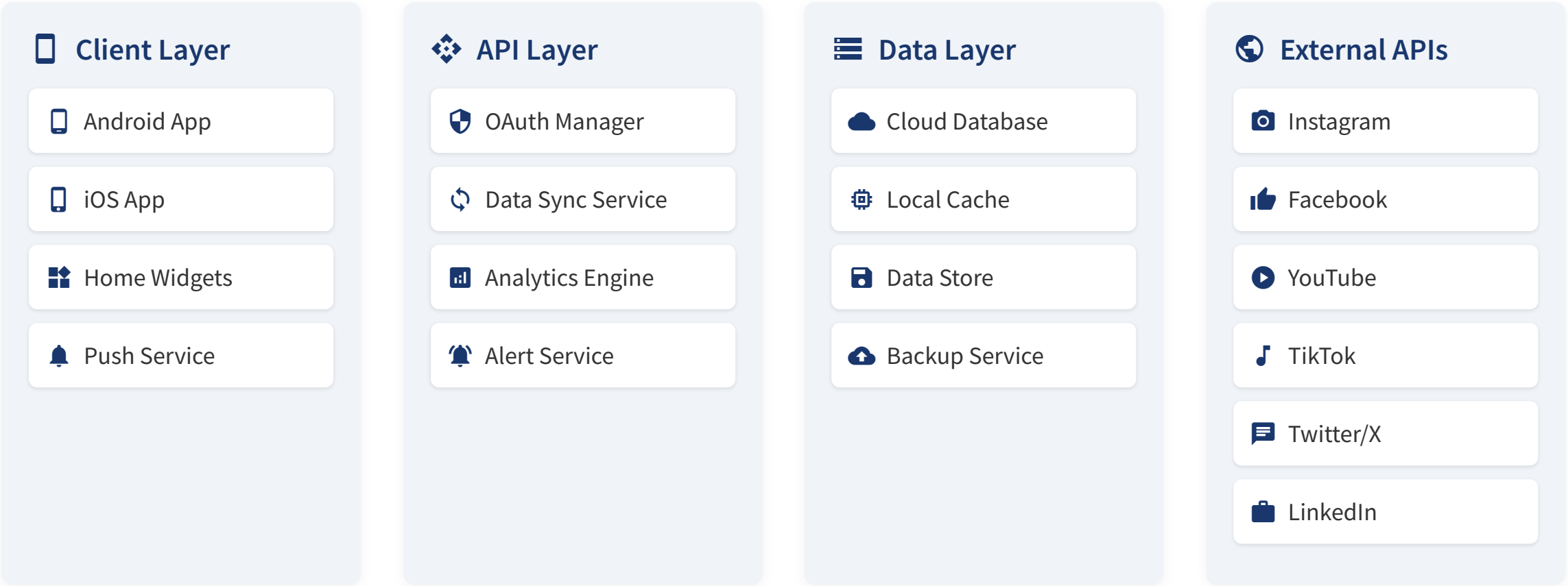



Social Media Analytics App




Technical Architecture Overview

System components and data flow for cross-platform social media analytics




 **Architecture Pattern**

MVVM for mobile apps with **Repository Pattern** for data access

 **Data Flow**

Real-time sync from social APIs → Process → Store → Cache → Display

 **Update Strategy**

Background sync every 30min with **incremental updates**

API Integration Specifications

Social media platform integration requirements and limitations



Instagram

- Auth:** OAuth 2.0 (Business/Creator)
- Endpoint:** Graph API v18.0
- Metrics:** Engagement, Reach, Impressions

200 calls/hour per user



Facebook

- Auth:** OAuth 2.0 (Page Access)
- Endpoint:** Graph API v18.0
- Metrics:** Page Insights, Post Performance

200 calls/hour per app



YouTube

- Auth:** OAuth 2.0 (Channel Access)
- Endpoint:** YouTube Analytics API
- Metrics:** Views, Watch Time, Subscribers

100 units/day per project



TikTok

- Auth:** OAuth 2.0 (Creator Access)
- Endpoint:** TikTok Research API
- Metrics:** Video Views, Likes, Comments

10 queries/second



Twitter/X

- Auth:** OAuth 2.0 (Bearer Token)
- Endpoint:** Twitter API v2
- Metrics:** Impressions, Engagements, Followers

300 requests/15-min window



LinkedIn

- Auth:** OAuth 2.0 (Company Page)
- Endpoint:** LinkedIn Marketing API
- Metrics:** Engagement, Clicks, Impressions

100 requests/day per member

Integration Implementation Notes

Token Refresh: Implement automatic token refresh before expiry with 5-minute buffer

Rate Limiting: Use exponential backoff strategy when hitting API limits

Data Sync: Implement incremental sync with last-sync timestamps to minimize API calls

Data Models and Schemas

Core data structures for storing social media analytics



Account Model

- String** **accountId** Unique identifier
- String** **platform** Instagram, Facebook, etc.
- String** **accessToken** Encrypted OAuth token
- Date** **tokenExpiry** Token expiration date
- Boolean** **isActive** Account sync status



Post Model

- String** **postId** Unique identifier
- String** **accountId** Reference to Account
- String** **contentUrl** Media URL
- String** **caption** Post caption/title
- Date** **postDate** Publication date
- String** **type** Image, video, etc.



Metrics Model


- String** **metricId** Unique identifier
- String** **accountId** Reference to Account
- Number** **followers** Current follower count
- Number** **engagement** Total engagement
- Number** **impressions** Total impressions
- Date** **timestamp** Collection date



PostMetrics Model


- String** **metricId** Unique identifier
- String** **postId** Reference to Post
- Number** **likes** Like count
- Number** **comments** Comment count
- Number** **shares** Share count
- Number** **reach** Unique viewers
- Date** **timestamp** Collection date

Data Flow & Storage Strategy

 **API Fetch:** Incremental sync every 30min



Transform: Normalize platform-specific data




 **Storage:** Cloud DB + Local cache (7 days)

Widget Implementation Specifications


Technical requirements for iOS and Android home screen widgets

iOS Widgets

Small Widget





-  **Size:** 2×2 grid units
-  **Data:** Single metric + trend
-  **Refresh:** 15-30 min

Medium Widget

-  **Size:** 4×2 grid units
-  **Data:** 2 metrics + mini chart
-  **Refresh:** 15-30 min




Large Widget

-  **Size:** 4×4 grid units
-  **Data:** Summary + top post
-  **Refresh:** 15-30 min




-  **Implementation:** WidgetKit + TimelineProvider
-  **Data Storage:** App Group UserDefaults
-  **Interaction:** WidgetURL to open app
-  **Battery:** Background fetch limit 30 sec

Android Widgets


Compact Widget





-  **Size:** 4×1 cells (min)
-  **Data:** Single metric + trend
-  **Refresh:** 30 min default

Medium Widget

-  **Size:** 4×2 cells
-  **Data:** 2 metrics + mini chart
-  **Refresh:** 30 min default

Large Widget

-  **Size:** 4×4 cells
-  **Data:** Summary + top post
-  **Refresh:** 30 min default

-  **Implementation:** AppWidgetProvider + RemoteViews
-  **Data Storage:** SharedPreferences + WorkManager
-  **Interaction:** PendingIntent to open app
-  **Battery:** Background restrictions apply

Cross-Platform Implementation Notes

Data Sync: Shared backend API with platform-specific data caching

Refresh Strategy: Configurable intervals with fallback to cached data





Offline Mode: Display last cached data with timestamp indicator

Security Requirements

Data protection and user privacy implementation guidelines







Data Protection

-  **End-to-end encryption** for all API communications
-  **Encrypted local storage** using platform-specific keychain
-  **OAuth 2.0** with PKCE for social platform authentication
-  **Token rotation** with secure refresh mechanism







Privacy Controls

-  **Granular permissions** for each connected account
-  **Widget privacy mode** to hide sensitive metrics
-  **Data deletion** with complete removal from all systems
-  **Data export** in standard formats for user portability







Code Security

-  **Code obfuscation** using ProGuard (Android) and Swift optimizations
-  **Static analysis** with automated vulnerability scanning
-  **Certificate pinning** for API communications
-  **Secure dependencies** with regular vulnerability updates



Access Control

-  **Biometric authentication** for sensitive operations
-  **Session timeout** with automatic logout after inactivity
-  **Role-based access** for team collaboration features
-  **Audit logging** for all data access and modifications



Regulatory Compliance

-  **GDPR** - EU data protection
-  **CCPA** - California privacy
-  **SOC 2** - Security controls

Performance Considerations

Optimization strategies for responsive app performance



Data Loading

- ↺ Lazy loading for analytics data
- ≡ Pagination for post lists
- ⚡ Data compression for API responses
- ↺ Background sync with incremental updates



Memory Management

- 🖼 Image caching with LRU strategy
- 📄 Local database with data pruning
- 🗑 Memory leaks prevention
- 🗑 Auto-cleanup of old data



Battery Optimization

- 🕒 Batch API calls to reduce wake-ups
- 📶 WiFi-only for large data sync
- 🔌 Doze mode compatibility
- 🔋 Adaptive refresh based on battery level



Network Efficiency

- 📶 Offline mode with cached data
- 📶 Network-aware data loading
- 🔄 Retry logic with exponential backoff
- ☁ CDN integration for media assets



UI Responsiveness

- 🌀 60 FPS animations
- 👆 Touch feedback under 100ms
- 📄 View recycling for lists
- 📄 View hierarchy optimization



Monitoring

- 📊 Performance metrics collection
- 🔍 Crash reporting integration
- 🕒 API response time tracking
- 🔍 Memory usage monitoring



Performance Targets

- 🕒 App Launch: < 2 seconds
- ↺ Data Sync: < 5 seconds
- 🔋 Battery Impact: < 5% daily

Testing Requirements

Quality assurance strategy for reliable deployment



Unit Testing

- ✓ 80%+ code coverage
- ⚙️ API mock for all endpoints
- Σ Data models validation
- 🔄 CI/CD integration



UI Testing

- 📱 Espresso for Android
- 📱 XCUITest for iOS
- 🧩 Widget interaction testing
- 📺 Screen size compatibility



Integration Testing

- 🔗 Social API integration
- 🔄 Data sync validation
- ☁️ Backend connectivity
- 🛡️ OAuth flow testing



Performance Testing

- 🕒 Load time benchmarks
- 🧠 Memory profiling
- 🔋 Battery consumption
- 🌐 Network throttling



Security Testing

- 🔒 Data encryption validation
- 🔑 Token storage security
- 🛡️ Vulnerability scanning
- 🛡️ Data privacy compliance



User Testing

- 👤 Usability studies
- ♿️ Accessibility testing
- 🧠 User journey validation
- 💬 Beta testing program



Quality Metrics & Deployment

- 🐛 Critical Bugs: Zero tolerance
- 🚀 Release Cycle: 2-week sprints
- 🔄 Deployment: Staged rollout