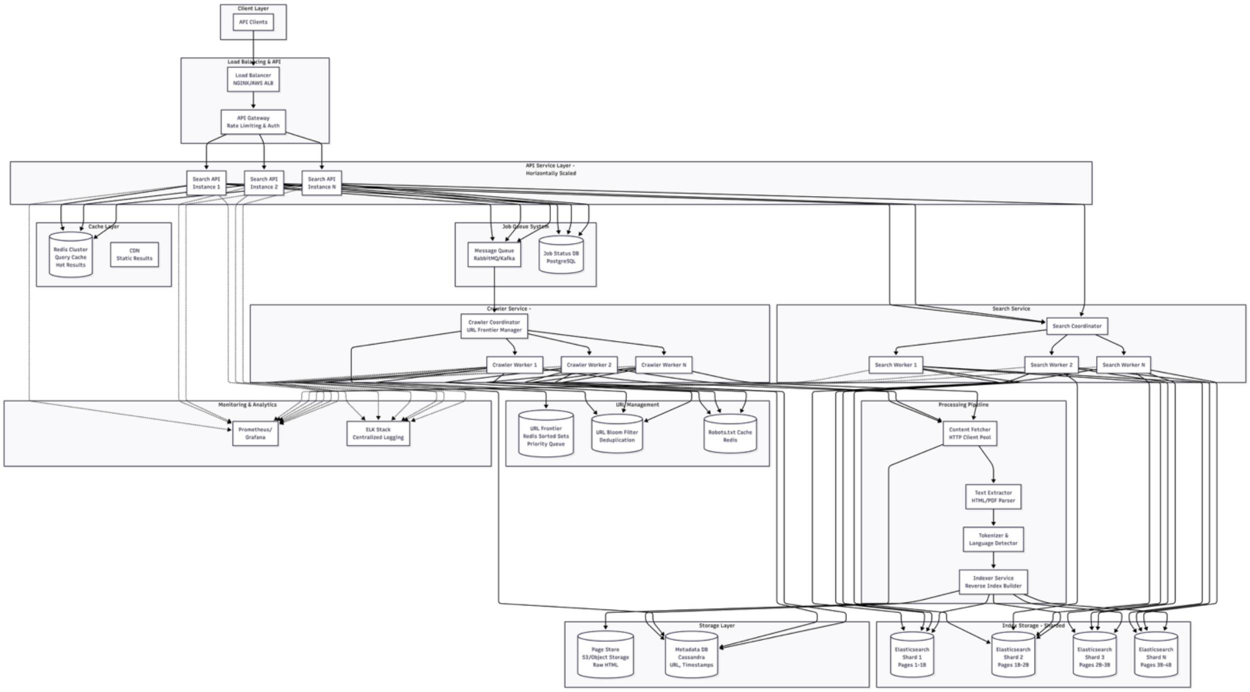
High Level Architecture and diagram flow



The key Component in Architecture diagram:

- Load balancing and API gateway layer
- Horizontally scaled API services
- Multi-layer caching (Redis + CDN)
- Distributed search service with Elasticsearch sharding
- Job queue system for crawl management
- Distributed crawler service with URL management
- Processing pipeline (fetch, extract, tokenize, index)
- Storage layers for pages and metadata
- Monitoring and logging infrastructure

Key scaling highlights:

- 100-400 API instances for query load
- 500-2000 crawler workers
- 40-60% cache hit ratio saving billions of queries
- 20-50 Elasticsearch shards with replicas
- Geographic distribution for latency reduction

API Specification

Complete RESTful API with:

- **5 endpoints**: Search, Submit Re-crawl, Get Status, List Jobs, Cancel Job
- Detailed request/response formats
- Authentication and rate limiting
- Webhook notifications
- Error handling and HTTP status codes
- Pagination and filtering

API Implementation Code (Fast API/Python)

Production-ready code structure featuring:

- **Separation of concerns**: Services, models, dependencies separated
- **Scalability patterns**: Caching, async processing, distributed queuing
- **Service layer architecture**: CacheService, SearchService, JobQueueService.
- **Rate limiting**: Token bucket algorithm with distributed support
- **Background tasks**: Webhook notifications, job monitoring

Key Scalability Features Demonstrated

- **Caching Strategy**: Multi-layer (Redis + CDN) reducing 40-60% of backend load
- **Horizontal Scaling**: All services can scale independently
- **Sharding**: Data distributed across multiple Elasticsearch nodes
- **Async Processing**: Job queue pattern for re-crawl requests
- **Load Balancing**: Traffic distribution across API instances
- **Rate Limiting**: Prevents abuse and ensures fair usage
- **Monitoring**: Built-in health checks and metrics

Search Engine API Specification v1.0

Base URL

```
https://api.searchengine.com/v1
```

Authentication

All API requests require authentication using API keys passed in the header:

```
Authorization: Bearer YOUR_API_KEY
```

Rate Limits:

- Standard tier: 1,000 requests/minute
- Premium tier: 10,000 requests/minute
- Enterprise tier: Custom limits

Rate limit headers included in all responses:

```
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 995
X-RateLimit-Reset: 1638360000
```

Endpoints

1. Search for Pages

Endpoint: GET /search

Description: Search the indexed web pages with advanced filtering and ranking options.

Query Parameters:

Parameter	Type	Required	Description
q	string	Yes	Search query (max 500 characters)
page	integer	No	Page number for pagination (default: 1)
per_page	integer	No	Results per page (default: 10, max: 100)
language	string	No	Filter by language code (e.g., 'en', 'es')
date_from	string	No	Filter results from date (ISO 8601: YYYY-MM-DD)
date_to	string	No	Filter results to date (ISO 8601: YYYY-MM-DD)

sort	string	No	Sort order: 'relevance' (default), 'date', 'popularity'
safe_search	boolean	No	Enable safe search filtering (default: false)
fields	string	No	Comma-separated fields to return (e.g., 'title,url,snippet')

Request Example:

```
GET /search?q=machine+learning&page=1&per_page=20&language=en&sort=relevance
Authorization: Bearer abc123xyz789
```

Response: 200 OK

```
{
  "query": "machine learning",
  "total_results": 15420000,
  "page": 1,
  "per_page": 20,
  "total_pages": 771000,
  "search_time_ms": 145,
  "results": [
    {
      "id": "doc_8f7a3b2c",
      "url": "https://example.com/ml-intro",
      "title": "Introduction to Machine Learning",
      "snippet": "Machine learning is a subset of artificial intelligence that enables systems to learn...",
      "crawled_at": "2024-12-10T14:23:45Z",
      "last_modified": "2024-12-08T10:15:30Z",
      "language": "en",
      "score": 0.95,
      "metadata": {
        "author": "Jane Doe",
        "domain": "example.com",
        "content_type": "text/html"
      }
    }
  ],
  "pagination": {
    "next": "/search?q=machine+learning&page=2&per_page=20",
    "previous": null,
    "first": "/search?q=machine+learning&page=1&per_page=20",
    "last": "/search?q=machine+learning&page=771000&per_page=20"
  }
}
```

Error Responses:

400 Bad Request - Invalid parameters

```
{
  "error": "bad_request",
  "message": "Query parameter 'q' is required",
  "details": {
    "parameter": "q",
    "issue": "missing_required_field"
  }
}
```

429 Too Many Requests - Rate limit exceeded

```
{
  "error": "rate_limit_exceeded",
  "message": "Rate limit of 1000 requests/minute exceeded",
  "retry_after": 45
}
```

2. Request Re-crawl

Endpoint: POST /recrawl

Description: Submit a URL for priority re-crawling with 1-hour SLA.

Request Headers:

```
Content-Type: application/json
Authorization: Bearer YOUR_API_KEY
```

Request Body:

Field	Type	Required	Description
url	string	Yes	Full URL to re-crawl (must be previously indexed)
priority	string	No	Priority level: 'standard', 'high', 'urgent' (default: 'high')
callback_url	string	No	Webhook URL for job completion notification
force	boolean	No	Force re-crawl even if recently crawled (default: false)

Request Example:

```
POST /recrawl
Authorization: Bearer abc123xyz789
Content-Type: application/json

{
  "url": "https://example.com/updated-article",
  "priority": "urgent",
  "callback_url": "https://client.com/webhook/crawl-complete",
  "force": true
}
```

Response: 202 Accepted

```
{
  "job_id": "recrawl_9d8c7b6a5f4e3d2c",
  "status": "queued",
  "url": "https://example.com/updated-article",
  "priority": "urgent",
  "sla_deadline": "2024-12-12T15:30:00Z",
  "estimated_completion": "2024-12-12T14:45:00Z",
  "created_at": "2024-12-12T13:30:00Z",
  "callback_url": "https://client.com/webhook/crawl-complete",
  "status_url": "/recrawl/recrawl_9d8c7b6a5f4e3d2c"
}
```

Error Responses:

400 Bad Request - Invalid URL

```
{
  "error": "invalid_url",
  "message": "The provided URL is not valid or not previously indexed",
  "details": {
    "url": "https://example.com/updated-article",
    "issue": "url_not_indexed"
  }
}
```

403 Forbidden - Insufficient quota

```
{
  "error": "quota_exceeded",
  "message": "Monthly re-crawl quota exceeded",
  "details": {
    "used": 1000,
    "limit": 1000,
    "resets_at": "2025-01-01T00:00:00Z"
  }
}
```

429 Too Many Requests - Rate limit exceeded

```
{
  "error": "rate_limit_exceeded",
  "message": "Too many concurrent re-crawl requests",
  "retry_after": 60
}
```

3. Get Re-crawl Job Status

Endpoint: GET /recrawl/{job_id}

Description: Retrieve the current status of a re-crawl job.

Path Parameters:

Parameter	Type	Required	Description
job_id	string	Yes	Unique job identifier

Request Example:

```
GET /recrawl/recrawl_9d8c7b6a5f4e3d2c
Authorization: Bearer abc123xyz789
```

Response: 200 OK

Status: Queued

```
{
  "job_id": "recrawl_9d8c7b6a5f4e3d2c",
  "status": "queued",
  "url": "https://example.com/updated-article",
  "priority": "urgent",
  "sla_deadline": "2024-12-12T15:30:00Z",
  "estimated_completion": "2024-12-12T14:45:00Z",
  "created_at": "2024-12-12T13:30:00Z",
  "started_at": null,
  "completed_at": null,
  "queue_position": 3
}
```

Status: Processing

```
{
  "job_id": "recrawl_9d8c7b6a5f4e3d2c",
  "status": "processing",
  "url": "https://example.com/updated-article",
  "priority": "urgent",
  "sla_deadline": "2024-12-12T15:30:00Z",
  "created_at": "2024-12-12T13:30:00Z",
  "started_at": "2024-12-12T13:35:00Z",
  "completed_at": null,
  "progress": {
    "stage": "fetching",
    "percentage": 35
  }
}
```

Status: Completed

```
{
  "job_id": "recrawl_9d8c7b6a5f4e3d2c",
  "status": "completed",
  "url": "https://example.com/updated-article",
  "priority": "urgent",
  "sla_deadline": "2024-12-12T15:30:00Z",
  "sla_met": true,
  "created_at": "2024-12-12T13:30:00Z",
  "started_at": "2024-12-12T13:35:00Z",
  "completed_at": "2024-12-12T14:12:00Z",
  "duration_seconds": 2220,
  "result": {
    "success": true,
    "document_id": "doc_8f7a3b2c",
    "indexed_at": "2024-12-12T14:12:00Z",
    "content_changed": true,
    "http_status": 200
  }
}
```

Status: Failed

```
{
  "job_id": "recrawl_9d8c7b6a5f4e3d2c",
  "status": "failed",
  "url": "https://example.com/updated-article",
  "priority": "urgent",
  "sla_deadline": "2024-12-12T15:30:00Z",
  "sla_met": false,
  "created_at": "2024-12-12T13:30:00Z",
  "started_at": "2024-12-12T13:35:00Z",
  "completed_at": "2024-12-12T14:05:00Z",
  "result": {
    "success": false,
    "error_code": "fetch_timeout",
    "error_message": "Failed to fetch URL after 3 retries",
    "http_status": 0,
    "retry_count": 3
  }
}
```

Error Responses:

404 Not Found - Job not found

```
{
  "error": "job_not_found",
  "message": "Re-crawl job with ID 'recrawl_9d8c7b6a5f4e3d2c' not found"
}
```

4. List Re-crawl Jobs

Endpoint: GET /recrawl

Description: List all re-crawl jobs for the authenticated user.

Query Parameters:

Parameter	Type	Required	Description
status	string	No	Filter by status: 'queued', 'processing', 'completed', 'failed'
page	integer	No	Page number (default: 1)
per_page	integer	No	Results per page (default: 20, max: 100)
date_from	string	No	Filter jobs created from date (ISO 8601)
date_to	string	No	Filter jobs created to date (ISO 8601)

Request Example:

```
GET /recrawl?status=completed&page=1&per_page=20
Authorization: Bearer abc123xyz789
```

Response: 200 OK

```
{
  "total_jobs": 145,
  "page": 1,
  "per_page": 20,
  "total_pages": 8,
  "jobs": [
    {
      "job_id": "recrawl_9d8c7b6a5f4e3d2c",
      "status": "completed",
      "url": "https://example.com/updated-article",
      "created_at": "2024-12-12T13:30:00Z",
      "completed_at": "2024-12-12T14:12:00Z",
      "sla_met": true
    }
  ],
  "pagination": {
    "next": "/recrawl?status=completed&page=2&per_page=20",
    "previous": null
  }
}
```

5. Cancel Re-crawl Job

Endpoint: DELETE /recrawl/{job_id}

Description: Cancel a queued or processing re-crawl job.

Path Parameters:

Parameter	Type	Required	Description
job_id	string	Yes	Unique job identifier

Request Example:

```
DELETE /recrawl/recrawl_9d8c7b6a5f4e3d2c
Authorization: Bearer abc123xyz789
```

Response: 200 OK

```
{
  "job_id": "recrawl_9d8c7b6a5f4e3d2c",
  "status": "cancelled",
  "message": "Re-crawl job successfully cancelled",
  "cancelled_at": "2024-12-12T13:45:00Z"
}
```

Error Responses:

400 Bad Request - Cannot cancel completed job

```
{
  "error": "invalid_operation",
  "message": "Cannot cancel job with status 'completed'"
}
```

Webhook Notifications

When a `callback URL` is provided with a re-crawl request, the system will POST to that URL upon job completion.

Webhook Payload:

```
{
  "event": "recrawl.completed",
  "timestamp": "2024-12-12T14:12:00Z",
  "job_id": "recrawl_9d8c7b6a5f4e3d2c",
  "status": "completed",
  "url": "https://example.com/updated-article",
  "result": {
    "success": true,
    "document_id": "doc_8f7a3b2c",
    "indexed_at": "2024-12-12T14:12:00Z",
    "content_changed": true
  }
}
```

Webhook Security:

- All webhooks include signature in `X-Signature` header
- Signature = HMAC-SHA256(payload, webhook_secret)
- Verify signature before processing webhook

HTTP Status Codes

Code	Description
200	Success
201	Resource created
202	Request accepted for processing
400	Bad request - invalid parameters
401	Unauthorized - invalid or missing API key
403	Forbidden - insufficient permissions or quota
404	Not found - resource doesn't exist
429	Too many requests - rate limit exceeded
500	Internal server error

503	Service unavailable - temporary issue

Error Response Format

All error responses follow this structure:

```
{
  "error": "error_code",
  "message": "Human-readable error message",
  "details": {
    "additional": "context"
  },
  "request_id": "req_abc123xyz",
  "documentation_url": "https://docs.searchengine.com/errors/error_code"
}
```

Pagination

All list endpoints support pagination using `page` and `per_page` parameters.

Pagination Response:

```
{
  "pagination": {
    "next": "/endpoint?page=2",
    "previous": "/endpoint?page=0",
    "first": "/endpoint?page=1",
    "last": "/endpoint?page=100"
  }
}
```

Rate Limiting

Rate limits are enforced per API key based on tier:

Tier	Search Requests/min	Re-crawl Requests/hour	Concurrent Re-crawls
Standard	1,000	100	5
Premium	10,000	1,000	50
Enterprise	Custom	Custom	Custom

Rate Limit Headers:

```
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 995
X-RateLimit-Reset: 1638360000
Retry-After: 45
```

Best Practices

1. **Implement exponential backoff** for retry logic on 429 and 5xx errors
2. **Cache search results** on your side to reduce API calls
3. **Use pagination** efficiently - don't request all pages at once
4. **Monitor SLA compliance** for critical re-crawl requests
5. **Validate webhook signatures** before processing
6. **Use appropriate priority levels** for re-crawl requests