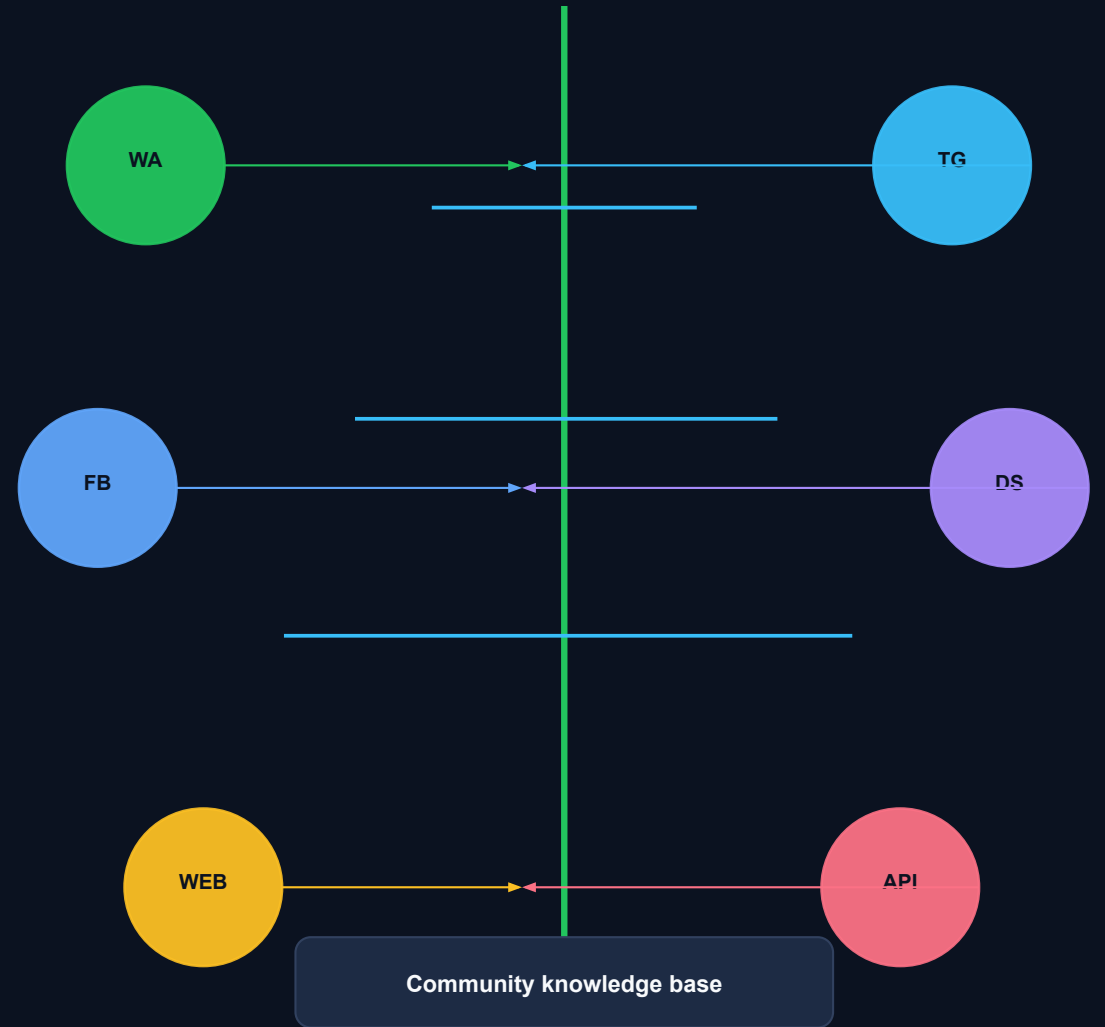


# Insignia TOWER

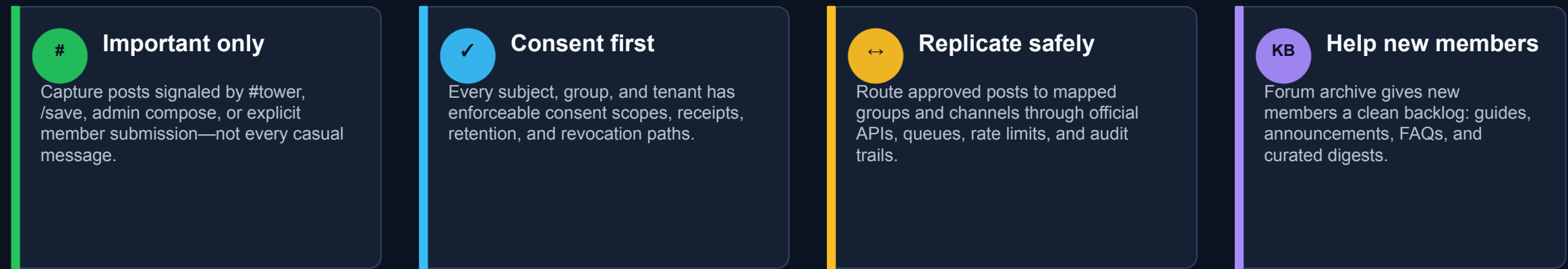
## TwO Way Extractor Replicator

Consent-first important-post synchronization across WhatsApp groups, social communities, and a durable Insignia knowledge forum.

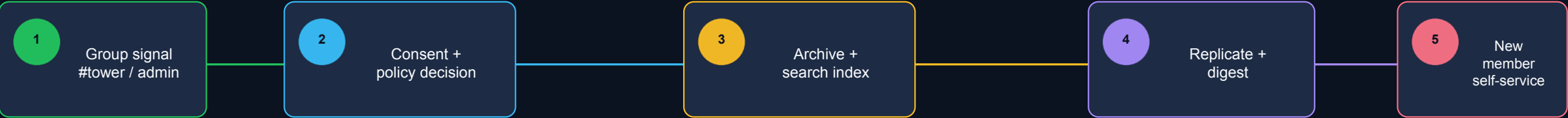


# Product north star

TOWER turns fragmented group announcements into a trusted, searchable, consented knowledge layer.



## TOWER VALUE LOOP



# WhatsApp reality check

The product should avoid Web automation as a core path and use official, consented capture patterns.

## Compliant Phase 1 path

Official WhatsApp Business Platform as the primary adapter surface  
Groups API only where the tenant is eligible and group design fits limits  
Admin/member-initiated submission or #tower capture for important posts  
Portal-to-channel publishing using approved templates and rate limits

build first

## Non-product path

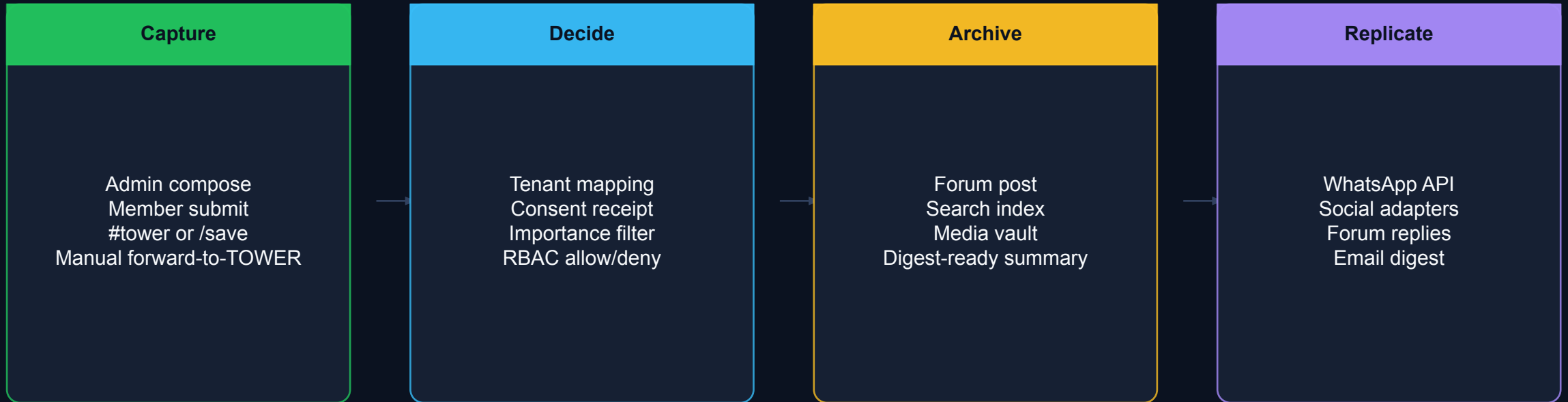
No WhatsApp Web session scraping as the commercial architecture  
No unconsented capture of private group chatter  
No claims that TOWER breaks or bypasses end-to-end encryption  
No dependency on brittle bot accounts that may be banned or blocked

explicitly avoid

E2E-respecting endpoint capture: users/admins deliberately submit or platform webhooks legally provide the content.

# Phase 1 MVP: important-post fabric

Deliver immediate community value while preserving room for Phase 2 enterprise controls.



## In scope

Portal, consent registry, archive/forum, search, manual + official API adapters, audit trail.

## Out of scope

Private scraping, unconsented mirroring, "sync everything" bots, informal E2E bypass claims.

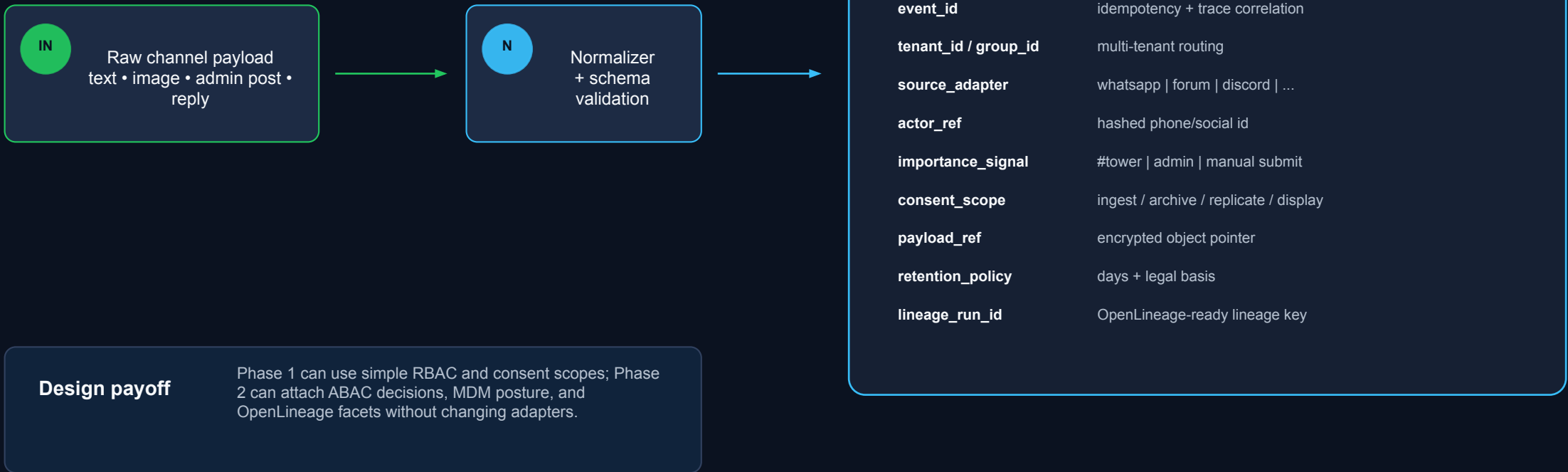
# System architecture

Provider-agnostic core with channel adapters, consent enforcement, and event-driven replication.



# Canonical TOWER event model

Normalize every channel payload so consent, lineage, policy, and routing can evolve independently.



# Data flow: save once, publish everywhere

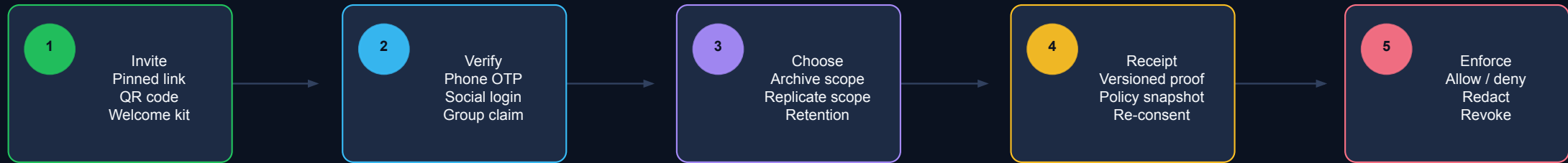
Every important post is processed through the same consented, auditable path.



**Operations rail** trace\_id + audit\_id follow the post from ingest to archive to outbound delivery; lineage events track each derived dataset and index.

# Opt-in and consent architecture

Consent is not a checkbox—it is a versioned service that gates every archive and replication decision.



## ConsentRecord schema

subject\_id • tenant\_id • group\_id • phone\_hash • scopes[] • retention\_days •  
policy\_version • status • proof\_event\_id • effective\_at • revoked\_at

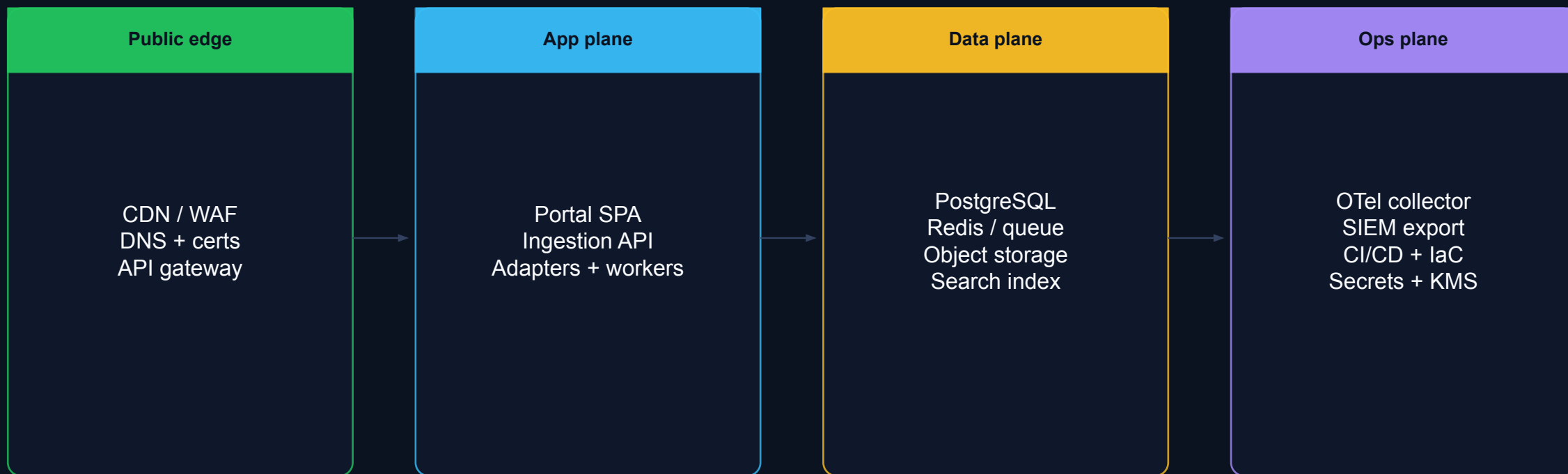
## Revocation pattern

Users can revoke by portal, STOP-style message, or admin action.  
Revocation blocks future capture, hides display when policy requires, and  
creates an immutable audit event.



# Hosting on existing Insignia Cloud

Provider-neutral blueprint that fits Kubernetes/OpenShift-style environments and managed cloud services.



## Assumption

If Insignia Cloud already uses AWS, Azure, GCP, IBM Cloud/OpenShift, or private Kubernetes, keep this control pattern and swap service equivalents.

# Phase 2 target: full ZTA control plane

Move from “secure MVP” to continuous trust evaluation across identity, device, service, data, and consent.



Policy decision = subject attributes + device posture + service identity + resource attributes + consent scope + request context.

# ABAC model for TOWER

Phase 2 externalizes authorization into a policy service without rewriting product flows.

## Subject

role  
verified\_phone  
member\_since  
admin\_device\_posture  
consent\_status

## Resource

tenant\_id  
group\_id  
post\_classification  
retention\_window  
visibility\_scope

## Action

ingest  
archive  
replicate  
read  
delete\_or\_hide

## Context

channel  
risk\_score  
time  
geo/device  
legal\_basis

### Example natural-language policy

A verified member may archive their own #tower post when archive consent is active; an admin on an MDM-compliant device may replicate approved posts to mapped channels.

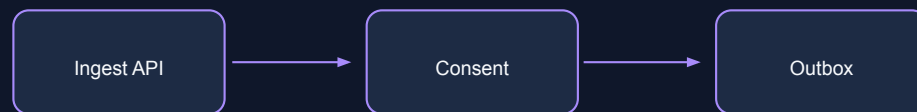
PEP in adapters and APIs → PDP policy decision → permit / deny / redact / quarantine

# Observability and lineage by design

Operational visibility and data provenance are first-class product capabilities, not afterthoughts.

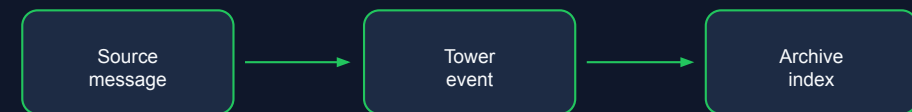
## OpenTelemetry operating model

trace\_id generated at ingestion and carried through consent, archive, search, and outbox  
metrics: ingestion lag, policy denies, consent revocation latency, delivery success, retry depth  
logs: structured JSON with PII redaction and tenant-aware sampling



## OpenLineage operating model

model source group, normalized event stream, archive table, media vault, and search index as datasets  
emit run events for ingest, transform, redact, archive, index, replicate, export  
answer: where did this archived post come from, what derived artifacts exist, who changed it?



# Roadmap

Build a useful Phase 1 product in slices while reserving interfaces for enterprise-grade Phase 2.

0–30 days

## Product spine

Tenant model, SPA, auth, consent records, forum archive, canonical event schema.

31–60 days

## WhatsApp path

Business Platform adapter, inbound submit flows, webhook verification, admin publish, templates.

61–90 days

## Community value

Search, digests, media handling, DLP queue, audit dashboards, social adapter scaffold.

Phase 2

## Enterprise upgrade

mTLS mesh, MDM posture, ABAC PDP/PEP, OpenLineage, advanced tenant isolation.

# Risks and product guardrails

The right product promise is “useful community memory,” not invisible mirroring.



# Implementation decision package

Recommended defaults for the first engineering build.

## Frontend

Next.js/React target; standalone SPA prototype included for stakeholder walkthroughs.

## Backend

TypeScript/NestJS or Go services; adapter interface isolates channel-specific logic.

## Messaging

Managed queue or NATS JetStream for Phase 1; Kafka/Redpanda-compatible event model for Phase 2.

## Data

PostgreSQL + RLS, Redis, object storage, OpenSearch/Meilisearch; tenant-aware encryption.

## Security

OIDC, KMS, signed webhooks, audit log, OTel baseline; reserve PEP/PDP hooks for ABAC.

## First PoC

One tenant, one WhatsApp-approved path, one web forum, one consent flow, one outbox worker.

Build Phase 1 as a compliant platform, not a workaround. The same event and consent contracts become the Phase 2 ZTA/ABAC foundation.

# Reference sources

External sources used to ground platform constraints and architecture standards.

## WhatsApp Business Developer Hub

Business Platform docs, webhooks, rate limits, opt-in and pricing resources.

## Meta for Developers: WhatsApp Groups API

Current official Groups API docs/snippets show eligibility and group constraints; validate in Meta console during PoC.

## WhatsApp Business Messaging Policy / Terms

Opt-in, message templates, 24-hour service window, stop/opt-out expectations.

## NIST SP 800-207

Zero Trust Architecture principles: no implicit trust; protect users, assets, and resources.

## OpenTelemetry

Vendor-neutral framework for traces, metrics, and logs.

## OpenLineage

Open standard for lineage events across jobs, runs, and datasets.

## Open Policy Agent

ABAC policies using subject, object, action, and contextual attributes.