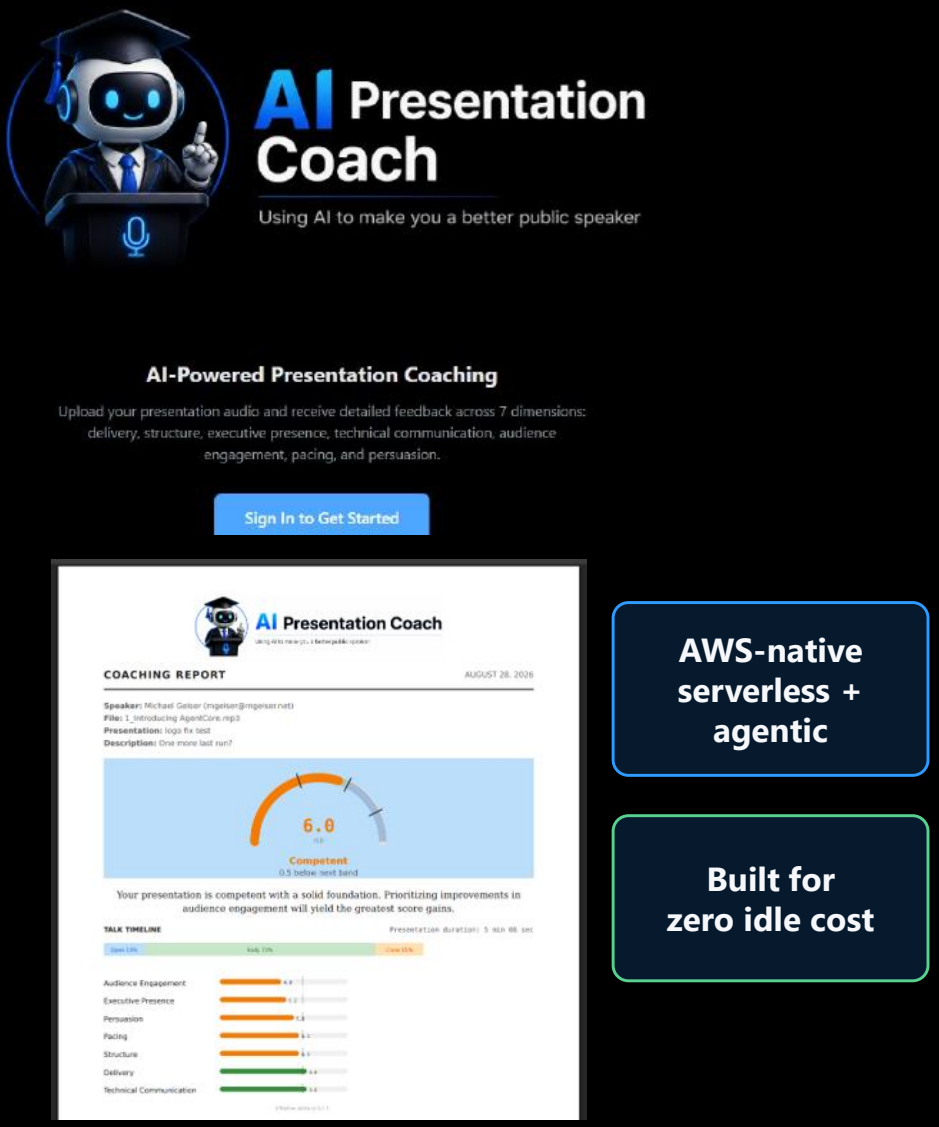
AI Presentation Coach

Using AI to make you a better public speaker

An AWS-native event based agentic presentation coaching platform built with Kiro and AI-DLC practices.

This is not just a demo, it is a useful pattern for taking agentic systems from idea to PoC to Production via governed software delivery.

Created and developed by Michael Geiser



AWS-native serverless + agentic

Built for zero idle cost

Dark theme retained from the working deck

EXECUTIVE LENS

The CTO readout

The AI Presentation Coach (“PresCoach”) is a working AWS-native product pattern, not a lab experiment. It accepts audio, prepares it, then hands it to independent Agentic AI evaluators that produce a coaching report.

7
AI coaching dimensions

20+
AWS native services used

>900
Tests through all workflows

$33.69
Spend for environment and AI tools used

($6.59 AWS + $27.10 Kiro)

Architecture decision worth a CTO review

Where to harden before real users

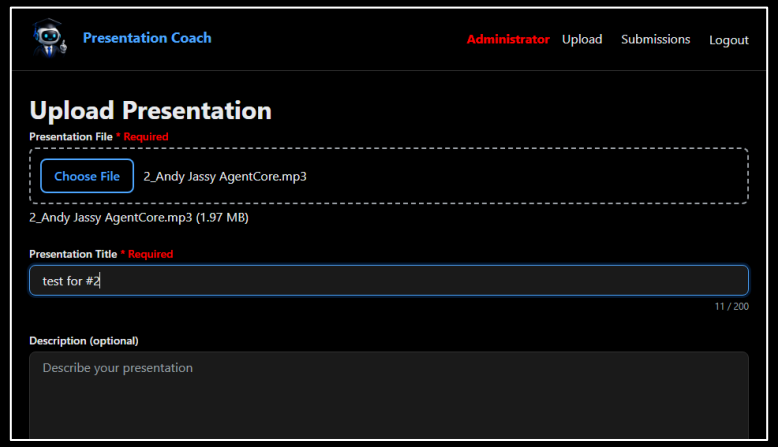
How to scale without cost traps

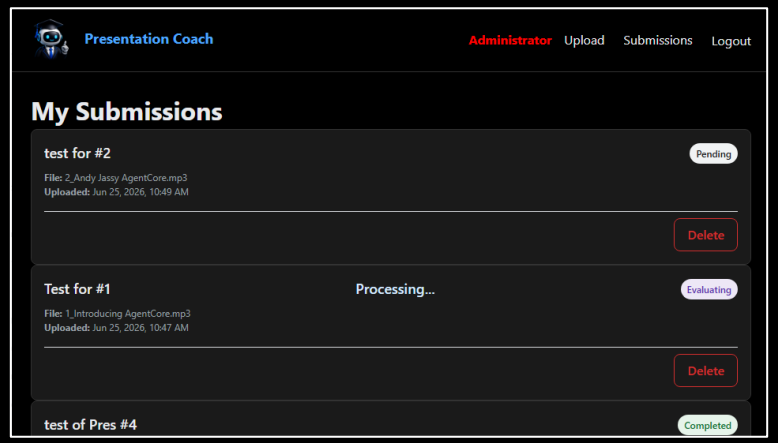
What AI-DLC actually changed

Source deck reported development cost; project overview and retrospective supplied architecture, services, test count, and operating model.

WHAT IT DOES

Using the product is simple, the system behind it is not





User-facing flow

The deliberate split: deterministic processing does the mechanical work, while the probabilistic agentic layer handles judgment. That separation is what makes the design easier to govern.

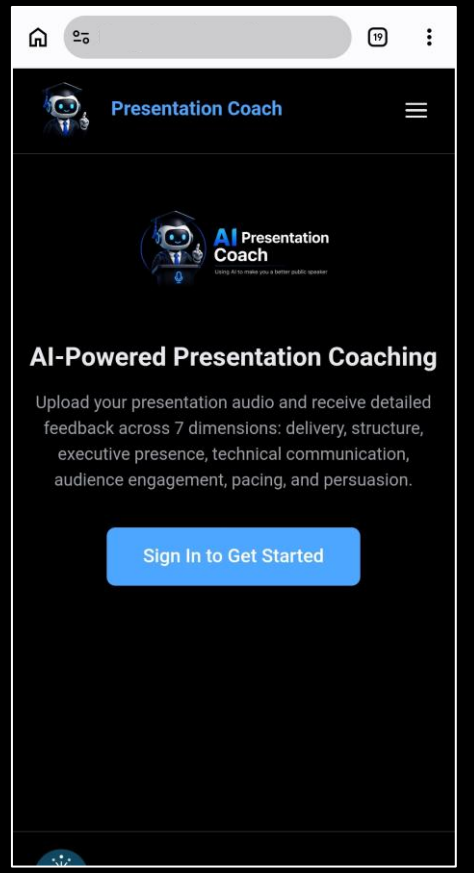
Let's Deep Dive into the App UI Frontend & Backend, processing, and Agentic layers and then review the AI-DLC behind it.

Accepted audio formats and video flag behavior come from the end-to-end processing flow.

The Web App

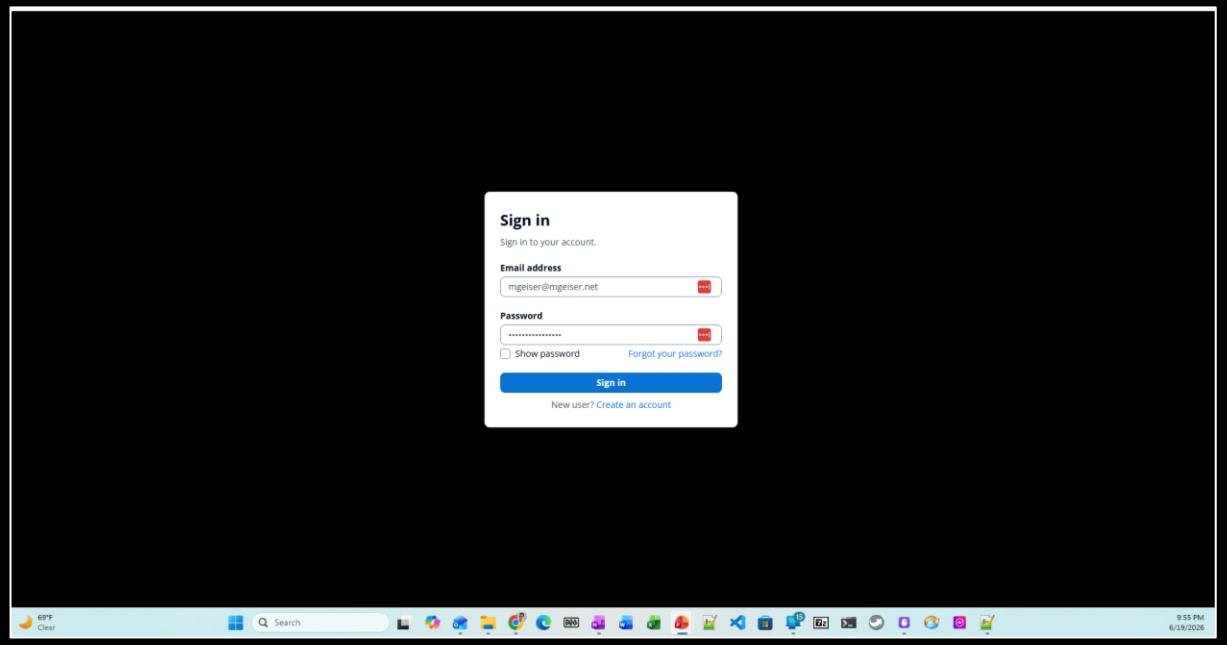
Responsive Web App

Web



Mobile

Cognito Based User IAM



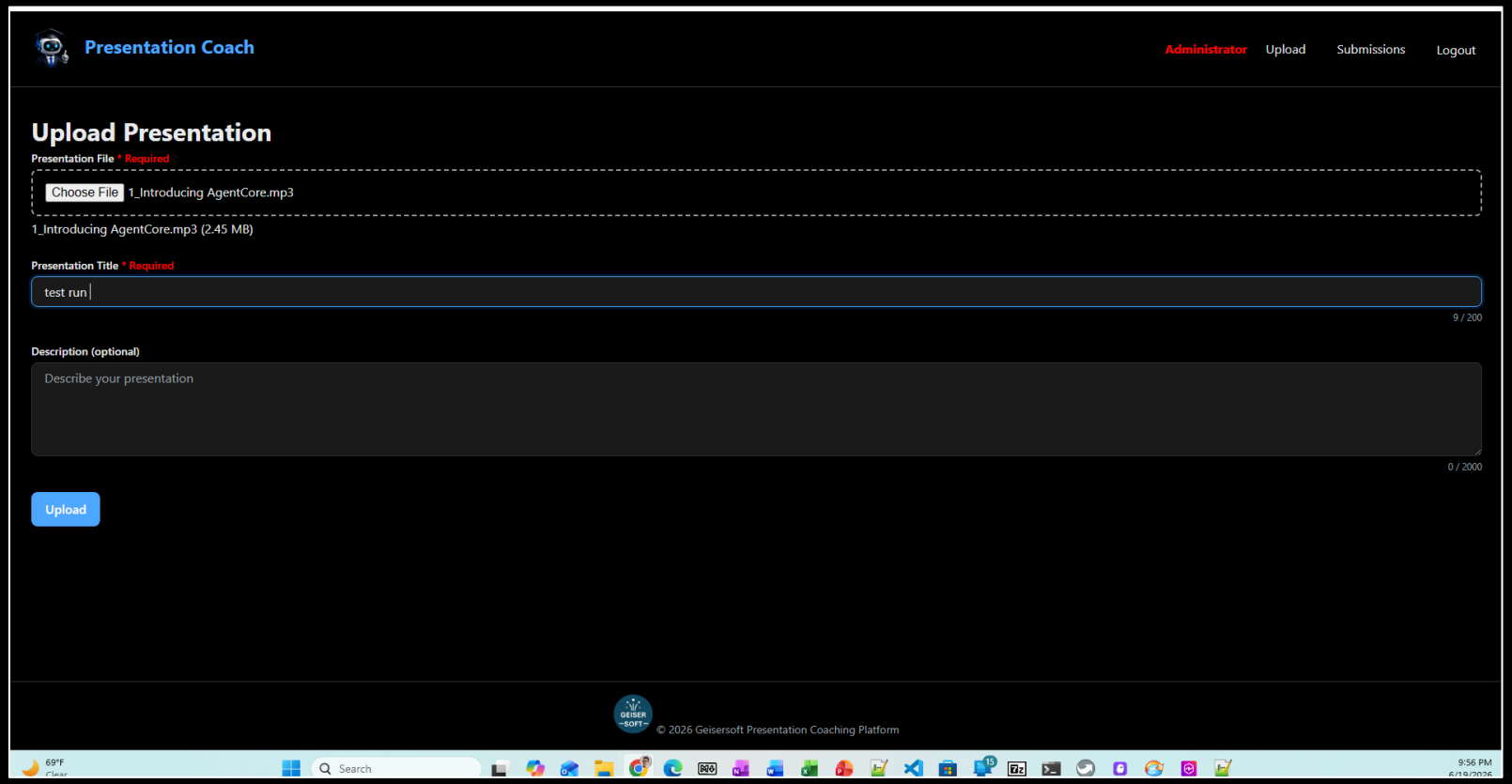
All IAM functionality, including MFA, is a *no-cost* managed service

Groups to define role
- Users
- Administrators

Other Cognito Functions
- User self signup
- Password reset
- Password recovery
- MFA (not turned on yet)

Managed Service
- No Infrastructure or services to admin
- No code to write or maintain
- No cost for under 50,000 users

Audio File Upload

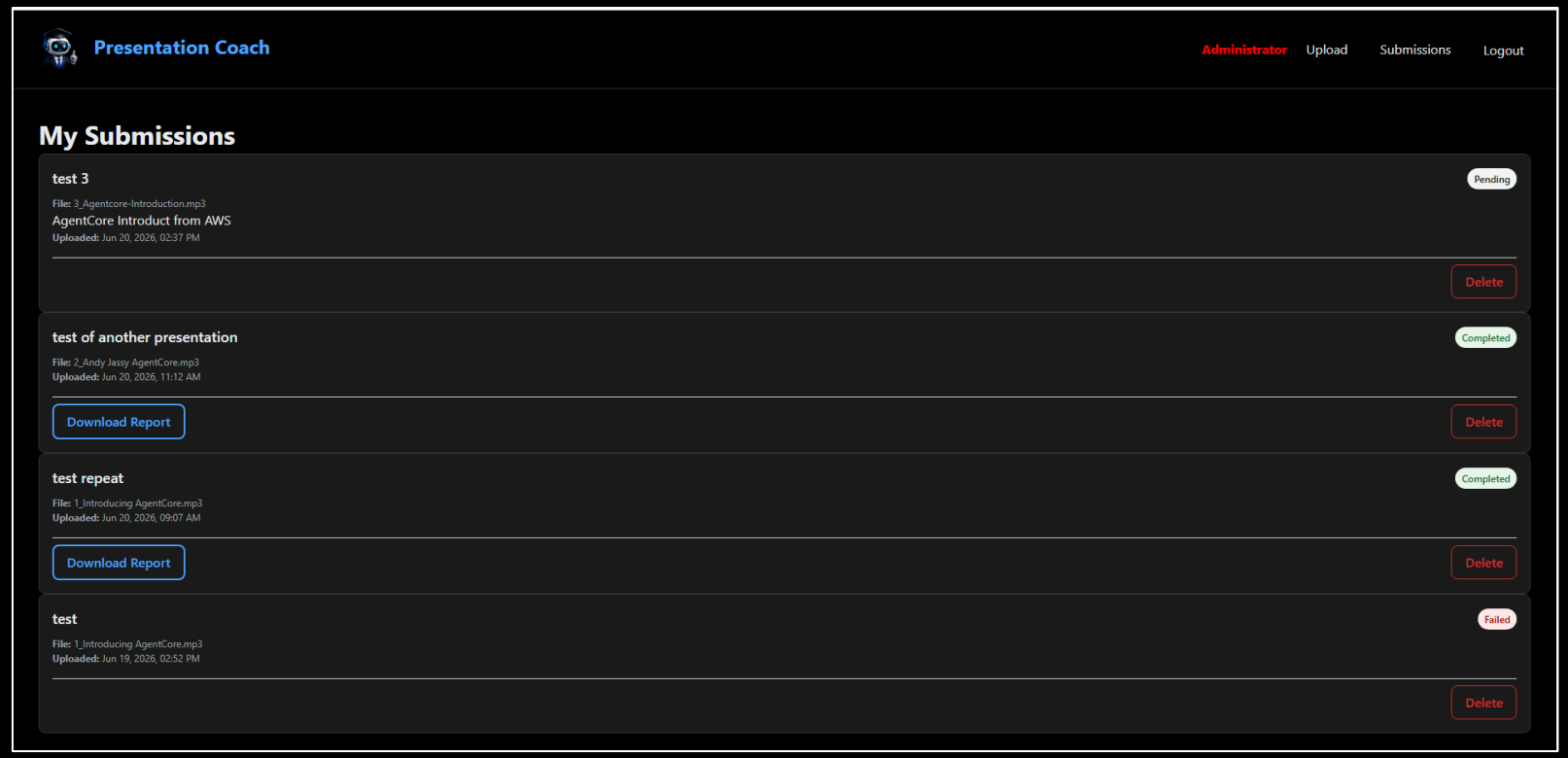


Features

- File picker
- Title
- Description

- Video File upload on roadmap

Processing

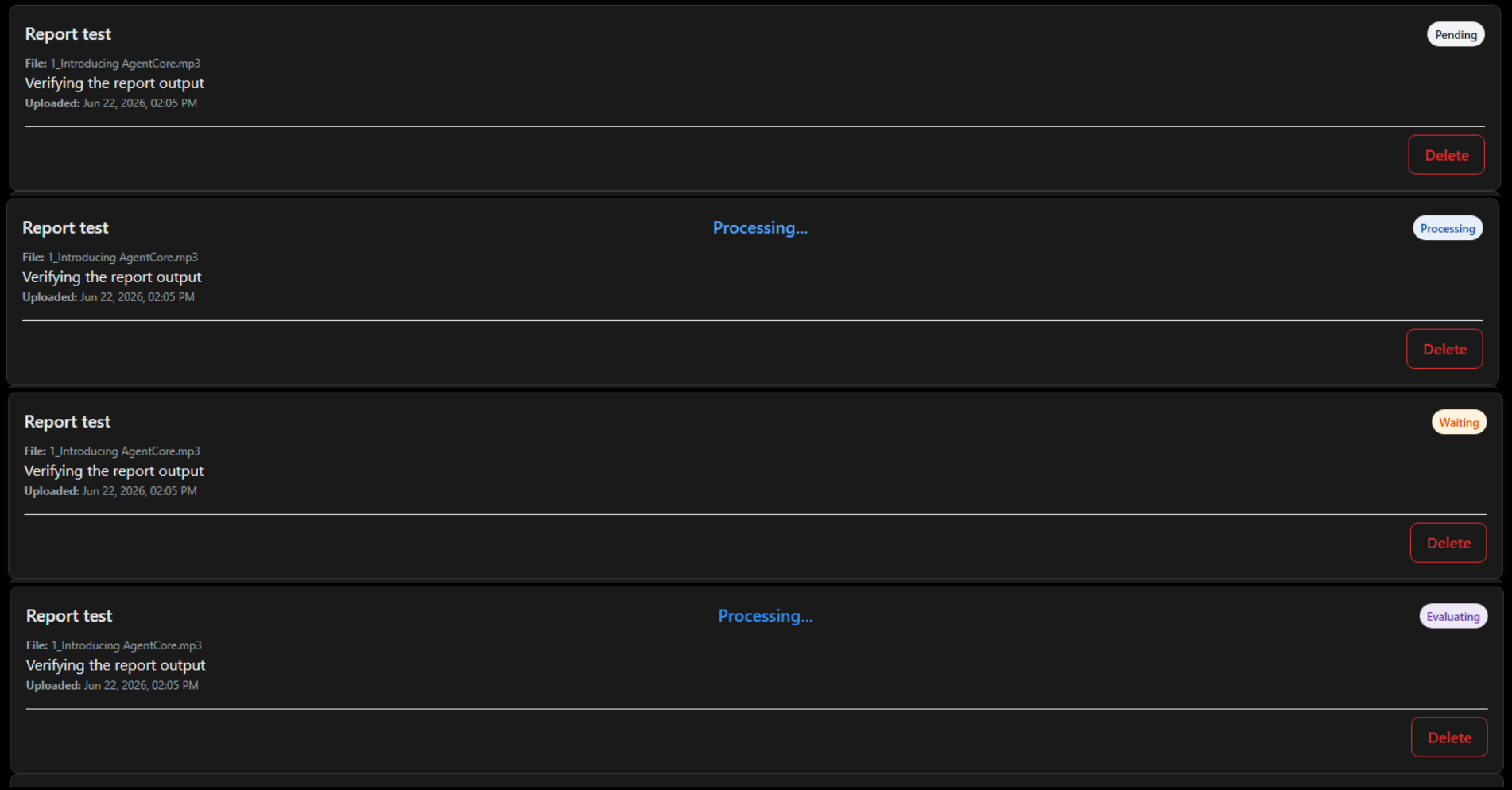


Features

- Multiple steps
- Realtime User status updates

Statuses

Start of Presentation Coach Process



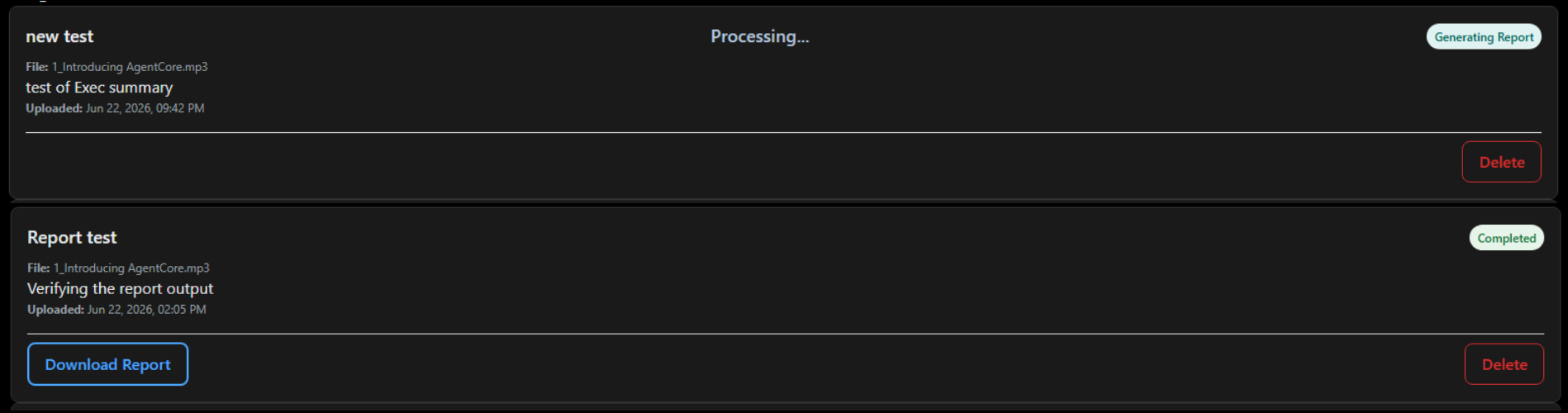
Pending: User uploaded unprocessed audio file

Processing: Transcription and prep

Waiting: Queued for Agentic Evaluation

Evaluation: Agentic analysis in progress

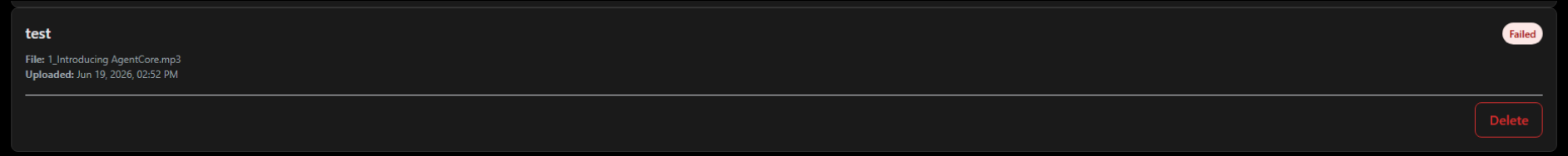
Statuses



Generating Report: Final report is being created

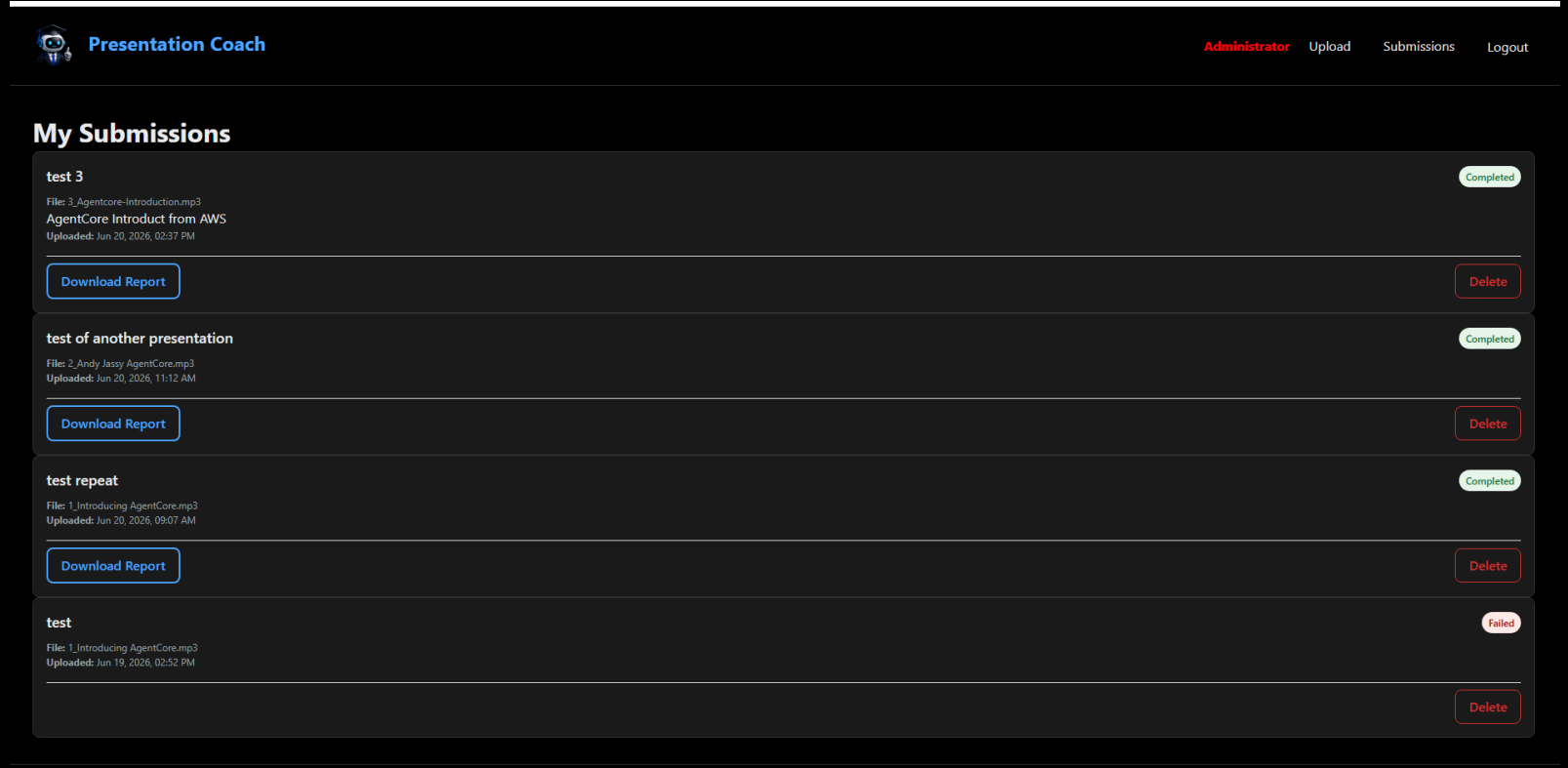
Completed: Report is ready for download

End of "Happy Path" Presentation Coach Process



Failed: An error has halted processing

Report is Ready to Read



Features

- Current status of analysis
- Data entered by User displayed
- Download report link
- Delete link

AI Presentation Coach

Using AI to make you a better public speaker

COACHING REPORT

AUGUST 28, 2026

Speaker: Michael Geiser (mgeiser@mgeiser.net)
File: 1_Introducing AgentCore.mp3
Presentation: logo fix test
Description: One more last run?

Competent
0.5 below next band

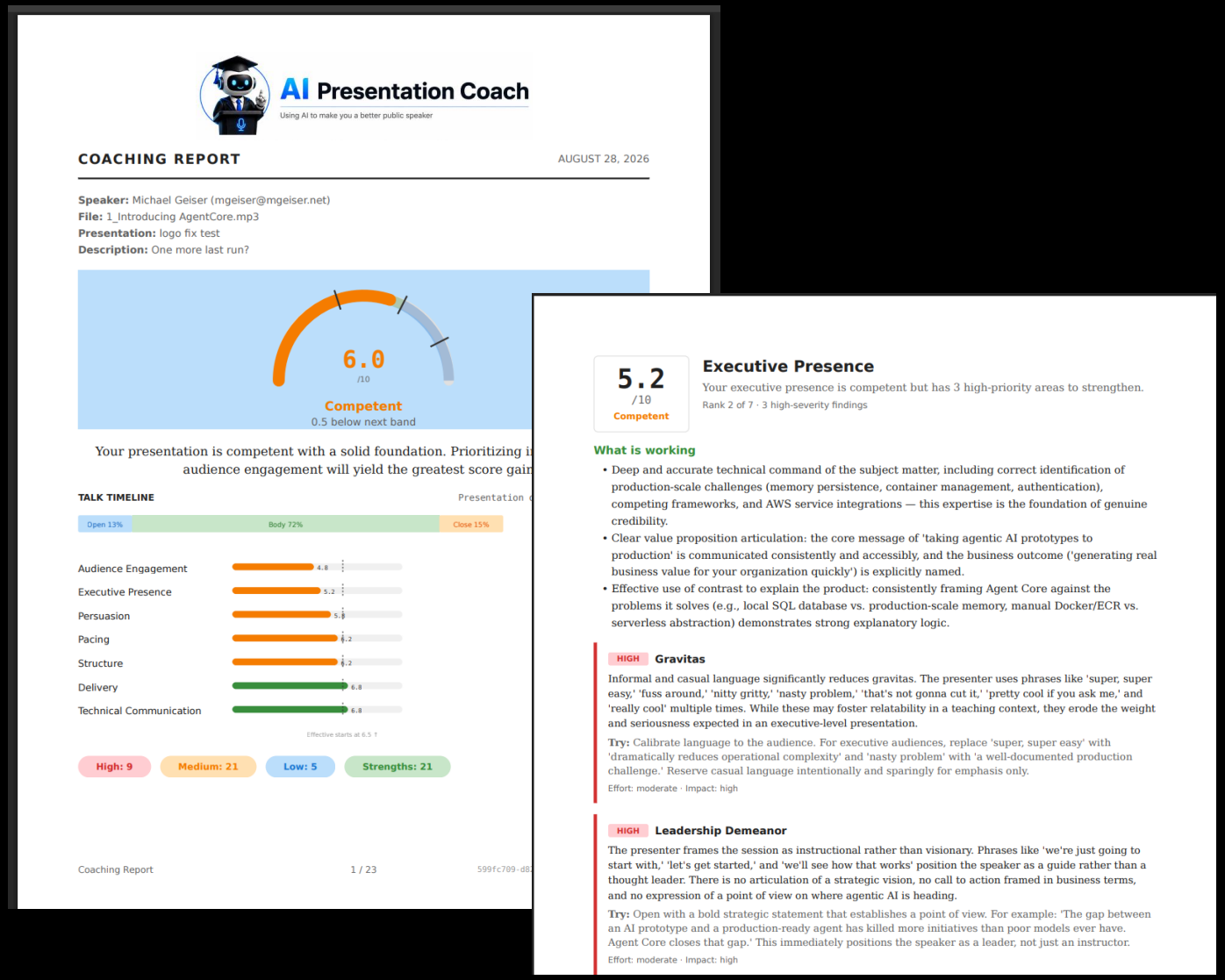
Your presentation is competent with a solid foundation. Prioritizing improvements in audience engagement will yield the greatest score gains.

TALK TIMELINE

Presentation duration: 5 min 06 sec

High: 9 **Medium: 21** **Low: 5** **Strengths: 21**

The Presentation Coaching Report



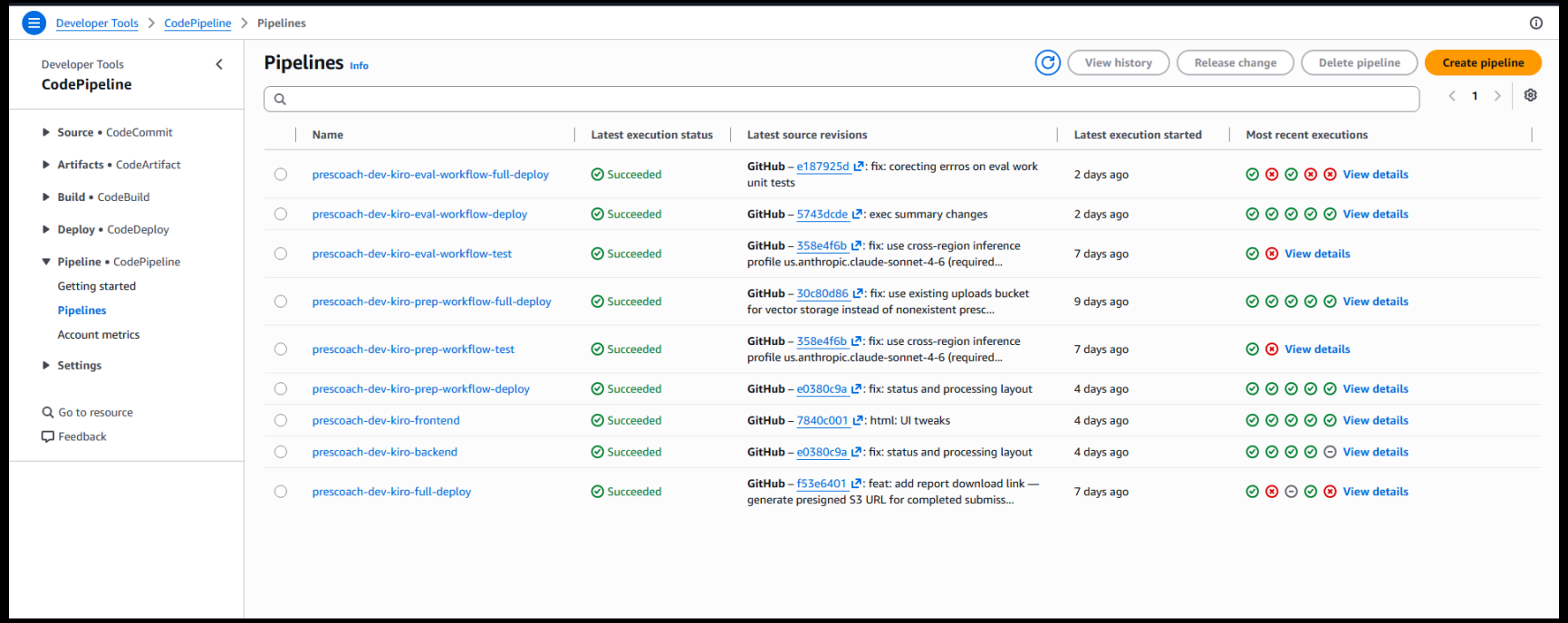
Features

Detailed feedback across 7 dimensions:

- Delivery
- Structure
- Executive presence
- Technical communication
- Audience engagement
- Pacing
- Persuasion

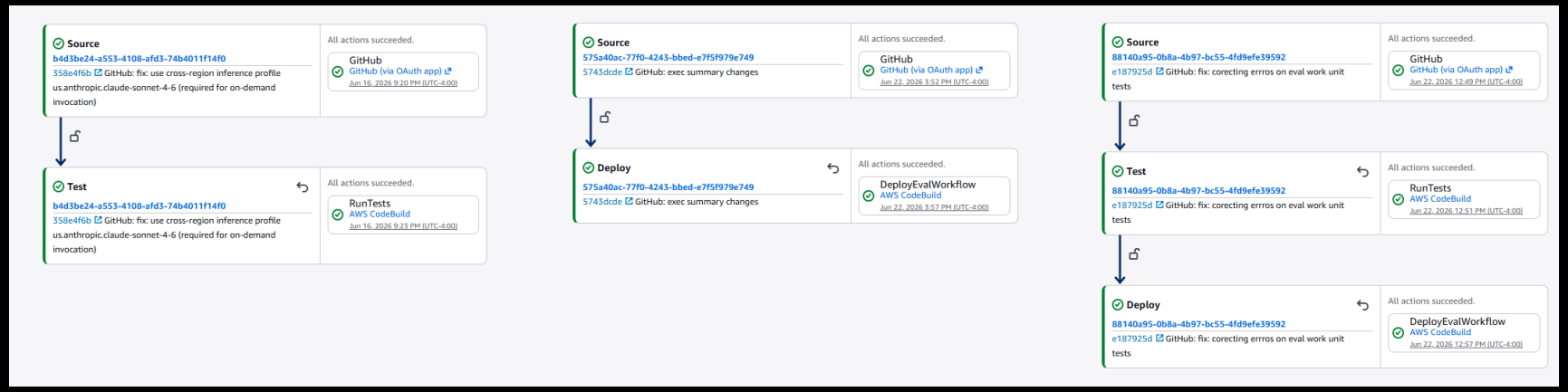
Draft: Report format will be worked on during 60-day Path to Production.

CI/CD Pipelines



CICD Pipelines pull from Github and allow deploys at multiple logical unit levels:

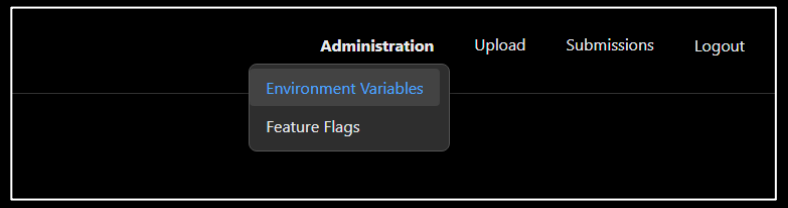
- Frontend/Backend
- Audio/Video Processing
- Agentic AI Evaluation



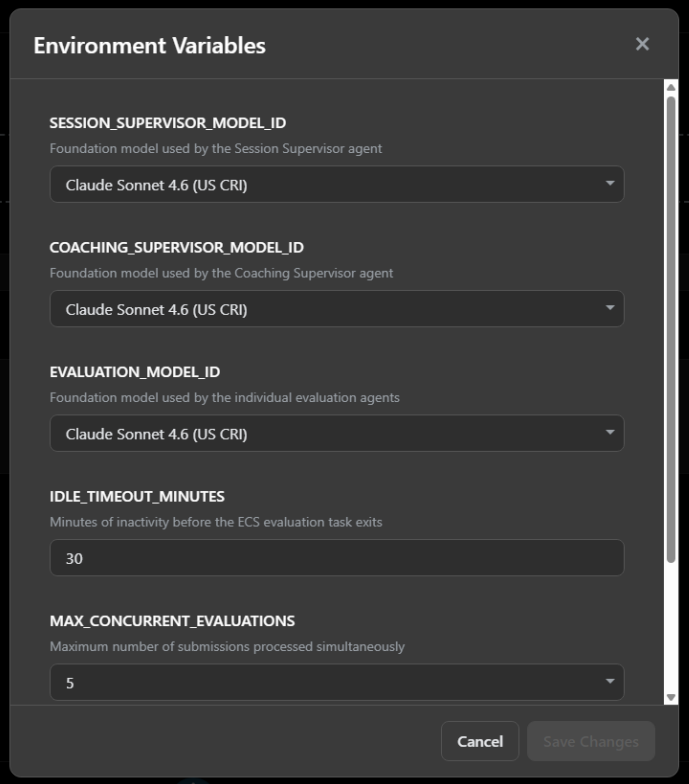
Each Pipeline can deploy

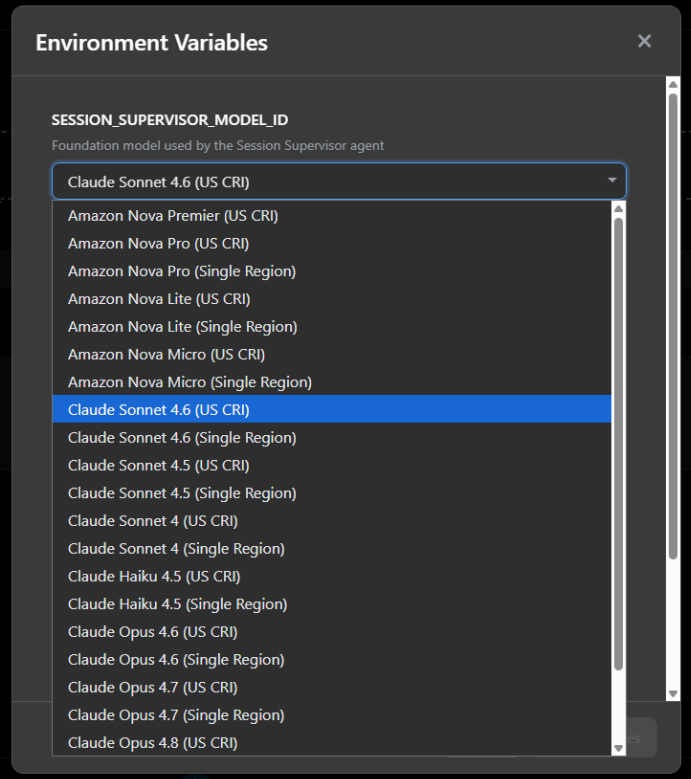
- Tests only (“test”)
- Code only (“deploy”)
- Tests & code (“full-deploy”)

Administration – Environment Variables



"Adminitration" menu only available to users with Administrator role in Amazon Cognito.





Features

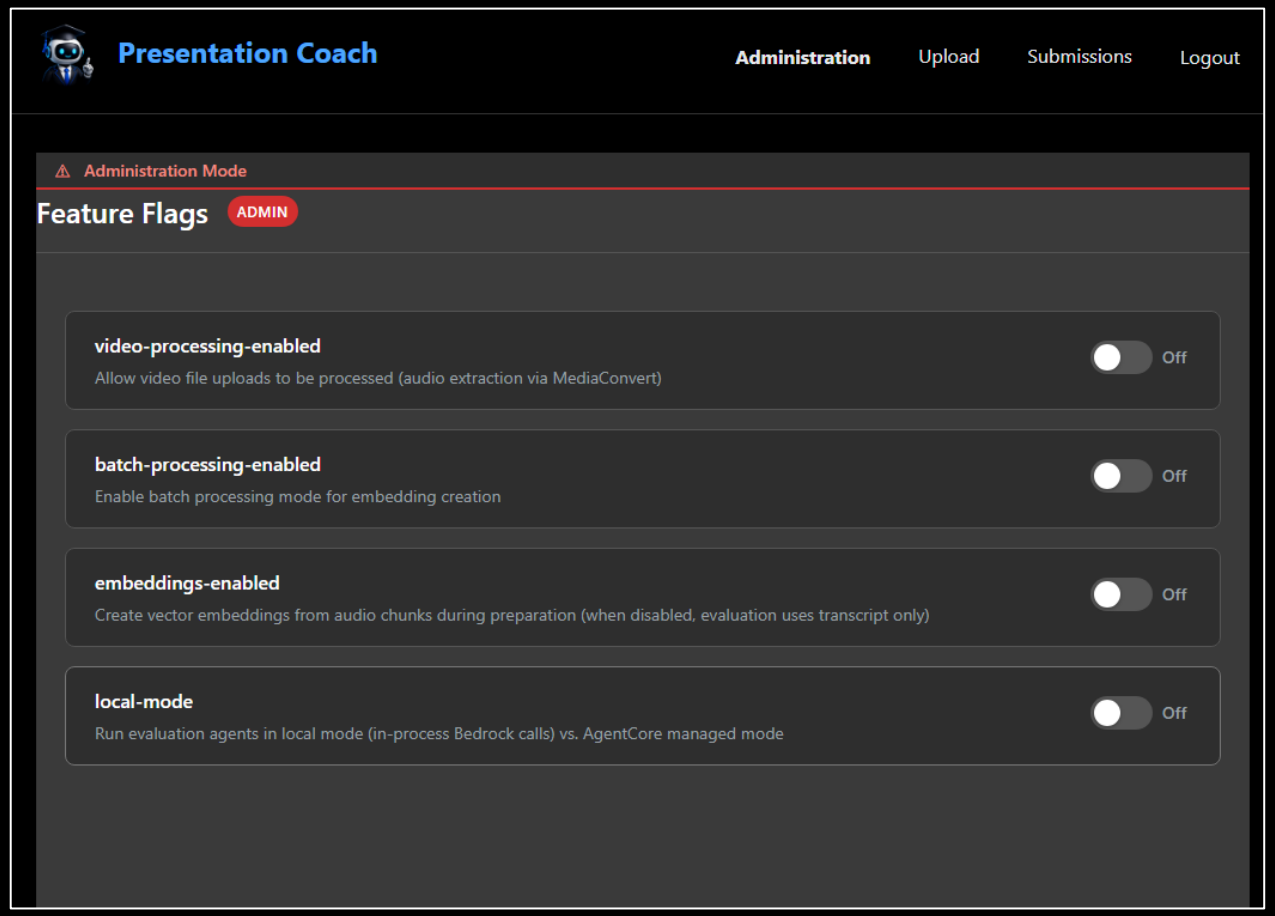
Full control over all environment variables

- Models used
- Idle timeout for ECS tasks
- AI Concurrency limits
- IAM settings

App designed as single tenant but multi-tenant was considered in design

Tested and approved models only are allowed to be selected for the app.

Administration – Feature Flags



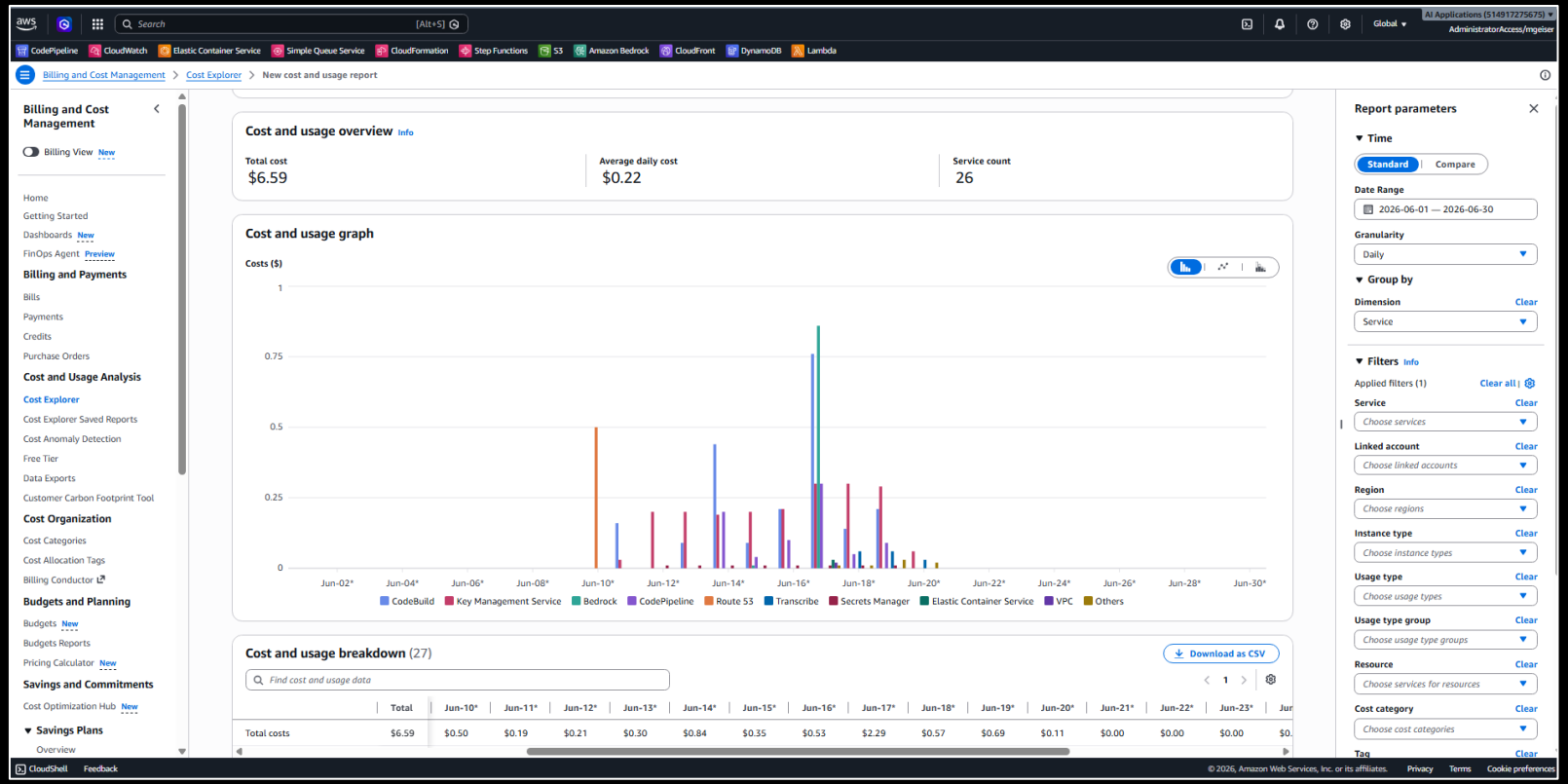
Features

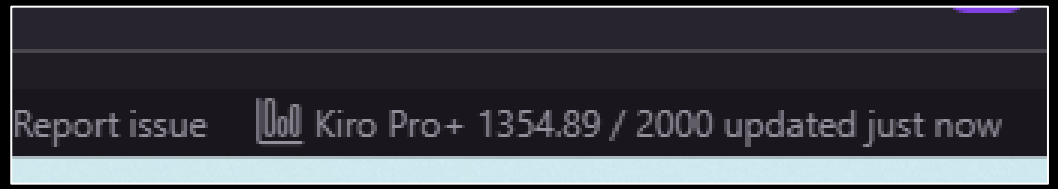
Full control over Feature Flagged scope:

- Video Processing: User could upload video files to be processed in addition to audio files
- Batch Processing: Batch processing allows for lower cost asynchronous processing of presentations for vector stores.
- Embeddings: turns on and off vectorization of presentations.
- Local Mode: using a simplified (less resilient) agentic processing implementation

Observability and Monitoring – (eta: Aug 15, 2026)

Development Costs





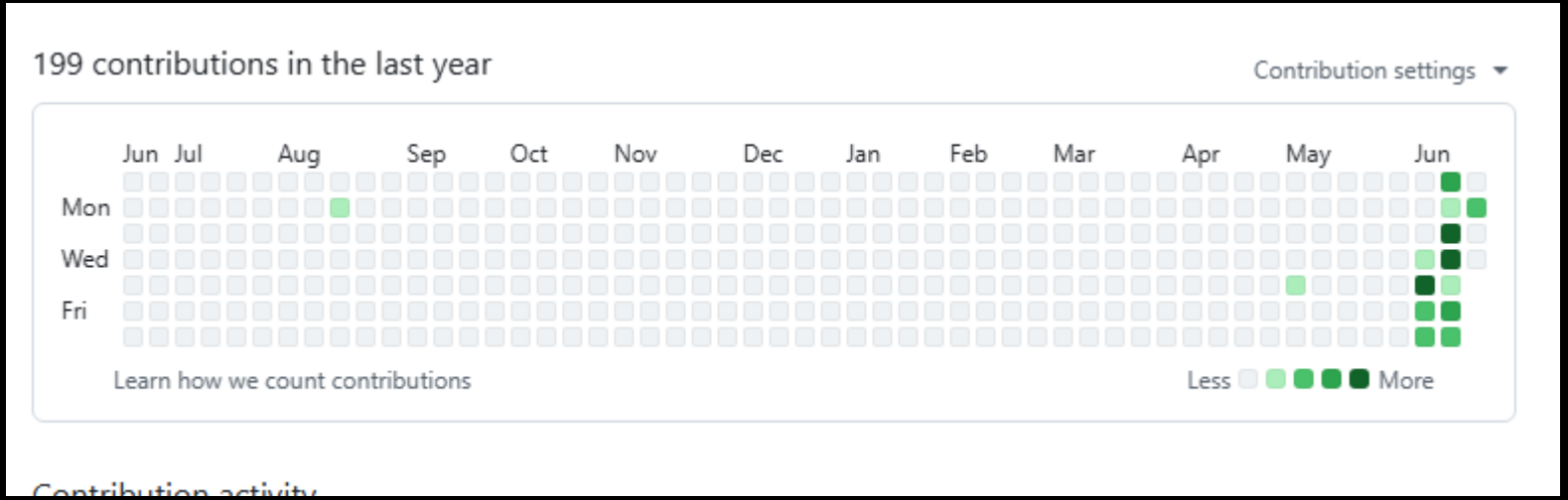
Features

Total costs of the development

- AWS Account: $6.59
- Kiro AI-DLC: 1355 Credits ($27.10)
- LLM usage was not included in these costs. They usually show up each day but were delayed in June.

Github Repo Activity

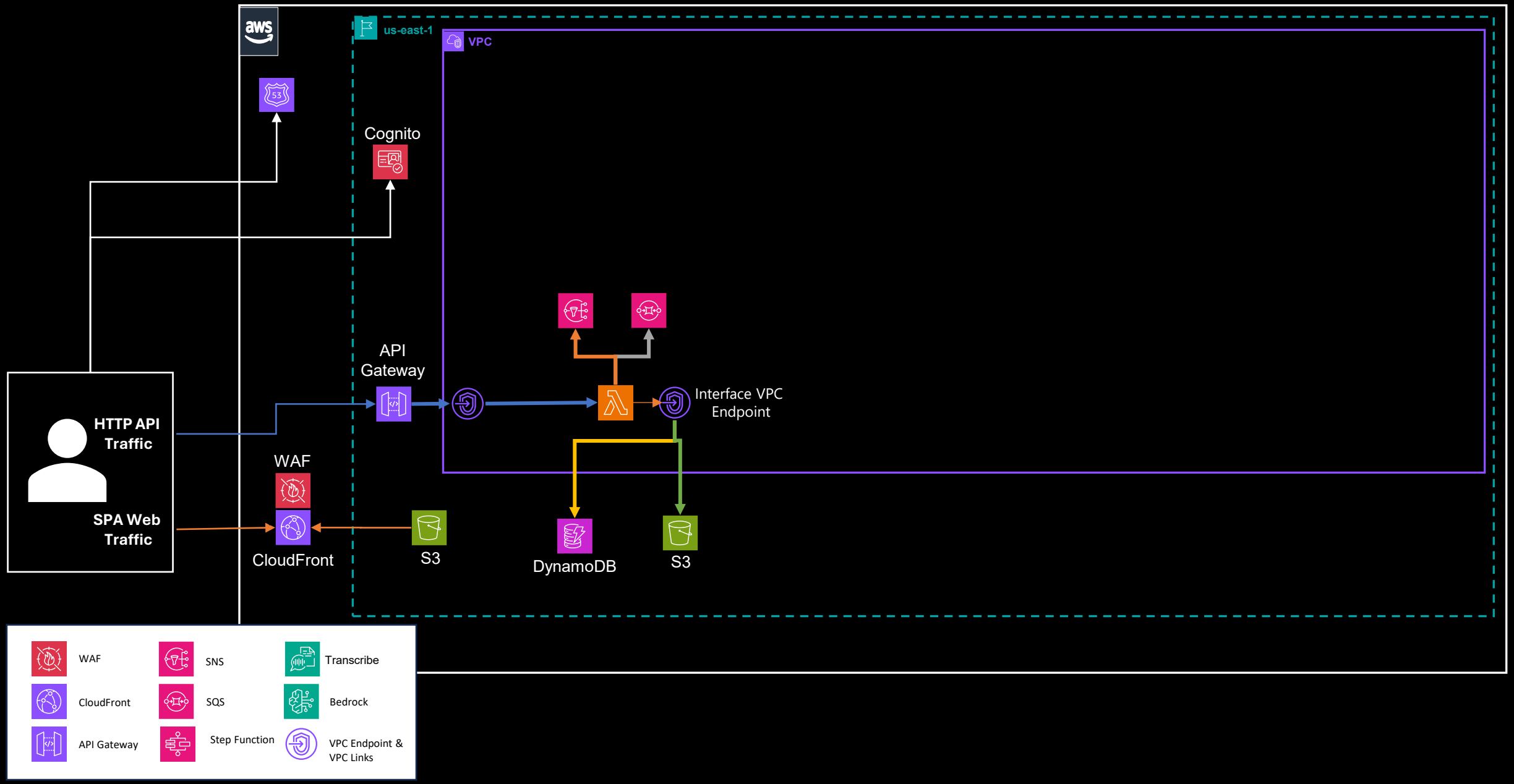
Initial Build



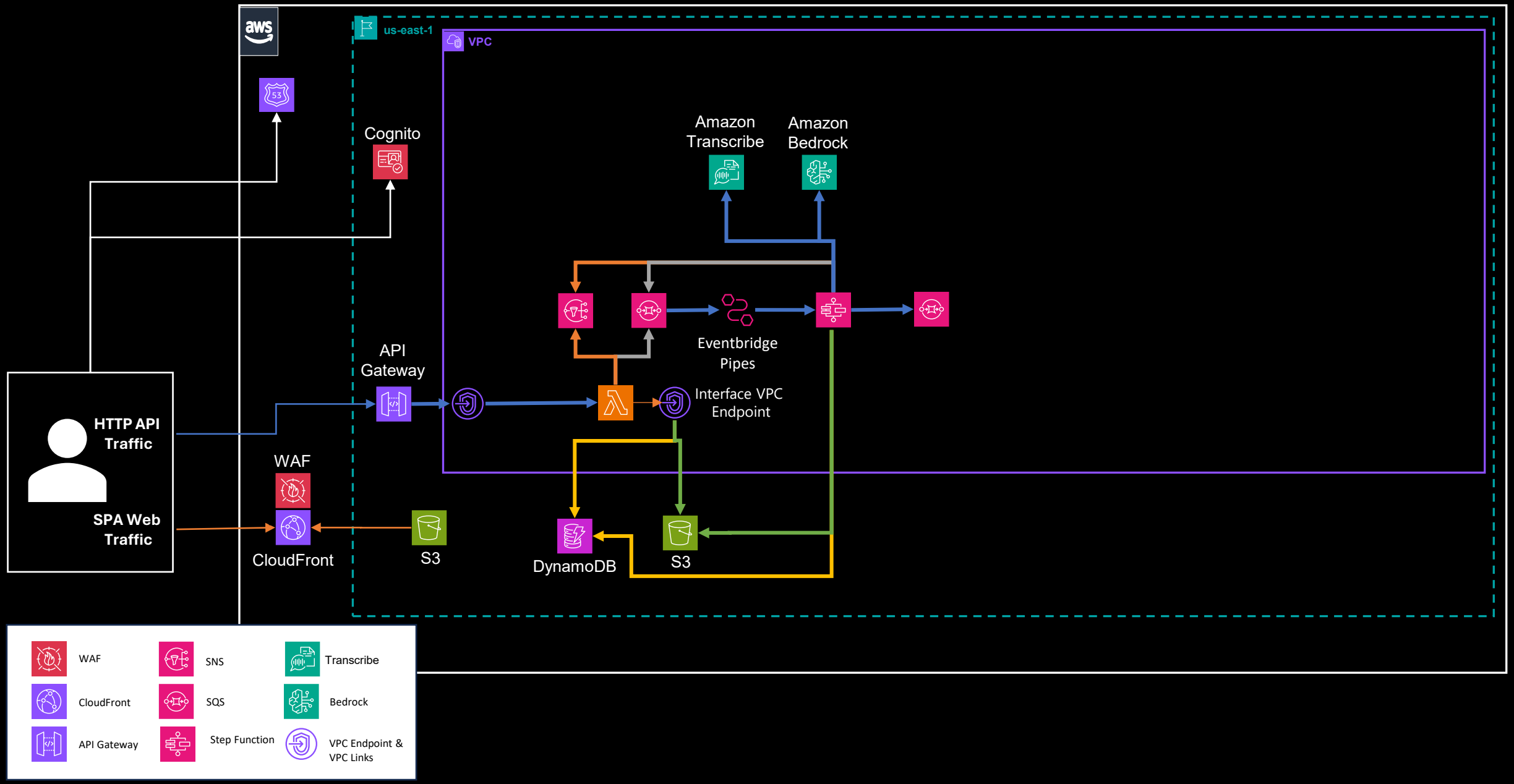
https://github.com/michaelgeiser/kiro-agentcore-app

The Architecture

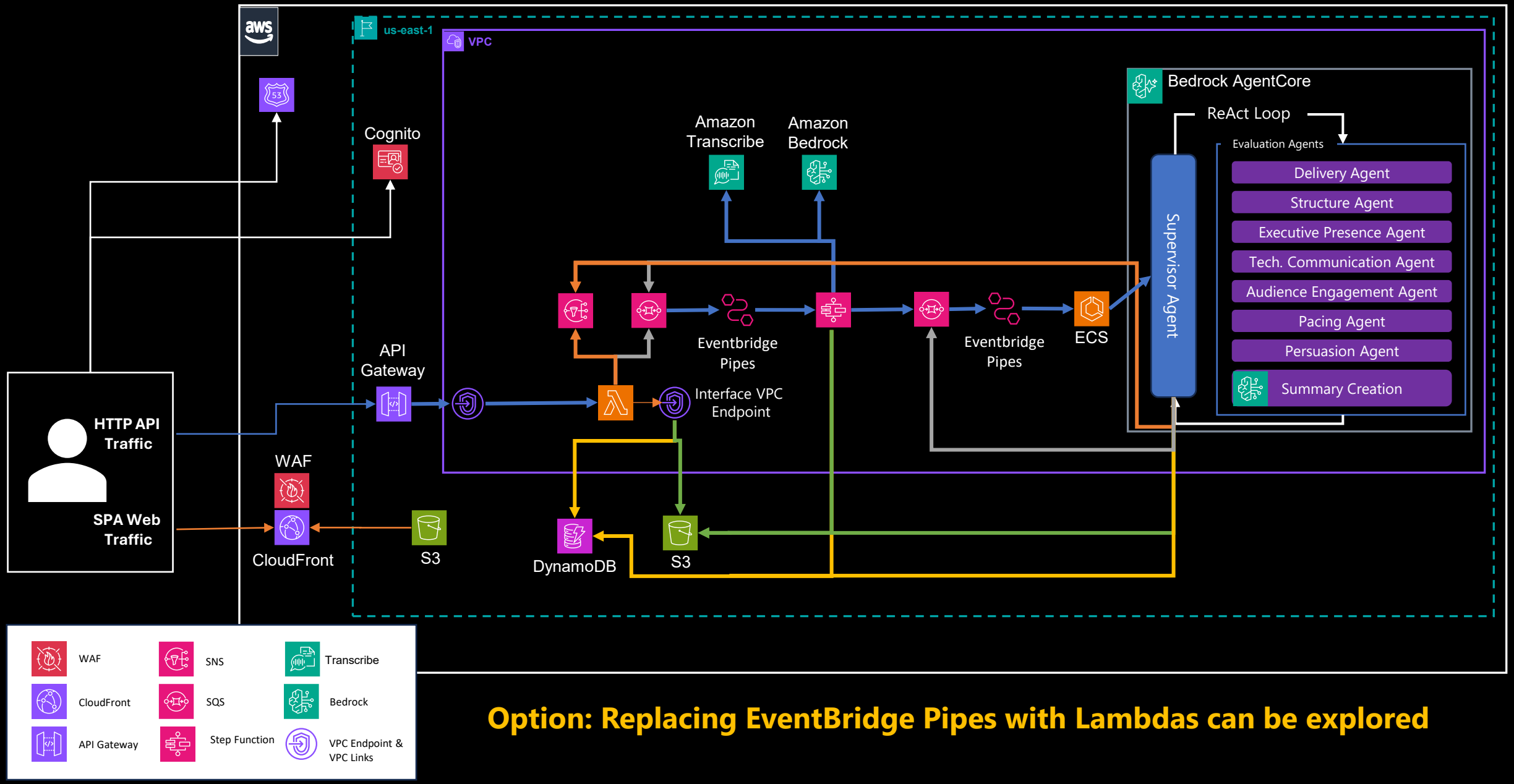
Web App SPA and Backend Architecture



Audio File Preparation Architecture

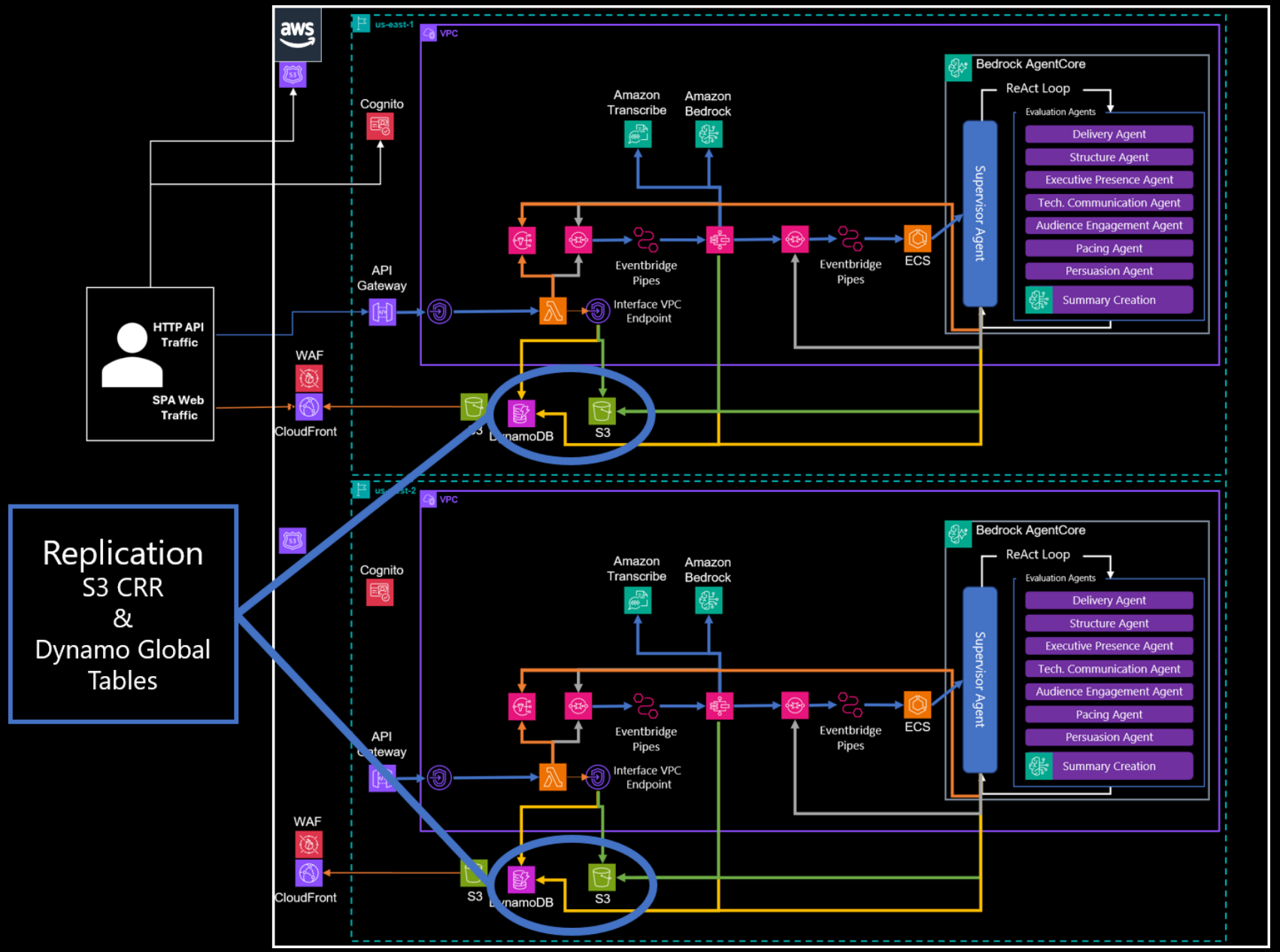


Evaluation with Agentic AI and Report Creation



Option: Replacing EventBridge Pipes with Lambdas can be explored

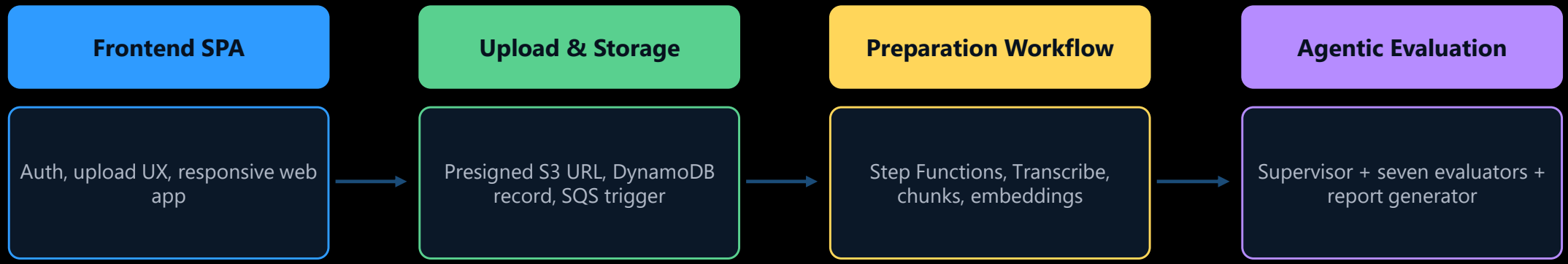
Multi-Region Ready if the Business Case Exits



DELIVERY MODEL

Build approach: AI-DLC turned the idea into deployable work units

The useful lesson is not that code was generated quickly. The useful lesson is that the work was decomposed into independently testable “bolts” with clear contracts.



<7 days
elapsed build window

9
CI/CD pipelines

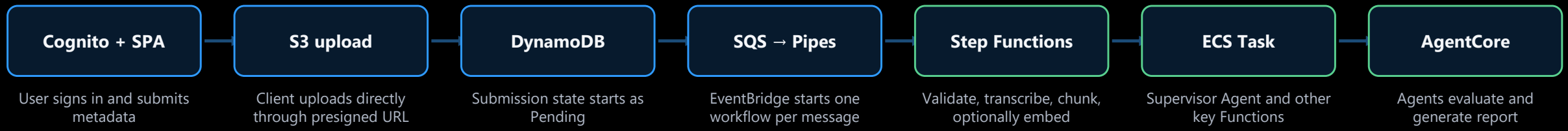
0
servers managed

~$0
idle architecture cost

Retrospective describes four work units, bolt-based interfaces, CI/CD pipelines, and zero-idle service selection.

RUNTIME FLOW

End-to-end processing is event-driven and observable



The handoff contract matters more than the tool choice. The upload service does not know how evaluation works. The evaluator does not care how the file arrived. That keeps each boundary testable and replaceable.

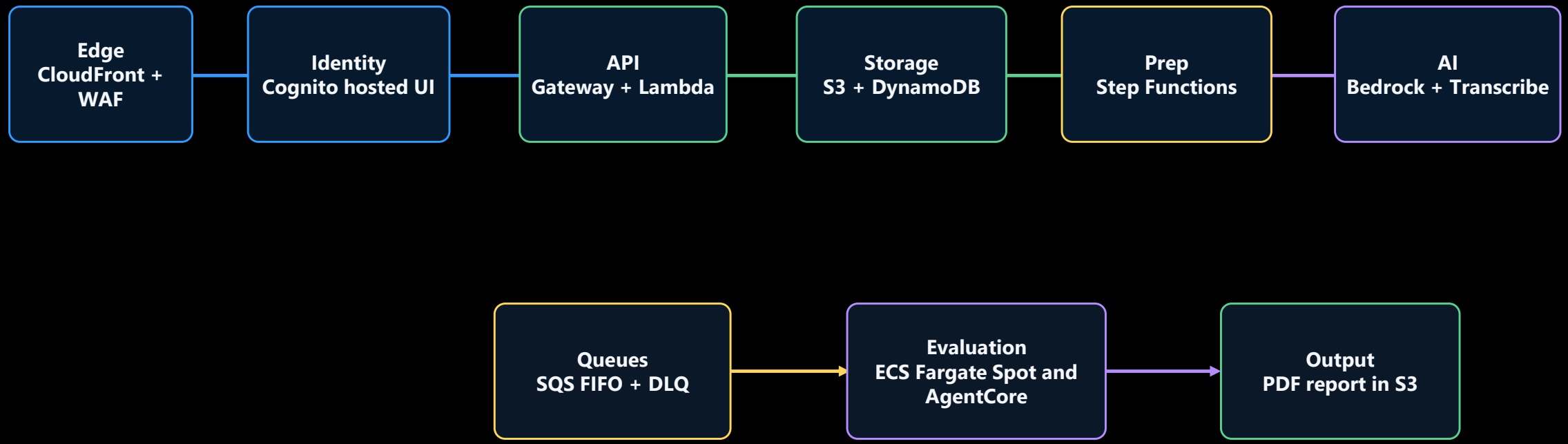
State transitions
Pending → Processing → Completed or Failed

Failure path
retries → HandleFailure → SNS + DLQ

Flow reflects the documented happy path, state machine sequence, S3 objects, DynamoDB transitions, retries, SNS, and DLQs.

AWS DESIGN

Architecture, viewed like a CTO would review it



Why this is a good MVP architecture

- Files never pass through Lambda
- Async processing gives back-pressure
- Feature flags allow runtime tuning
- Least-privilege IAM can be reviewed per work unit

What I would challenge before production

- Model quotas and throttling behavior
- Regional failure stance and user-pool strategy
- PII retention and report lifecycle policy
- Evaluation result explainability and traceability

Architecture summarizes services described in project overview, flow, and retrospective.

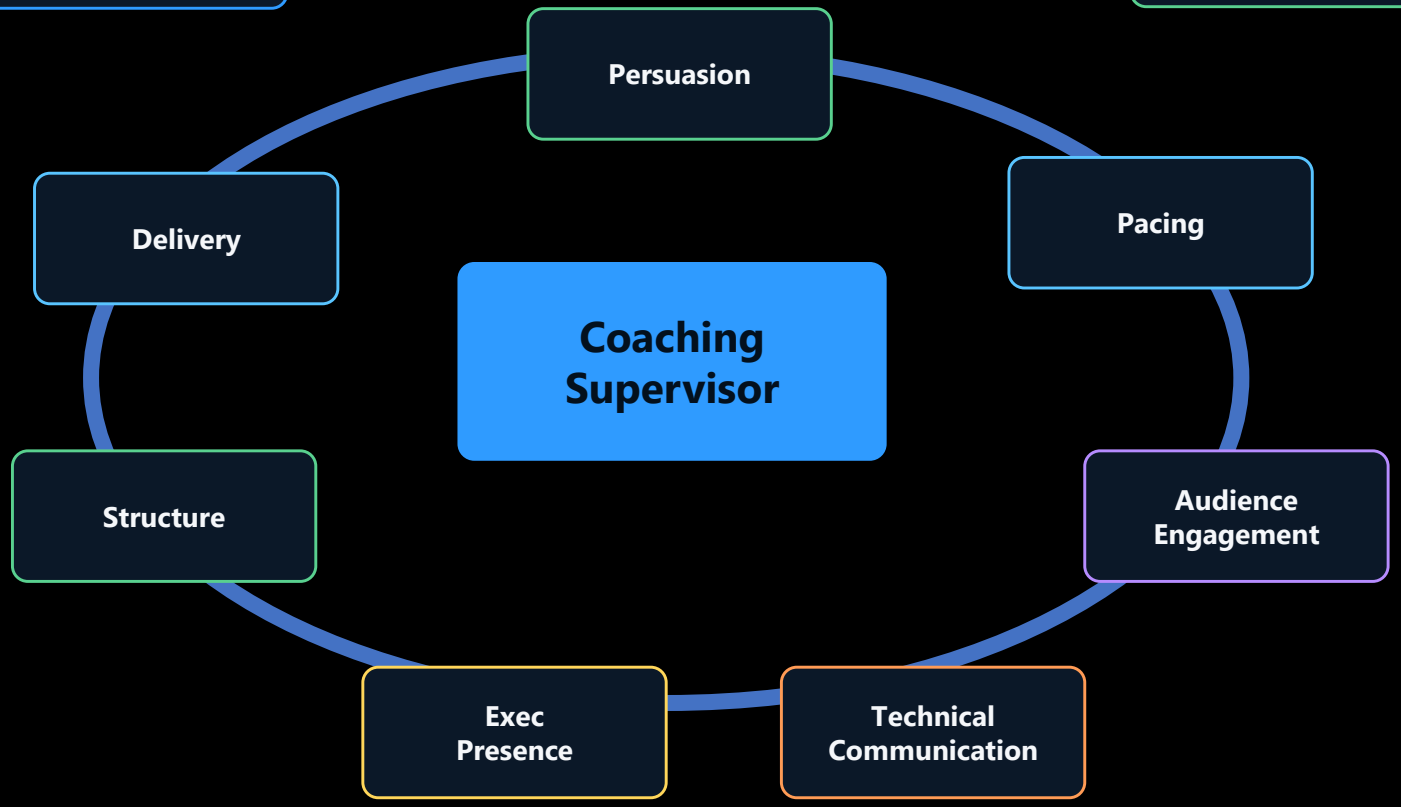
AI EVALUATION

The agentic layer is composable by design

This is the part that justifies an agentic approach. The value comes from multiple independent lenses reviewing the same asset, not from one big prompt pretending to be a committee.

Agent definitions live in a JSON manifest and can be enabled or disabled without code changes.

The report generator assembles structured results into a 10-18 page PDF coaching report.



COST AND SCALE

The economics are exactly why serverless fits

For spiky, early-stage usage, the cost goal is not “cheapest at theoretical peak.” It is “do not pay for infrastructure while product-market fit is still loading” but will scale economically as usage takes off.

LLM selection must be evaluated and optimized. Costs and output quality differ for LLMs such as Claude Sonnet 4.6 vs Amazon Nova Pro and need to be evaluated.

Infrastructure only: LLM costs must be determined; cost and output will vary based on LLMs used.

Bedrock dominates the bill at scale

Zero idle for Lambda, API Gateway, Step Functions, SQS, SNS, Bedrock

S3 vector store keeps MVP cost near zero

S3 vector store keeps MVP cost near zero

Cost scenarios and zero-idle principle come from project retrospective cost analysis.

COST DECISION

S3 as the first vector store is the right kind of cheap

This is one of those choices where CTO discipline matters. For the MVP, S3 is not the “best vector database.” It is the best way to avoid paying for one before the workload proves it deserves one; it is **fit to purpose** engineering.

	S3 JSON embeddings	OpenSearch Serverless / pgvector
Idle cost	$0 storage only	~$175/month minimum
Read pattern	Per-submission retrieval	Similarity search across corpora
Latency	~50-100 ms	~10 ms query path
Move when	MVP and variable load	>50K presentations/month or cross-submission search

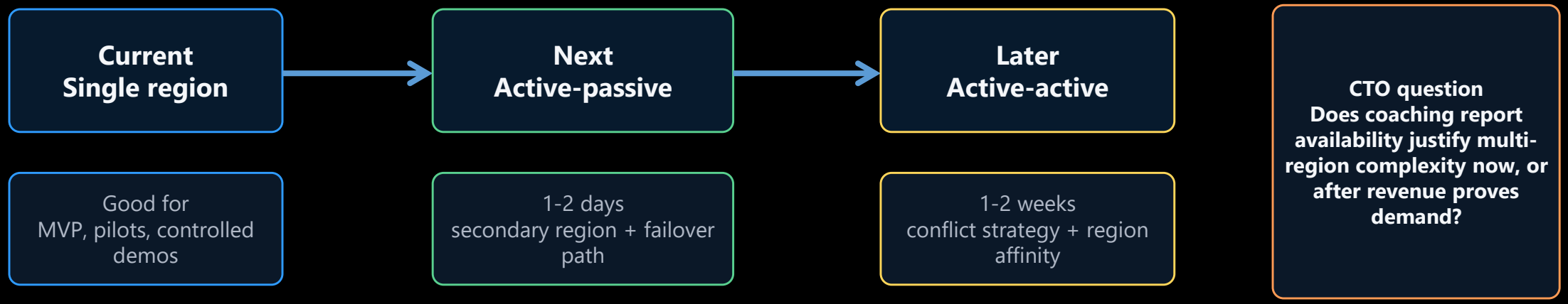
Design win: vector-store-type and vector-store-endpoint are SSM parameters, so the upgrade path is operationally clean.

S3 vs OpenSearch cost and upgrade guidance come from the retrospective cost analysis.

RESILIENCE

Operational resilience is solid for MVP and launch, but the regional stance is a decision

Current state is single-region in us-east-1 and multi-AZ compatible by design and construction.
Single Region is acceptable for an MVP, but production requires an explicit answer to how much regional failure the business will to tolerate. Multi-region is cost effective and does not require re-architecure work.



Low-friction upgrades to multi-region resilience: DynamoDB Global Tables, S3 CRR, second regional API Gateway, replicated Step Functions and Lambda stacks, CloudFront origin routing.

Multi-region path and estimated effort come from the retrospective.

RISK REGISTER

The production-hardening backlog is manageable, not mysterious

The system is deployed and the end-to-end pipeline runs. The remaining work is application-level correctness plus the usual enterprise-grade hardening. Good news: no smoke machine required.

Known app fixes	Use Web Sockets instead of polling for Submissions page while a presentation is processing.
Security	Add additional guardrails, add admin approvals for new users, enhance WAF rules, define data retention policies.
Reliability	Refine throttling policy by user and entire system, ECS Spot interruption handling, DLQ message runbooks, retry budgets, test idempotency fully.
Observability	Add Correlation Id and additional logging to trace a submission across Cognito, API, S3, Step Functions, queues, ECS, Bedrock, and report generation.
FinOps	Refine budgets, Bedrock spend alarms, lifecycle policies, model-choice controls, chargeback by user or tenant.

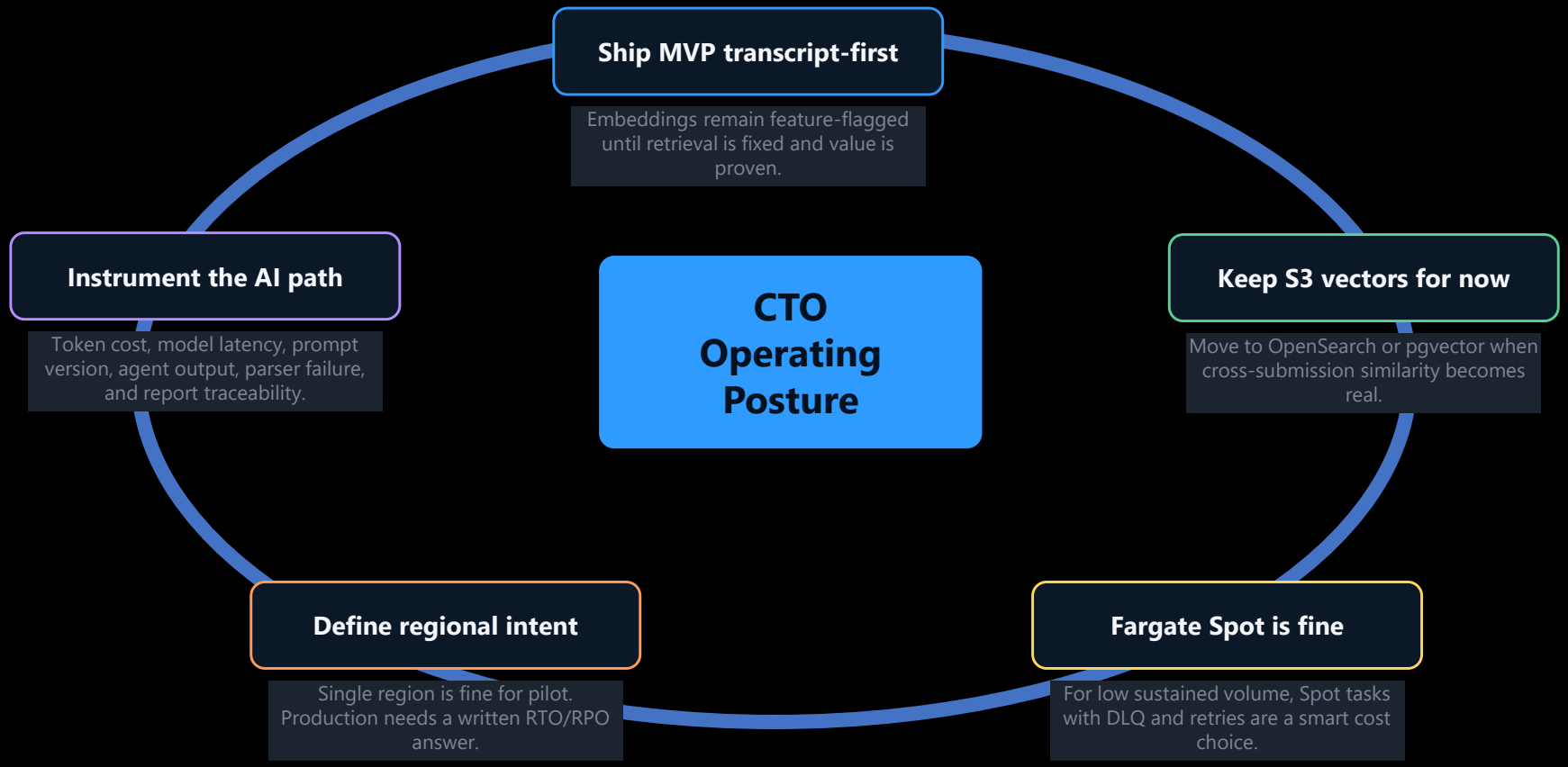
Evaluate items and prioritize:
In this case the Security and Reliability issues have the highest priority

Known application issues come from project overview and source deck. Hardening items are CTO recommendations derived from the architecture.

WHERE I WOULD SPEND TIME

CTO decision radar

The architecture is good enough to advance. The discussion should move from “can this work?” to “what operating posture do we want?”



This slide distills CTO decisions from the supplied architecture, flow, cost, and current-state documents.

NEXT MOVES

A Practical 15/30/60 Day Path to Production

Do not boil the architecture ocean. Fix the known seams, prove the evaluator quality, then harden only the parts that have a user or revenue reason to exist.

15 days

Stabilize MVP

- Observability and monitoring enhancement
- Administrator Menus
- Refine throttling policy
- ECS Spot interruption handling,
- Budget alarms on Bedrock usage

30 days

Prove quality

- Golden set for coaching quality
- Prompt and model versioning
- Agent output schema enforcement
- Report scoring calibration

60 days

Prepare production posture

- Retention and PII controls
- Other Responsible AI controls
- RTO/RPO decisions
- Quotas and load test
- Tenant or user-level cost attribution

Exit criteria
A CTO should be able to trace one submission from upload to report, explain one agent's score, and predict the cost of the next 1,000 runs.

Path is a recommendation built from the documented current state and CTO operating concerns.

CLOSE

The CTO-to-CTO takeaway

PresCoach is a strong pattern because it combines three things that usually get separated: a useful product experience, an AWS-native cost model, and an AI-DLC delivery method that keeps architecture boundaries visible.

What is working
Clear work-unit boundaries, async flow, managed services, zero idle cost, composable agents.

What to fix
Retrieval, parser handling, traceability, hardening of AI and data controls.

What to decide
Regional posture, vector store timing, quality gates, model spend controls, production-readiness threshold.

Bottom line: advance it but make the next phase *boring* in the best possible enterprise way. Boring is where systems go to become dependable.

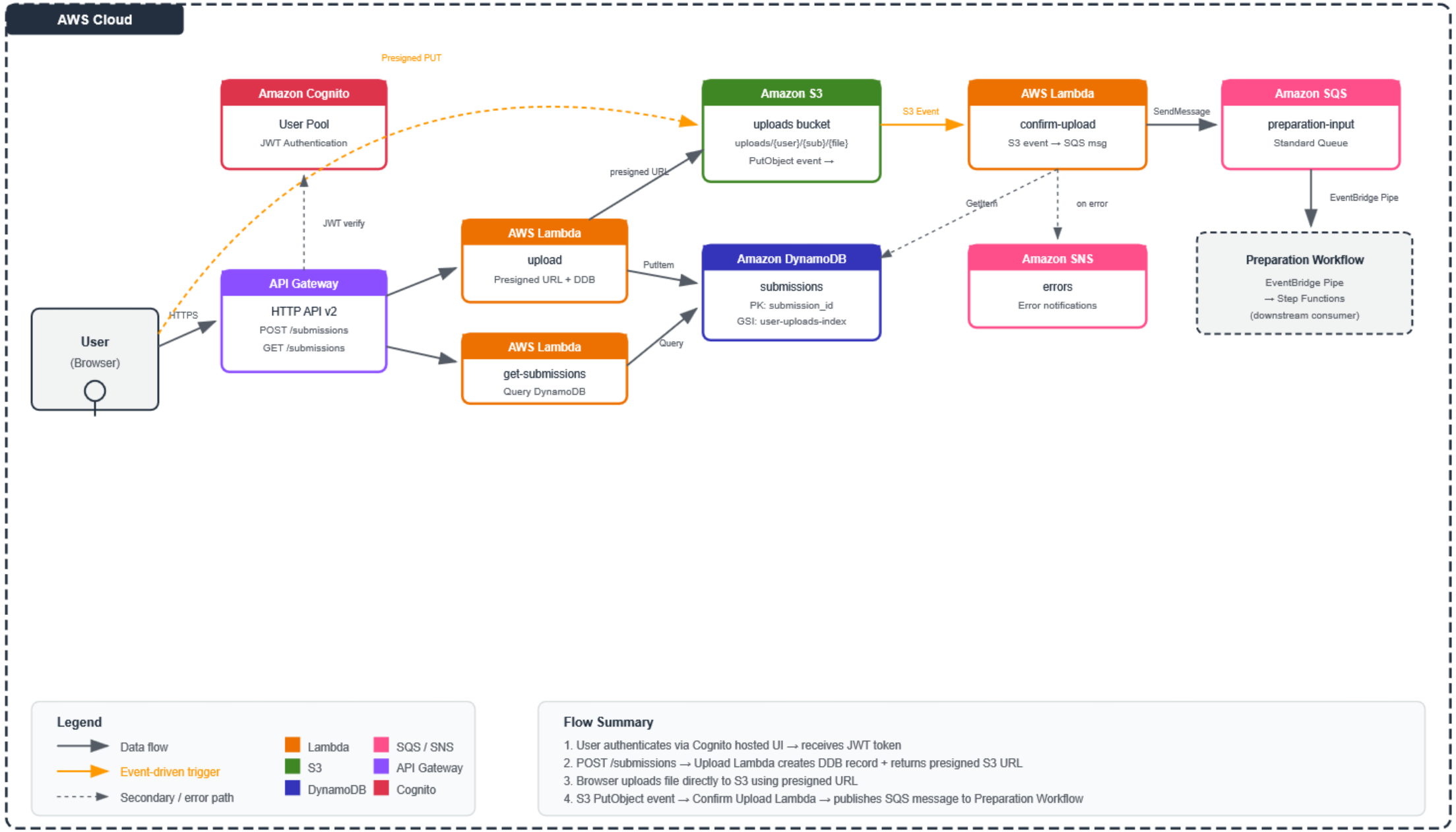
Prepared from the source PowerPoint and three supporting PDFs.

Appendix Material

Architecture Flow Diagrams

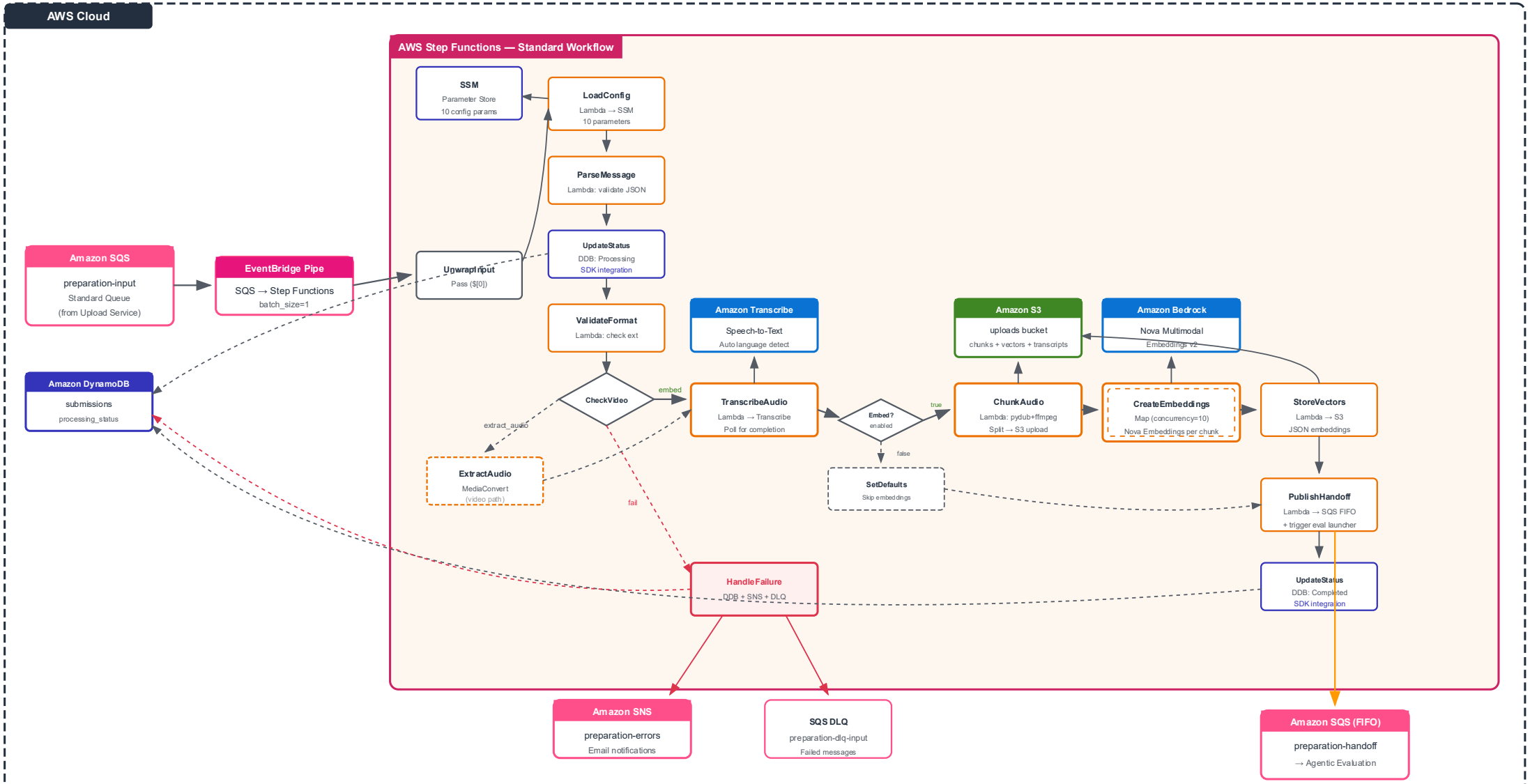
Upload and Storage Service — Architecture Diagram

prescoach-dev-kiro



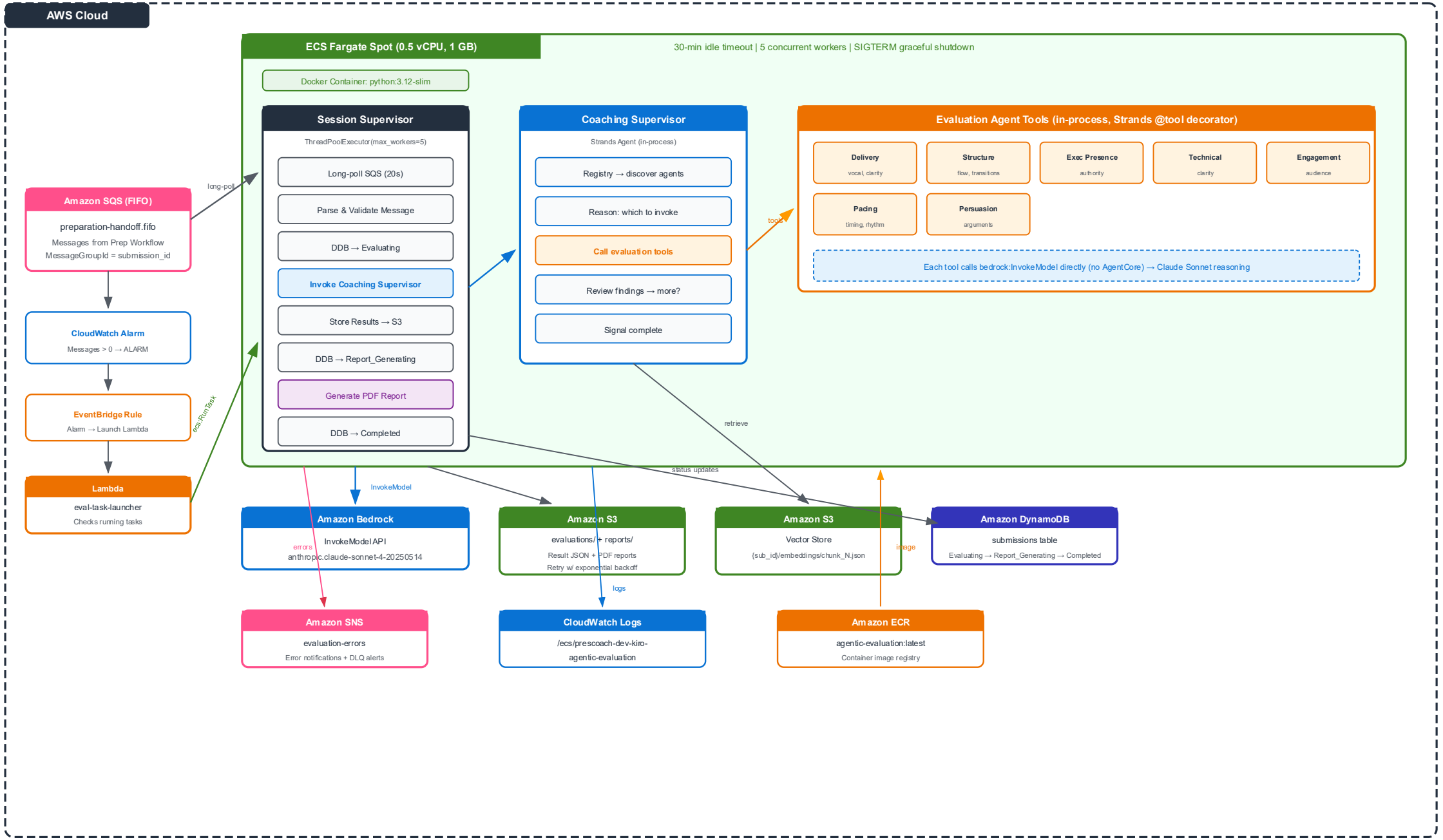
Preparation Workflow — Architecture Diagram

prescoach-dev-kiro | Step Functions Standard Workflow | 10 Lambda Functions



Agentic Evaluation — LOCAL MODE Architecture

ECS Fargate Spot | Strands SDK Local Execution | Direct Bedrock InvokeModel



LOCAL MODE — Key Characteristics

- All agents run in-process inside a single ECS Fargate Spot container (no AgentCore deployment required)
- Strands SDK creates Agent objects locally — each evaluation tool calls bedrock:InvokeModel directly
- Session state is in-memory only (lost when task exits) — no persistent memory across task launches
- Cost-optimized: Fargate Spot (70% discount), 30-min idle timeout, concurrent processing of 5 submissions
- Cold start: ~45-90s (container pull + startup). Kept warm for 30 min after last message.

Cost Estimate

Per evaluation task (5 min): ~$0.0007

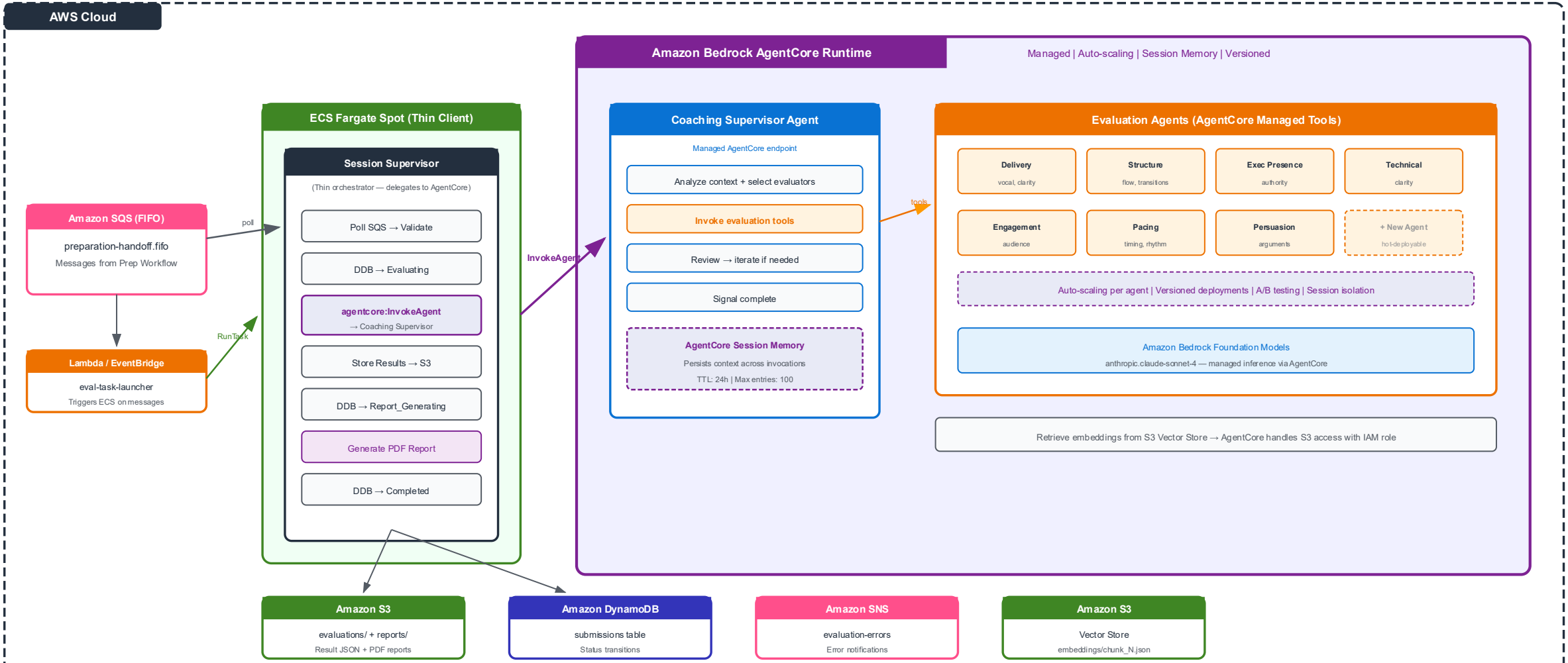
Monthly @ 5/day: ~$0.10

Monthly @ 100/day: ~$2.10

(Bedrock model costs are additional)

Agentic Evaluation — AGENTCORE MODE Architecture

Amazon Bedrock AgentCore Runtime | Managed Agents | Session Memory | Auto-Scaling

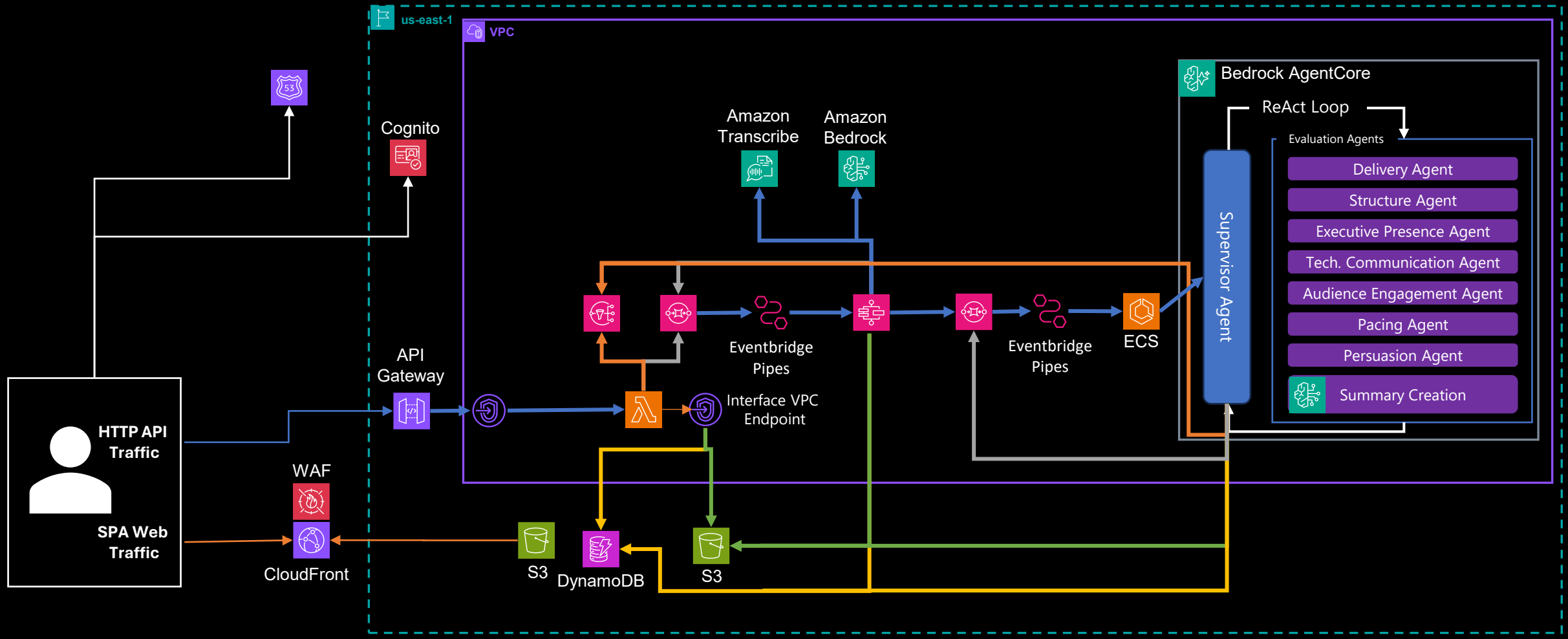


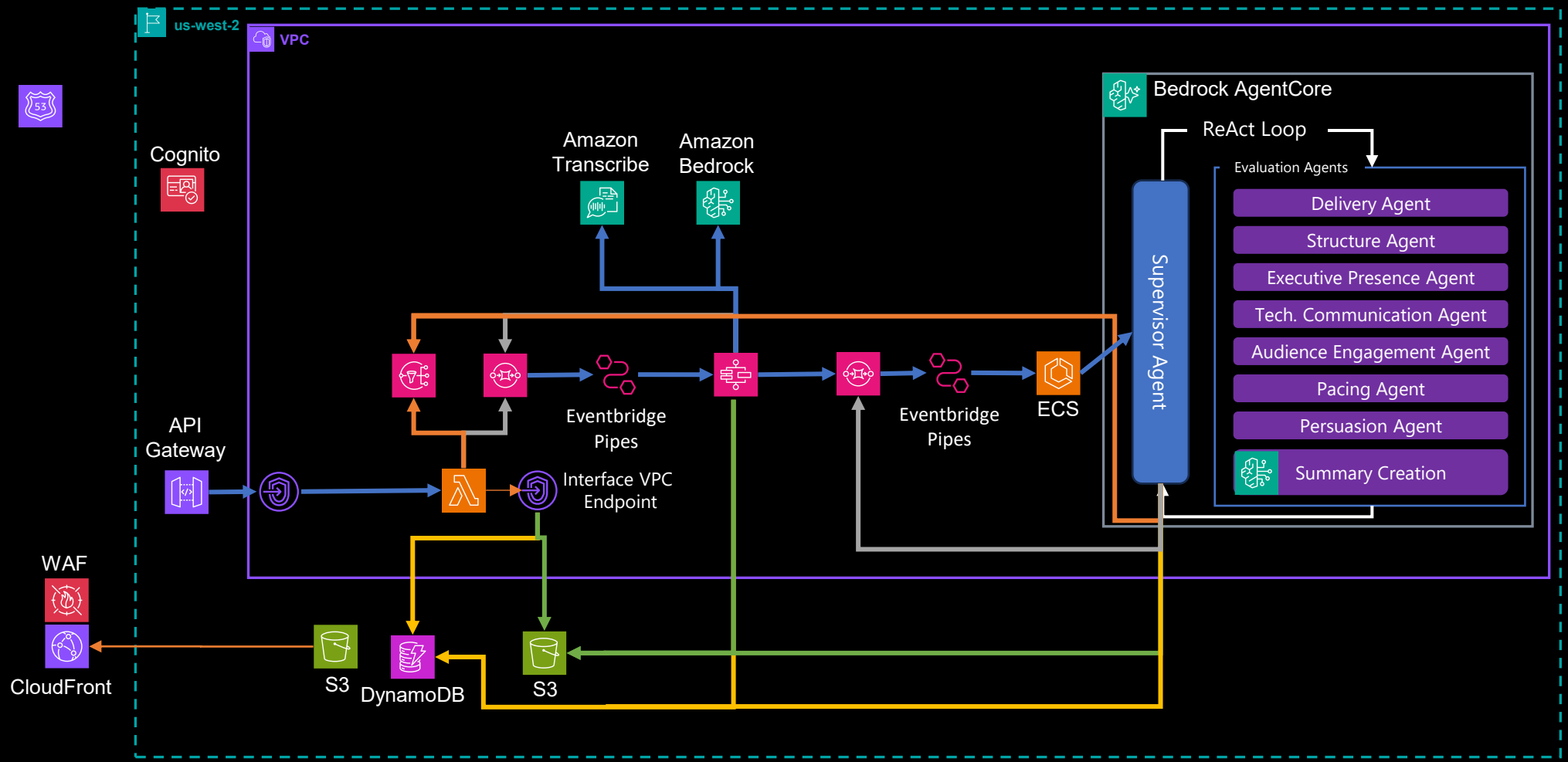
AGENTCORE MODE — Key Differences from Local Mode

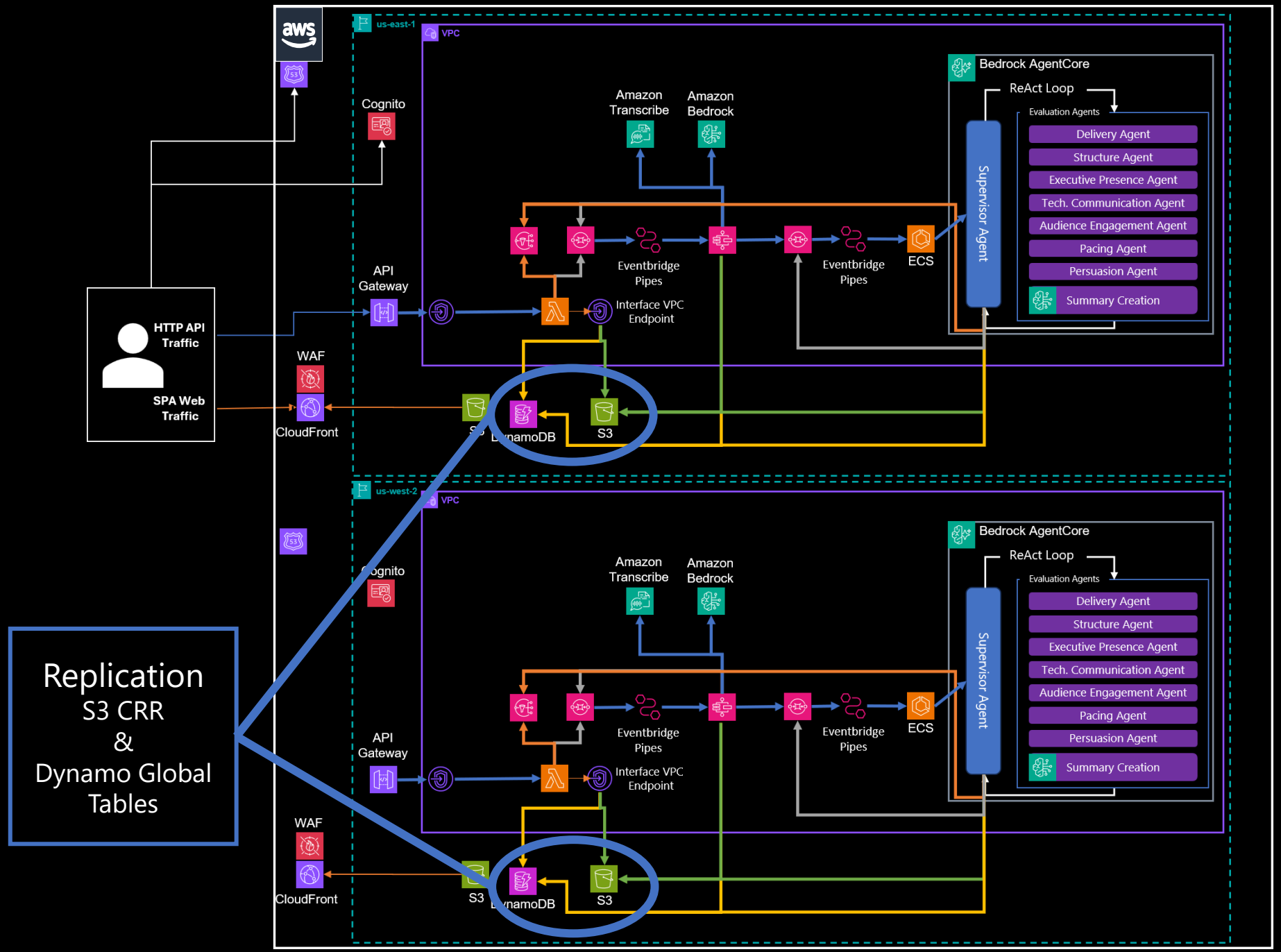
- ECS container is a thin client — only handles SQS polling, DDB updates, S3 storage. All AI reasoning happens in AgentCore.
- Session memory persists across task launches — context is maintained even if the ECS task exits and a new one starts.
- Each evaluation agent can scale independently — high-demand agents get more capacity without scaling the whole container.
- Agent versioning: deploy new evaluator versions with canary rollout, A/B test different prompts without code changes.
- Switch by setting LOCAL_MODE=false + registering agents with AgentCore. Application code stays identical (Strands handles both).

When to Use This Mode

- Need persistent session memory across tasks
- Workload exceeds 100+ evaluations/day
- Want per-agent auto-scaling
- Need agent versioning / A/B testing
- AgentCore is GA with favorable pricing


us-east-1
VPC
Cognito
Amazon Transcribe
Amazon Bedrock
Bedrock AgentCore
ReAct Loop
Evaluation Agents
Delivery Agent
Structure Agent
Executive Presence Agent
Tech. Communication Agent
Audience Engagement Agent
Pacing Agent
Persuasion Agent
Summary Creation
Supervisor Agent
Eventbridge Pipes
Eventbridge Pipes
ECS
API Gateway
Interface VPC Endpoint
HTTP API Traffic
SPA Web Traffic
WAF
CloudFront
S3
DynamoDB
S3


us-west-2
VPC
Cognito
Amazon Transcribe
Amazon Bedrock
Bedrock AgentCore
ReAct Loop
Evaluation Agents
Delivery Agent
Structure Agent
Executive Presence Agent
Tech. Communication Agent
Audience Engagement Agent
Pacing Agent
Persuasion Agent
Summary Creation
Supervisor Agent
Eventbridge Pipes
Eventbridge Pipes
ECS
API Gateway
Interface VPC Endpoint
WAF
CloudFront
S3
DynamoDB
S3


aws
us-east-1
VPC
Cognito
Amazon Transcribe
Amazon Bedrock
Bedrock AgentCore
ReAct Loop
Evaluation Agents
Delivery Agent
Structure Agent
Executive Presence Agent
Tech. Communication Agent
Audience Engagement Agent
Pacing Agent
Persuasion Agent
Summary Creation
Supervisor Agent
Eventbridge Pipes
Eventbridge Pipes
ECS
API Gateway
Interface VPC Endpoint
HTTP API Traffic
SPA Web Traffic
WAF
CloudFront
S3
DynamoDB
us-west-2
VPC
Cognito
Amazon Transcribe
Amazon Bedrock
Bedrock AgentCore
ReAct Loop
Evaluation Agents
Delivery Agent
Structure Agent
Executive Presence Agent
Tech. Communication Agent
Audience Engagement Agent
Pacing Agent
Persuasion Agent
Summary Creation
Supervisor Agent
Eventbridge Pipes
Eventbridge Pipes
ECS
API Gateway
Interface VPC Endpoint
WAF
CloudFront
S3
DynamoDB
Replication
S3 CRR
&
Dynamo Global Tables