1. Problem Being Solved

Modern companies communicate with customers across:

- web forms
- emails
- WhatsApp / chat
- multiple meetings

Today, this context is:

- fragmented across tools
- lost between meetings
- re-asked repeatedly
- poorly transferred between Sales → Engineering → PM

Result:
Missed requirements, scope creep, poor proposals, repeated clarification meetings.

---

2. Vision of the Proposed System

This architecture designs an **AI-first, context-aware enterprise communication system** where:

- AI participates **before**, **during**, and **after** meetings
- AI remembers **everything that matters**, across all channels
- AI proactively prepares humans (instead of waiting for prompts)
- Context improves over time — the system learns, not just the model

The system behaves like a **continuously learning project assistant**, not a chatbot.

---

3. Core Architectural Principle

Memory > Model

The intelligence comes from:

- how context is stored

- how it is retrieved
- how it is versioned
- how it is validated

LLMs reason over **carefully curated context packs**, not raw history.

---

1. High-Level System Blocks

Channels → Ingestion → Memory Platform → AI Intelligence → Outputs → Humans → back to Channels

Key Blocks

1. **Channels**
    - Web / Forms
    - Email
    - WhatsApp / Chat
    - Meetings (Zoom / Teams / Meet)
2. **Ingestion Gateway**
    - Normalizes all inputs
    - Tags metadata
    - Routes events to storage and workflows
3. **Context & Memory Platform**
    - Structured DB (state)
    - Vector DB (semantic memory)
    - Memory Graph (decisions, dependencies, risks)
    - Artifact Store (raw files)
4. **AI Intelligence Services**
    - Scheduling
    - Pre-Meeting Intelligence

    - Live Meeting Bot
    - Post-Meeting Update
    - Engineering, Scope & Proposal generation
5. **Outputs + Humans**
    - Internal dashboard
    - CRM
    - Proposal documents

---

2. Why Multiple Memory Types Are Needed

Memory Type	Purpose
Structured DB	What is open/closed, owners, status
Vector DB	"Find where this was discussed before"
Memory Graph	Decisions → risks → dependencies
Artifact Store	Source of truth (audio, PDFs, emails)
Context Ledger	What changed, when, and why

This separation prevents hallucination and enables explainability.

STEP 1 — Customer Request Intake

What happens

- Customer sends request via web/email/WhatsApp
- System extracts intent, requirements, urgency
- Sales receives an AI-generated internal brief

Tech Stack

- Web: Next.js / React
- Email: Gmail API / Microsoft Graph

- WhatsApp: WhatsApp Business API / Twilio
- LLM extraction: GPT-4.1 / GPT-4o / GPT-5
- Storage: Postgres + Vector DB + Object Store

Why

- Ensures every request enters the same pipeline
- Prevents context loss at the first interaction

STEP 2 — External Enrichment

What happens

- System fetches customer website
- AI summarizes industry, maturity, tech hints
- Marks all enrichment as *inferred*, not confirmed

Tech Stack

- Website extraction: Diffbot / Mercury Parser
- Summarization: GPT-4.1 / GPT-4o
- Embeddings: OpenAI text-embedding-3-*
- Storage: Vector DB + Knowledge Graph

Why

- Sales and Engineering start meetings informed
- Reduces basic "who are you?" questions

STEP 3 — Sales Notification & Meeting Draft

What happens

- AI creates a short Sales Brief
- CRM opportunity is created
- Meeting intent is defined (discovery/follow-up)

Tech Stack

- CRM: Salesforce / HubSpot
- Notifications: Slack API / Teams API
- LLM: GPT-4.1 (summarization only)

Why

- AI assists humans, does not replace decision-making

STEP 4 — AI-Based Scheduling

What happens

- Finds meeting times using:
    - availability
    - time zones
    - **personal preferences**
- Suggests optimal slots to customer

Tech Stack

- Calendar APIs: Google Calendar / Microsoft Graph
- Constraint solving: OR-Tools
- LLM (language only): GPT-4o

Why

- Calendars show availability, not preference
- Scheduling is deterministic, not hallucinated

STEP 5 — Pre-Meeting Context Intelligence

What happens

- Runs at T-24h (deep brief) and T-2h (delta brief)
- Retrieves all relevant history

- Detects:
    - gaps
    - contradictions
    - risks
- Produces role-specific questions + agenda

Tech Stack

- Vector DB: Pinecone / Weaviate / Milvus
- Graph DB: Neo4j / Neptune
- Orchestration: LangGraph + Temporal
- LLM reasoning (RAG): GPT-5 / GPT-4.1

Why

- This is the system's **core intelligence**
- Humans walk into meetings prepared

PAGE 4 — Workflow Steps 6–10 (During & After Meetings)

STEP 6 — Live Meeting Bot (During Meeting)

What happens

- AI joins meeting
- Records audio
- Transcribes speech
- Identifies speakers and roles

Tech Stack

- Meeting APIs: Zoom / Teams
- STT: Whisper / Deepgram / AssemblyAI
- Diarization: pyannote
- Optional products: Otter.ai, Fireflies, Gong

Why

- Meetings are the highest-value data source
- Accuracy here affects everything downstream

STEP 7 — Post-Meeting Update

What happens

- AI summarizes meeting
- Extracts decisions, answers, open questions
- Updates memory with versioned changes

Tech Stack

- LLM extraction: GPT-4.1 / GPT-5
- Validation: JSON schema + rule checks
- Storage: DB + Graph + Ledger

Why

- Converts conversation into structured truth
- Prevents repeated discussions in future meetings

STEP 8 — Engineering Context Usage

What happens

- Engineers receive a clean technical brief
- AI suggests architecture options + assumptions

Tech Stack

- RAG pipeline: LlamaIndex / LangChain
- LLM: GPT-4.1 / GPT-5
- Knowledge sources: internal design docs

Why

- Engineers start from context, not guesswork

STEP 9 — Technical Questionnaire

What happens

- AI detects missing technical details
- Generates prioritized clarification questions

Tech Stack

- LLM: GPT-4.1 / GPT-5
- Checklists + Graph queries

Why

- Prevents late-stage surprises

STEP 10 — Follow-Up Meetings

What happens

- Steps 5 → 6 → 7 repeat
- Only unresolved or changed items are focused

Why

- Context accumulates
- Meetings become shorter and sharper

PAGE 5 — Finalization, Governance & Market Reality

STEP 11 — Scope Definition

What happens

- AI drafts in-scope / out-of-scope
- Humans approve and lock baseline

Tech Stack

- LLM: GPT-4.1 / GPT-5
- Graph validation

STEP 12 — Project Planning

What happens

- Timeline, milestones, roles generated
- PM validates against real capacity

Tech Stack

- LLM: GPT-4.1
- Optional integration: Jira / Azure DevOps

STEP 13 — Proposal Generation

What happens

- AI fills proposal template
- No pricing invented
- Human approval required

Tech Stack

- LLM: GPT-4.1 / GPT-5
- Templates: Google Docs API / docxtpl
- Export: PDF/DOCX tools

STEP 14 — Continuous Context Loop

What happens

- New emails/chats restart the loop
- AI checks if context changes
- Triggers new briefs if needed

Governance, Safety & Reality Check

- RAG is mandatory for reasoning steps
- Fine-tuning is optional and future-stage
- Humans approve customer-facing outputs
- Context ledger enables audits

Market Products vs This Architecture

Product	**Coverage**
Otter / Fireflies	Meeting capture only
Gong / Chorus	Sales conversations
Notion AI	Document assistance
Salesforce Einstein	CRM insights

None provide:

- cross-channel memory
- decision graphs
- proactive pre-meeting intelligence

That’s the unique value of this system.