**Inbox Intelligence Engine – Technical Module Documentation**

## 🧠 Overview

The Inbox Intelligence Engine is a core module designed to intelligently parse, identify, and map incoming customer conversations (in Messenger, Instagram DMs, and WhatsApp) to known customer identities, previous orders, and website accounts.

This engine supports personalization, order syncing, and real-time contextual replies — all by understanding who the user is based on their past history.

## 🎯 Core Goals

* Identify users based on message content, name, phone, email, or order references.
* Track past interaction history across platforms.
* Link social media profiles to WooCommerce/Shopify customers.
* Extract actionable data from conversations (e.g., complaints, interests, feedback).
* Trigger workflows based on message content.

## 🔧 Technical Architecture

### 1. **Message Collector Layer**

* **Meta API Webhooks** to collect:
  + Messenger thread messages
  + Instagram DM messages
  + Story replies, mentions, etc.
* **WhatsApp Business API** (via provider like Twilio/360Dialog)
* Standardized schema:

{  
 "platform": "facebook\_messenger",  
 "message\_id": "abc123",  
 "sender\_name": "Rafi",  
 "sender\_id": "fb:1000123",  
 "text": "Hi, I placed an order #4523 but didn’t get a call",  
 "timestamp": "2025-07-10T06:15:00Z"  
}

### 2. **Entity Extraction Engine**

#### NLP Pipeline

* Name detection (match to user database)
* Phone/email regex match
* Order reference pattern (e.g., #1234, Order ID 9911)
* Sentiment & intent classifier: complaint, feedback, inquiry, product\_request
* Customer classifier confidence score

result = {  
 "matched\_customer\_id": 4215,  
 "confidence": 0.88,  
 "intent": "complaint",  
 "order\_id": 9911,  
 "matched\_fields": ["phone", "name"]  
}

### 3. **Customer Mapping Graph**

* Build a **Customer Identity Graph**:
  + FB sender ID ↔ website customer ID
  + WhatsApp number ↔ previous order
  + Email from inbox ↔ user profile
  + Cross-links from CRM uploads
* Data store: Neo4j or graph-based PostgreSQL table
* Visual in Admin Panel for merged profiles

### 4. **Inbox Enhancer UI**

#### Sidebar Component

* Shown inside unified Inbox UI:
  + Customer Name + Tier (Gold/Silver)
  + Location (from order history)
  + Past purchases
  + Tags (e.g., VIP, Regular, Refund Complainer)
  + Message insights (intent, sentiment)

#### Smart Reply Suggestions

* GPT-generated replies based on past orders
* Action button to:
  + Mark as complaint
  + Trigger re-delivery workflow
  + Send discount code

## 🔁 Workflow Example

1. Rafi sends DM: *“My order 9911 didn’t arrive. Can I reorder or get help?”*
2. Message ingested via webhook
3. NLP detects order ID, sentiment = complaint
4. Matched to WooCommerce Customer rafi.hossain@gmail.com
5. Sidebar shows past 3 orders, last refund history
6. GPT recommends reply: *“Hi Rafi! I’m so sorry about your experience. Let me look into order 9911 and make this right. Can I offer you a free replacement or refund?”*
7. Agent selects reply + triggers delivery retry

## 🧰 Tech Stack

* **Backend**: Python (FastAPI)
* **NLP**: spaCy + OpenAI + regex pipeline
* **Vector Store**: FAISS for matching message context
* **Customer Store**: PostgreSQL + Neo4j for mapping
* **Frontend**: React (inbox plugin), connects to shared CRM

## 🔒 Privacy & Compliance

* All messages encrypted in storage
* GDPR-compliant consent + data usage log
* Opt-out handling and masking for sensitive queries

## 🔑 Superadmin Controls

* Enable/Disable this module per brand
* Define platform access (FB only, or WhatsApp too)
* Monitor mapping accuracy rates
* Set confidence thresholds for automatic actions

## 🧪 Add-On Ideas

* Conversation summary timeline
* Auto ticket creation for unresolved DMs
* Voice note transcription (via Whisper/OpenAI)
* UGC sentiment graph (weekly mood tracker)

✅ Ready for integration with:

* Gamification Module (for identity mapping)
* Product Recommender (based on messages)
* Loyalty Program (auto reward for helpful DMs)
* Complaint Tracker