Contents

Samsung App Manager - Project Documentation

Overview

The Samsung App Manager is a system designed to manage and monitor Android applications available on the Samsung Galaxy Store. This system addresses the challenges of maintaining up-to-date APK versions for various client applications, particularly focusing on tracking version updates and facilitating the APK extraction process.

Business Context

- **Problem**: Inability to consistently access the latest app versions from the Samsung store
- **Impact**:
  - Delayed implementation of new features (e.g., samsung_referrer for Singular)
  - Compatibility issues with client updates (e.g., Brigit)
  - User experience problems (e.g., Sezzle’ s outdated version blocking)
- **Solution**: In-house APK management system with monitoring capabilities

System Architecture

Core Features

1. App Management Interface

- Add/Edit managed applications
- View application details and version history

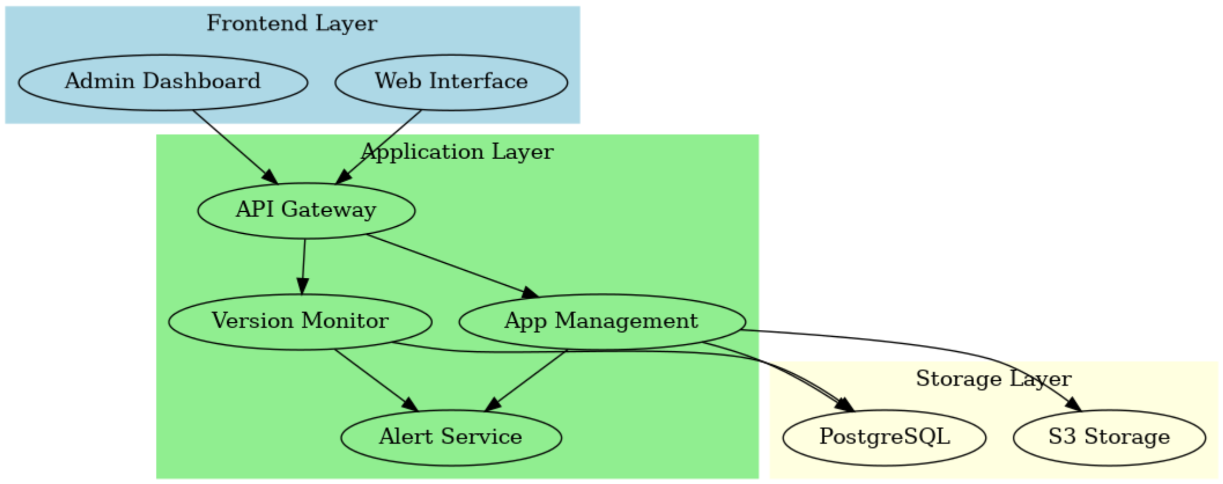

Figure 1: System Architecture

- Manual APK upload capability
- Download stored APKs

2. Version Monitoring System

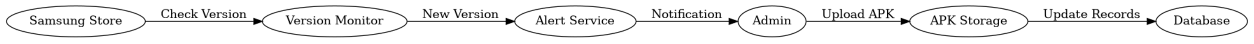

Figure 2: Data Flow

3. APK Storage System

- Secure storage for APK files
- Version tracking and history
- File integrity verification
- Access control and audit logging

Technical Implementation

Component Architecture

API Endpoints

App Management

```
POST   /api/apps              # Create new app entry
GET    /api/apps              # List all managed apps
GET    /api/apps/:id          # Get specific app details
PUT    /api/apps/:id          # Update app information
DELETE /api/apps/:id          # Remove app from management
```

Version Management

```
POST   /api/apps/:id/versions # Upload new APK version
GET    /api/apps/:id/versions # List all versions
GET    /api/versions/:id      # Get specific version details
DELETE /api/versions/:id      # Remove version
```

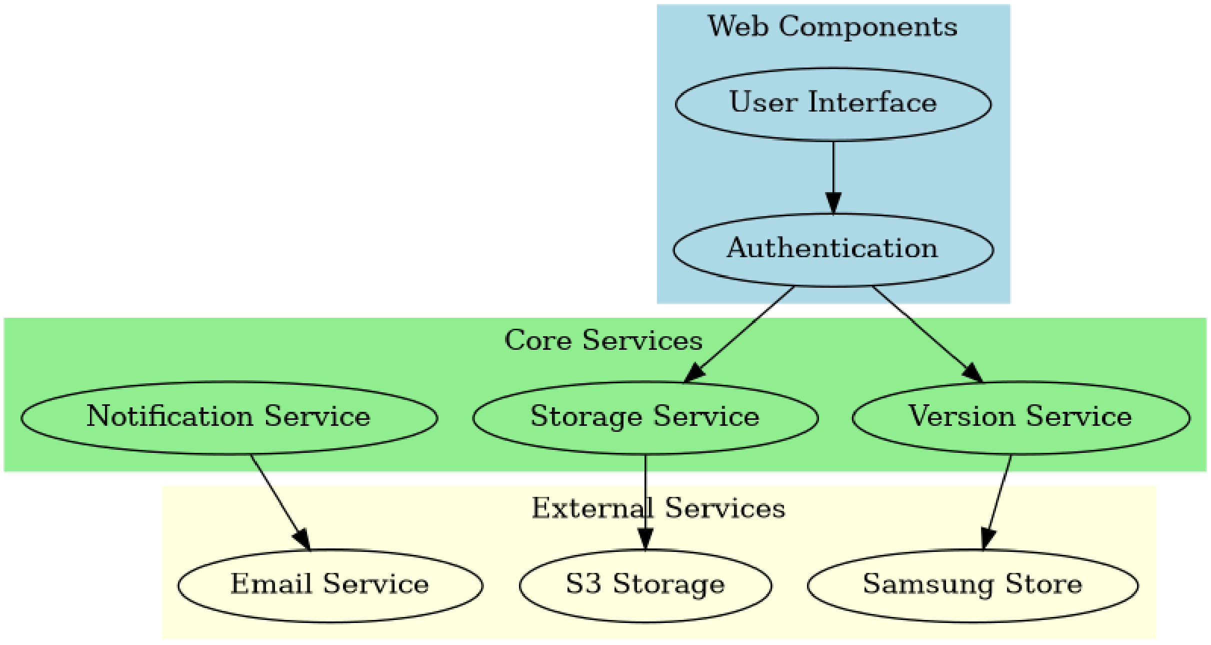

Figure 3: Component Diagram

Monitoring

```
GET    /api/apps/:id/status   # Check current status
POST   /api/apps/:id/check    # Force version check
GET    /api/alerts            # List all alerts
PUT    /api/alerts/:id        # Update alert status
```

Security Considerations

1. **APK Verification**
   - Hash verification for file integrity
   - Signature validation
   - Malware scanning integration
2. **Access Control**
   - Role-based access control
   - Audit logging
   - API authentication
3. **Data Protection**
   - Encrypted storage
   - Secure file transfer
   - Regular backup procedures

Monitoring and Alerts

Alert Types

1. New version available
2. Version mismatch
3. Download failures
4. Storage warnings

Monitoring Metrics

- Version check frequency
- Alert response time
- Storage utilization
- API performance

Future Enhancements

1. **Automation**
    - Automated APK extraction
    - Batch processing capabilities
    - Integration with CI/CD pipelines
2. **Analytics**
    - Version adoption tracking
    - Update pattern analysis
    - Usage statistics
3. **Integration**
    - MMP integration
    - Automated deployment to test environments
    - Extended store support (beyond Samsung)

Development Guidelines

Code Structure

- MVC architecture for web interface
- Microservices for core functionalities
- Event-driven architecture for monitoring

Best Practices

- Comprehensive error handling
- Extensive logging
- Regular security audits
- Performance optimization

Testing Strategy

- Unit tests for core functions
- Integration tests for API endpoints
- End-to-end testing for critical flows
- Security testing

Deployment

Requirements

- Linux-based server
- PostgreSQL database
- Redis for caching
- S3-compatible storage

Configuration

- Environment-based settings
- Feature flags

- Monitoring setup
- Backup configuration

Maintenance

Regular Tasks

- Log rotation
- Database optimization
- Storage cleanup
- Security updates

Backup Strategy

- Daily database backups
- APK archive backups
- Configuration backups
- Disaster recovery procedures