Title: URL Shortener

Description: A scalable, globally distributed URL shortening service designed for low-latency redirects, high availability, and real-time click analytics.

Name: Meka.Nithisha

Registration number: 12400718

Roll No.: 25

1. Stakeholder Analysis & Prioritization

Stakeholders

- **Marketers:** Need custom links, branded domains, campaign analytics.
- **App Developers:** Need stable APIs, automation, integrations.
- **End Users:** Require fast global redirects and reliable uptime.
- **Analytics Consumers:** Need aggregated click insights.
- **Security/Compliance:** Need abuse controls, domain safety, and governance.

Prioritization

Priority	Stakeholder Need
Critical	Redirect latency, correctness, availability
High	Accurate analytics, strong API reliability
Medium	Custom alias features, dashboard polish
Low	Advanced segmentation, export tooling

2. Functional Requirements

1. **Create Short Link**
    - Auto-generated key or custom alias
    - TTL, tags, optional metadata
2. **Resolve Link (Redirect)**
    - GET /s/{id} → 302 redirect
    - Must be globally fast (p95 < 50 ms)
3. **Track Clicks**
    - Device → via User-Agent
    - Geography → via IP lookup
    - Referrer tracking
4. **Analytics**
    - Total clicks, unique clicks
    - Time-series charts (hour/day)
    - Top countries/referrers/devices
5. **Admin Controls**

    - Link deletion, TTL change
    - Disable malicious links
6. **API + Dashboard**
    - CRUD for links
    - Analytics retrieval

3. Non-Functional Requirements

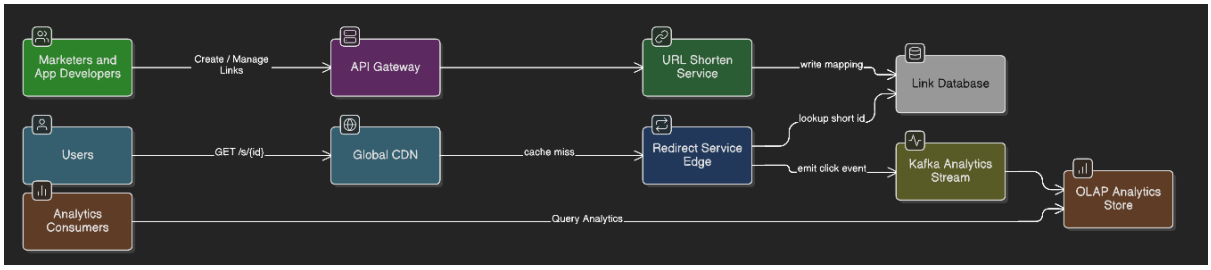
- **Availability:** 99.99% for redirects
- **Performance:** p95 < 50ms, p99 < 100ms
- **Scalability:** 1B+ clicks/day, 100k+ peak QPS
- **Durability:** Never lose link mappings
- **Strong consistency** for redirect mapping
- **Eventual consistency** for analytics
- **Security:** AuthN/AuthZ, rate limiting, domain safety
- **Global delivery:** Anycast + multi-region

4. Constraints & Assumptions

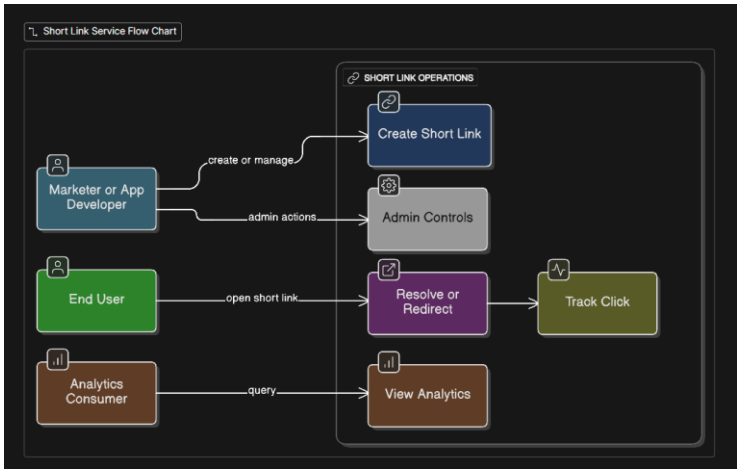
- Redirect path is read-heavy (~95% handled by CDN).
- Analytics path is write-heavy and asynchronous.
- Multi-region replica latency varies → write operations centralized.
- Analytics can tolerate 1–5 minute delay.

2. Architecture & Diagrams:-

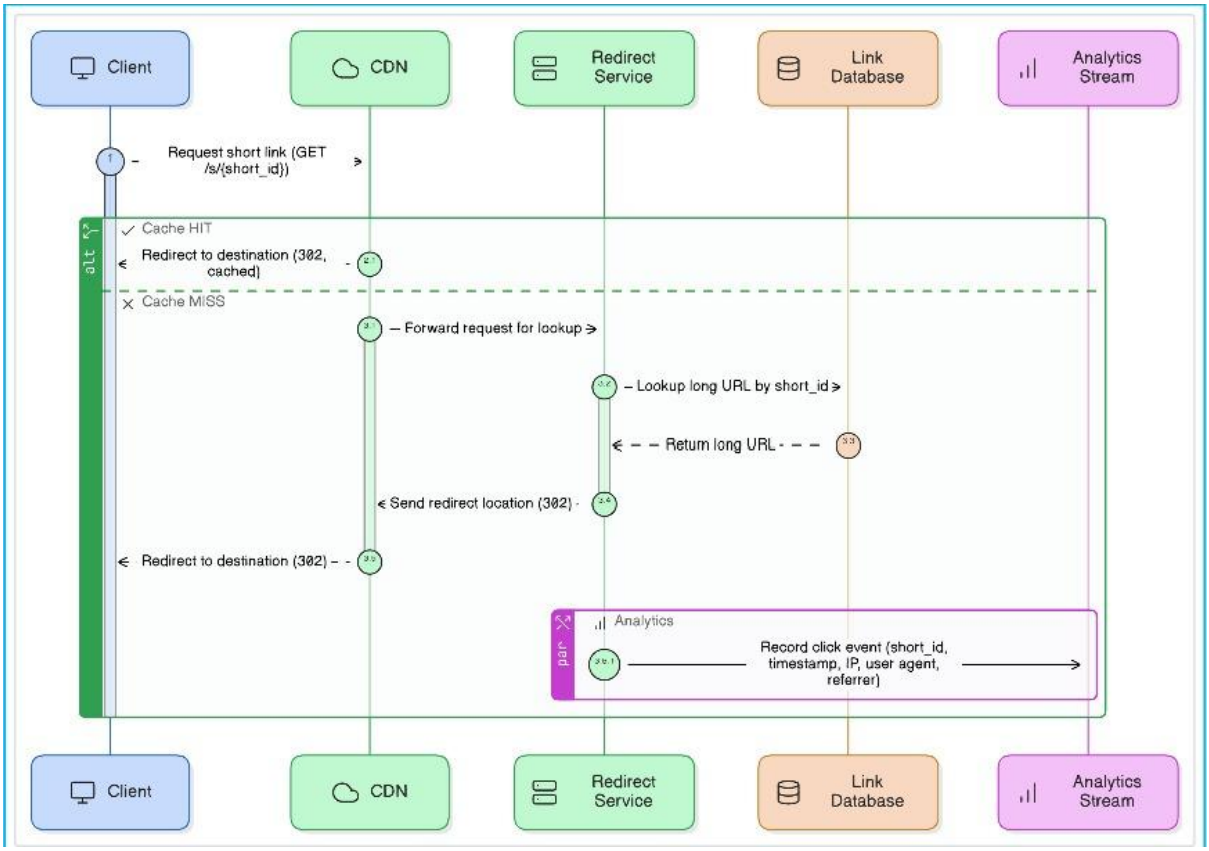
2.1 System Context Diagram (Simplified ASCII)



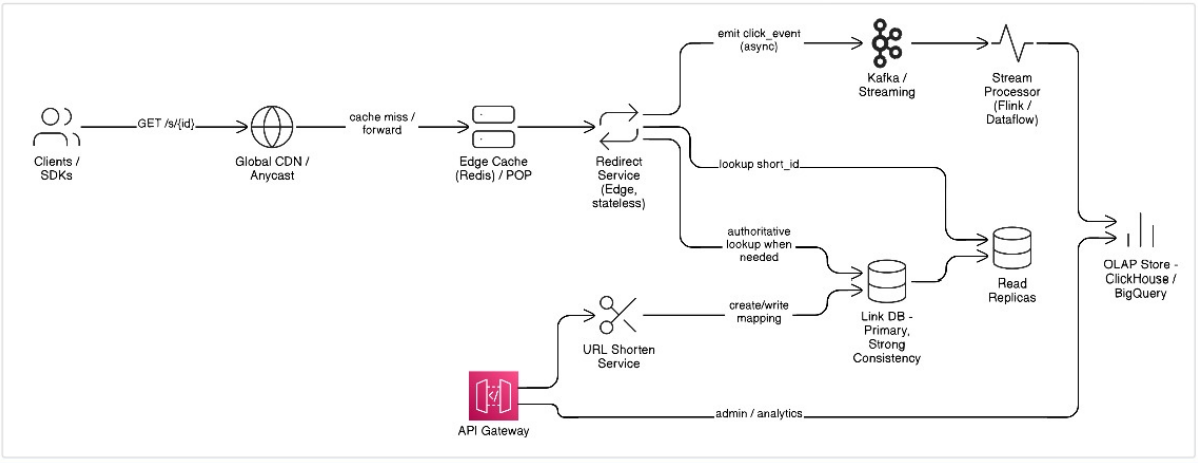
2.2 Use-Case Diagram



2.3 Sequence Diagram – Redirect



2.4 High-Level Architecture



3. Engineering Notes

3.1 Capacity Planning

Assumptions

- 100M short links
- 1B clicks/day → ~12k clicks/sec avg

- Peak traffic → 100k+ QPS

Cache Efficiency

- CDN handles ~95%
- Edge cache handles ~90% of remaining 5%
- DB receives ≈ 0.5% of total traffic (~500 QPS)

Data Volume

- Raw analytics: 1B events × 200 bytes ≈ 200 GB/day
- OLAP compression reduces to 40–60 GB/day

3.2 API Specs

POST /v1/links

Request:

{ long_url, custom_alias?, ttl?, metadata? }

Response:

{ short_id, short_url }

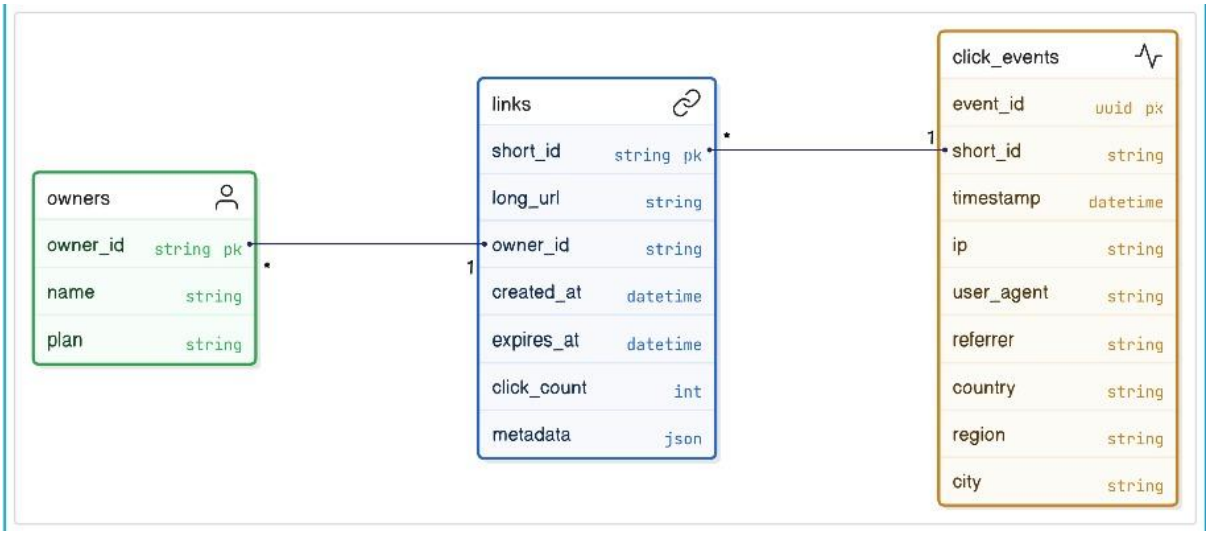
GET /s/{short_id}

Returns 302 redirect.

GET /v1/links/{id}/stats

Returns analytics aggregates.

3.3 Data Model



3.4 Consistency

- Link resolution → **Strong consistency**
- Analytics → **Eventual consistency (1–5 mins)**

- Cache invalidation via TTL + purge API

3.5 Caching & Indexing

- **CDN caching:** 30–60 seconds
- **Edge Redis caching:** 1–24 hours
- **Local LRU cache:** microsecond lookups
- Index:
    - PK: short_id
    - Secondary index: owner_id

3.6 Rate Limiting

- Per-IP throttling
- Per-token API limits
- Custom alias namespace protection
- Malicious domain blocklist

3.7 Resiliency

- **Circuit breakers** around DB + Kafka
- **Retry with exponential backoff**
- **Regional failover** through Anycast routing
- Redirect service continues even if analytics pipeline is degraded

3.8 Observability

- Metrics: latency, QPS, cache hit-rate, DB latency, stream lag
- Logs: structured JSON with PII redaction
- Traces: OpenTelemetry across redirect → DB → analytics

4. Quality Targets & Trade-Offs

4.1 SLOs

Category	Target
Redirect availability	**99.99%**
Redirect latency p95	**< 50ms**
Analytics freshness	**95% < 5 minutes**

Category	Target
API availability	99.9%

4.2 Scalability Plan

- Horizontal scaling of stateless services
- Sharding link DB by short_id prefix
- Kafka partitioning by short_id hash
- OLAP partitioning by date + hash
- CDN/edge expansion to lower global latency

4.3 Trade-Off Discussion

- **Strong consistency** increases redirect correctness, but adds write latency from remote regions.
- **Eventual consistency** in analytics allows massive scale at lower cost.
- **CDN caching** drastically improves performance but requires purge workflows.
- **Primary-region writes** simplify correctness; replicas handle global reading.