BigCorp Integration Technical Specification

Version: 1.0.0 Last Updated: 2024-03-15 Status: Approved

Overview

Technical specification for BigCorp’s enterprise integration requirements, focusing on bulk data operations, security, and performance requirements.

API Endpoints

Bulk Import

```
POST /api/v1/imports/bulk
Content-Type: application/vnd.bigcorp.v1+json
Authorization: HMAC {signature}

{
  "metadata": {
    "source": "bigcorp-crm",
    "version": "1.0.0",
    "timestamp": "2024-03-15T00:00:00Z"
  },
  "records": [
    {
      "id": string,
      "type": "customer" | "order" | "product",
      "attributes": Record<string, any>,
      "customFields": Record<string, string>
    }
  ]
}
```

Import Status

```
GET /api/v1/imports/:jobId
Authorization: HMAC {signature}

Response:
{
  "status": "pending" | "processing" | "completed" | "failed",
  "progress": {
    "total": number,
    "processed": number,
    "failed": number
  },
  "errors": Array<{
    "record": string,
    "error": string
  }>
}
```

Validation

```
POST /api/v1/imports/validate
Content-Type: application/vnd.bigcorp.v1+json
Authorization: HMAC {signature}

// Same schema as bulk import
```

Performance Requirements

Rate Limits

- Bulk endpoints: 1000 requests/minute
- Status endpoints: 5000 requests/minute
- Validation endpoints: 2000 requests/minute

Concurrency

- Maximum 5 concurrent bulk imports per customer
- Each import job can process up to 500k records
- Expected average processing time: 10 minutes per 100k records

Resource Limits

- Maximum payload size: 50MB
- Custom fields: 50 per record
- Field name length: 256 characters
- Field value length: 1024 characters

Security Requirements

Authentication

- HMAC authentication required for all endpoints
- Keys rotated every 90 days
- Failed authentication results in 401 response

Encryption

- All data encrypted at rest using AES-256
- TLS 1.3 required for all API connections
- Customer-specific encryption keys for stored data

Audit

- All bulk operations logged with:
  - Timestamp
  - User ID
  - Operation type
  - Record count
  - Source IP
  - Request ID

Architecture

Processing Pipeline

1. Request validation
2. HMAC verification
3. Rate limit check
4. Payload validation
5. Job creation
6. Queue message publishing
7. Async processing
8. Status updates via WebSocket

Infrastructure

- AWS SQS for job queue
- Lambda for processing
- Redis cluster for rate limiting
- Primary/Secondary architecture for HA
- Multi-AZ deployment

Error Handling

- Retry logic for failed records
- Dead letter queue for unprocessable records
- Automatic notification on failure thresholds
- Error aggregation in status response

Monitoring

- Real-time metrics dashboard
- Processing speed per record type

- Error rate monitoring
- Queue depth alerts
- Rate limit usage tracking