High-level Overview

Workflow Decisioning Bridge Service

Thomas Feduk, Jr.
thomasfeduk@gmail.com
June 1st 2025

nintex > aws > ChatGPT

This document serves as a brief and high-level overview of one variation of many AI bridge services I have designed in AWS for enabling various existing workflow systems (Nintex in this example) to delegate complex decision logic to LLM processors such as OpenAI.

Note: This is not a comprehensive Technical Design Document or Design Proposal Document. It is only a high-level design for easy reading and to convey the core concept flows. To keep the document short, it intentionally does not include any of the following:

- Multi-region HA (High-Availability) failover
- Details on recovery or restoration in the event of failures
- Deployment mechanisms and pipelines
- Testing infrastructure and harnesses
- Error handling, monitoring, logging or alerting processes
- Detailed security/IAM handling
- Nuanced infrastructure specifications such as API Gateway types, stages, generation methods Lambda and SQS attributes, DynamoDB properties etc.
- API specifications or LLM Prompt details/ELT conversion rules

Typically if this were a formal proposal I would first draft a comparison of the various alternative approaches:

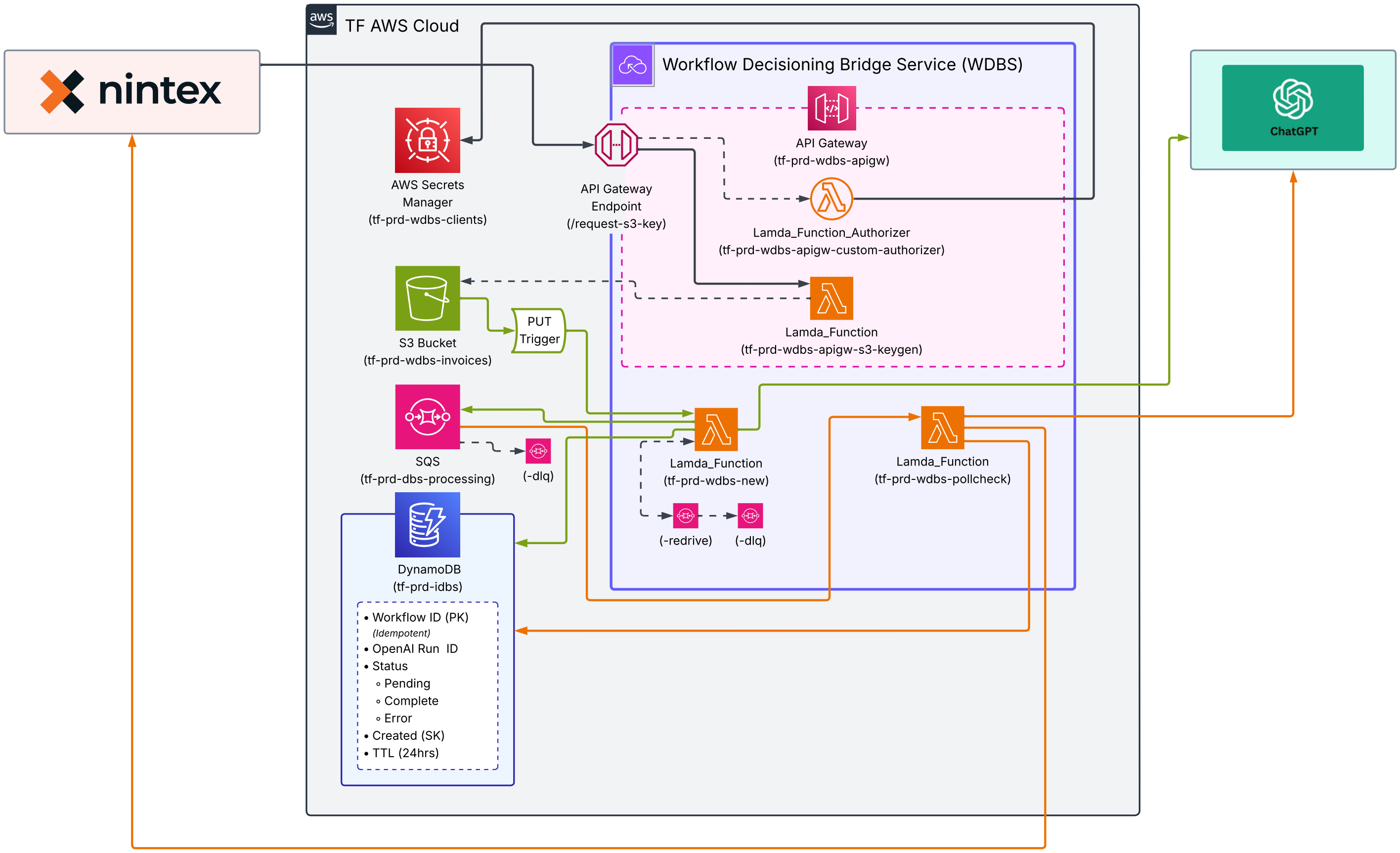
- Per-event vs batched
- Full serverless vs Fargate/ECS vs Kubernetes vs EC2
- Step-Function vs SQS

For this document I simply chose a preference I felt was balanced with price given a set of assumptions based on most common business use cases: A fully serverless implementation due to nature of pay-as-you-go infrastructure priced incredibly inexpensive.

A proper Design Proposal Document would also provide detailed comparative cost breakdown of the differently priced models and alternatives, including pre-processing price and token optimizations that could be applied (such as pdf cropping to known areas of decision, text pre-extraction to reduce token count rather than utilizing full vision etc.).

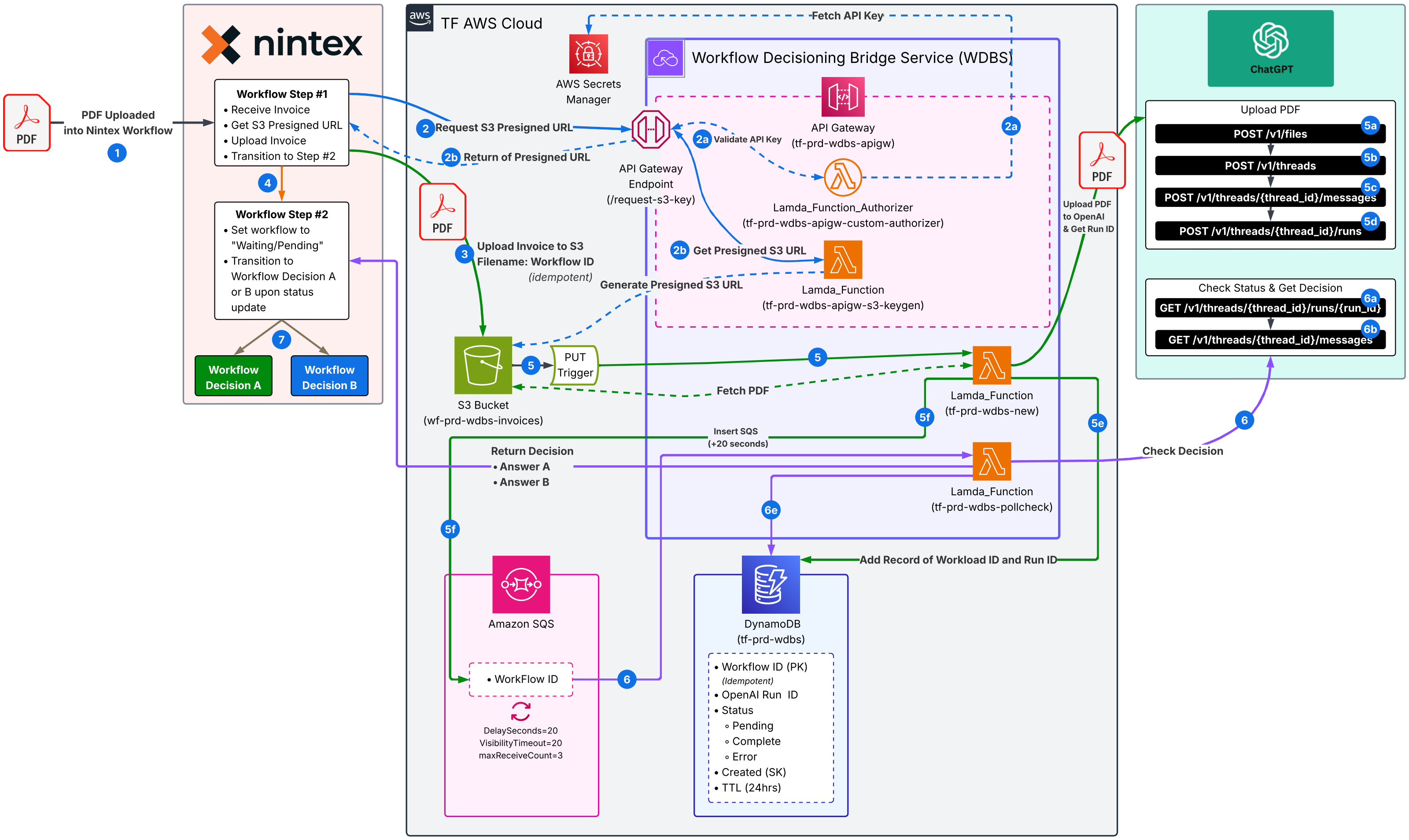
Workflow Decisioning Bridge Service — Architecture Diagram (Serverless)

Thomas Feduk 6/1/25


nintex
TF AWS Cloud
Workflow Decisioning Bridge Service (WDBS)
ChatGPT
AWS Secrets Manager (tf-prd-wdbs-clients)
API Gateway Endpoint (/request-s3-key)
API Gateway (tf-prd-wdbs-apigw)
Lamda_Function_Authorizer (tf-prd-wdbs-apigw-custom-authorizer)
Lamda_Function (tf-prd-wdbs-apigw-s3-keygen)
S3 Bucket (tf-prd-wdbs-invoices)
PUT Trigger
SQS (tf-prd-dbs-processing)
(-dlq)
Lamda_Function (tf-prd-wdbs-new)
(-redrive)
(-dlq)
Lamda_Function (tf-prd-wdbs-pollcheck)
DynamoDB (tf-prd-idbs)
• Workflow ID (PK)
(Idempotent)
• OpenAI Run ID
• Status
◦ Pending
◦ Complete
◦ Error
• Created (SK)
• TTL (24hrs)

Workflow Decisioning Bridge Service — Flow Diagram (Serverless)

Thomas Feduk 6/1/25



Flow Steps

1 PDF is ingested into Nintex

2 Nintex requests S3 Presigned URL from WDBS

- 2a WDBS API Gateway authorizer fetches API key from Secrets Manager to check auth
- 2b WDBS -keygen Lambda creates S3 Presigned URL and returns to Nintex

3 Nintex makes a PUT HTTP request to S3 using the Presigned URL

4 Nintex transitions the workflow to "Pending" state

5 S3 PUT trigger invokes the WDBS -new Lambda which uploads the PDF to OpenAI

- 5a File is submitted via POST call to /files
- 5b A new OpenAI thread is created via POST call to /threads
- 5c The engineered prompt query is submitted via POST call to /messages
- 5d The thread is started via POST call to /run
- 5e The entry is recorded in DynamoDB for status handling and idempotency protection
- 5f An SQS is created with DelaySeconds+VisibilityTimeout=20 seconds and maxReceiveCount=3 to DLQ

6 Delayed SQS poll invokes -pollcheck Lamda to update Nintex Workflow with decision

- 6a A GET call to /runs is made to confirm the prompt query has completed
- 6b The prompt response/decisoin is read via a GET call to /messages
- 6c The status is updated in DynamoDB and evicted from the SQS
- 6d The decision is ETL'd and submitted to the pending Nintex Workflow webhook endpoint

7 Nintex receives the decision via a webhook call and transitions the Workflow to Decision A or B

Price Tables

OpenAI API Pricing

Model	Input Cost (per 1K tokens)	Output Cost (per 1K tokens)	Input Cost (per 1M tokens)	Output Cost (per 1M tokens)
GPT-3.5 Turbo	$0.0015	$0.0020	$1.50	$2.00
GPT-4	$0.0300	$0.0600	$30.00	$60.00
GPT-4 Turbo	$0.0100	$0.0300	$10.00	$30.00
GPT-4o	$0.0050	$0.0200	$5.00	$20.00
GPT-4o Mini	$0.00015	$0.00060	$0.15	$0.60
GPT-4.1	$0.0020	$0.0080	$2.00	$8.00
GPT-4.1 Mini	$0.00040	$0.00160	$0.40	$1.60
GPT-4.1 Nano	$0.00010	$0.00040	$0.10	$0.40
OpenAI o3	$0.0100	$0.0400	$10.00	$40.00
OpenAI o4-mini	$0.00110	$0.00440	$1.10	$4.40
OpenAI o1	$0.1500	$0.6000	$150.00	$600.00

- **Input tokens** refer to the tokens in your prompt or input text.
- **Output tokens** are the tokens generated by the model in response.
- **GPT-4o Mini** offers the most cost-effective solution for lightweight tasks, especially suitable for high-volume processing like invoice parsing.
- **GPT-4 Turbo** provides a balance between performance and cost, ideal for more complex tasks requiring higher accuracy.
- **OpenAI o1** is designed for advanced reasoning tasks but comes at a higher cost, making it suitable for specialized applications.

AWS Infrastructure Pricing

Service	Pricing Unit	Approximate Cost	Notes
S3 – PUT/GET	$0.005 per 1,000 requests	$0.000005 per request	Applies to uploads (PUT) or downloads (GET)
S3 – Storage	$0.023 per GB/month	$0.000023 per MB/month	Standard storage class
Lambda	$0.20 per 1M invocations	$0.0000002 per invocation	Plus execution time below
	$0.00001667 per GB-s (128MB = 0.125 GB)	~ $0.000002 per 1s execution at 128MB	Compute duration cost
API Gateway – REST	$3.50 per 1M requests	$0.0000035 per request	Public-facing REST API
SQS – Standard	$0.40 per 1M requests	$0.0000004 per message	Includes both send and receive
DynamoDB – On-demand	$1.25 per 1M write units / $0.25 per 1M read units	$0.00000125 per write / $0.00000025 per read	No provisioning needed
Secrets Manager	$0.40 per secret per month	$0.40 flat + $0.05 per 10,000 API calls	First 30 days per secret free

Monthly Cost Estimate Breakdown

Assumptions	
Invoices per month	100,000
Input tokens per invoice	1,500
Output tokens per invoice	3–5 tokens max (1 word only)
Avg. invoice file size	500 KB
Model	**GPT-4o with vision**
Assumptions	No retries, no DLQ, average Lambda durations (0.5–1s), one poll per invoice

Monthly Cost Breakdown for 100,000 Invoices

Service	Description	Usage Estimate	Unit Cost	Monthly Cost
OpenAI GPT-4o	Input: 1,500 tokens/invoice	150M tokens @ $0.005/1K	$0.0075/invoice	$750.00
	Output: ~5 tokens (1 word) per invoice	500K tokens @ $0.02/1K	$0.0001/invoice	$10.00
S3 (PUT Requests)	Invoice uploads (via presigned URL)	100,000 PUTs	$0.005 per 1,000	$0.50
S3 (Storage)	~500 KB per invoice	~50 GB @ $0.023/GB	—	$1.15
Lambda	new: 3s, pollcheck: 1s, keygen: 0.5s	300,000 calls @ 128MB	Varies by duration	$0.94
SQS	One message per invoice	100,000 msgs @ $0.40/1M	$0.0000004 per message	$0.04
DynamoDB	Track run_id, workflow status	~200K ops (read+write)	~$1.25 per million writes	$0.25
API Gateway	Presigned URL requests from Nintex	100,000 requests @ $3.50/1M	$0.0000035 per request	$0.35
Secrets Manager	Auth key for API Gateway authorizer	1 secret (frequent use)	$0.40/month + access fees	$0.40

Per-Invoice Estimate @ 1,500 tokens per invoice:

- **Total: ~$0.0076383**
- **Breakdown:**
  - OpenAI: ~$0.0076/ea *(~$760.00/mo)*
  - AWS Services: ~$0.0000383/ea ($3.83/mo)

Total Estimated Monthly Cost @ 1,500 tokens/invoice: **~$763.83**

Per-Invoice Estimate @ 2,000 tokens per invoice:

- **Total: ~$ 0.0101383**
- **Breakdown:**
  - OpenAI: ~$0.01010/ea *(~$1,010.00/mo)*
  - AWS Services: ~$0.0000383/ea *(~$3.83/mo)*

Total Estimated Monthly Cost @ 2,000 tokens/invoice: **~$1,013.83**