DABBLER CONTENT ENGINE

Formal Architecture Specification

1. Introduction

1.1 Purpose

This document defines the architecture, structure, and operational model of the Dabbler Content Engine. The Content Engine is a core platform module responsible for managing all content displayed within the application feed, including user-generated posts, system announcements, sponsored content, and game/achievement integrations.

This document serves as the authoritative reference for:

- Backend development
- Feed logic design
- Permission & RLS enforcement
- Achievement integration
- Monetization strategy
- Future extensibility

1.2 Scope

The Content Engine includes:

- Social posts (moment, dab, kick-in)
- Comments
- Likes and reactions
- Reposts
- Themes
- Hashtags (future-ready)
- Admin announcements

- Sponsored content
- Achievement-generated content
- Game lifecycle-generated content
- Premium feature gating
- Feed ranking logic integration

---

2. System Position in Overall Architecture

The Content Engine integrates with:

- Authentication System Phase_2_Authentication_System_(...
- Core Infrastructure Layer Phase_1_Foundation__Core_Infras...
- Games Core System
- Rewards & Achievements System
- Notification Pipeline
- Feed Ranking Configuration

The module operates as a domain-level core service and is not an optional feature layer.

---

3. Core Architectural Principles

1. **Single Unified Content Stream**

   All feed items (social, system, sponsored) originate from a unified content structure.

2. **Separation of Identity and Ownership**

   Content is authored by persona (profile) but owned by user.

3. **Layered Classification**

   Content behavior is controlled via:

   - Content Class (ownership layer)
   - Post Type (social behavior layer)
   - Origin Type (source layer)

4. **Feature Extensibility**

   Schema must support future premium and monetization without structural refactoring.

5. **Event-Driven Integration**

   Content emits events to support achievements, notifications, and analytics.

---

4. Identity Model

4.1 User vs Profile

- User = authentication identity.
- Profile = persona identity (Player, Organiser, Socialiser, Host).

Each content item contains:

- author_profile_id
- author_user_id
- persona_type snapshot

Rationale

- Prevent self-like/reaction
- Block all personas of a user at once
- Maintain historical persona consistency

---

5. Content Classification Model

5.1 Content Class

Defines ownership and behavioral group.

Values:

- social
- system
- sponsored

Behavior Matrix

Content Class	Blockable	Persona Rules Apply	Engagement Allowed
social	Yes	Yes	Yes
system	No	No	Configurable
sponsored	No	No	Configurable

System and sponsored content bypass user block restrictions.

5.2 Post Type

Defines social intent.

- moment – real-time activity
- dab – expression or celebration
- kick_in – invitation or recruitment

Applies primarily to social content.

5.3 Origin Type

Defines content source.

- manual
- game
- achievement
- venue
- admin
- system
- repost
- future extensible

Used to link content to domain entities.

---

6. Persona-Based Posting Rules

Posting capability depends on persona type.

Persona	moment	dab	kick_in
Player	Yes	Yes	Yes
Organiser	No	Yes	Yes
Socialiser	Yes	Yes	No
Host	No	Yes	No

These rules must be enforced at domain/service level.

---

7. Content Structure

Each content item includes:

Identity

- Author profile
- Author user
- Persona snapshot

Classification

- Content class
- Post type
- Origin type
- Origin reference ID

Body & Metadata

- Text body
- Sport reference
- Venue reference
- Tagged location
- Event time
- External links

Location Snapshot (Immutable)

- author_city

- author_country
- author_neighbourhood
- coordinates

Location is frozen at creation time.

---

8. Visibility Model

Supported visibility levels:

- public
- followers
- circle
- private

Mention Override Rule

Mentioned users may view content even if outside normal circle visibility.

System Override Rule

System and sponsored content cannot be blocked by users.

---

9. Engagement Model

9.1 Likes

- Binary endorsement.
- One per user per content.
- Cannot like own content.
- Separate from reactions.

Used for analytics and achievement triggers.

9.2 Reactions (Vibe System)

Supported reactions include:

- Laugh
- Sad
- Angry
- Fire
- Celebrate
- Thanks
- Support
- Here for you
- Yay
- Clapping
- Shocked

Characteristics:

- Multiple reaction types supported.
- Separate from likes.
- Reaction counts stored separately.

9.3 Comments

- Threaded structure.
- Enable/disable per content.
- Mention support.
- Soft deletion supported.

9.4 Reposts

- Wraps original content.
- Engagement remains on original.
- Supports commentary.

9.5 Views

Supports:

- View count
- Optional unique-per-user tracking

Used for ranking and analytics.

---

10. Content Expiration

Kick-in posts linked to games expire automatically at game start time.

Content supports:

- expires_at
- is_active

Expiration reduces ranking priority or archives content.

---

11. Interaction Controls

Each content item can configure:

- allow_comments
- allow_likes
- allow_reacts
- allow_reposts

Premium features may override default restrictions.

---

12. Premium & Tier System

Premium gating is capability-based, not schema-based.

Free User Limitations (Examples)

- Limited mentions
- Cannot disable comments
- No theme selection

- Limited hashtags

Premium Capabilities (Examples)

- Custom theme
- Disable comments
- Extended mentions
- Schedule post
- Priority boost
- Advanced analytics

Enforcement occurs in service layer.

---

13. Themes System

Themes are stored independently and referenced by content.

Themes include:

- Background type
- Color / gradient
- Image
- Font style
- Premium flag
- System flag

Allows:

- Seasonal themes
- Achievement themes
- Sponsor-branded themes
- Premium-exclusive themes

---

14. Blocking Model

Blocking is user-level.

If User A blocks User B:

- All content authored by B is hidden.
- All personas of B are hidden.
- Engagement by B is hidden.

System content is unaffected.

---

15. Feed Architecture

Feed is composed of:

- Social content
- Reposts
- Admin content
- Sponsored content
- Venue highlights
- Game lifecycle posts
- Achievement posts

Ranking Factors

- Recency
- Engagement velocity
- Sport relevance
- Region relevance
- Persona relevance
- Kick-in urgency
- Premium boost
- Sponsored placement
- Hashtag relevance (future)

Feed injection logic handled server-side.

---

16. Achievement Integration

Content emits domain events:

- content.created
- content.liked
- content.reacted
- content.shared
- content.viewed
- content.hashtag_used

Achievement system listens and awards:

- First post
- First kick-in
- 50 likes
- 100 reactions
- Trending hashtag creator
- Most reposted organiser

Content engine acts as gamification amplifier.

---

17. Hashtag System (Future-Ready)

Supports:

- Unicode (Arabic + multilingual)
- No emoji
- Follow hashtag
- Mute hashtag
- Tag notification subscription
- Region-scoped trending
- Sport-scoped trending
- Achievement triggers

Designed to integrate without structural refactor.

---

18. Admin & Sponsored Content Rules

Admin Content

- Always visible
- Cannot be blocked
- May be pinned
- May disable engagement

Sponsored Content

- content_class = sponsored
- Region and sport targeting
- May restrict interaction
- May expire
- Tracked separately in analytics

Transparency in UI required.

---

19. Extensibility

The architecture supports:

- AI-generated content
- Paid promotion boosts
- Scheduled publishing
- Premium-only content
- Campaign content
- Multi-sport expansion
- Marketplace integration
- Tournament broadcasts

- Leaderboard spotlight posts

No schema restructuring required for future expansion.

---

20. Architectural Summary

The Dabbler Content Engine is:

- Multi-persona aware
- User-level secure
- Block-aware
- System-override capable
- Event-driven
- Premium extensible
- Monetization ready
- Achievement-integrated
- Feed-ranking optimized
- Region scalable
- Sport scalable
- Future-proof

It transforms Dabbler from a sports utility app into a full sport-social ecosystem platform.