshot) - **AI Server processes transiently**: Forgets everything after response - **Privacy preserved**: AI Server never stores cloud estate or credentials

2. Execution Flow

```
User: "Stop all dev EC2 instances in us-east-1"

Physical/Extend My Laptop (Client-Side):
  Search local RAG:
     Cloud Estate Inventory: Find all dev EC2 in us-east-1
     Chat History: Full conversation history (for LLM context)
     Executed Operations: Recent EC2 operations
  Prepare context from RAG results

→ Send to Escher AI Server:
    Query: "Stop all dev EC2 instances in us-east-1"
    Context:
        Full chat history
        Dev instances found: 5 instances (i-xxx, i-yyy, ...)
        Recent operations
```

```
AI Server:
  Intent: Stop resources
  Scope: EC2, us-east-1, tag=dev
  Context understanding: Full conversation allows LLM to understand intent
  RAG lookup: Stop EC2 playbook
  Safety check: High-risk operation (stops multiple instances)
  Generate execution plan

→ Return to Physical/Extend My Laptop:
    Response Type: "execution"
    Execution Plan:
        Step 1: List EC2 instances with tag=dev
        Step 2: Confirm instances with user
        Step 3: Stop instances (aws ec2 stop-instances...)
    Estimated Impact: 5 instances affected

Physical/Extend My Laptop Rust Execution Engine:
  Display execution plan to user
  Request user confirmation
  Execute playbook steps
  Store results in RAG (Executed Operations)
  Store audit log in RAG (Immutable Reports)
```

3. Scheduled Job Flow

```
Scheduled Job: "Stop all dev VMs at 8pm daily"

EventBridge/Cloud Scheduler → Extend My Laptop
↓
Extend My Laptop (Client-Side):
  Load RAG from S3/Blob/GCS:
      Cloud Estate: Find all dev VMs across clouds
      Chat History: Schedule creation context
      Executed Operations: Previous executions
  Prepare context from RAG results

→ Send to Escher AI Server:
    Query: "Execute scheduled job: Stop all dev VMs at 8pm"
    Context:
        Schedule creation chat history
        Dev VMs found: 15 VMs (5 AWS, 6 Azure, 4 GCP)
        Last execution: Yesterday, 15 VMs stopped successfully

AI Server:
  Intent: Execute scheduled operation
  RAG lookup: Stop VMs playbook (multi-cloud)
  Context understanding: Routine daily operation
  Generate execution plan for all clouds
  Return structured operations

→ Return to Extend My Laptop:
    Response Type: "execution"
    Multi-Cloud Operations:
        AWS: aws ec2 stop-instances --instance-ids...
        Azure: az vm stop --resource-group dev...
        GCP: gcloud compute instances stop...
    Expected Results: 15 VMs stopped

Extend My Laptop Rust Execution Engine:
  Execute multi-cloud operations in parallel
  Store results in RAG (Executed Operations)
  Store audit logs in RAG (Immutable Reports)
  Upload RAG to S3/Blob/GCS
  Shutdown (event-based lifecycle)
```

4. Playbook Generation Flow

```
User: "Create a disaster recovery playbook for my production environment"

Physical Laptop (Client-Side):
  Search local RAG:
      Cloud Estate: All production resources
      Chat History: Full conversation history
      Executed Operations: Existing backups, snapshots
  Prepare context from RAG results

→ Send to Escher AI Server:
    Query: "Create DR playbook for production"
    Context:
        Full chat history
        Production inventory: RDS, EC2, S3, ALB
        Existing DR: RDS backups enabled, no S3 replication

AI Server:
  Intent: Generate playbook
  Context understanding: User needs DR playbook, gaps identified
  RAG lookup: DR best practices, backup strategies
  Analyze context: Identify critical resources and missing DR
  Generate custom playbook addressing gaps

→ Return to Physical Laptop:
    Response Type: "playbook"
    Playbook Name: "Production DR Playbook"
    Steps:
       Step 1: Enable RDS snapshots  Already enabled
       Step 2: Replicate S3 buckets  Missing, critical
       Step 3: Create EC2 AMIs  Missing, recommended
       Step 4: Configure cross-region ALB
       Step 5: Set up Route53 failover
       Step 6: Test failover monthly
    Estimated Cost: $X/month
    Compliance: Meets RTO=4h, RPO=1h

Physical Laptop:
  Display playbook to user
  User reviews/modifies playbook
  Store playbook in RAG (Executed Operations)
  User can execute on-demand or schedule it
```

Playbook Management: - **Escher Playbook Library**: Server provides pre-built playbooks via RAG - **User Playbooks**: Users can create/modify and store locally or in cloud - **Playbook Override**: User playbooks override Escher-provided playbooks - **Playbook Storage**: Local (Local Only) or S3/Blob/GCS (Extend My Laptop)

↑ Back to Top

RAG Architecture

Client-Side RAG (Physical/Extend My Laptop - Rust)

Local Knowledge Base Collections:

```
              CLIENT-SIDE RAG (5 COLLECTIONS)

  1. Cloud Estate Inventory
     Current resource inventory across all clouds

  2. Chat History
     Conversational history with AI

  3. Executed Operations
     History of operations executed

  4. Immutable Reports
     Cost reports, audit logs, compliance reports
```

```
   (to avoid repeated API calls)

5. Alerts & Events
   Alert rules, alert history, scan results,
   auto-remediation settings, report templates,
   morning reports
```

Storage: - **Local Only**: Local vector store on laptop + periodic backup snapshots to S3/Blob/GCS (hourly) - **Extend My Laptop**: S3/Blob/GCS vector store (single source of truth)

Immutable Reports Collection: - **Cost Reports**: Daily snapshots from AWS Cost Explorer, Azure Cost Management, GCP Billing - Prevents repeated API calls (reduces cost) - Historical cost analysis without hitting cloud APIs - Daily sync scheduled (Manager gets updated data automatically) - **Audit Logs**: Immutable log of all operations - Daily sync ensures complete audit trail - Cannot be modified after creation (compliance requirement) - Stored in vector store for fast retrieval and AI analysis - **Compliance Reports**: Security scans, policy violations, CIS benchmarks - Generated on-demand or scheduled - Stored for historical comparison

Daily Sync for Manager Persona: - **Scheduled Job**: Daily sync at 2am (user-configurable) - **Syncs**: - Cost data (AWS Cost Explorer, Azure Cost Management, GCP Billing APIs) - Audit logs (all operations executed) - Compliance reports (CIS benchmarks, policy violations) - Security scans (public resources, encryption, IAM) - Idle resource detection (unused instances, volumes, snapshots) - Performance monitoring (resource utilization, bottlenecks) - Interactive morning report generation (aggregated insights with Q&A) - **Benefit**: Manager wakes up to fresh data and actionable morning report - **Cost Optimization**: Single daily API call instead of repeated queries

Server-Side RAG (Escher AI Server)

```
              SERVER-SIDE RAG (GLOBAL KNOWLEDGE)

 • All cloud provider APIs, CLI commands
 • Playbooks for common operations
 • Best practices and security advisories
 • Multi-cloud equivalents and migration patterns

 Purpose: Provides cloud operations expertise
 Updates: Escher continuously updates with new features
```

Combined Power

```
Client RAG (User Context)  +  Server RAG (Cloud Expertise)
        ↓                              ↓
   Estate Inventory              Playbook Library
   Chat History                  CLI Commands
   Operations History            Best Practices
   Immutable Reports             Multi-Cloud Knowledge
   Alerts & Events               Security Advisories
        ↓                              ↓
              → Intelligent, Context-Aware ←
                     Multi-Cloud Operations
```

Version 2 Release: Central Immutable Reports

Beta/V1 Release (Current): - Immutable reports stored in user's control: - **Local Only**: Local vector store on physical laptop - **Extend My Laptop**: Vector store in S3/Blob/GCS (user's cloud) - Privacy-first: No reports leave user's environment

V2 Release (Future): - **Optional**: User can choose to sync immutable reports to Escher-managed central location - **Benefits**: - Cross-device access to reports - Team collaboration on reports - Longer retention without user cloud costs - Advanced analytics across historical reports - **User Choice**: Opt-in only, users control what reports are synced - **Privacy**: Reports are encrypted, user controls access - **Migration**: Users can migrate from V1 (local) to V2 (central) anytime

Privacy & Security Model

What AI Server Receives: - Natural language queries - Cloud estate snapshots (for context - processed transiently, not stored) - Operation results (for generating recommendations - processed transiently)

What AI Server NEVER Receives: - Cloud credentials (AWS keys, Azure service principals, GCP service accounts) - Sensitive data from cloud resources (database contents, file contents, secrets) - User identity information

What AI Server NEVER Stores: - User data - Cloud estate information - Chat history - Operation history - Any user-specific state

Processing Model:

```
Request arrives → Load from RAG → Process with LLM → Generate response → Return → Forget eve
```

Every request is independent. The AI Server has no memory between requests.

↑ Back to Top

User Personas

1. Manager

- Reviews reports and analytics
- Sets budgets and cost policies
- Approves high-risk operations
- Schedules automated operations
- Monitors team activities
- Manages compliance requirements

2. Executor (Operations Engineer)

- Runs day-to-day operations conversationally
- Follows organizational policies
- Executes pre-approved playbooks
- **"Extend Me" Pattern**: Executes operations within manager-defined boundaries

↑ Back to Top

Cloud Management Operations

Operation Categories

```
                  ESCHER CLOUD OPERATIONS

  1. Resource Operations       Day-to-day tasks
  2. Cost Management           Real-time cost analysis
  3. Reports & Analytics       Infrastructure, cost, security
  4. Automation & Scheduling   Nightly shutdowns, backups
  5. Security & Compliance     Scanning, IAM, encryption
  6. Multi-Account Management Org-level visibility
  7. Collaboration & Workflows Approval, change tracking
  8. AI-Powered Operations     Conversational, smart
  9. Alerts & Monitoring       Real-time + scheduled
```

1. Resource Operations (Day-to-day)

Start/Stop/Restart resources: - **AWS**: EC2, RDS, Lambda - **Azure**: VMs, SQL Database, Functions - **GCP**: Compute Engine, Cloud SQL, Cloud Functions

Other Operations: - Resize/Scale (instance types, storage, compute) - Create/Delete resources - Configure (firewall rules, tags, settings) - Backup/Restore

- Snapshot management

Execution: Client-side with user credentials (cloud-specific SDKs/APIs)

---

2. Cost Management

- `Real-time cost analysis (current spend, trends)`
- `Budget tracking and alerts`
- `Cost optimization recommendations (rightsizing, unused)`
- `Resource utilization tracking`
- `Reserved instance analysis`
- `Savings plan recommendations`
- `Waste detection (idle resources, unattached volumes)`

Implementation: - Historical cost data stored in immutable reports collection - Daily snapshots from AWS Cost Explorer, Azure Cost Management, GCP Billing APIs - Budget alerts: Periodic (evaluated during daily sync, default 2am)

---

3. Reports & Analytics

Report Type	Contents
Infrastructure	Inventory, configuration, topology
Cost	By service, account, region, tag
Security	Vulnerabilities, policy violations, compliance
Performance	Resource utilization, bottlenecks
Change History	Audit trail of operations
Compliance	CIS benchmarks, custom policies

Implementation: - **Generation**: On-demand (user requests) and scheduled (daily sync at 2am) - **Storage**: Immutable reports collection in vector store (local or S3/Blob/GCS) - **Export Formats**: PDF, CSV, Excel, JSON (AI generates in requested format) - **Retention Policy**: User-configurable (default: 90 days for cost, 1 year for audit logs)

---

4. Automation & Scheduling

- `Scheduled Operations (nightly shutdowns, weekend starts)`

- `Automated Remediation (auto-stop idle, delete snapshots)`
- `Backup Schedules (automated backup execution)`
- `Compliance Enforcement (auto-tag, enforce encryption)`
- `Cost Optimization (automated cleanup of waste)`

Implementation: - **Local Only**: Physical laptop must be online (local cron/scheduler) - **Extend My Laptop**: Cloud schedulers (EventBridge/Logic Apps/Cloud Scheduler) - **Event-Driven**: EventBridge Events, Azure Event Grid, Cloud Pub/Sub trigger auto-remediation - **No AssumeRole needed**: Extend My Laptop uses credentials installed in SSM/Key Vault/Secret Manager

---

5. Security & Compliance

- `Security Scanning (misconfigurations, vulnerabilities)`
- `Compliance Checks (CIS, SOC2, HIPAA, custom policies)`
- `IAM Analysis (overprivileged roles, unused credentials)`
- `Encryption Validation (S3, EBS, RDS encryption status)`
- `Network Security (open ports, public resources)`
- `Continuous Monitoring (real-time security posture)`

---

6. Multi-Account/Subscription/Project Management

Org-Level Visibility: - **AWS**: Organizations, Accounts - **Azure**: Management Groups, Subscriptions - **GCP**: Organizations, Projects

Features: - Cross-Account Operations: Batch operations across cloud boundaries - Consolidated Reporting: Org-wide costs, compliance, security - Centralized Policy Enforcement: Consistent policies across all clouds - Account Governance: Account/subscription/project creation, access management

---

7. Collaboration & Approval Workflows

- `Operation Approval (Manager approves high-risk ops)`
- `Change Tracking (Audit log of all operations)`
- `Team Permissions (Role-based access control)`
- `Notification System (Alert team about ops, changes)`
- `Commenting (Team discussion on operations/reports)`

---

8. AI-Powered Operations

Conversational Queries: - “What’s my biggest cost driver across all clouds?” - “Show me all public storage buckets” (S3, Blob Storage, Cloud Storage) - “Which VMs are underutilized?”

Features: - Smart Recommendations: AI suggests cloud-specific optimizations - Anomaly Detection: Unusual spending, security events, performance issues - Playbook Generation: AI creates multi-step, multi-cloud operation plans - Natural Language Execution: - “Stop all dev VMs in Azure West US” - “Enable encryption on all GCP buckets in project X” - Context-Aware Responses: Understands user’s complete multi-cloud estate

---

9. Alerts & Monitoring

See Alert & Event Handling Architecture section above for complete details.

Real-Time Operational Alerts (Can’t Wait): - Critical event detection - Cloud-native alert sources - Auto-remediation with pre-approved options - Multi-channel notifications - Severity-based routing - Unified event schema

Scheduled Scan Alerts (Can Wait - Morning Report): - Daily proactive scans - Interactive morning report - AI-powered aggregation - Template-based customization - Actionable recommendations with 1-click fixes - Permanent storage for historical queries

↑ Back to Top

Architecture Questions to Resolve

Resolved - Where Escher Runs & Execution

1. Where Escher Runs

- **DECIDED**: Two options - Run on Your Laptop (Beta) and Extend to Your Cloud (Main Release)
- User chooses based on their needs

2. Immediate Operations (User-Initiated)

- **DECIDED**: Executed by Physical Laptop (Local Only) or Extend My Laptop (Extended mode)
- Uses user’s stored credentials

3. **Scheduled Operations (Automated)**
    - **DECIDED**:
        - **Local Only**: Physical laptop must be online (local cron/scheduler)
        - **Extend My Laptop**: Cloud scheduler triggers Extend My Laptop
        - User chooses model based on requirements
4. **State & Credentials Storage**
    - **DECIDED**:
        - **Local Only**: Stored on physical laptop + periodic backups to S3/Blob/GCS (hourly)
        - **Extend My Laptop**: Stored in user's cloud (S3/Blob/GCS for state, SSM/Key Vault/Secret Manager for credentials)
        - **Escher AI Server**: 100% stateless, stores nothing
5. **Continuous Monitoring & Automated Remediation**
    - **DECIDED**:
        - **Local Only**: Limited to when laptop online
        - **Extend My Laptop**: Event-driven via cloud schedulers

---

Resolved - Reports & Analytics

1. **Historical Data for Reports**
    - **DECIDED**: Stored in vector store as immutable reports collection
        - **Local Only**: Local vector store on laptop
        - **Extend My Laptop**: Vector store in S3/Blob/GCS (user's cloud)
        - **V2 Release**: Optional central immutable reports (opt-in)
        - Retention policy: User-configurable (default: 90 days for cost, 1 year for audit)
2. **Cost Data Collection**
    - **DECIDED**: Daily snapshots to avoid repeated API calls
        - Direct API calls to AWS Cost Explorer, Azure Cost Management, GCP Billing APIs
        - Daily sync scheduled (default 2am, user-configurable)
        - Stored in immutable reports vector store
        - Cost optimization: Single daily API call instead of repeated queries
        - Manager gets fresh data automatically every morning
3. **Audit Logs**
    - **DECIDED**: Immutable audit logs in vector store
        - Cannot be modified after creation (compliance requirement)

  - Daily sync ensures complete audit trail
  - Fast retrieval via vector store for AI analysis

4. **Report Generation**
   - **DECIDED**: Both on-demand and scheduled
     - On-demand: User requests via conversational interface
     - Scheduled: Daily sync for cost/audit logs
     - Export formats: PDF, CSV, Excel, JSON

Moderate - Collaboration & RBAC

1. **Multi-Account/Subscription/Project Access**
   - **OPEN**: How managed?
     - User installs credentials for each account/subscription/project?
     - Cross-account AssumeRole chains (AWS), Service Principals (Azure), Service Accounts (GCP)?
     - Both patterns supported?

2. **Team Collaboration**
   - **OPEN**:
     - How do Manager and Executor personas collaborate?
     - Approval workflows stored where (local vs cloud)?
     - Real-time notifications needed?

3. **Audit Trail**
   - **DECIDED**: Immutable audit logs in vector store
     - All operations logged in immutable reports collection
     - Storage: Local (Local Only) or S3/Blob/GCS (Extend My Laptop)
     - Immutable: Cannot be modified after creation
     - Daily sync at 2am
     - Retention: User-configurable (default: 1 year)

↑ Back to Top

"Extend Me" Pattern

Understanding Needed: - Manager defines operation templates/playbooks - Executor runs them with "extend me" command - Pre-approved operations with variable parameters - Reduces approval overhead for routine operations

Questions: - [ ] How are templates defined? - [ ] What parameters can Executor modify? - [ ] Approval workflow for template creation? - [ ] Audit trail for "extend me" executions?

↑ Back to Top

Next Steps - Architecture Discussion

Phase 1: Define Execution Model (COMPLETE)

1. Decided on two options (Run on Laptop / Extend to Cloud)
2. Defined scheduled operations execution
3. Clarified privacy model (AI Server 100% stateless)
4. Defined cloud extension provisioning

Phase 2: Define Data Architecture (COMPLETE)

1. Historical data retention strategy
2. Reports generation and storage model
3. Cost data collection and aggregation
4. Export formats
5. Daily sync for Manager persona

Phase 3: Define Personas & RBAC (PENDING)

1. Complete persona definitions (Manager, Executor, others?)
2. Permission model per persona
3. Approval workflows (local vs cloud-based)
4. "Extend me" pattern implementation details
5. Team collaboration mechanisms

Phase 4: Document Complete Operations (PENDING)

1. List all supported cloud operations by category
2. Define which operations need approval
3. Risk levels per operation type
4. Automation boundaries
5. Multi-account/subscription/project patterns

↑ Back to Top

Alignment Check

Fully Aligned & Documented

```
                 WHAT'S DECIDED

• Multi-cloud platform (AWS, Azure, GCP)
• Two ways to run (Laptop-only or Extend to Cloud)
• State and execution in user's control
• Escher AI Server 100% stateless
• Extend My Laptop provisioned in user's account
• Scheduled operations via cloud schedulers
• Multi-cloud management from single Extend My Laptop
• Rust execution engine for operations
• Immutable reports in vector store
• Daily sync for Manager persona (2am)
• Client-side RAG (5 collections)
• Server-side RAG (playbook library, cloud knowledge)
• V2 release plan (optional central immutable reports)
• Alert & Event Handling:
  - Real-time operational alerts
  - Scheduled scan alerts (morning report)
  - Unified event schema
  - Permanent storage in vector store
```

Partially Defined - Need Details

```
              NEEDS MORE DETAILS

• Multi-account/subscription/project credential management
• Collaboration and approval workflows
• "Extend me" pattern implementation
```

Not Yet Documented

```
                  TODO

• Complete list of supported cloud operations by category
• Personas & RBAC model details
• Budget management features
• Notification and alerting mechanisms
  (email delivery, Slack, PagerDuty)
```

↑ Back to Top

This document will be updated as we make architectural decisions.