

AI-Powered Newsletter Insights Generator Automation For MAX Smith KDP LLC

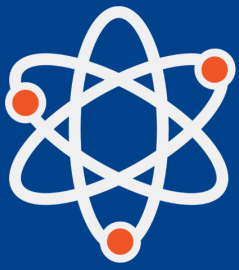


Table of Contents

Contents

1 Company Introduction.....	3
1.1 Services Portfolio.....	4
1.2 Implementation Approach.....	4
1.3 Working Methodology.....	5
2. Primary Goal for PoC Phase.....	6
3. Scope of work for PoC Phase.....	6
4. High-Level Solution Architecture for PoC Phase.....	7
4.1 High-Level Solution Architecture for PoC Phase.....	7
4.2 Technology Stack.....	10
5. Deliverables.....	10
6. Assumptions and Deliverables.....	11
7. Acceptance Criteria.....	11
8. Work Breakdown Structure & Estimates.....	12
9. Resource Plan-Development Cost:.....	13


1 Company Introduction

With a humble beginning in 2010, Fission Labs was born with a simple aim to leverage our engineering capabilities and help organizations overcome technological challenges.

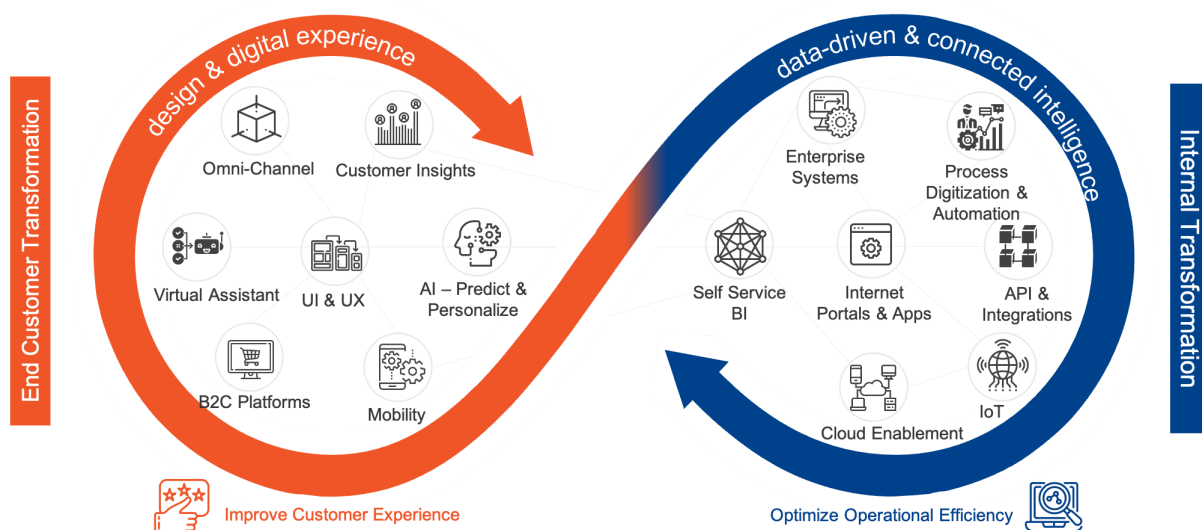
Vision  Fission Labs' mission is to help our clients achieve a sustainable digital transformation that's in sync with their business goals. We strive to reach our goals by harnessing innovation and engineering excellence as an instrument to create symphonies between business processes and underlying technologies.	A family of 450+ Engineers	Innovation Driven Active Clients  Culture
	250+ Products Delivered	Product Development  DNA
	Collaborate with thought leaders for innovation Global Partnerships	A fast-moving, collaborative work culture

HQ: Sunnyvale, CA
 Development Center: Dallas, TX
 Offshore Development Center: Hyderabad, India

SI Partners of



Our mission is to help our clients achieve a sustainable digital transformation that's in sync with their business goals. We strive to reach our goals by harnessing innovation and engineering excellence as an instrument to create symphonies between business processes and underlying technologies.



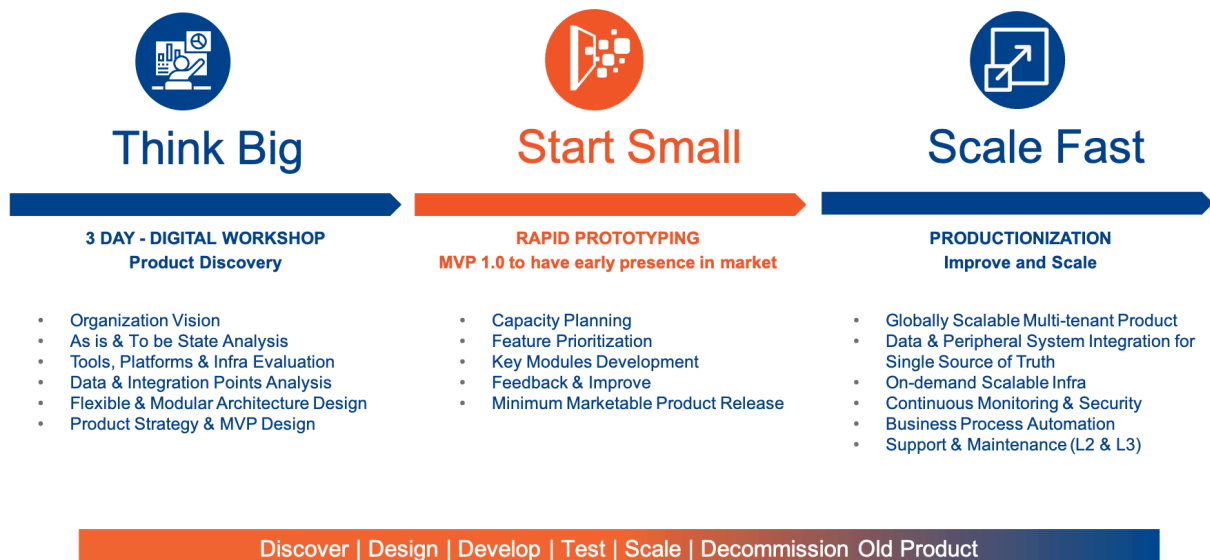
Fission Labs Inside Out Digital Transformation Strategy

1.1 Services Portfolio



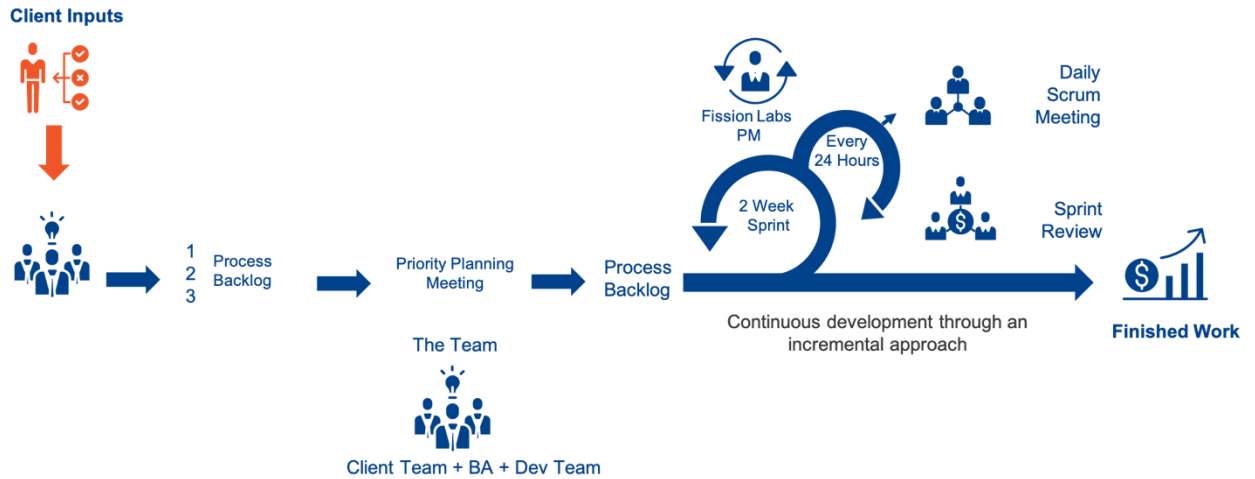
1.2 Implementation Approach

Our skilled leadership team, solution architects, and functional and audit experts help you boost efficiency and tap into a knowledge base so you can reduce operational costs, accelerate time to market, and boost your ROI.



1.3 Working Methodology

We follow agile scrum methodology, which is a sprint-based model that works on a continuous incremental delivery approach to facilitate collaboration, incorporate timely feedback, is adaptable, reduce costs, & enhances faster delivery.



2. Primary Goal for PoC Phase

This Statement of Work (SoW) outlines the scope, approach, deliverables, and responsibilities for implementing a PoC phase of an AI-powered newsletter and insight delivery platform. The objective is to validate a focused, end-to-end workflow that transforms Reddit data into meaningful GenAI-powered insights and sends them to subscribed users via email.

3. Scope of work for PoC Phase

In-Scope Features

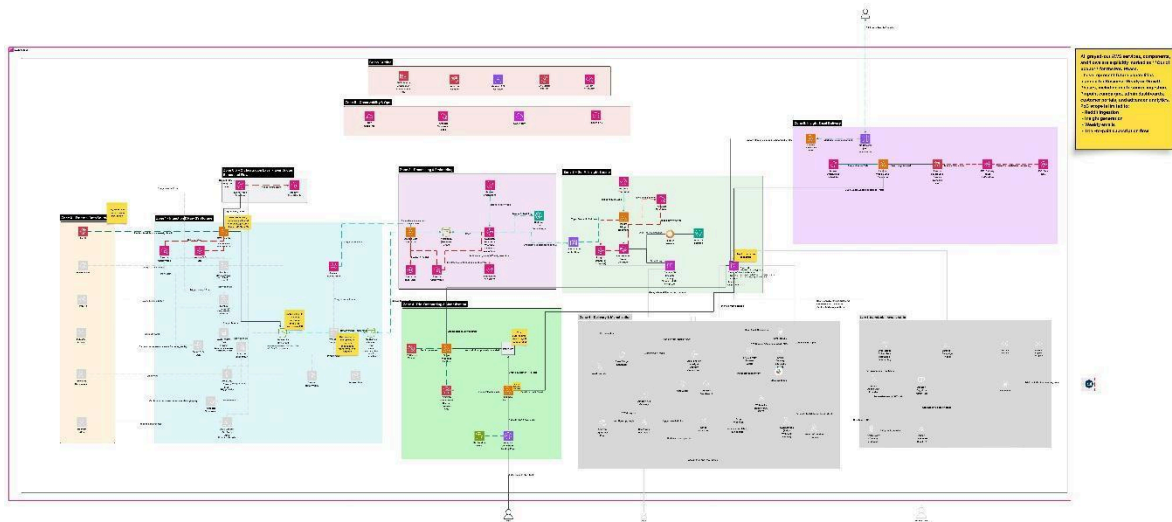
- **Reddit Data Ingestion:** Lambda scheduled weekly via EventBridge from **five** subreddits
- **ETL Pipeline:** Normalize and clean data, store in S3 Processed bucket.
- **Vectorization:** Use Bedrock Titan to generate embeddings.
- **Search Indexing:** Store vectors in OpenSearch.
- **Insight Generation:** Retrieve Top-K from OpenSearch and invoke Claude (or chosen Bedrock compatible LLM) via Bedrock.
- **Email Delivery:** Send weekly insights via SES.
- **Trial Onboarding:** Landing page collects email, invokes Stripe for 7-day free trial.
- **Subscription Management:** Stripe Checkout and Webhook to update plan status in DynamoDB.
- **Unsubscribe Handler:** Endpoint to mark user as unsubscribed in DynamoDB

Out-of-Scope (PoC Phase)

- Admin dashboard
- Customer self-service portal
- Additional data sources (e.g., Twitter, Hacker News)
- Amazon Pinpoint campaigns
- CI/CD automation pipelines

4. High-Level Solution Architecture for PoC Phase

4.1 High-Level Solution Architecture for PoC Phase



Group-Wise Architecture Summary

Group 0/1 – Data Ingestion (Reddit Only)

- **Objective:** Periodically collect posts from a specified subreddit.
- **Components:**
 - **Amazon EventBridge Scheduler:** Triggers ingestion weekly.
 - **AWS Lambda (Reddit Fetcher):** Uses Reddit API to fetch new posts.
 - **Amazon S3 (Raw Zone):** Stores raw Reddit JSON data.
- **Notes:**
 - Fully serverless, scalable, and event-driven.
 - Built for one data source (Reddit) in PoC; others to follow in future phases.

Group 2 – ETL & Semantic Embedding

- **Objective:** Clean, normalize, and vectorize Reddit posts for semantic analysis.
- **Components:**
 - **AWS Lambda (Data Normalizer):** Processes raw posts, stores cleaned data in S3.
 - **Amazon S3 (Processed Zone):** Stores structured, normalized content.
 - **AWS Step Functions (Embedding Orchestrator):** Orchestrates the following:
 - **Amazon Bedrock Titan Embeddings:** Converts posts into dense vector embeddings.
 - **Amazon OpenSearch (Vector Index):** Stores semantic embeddings for future retrieval.
- **Notes:**
 - Vectorization enables meaning-based similarity rather than keyword-based.

Group 3 – GenAI Insight Engine

- **Objective:** Retrieve relevant content and generate meaningful insights.
- **Components:**
 - **EventBridge Scheduler:** Weekly trigger to initiate insight generation.
 - **Lambda (Top-K Retriever):** Queries OpenSearch to retrieve top-K similar chunks.
 - **Step Functions (Insight Generator):**
 - Invokes **Claude** (or chosen LLM) **via Bedrock** to generate summaries/insights.
 - Stores insights in **Amazon DynamoDB (Insights Table)**.
- **Notes:**
 - Fully RAG-capable structure with OpenSearch as vector DB.

Group 4 – Subscription & Trial Onboarding

- **Objective:** Allow users to subscribe, trial, and upgrade to a paid plan.
- **Components:**
 - **React Landing Page:** Collects user emails for free trials.
 - **Lambda (/start-trial):** Invokes **Stripe Checkout** for 7-day trial.
 - **Stripe Webhooks → Lambda:** Updates **DynamoDB (Users Table)** with subscription status.
- **Notes:**
 - Simple one-plan trial-to-paid flow designed for PoC.

Group 5 – Email Delivery

- **Objective:** Send generated insights to subscribers.
- **Components:**
 - **Lambda (Weekly Email Sender):** Fetches insights, sends via **Amazon SES**.
 - **Unsubscribe Lambda:** Handles unsubscribe requests.
 - **SES DLQ via SNS → SQS:** Captures email delivery failures for diagnostics.

Group 6 – Monitoring, Logging & Ops

- **Objective:** Ensure observability, reliability, and secure operations.
- **Components:**
 - **Amazon CloudWatch:** Logging, metrics, alarms.
 - **AWS Secrets Manager:** Secure storage of API keys (e.g., Reddit, Stripe).
 - **Dead Letter Queues (SQS):** Captures unhandled Lambda or SES errors.

High-Level Process Flow Summary

1. Ingest Data

- EventBridge triggers Reddit Lambda
- Raw data stored in S3

2. Process & Embed

- Lambda cleans data
- Processed data stored in S3
- Step Function calls Titan Embeddings
- Vectors stored in OpenSearch

3. Generate Insights

- EventBridge triggers Lambda
- Top-K chunks retrieved from OpenSearch
- Step Function invokes Claude (or chosen LLM) with RAG context (Top-K)
- Insights stored in DynamoDB

4. Deliver to Users

- Lambda fetches insights
- SES sends email
- Optional DLQ and unsubscribe flows handled

5. User Onboarding

- User enters email on landing page
- Trial started via Stripe
- Subscription status tracked in DynamoDB via Webhook

Key Considerations

FloTorch Gateway Integration

As part of the proposed solution architecture, **FloTorch Gateway** will play a critical role in both the experimentation and operational phases:

- **Model Experimentation & Benchmarking:** During the initial implementation phase, FloTorch will serve as the unified gateway for evaluating and selecting the most suitable LLMs for this use case. This includes benchmarking models based on accuracy, latency, and cost-effectiveness in generating insights from the Reddit.
- **Runtime Gateway Post-Build:** Once the appropriate LLM is selected and the build phase is complete, FloTorch will continue to be integrated for:
 - **Model Observability & Logging**
 - **Flexibility to Swap/Upgrade Models**
 - **Centralized Monitoring of Agent Interactions**

This strategic inclusion ensures vendor neutrality, operational flexibility, and transparency in model performance—supporting long-term maintainability and adaptability of the solution.

4.2 Technology Stack

- **AWS Services:** Lambda, S3, EventBridge, Step Functions, Bedrock (Claude & Titan), OpenSearch, DynamoDB, SES, SNS, SQS, Secrets Manager, CloudWatch
- **Third Party:** Reddit API, Stripe API
- **Language:** Python
- **Frameworks:** FastAPI (for API Gateway integrations)
- **Frontend:** React.js (for landing page development)
- **LLM Experimentation & Gateway:** FloTorch
- **IaC:** Terraform

5. Deliverables

- End-to-end working deployment in client AWS account
- Weekly insights sent to subscribed users
- Working trial-to-paid Stripe flow
- Functional unsubscribe URL handler flow

6. Assumptions and Deliverables

- An AWS account with required permissions will be provided by the client.
- The client will provision a Stripe account and either share credentials or configure webhooks.
- The client will verify domain and sender email ID in SES.
- The client will specify a maximum **five** subreddit(s) for Reddit scraping.

UX Design

- UX design work for the landing page must start in the PoC phase with a strong collaboration with the client team and approved by the client before the implementation start date.

Costs

- All the platform costs related to the cloud, infrastructure costs, software, and third-party integration licenses, and component licenses should be borne by the client.

Code Repository / Domain

- Client will provide a repository/version control system for maintaining the application codebase (e.g., GitHub).
- Client will provide and authorize the use of web domains and SSL certificates.

7. Acceptance Criteria

- Trial flow working and emails received.
- Payments processed via Stripe.
- DynamoDB accurately reflects user plans

8. Work Breakdown Structure & Estimates

Milestone	Title	Description
M1	Reddit Lambda Setup	Scheduled ingestion from specified five subreddits
M2	ETL Pipeline	Data cleaning, chunking, S3 partitioning
M3	Embedding & Indexing	Generate vectors with Titan, index in OpenSearch
M3.5	LLM Experimentation with FloTorch	LLM Experimentation with FloTorch
M4	Insight Engine	Prompt composition, call Claude (or chosen LLM from M3.5), store insights
M5	Weekly Email Delivery	SES delivery, unsubscribe link, DLQ tracking
M6	Stripe Trial & Plan Management	Trial initiation, webhook → DynamoDB update
M7	Landing Page (React)	Email capture form with backend integration
M8	Monitoring & Logging	CloudWatch, DLQ setup, alert mechanisms
M9	Infrastructure as Code (Terraform)	Deploy core infra using modular, reusable IaC templates

Consolidated Poc Phase Estimation

PHASE	High-level Estimates (Person-days)						Total
	Lead Backend	Senior Frontend	Senior DevOps Engineer	Solution Architect	Senior QA Engineer	Project Manager	
Poc Phase Estimation	40	10	25	20	20	20	125

9. Resource Plan-Development Cost:

Role	Involvement in Person days	Effort In Person days	Monthly Cost (USD)	Per Day (USD)	Total Cost (USD)
Lead Backend Engineer(1)	30	40	\$6,000	\$273	\$10,909
Senior Frontend Engineer(0.5)	20	10	\$5,000	\$227	\$2,273
Senior DevOps Engineer(0.5)	30	25	\$5,000	\$227	\$5,682
Senior QA Engineer(0.5)	20	10	\$4,000	\$182	\$1,818
Solution Architect(0.5)	30	20	\$7,200	\$327	\$6,545
Project Manager(0.5)	30	20	\$6,500	\$295	\$5,909
Total cost (USD) before taxes					\$33,136
TAX					
Total Cost					

Note: Infrastructure, licenses, and hardware costs will be billed on actuals for both development and implementation phases to client