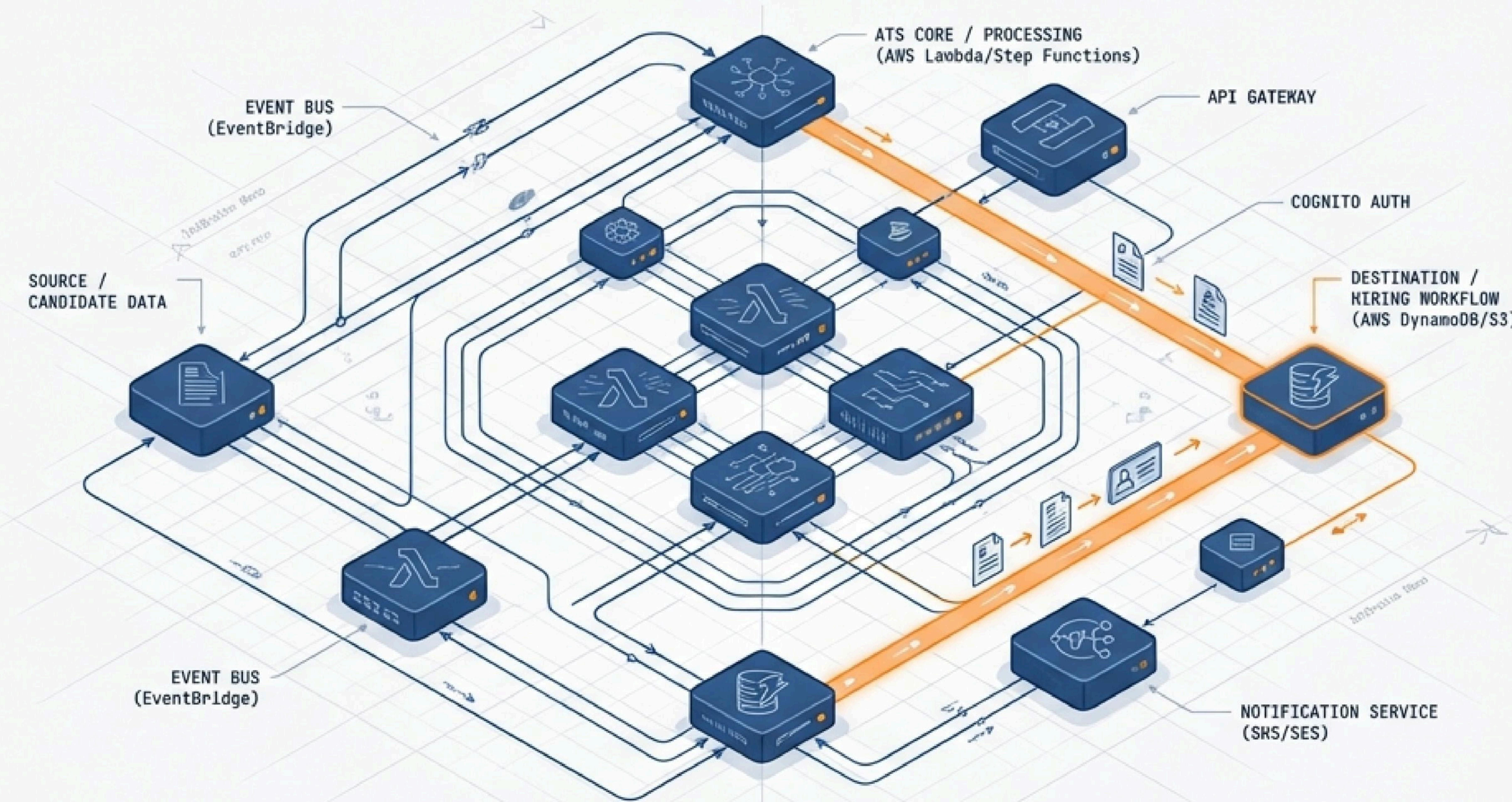


Smart ATS: Sistema Cloud-Native Serverless

Automazione intelligente per la selezione del personale
basata su architettura AWS Event-Driven.



Il Contesto: Dal Caos Manuale all'Automazione



La Sfida

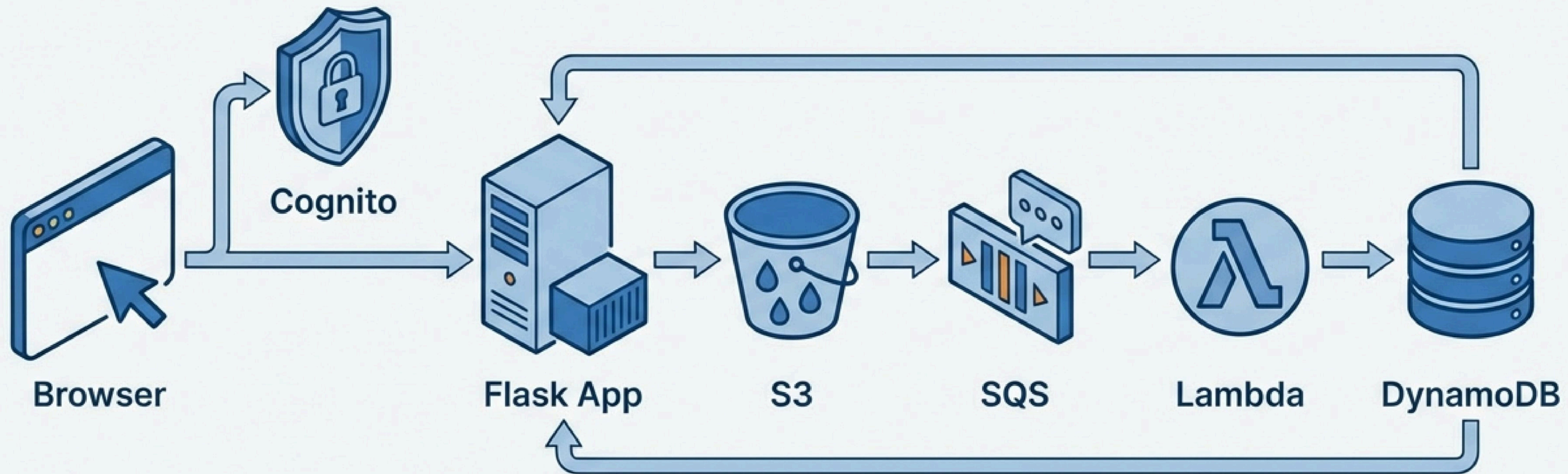
- Screening manuale di documenti non strutturati
- Processo lento, soggetto a bias e difficile da scalare



La Soluzione Smart ATS

- **Processing Asincrono:** Gestione alti volumi
- **Ranking Oggettivo:** Algoritmo basato su criteri
- **Architettura Reattiva:** Scalabilità automatica

Panoramica Architetturale



Frontend

Python Flask

Backend & Logic

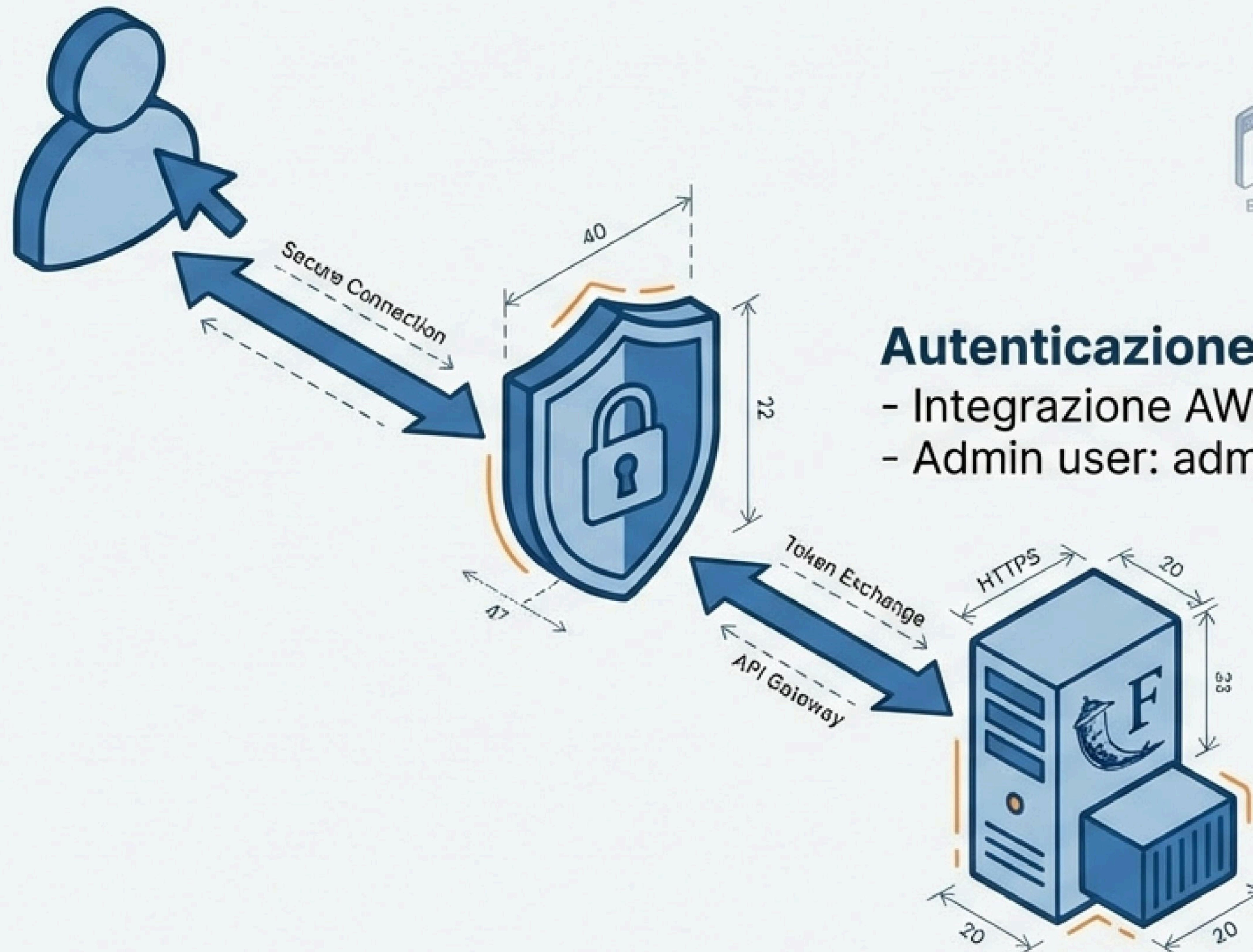
AWS Serverless
Event-Driven

Storage

NoSQL
Database

7 Servizi AWS integrati

Accesso e Sicurezza



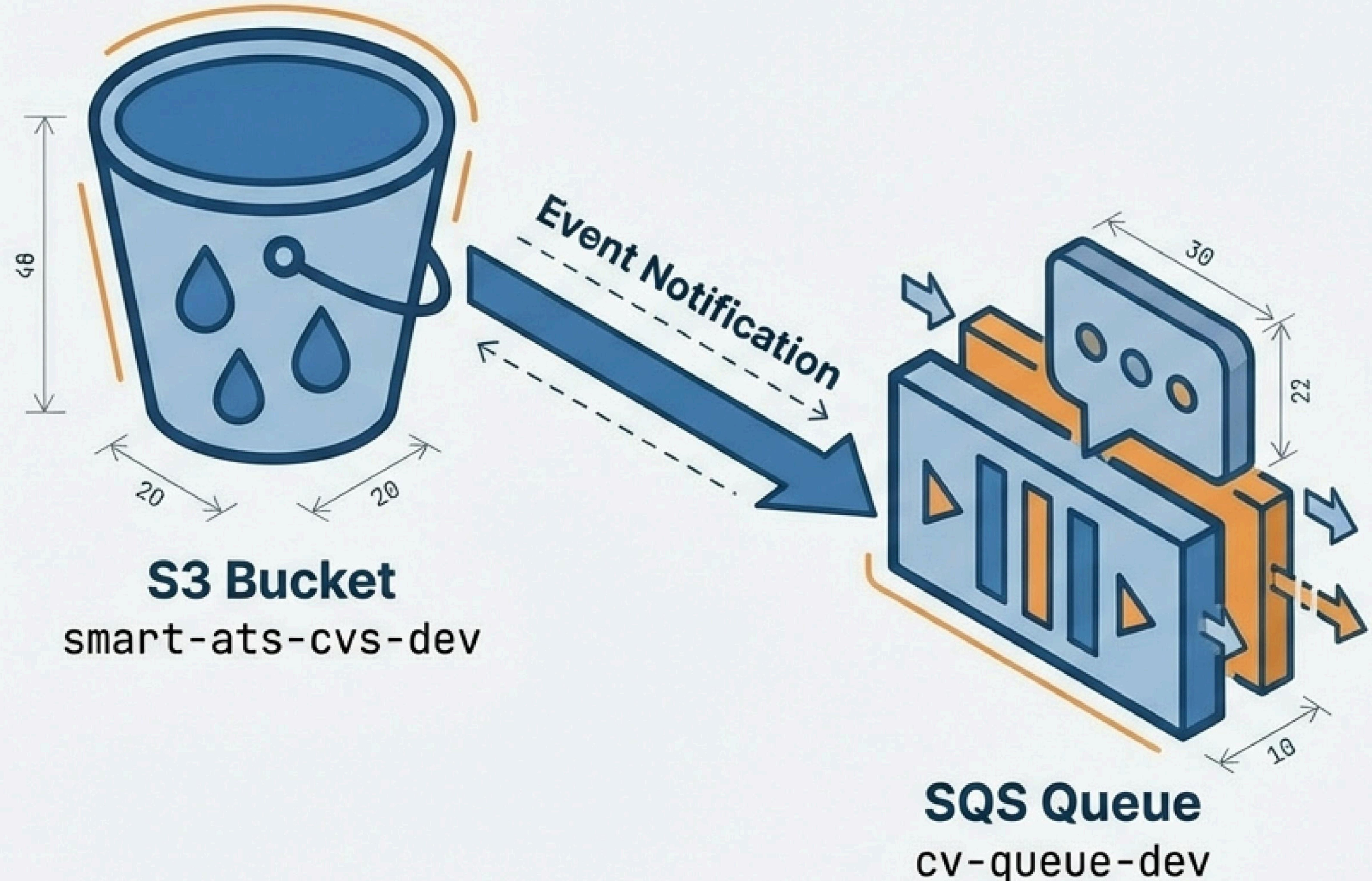
Autenticazione Gestita

- Integrazione AWS Cognito
- Admin user: admin@smartats.com

Interfaccia Recruiter

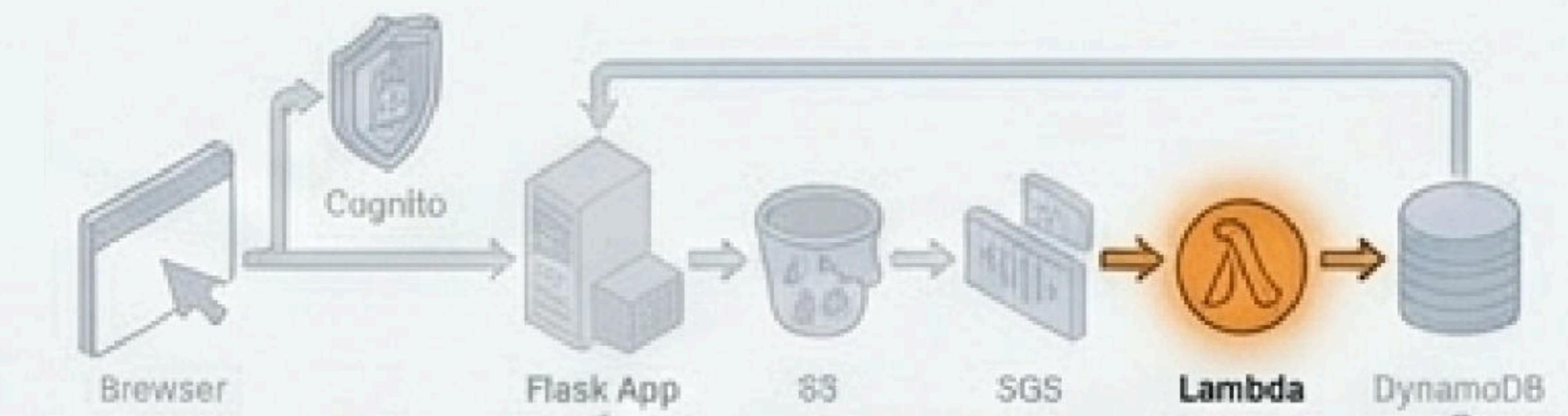
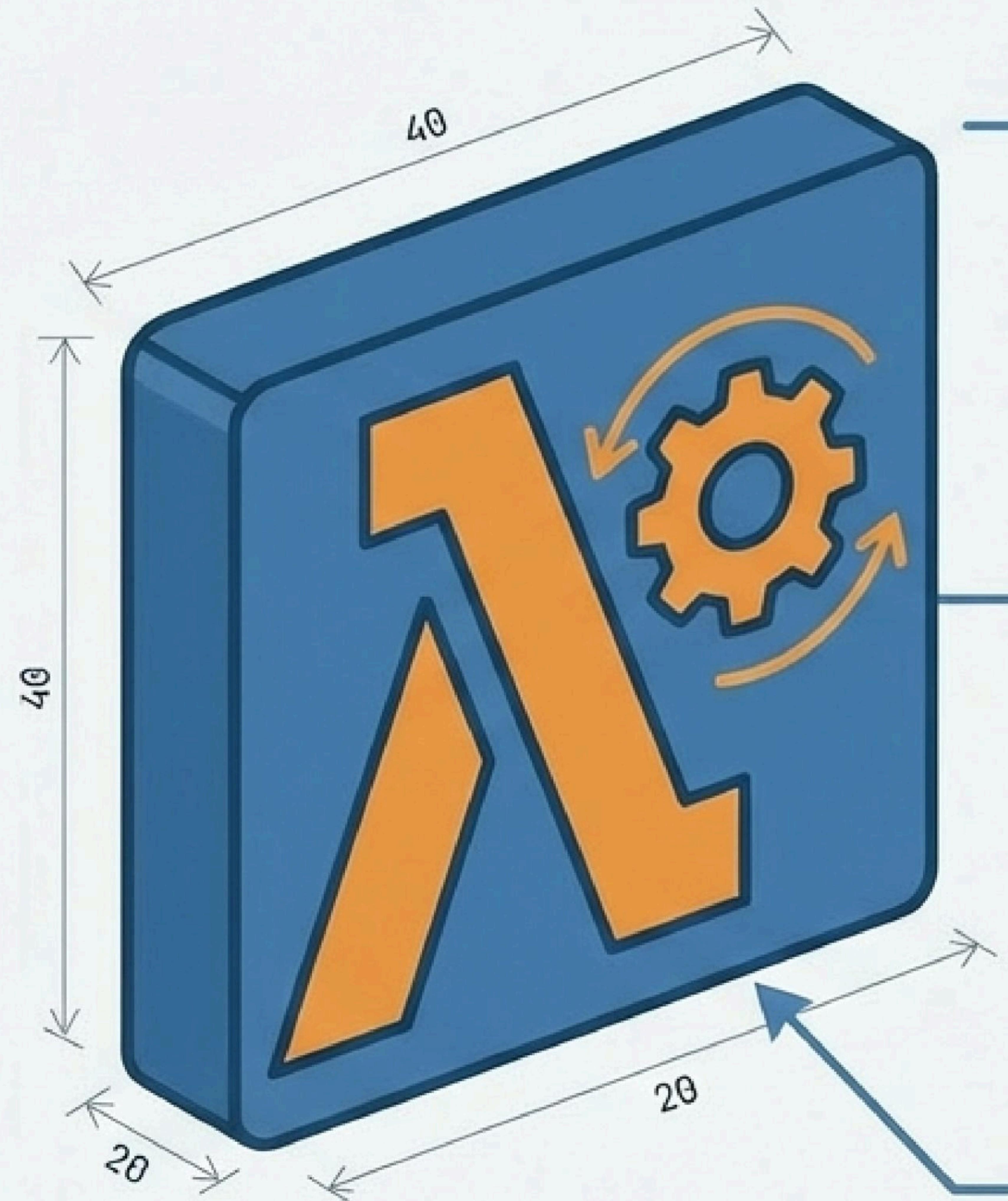
- Dashboard in Flask (Python)
- Containerizzata con Docker
- Funzionalità: Upload sicuro
- Funzionalità: Upload sicuro & Graduatoria Real-time

L'Innesco Asincrono: S3 e SQS



- 1. Event Notification:**
 - Upload su S3 genera evento.
- 2. Buffering:**
 - La coda SQS riceve l'evento.
- 3. Resilienza:**
 - Disaccoppiamento velocità upload vs. elaborazione.

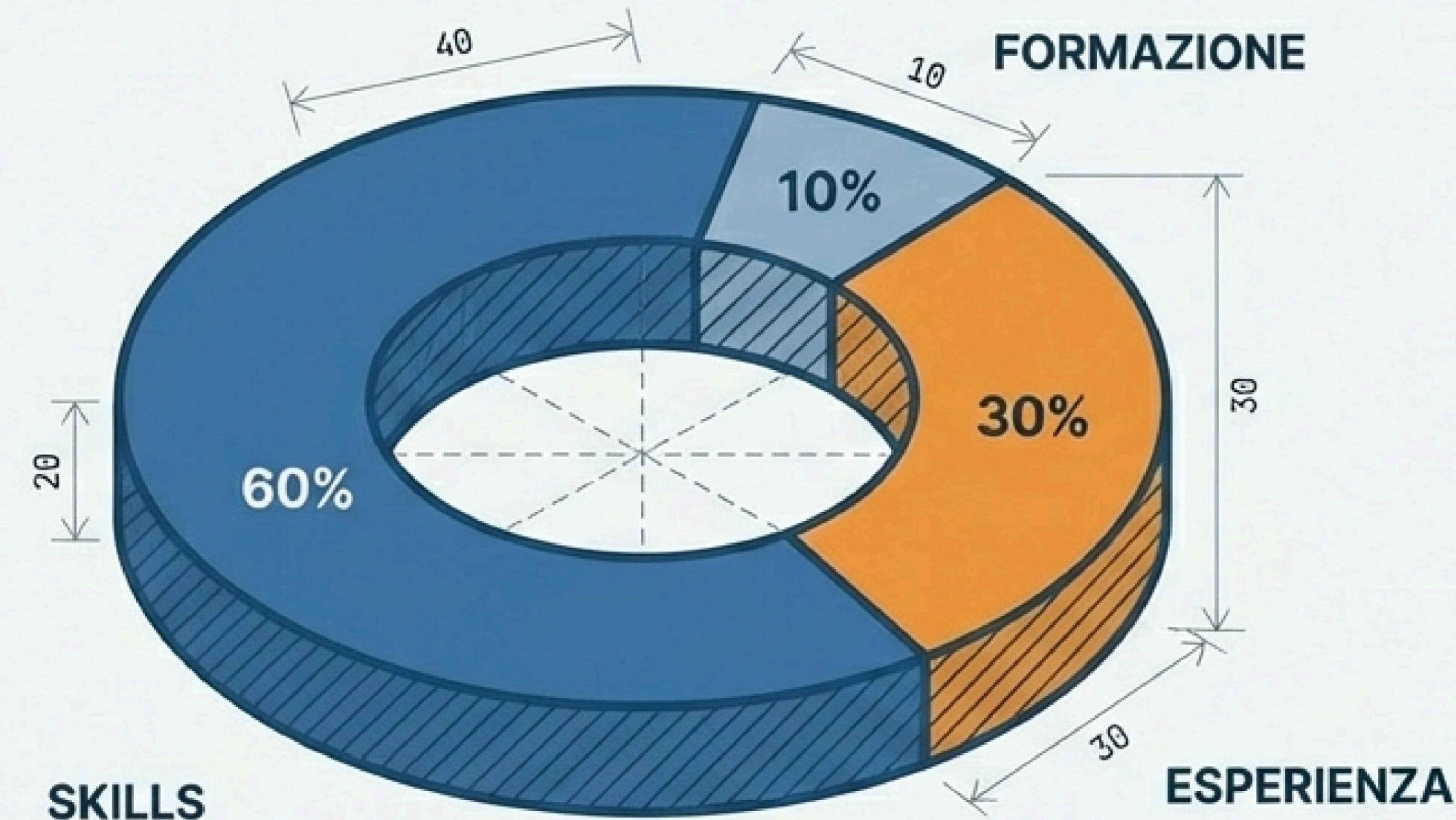
Elaborazione Serverless



```
def handler(event, context):  
    1. Trigger: Sveglia da coda SQS  
    2. Download: Recupero file (PDF/DOCX)  
    3. Parsing: Estrazione testo grezzo  
    4. Analysis: Identificazione entità
```

Tech: Python 3.x, Boto3 SDK

II Motore di Ranking



Logic Breakdown:

- Match competenze tecniche richieste (60%)
- Anni di lavoro rilevati nel testo (30%)
- Titoli di studio e certificazioni (10%)

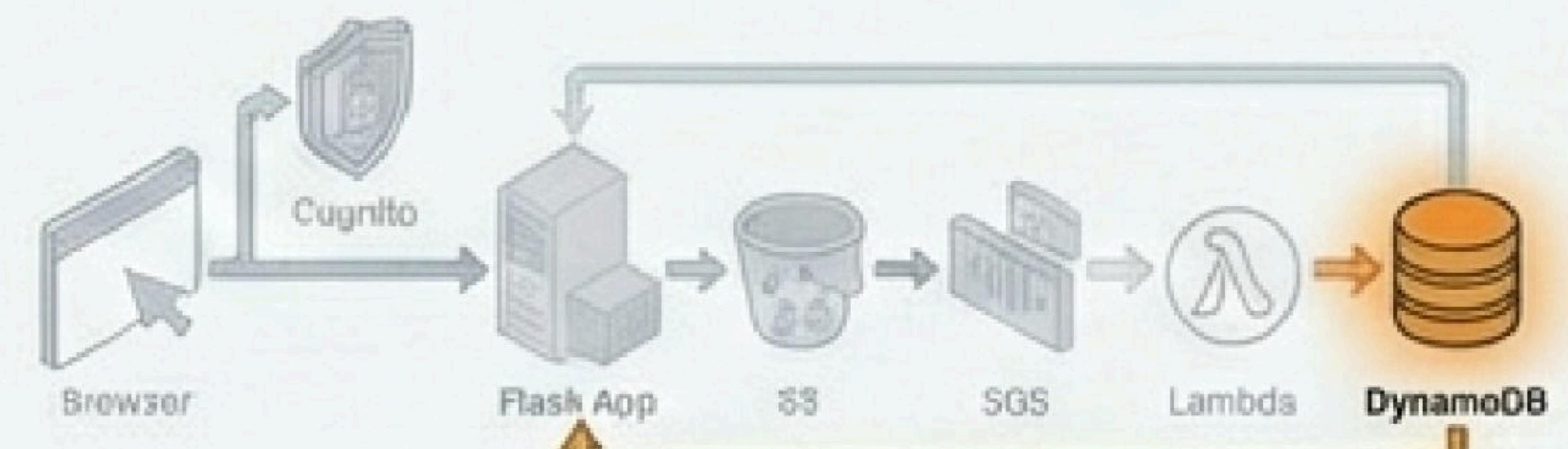
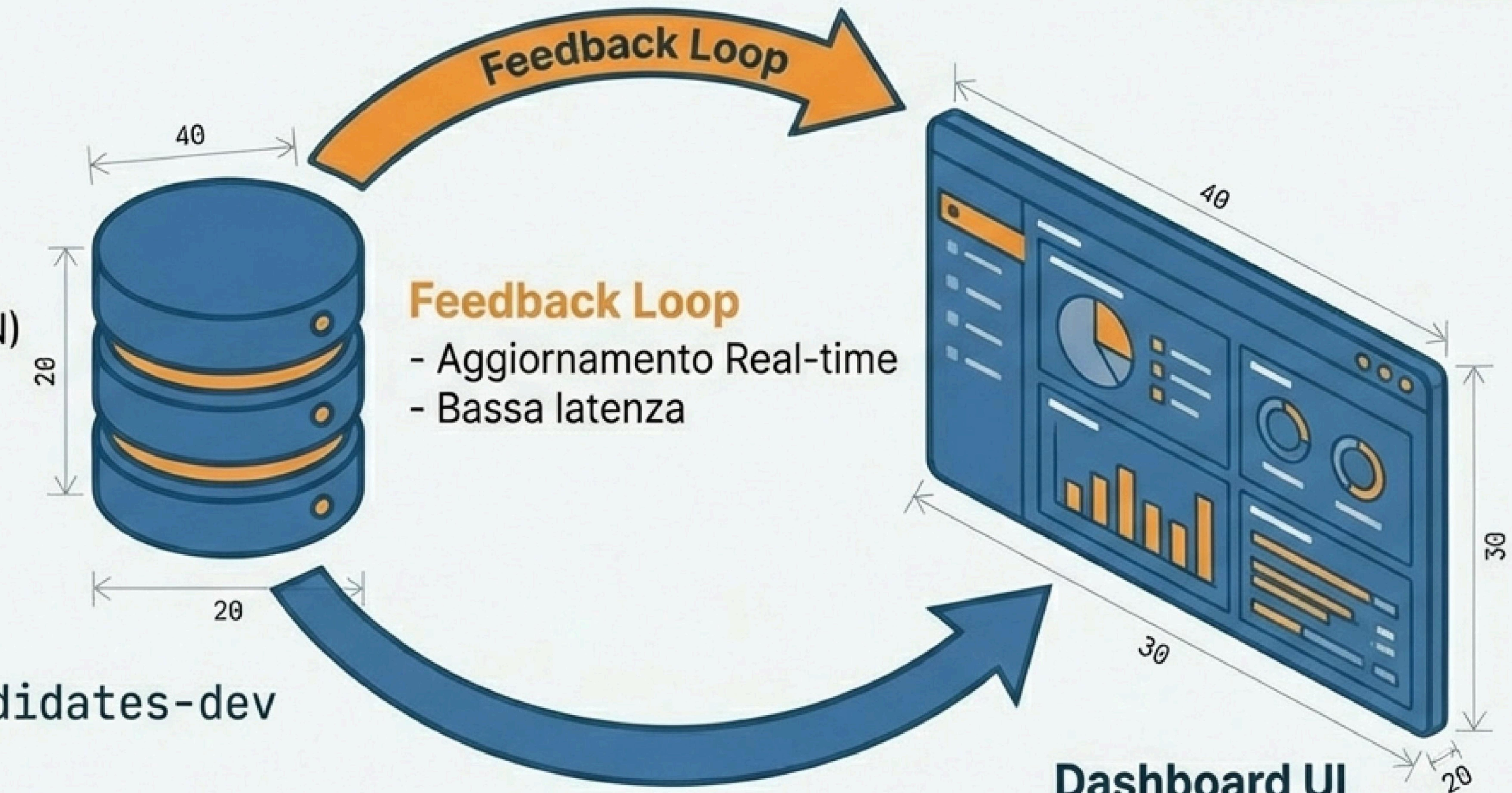
```
score = (skills_match * 0.6) + (experience_years * 0.3) + (education_level * 0.1)  
File: ranking_engine.py
```

Persistenza e Visualizzazione

NoSQL Strategy

- Schema flessibile (JSON)
- Key: CandidateID
- Attributes: TotalScore, ExtractedData

smart-ats-candidates-dev



Infrastructure as Code (IaC) con AWS SAM

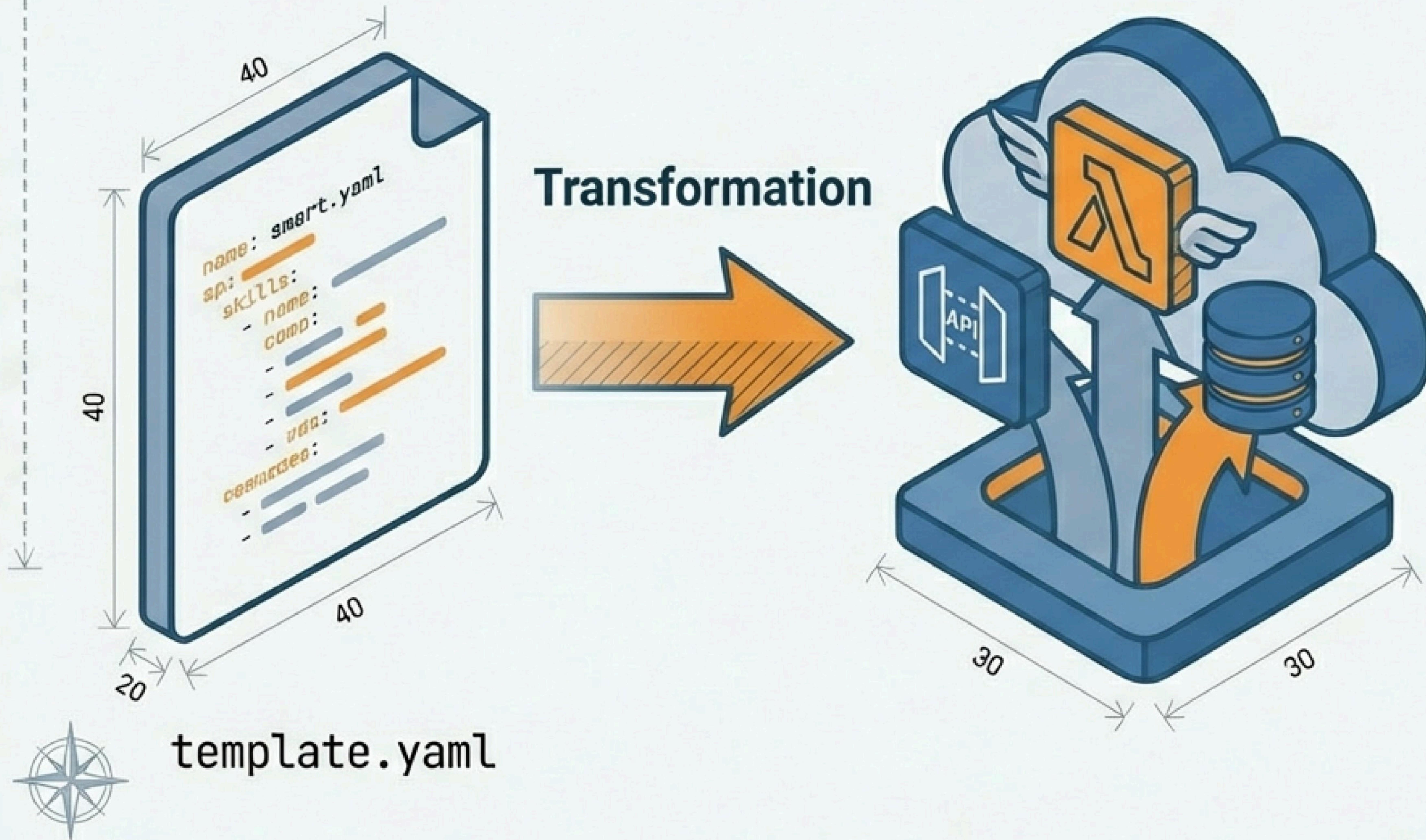


Definizione:
Intera infrastruttura
nel file template.yaml

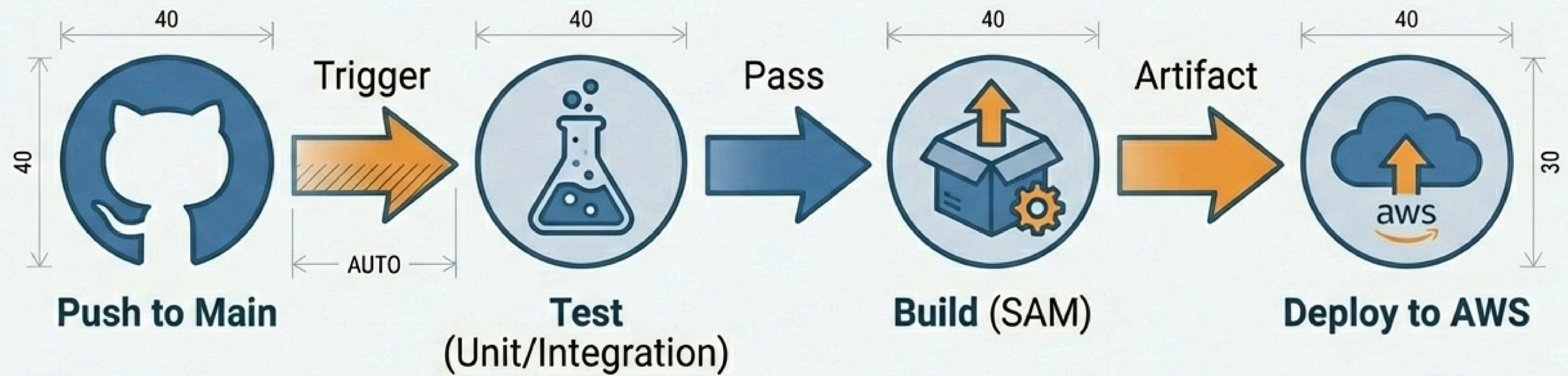
Riproducibilità:
Ambienti identici
(Dev/Prod)

Orchestrazione:


```
|> sam build  
|> sam deploy
```



Automazione e CI/CD Pipeline



Stack Tecnologico



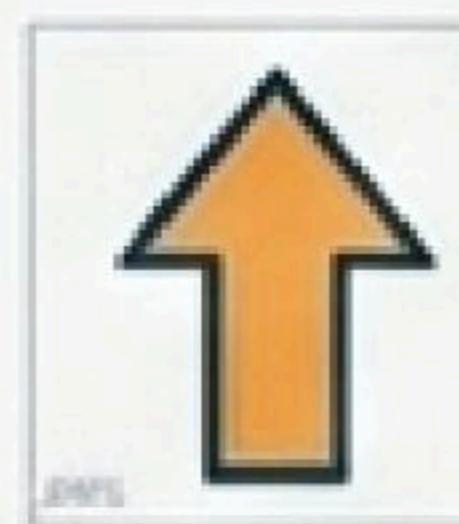
Conclusioni e Sviluppi Futuri



Risultati Chiave:



- Architettura Serverless
(Pay-per-use & Zero Idle Cost)



- Scalabilità Event-Driven e Resilienza

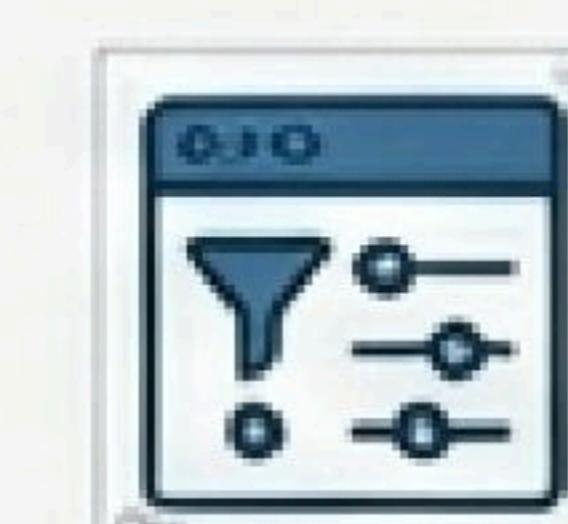


- Pipeline CI/CD completa e sicura

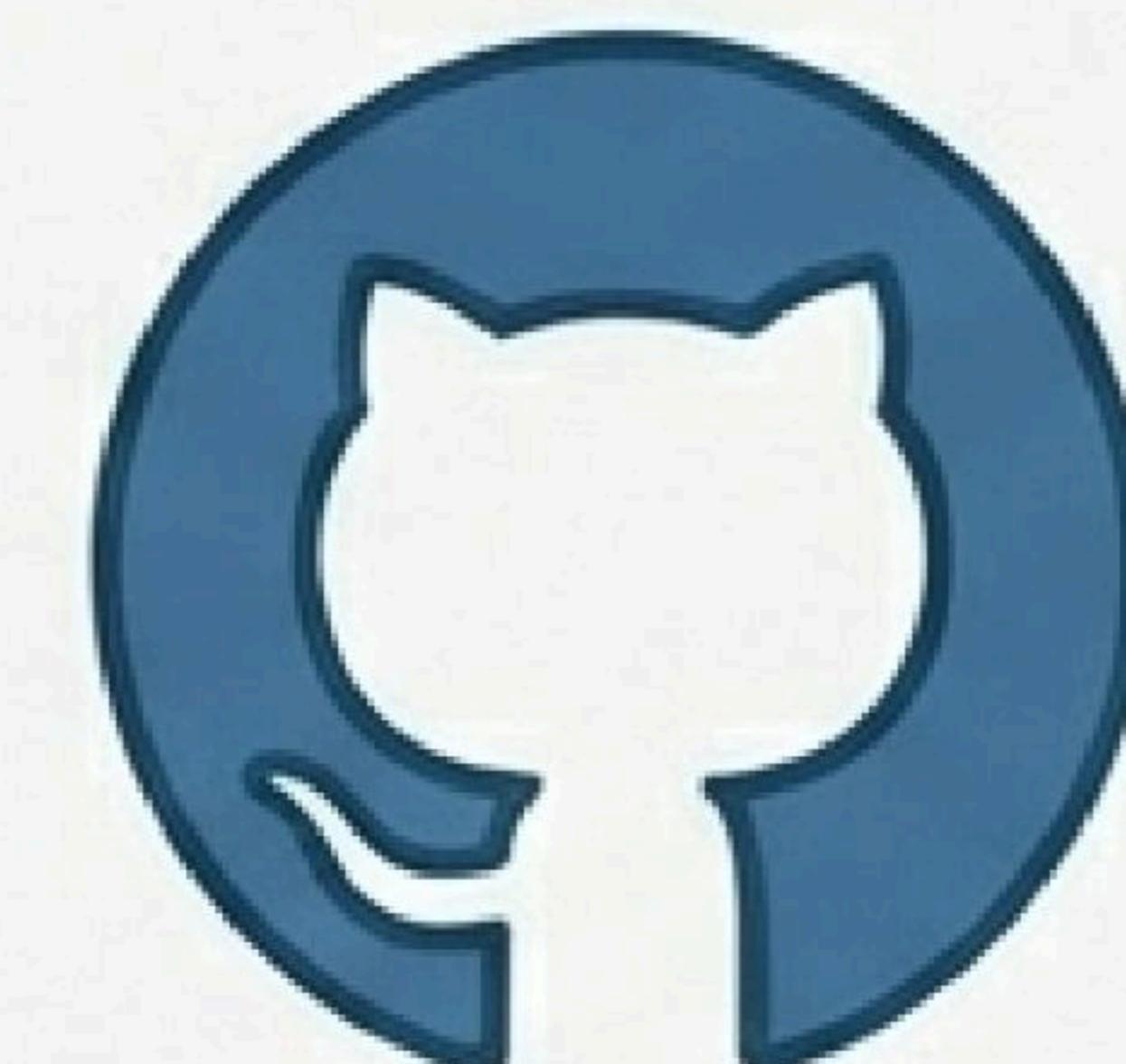
Sviluppi Futuri:



- Integrazione GenAI per analisi semantica



- Espansione Dashboard con filtri avanzati



github.com/salvlea/Sistemi_Cloud

160