System Design Document

High-Level Architecture

HTML Processing Service

Input: Raw HTML content
Output: Processed HTML with shortened URLs

Components:

- HTML Parser (e.g., JSDOM)
- URL Shortener Service
- URL Storage

URL Shortener Service Backend

Input: Original URL
Output: Shortened URL

Components:

- URL Generation (e.g., SHA-256 hash)
- URL Storage
- Redirection and Analytics Capture

Analytics Data Storage and Reporting System

Input: Click events
Output: Analytics data

Components:

- Analytics Storage
- Reporting API

Data Flow

1 HTML Processing Pipeline

graph TD A[Raw HTML] --> B(Parse HTML) B --> C{Identify Links} C --> D[Replace URLs] D --> E[Store Mappings] E --> F[Processed HTML]

2 Click Handling Flow

graph TD A[Short URL Click] --> B{Lookup Original} B --> C[Record Analytics] C --> D[302 Redirect] D --> E[User Follows Link]

From Raw HTML Input to Processed HTML with Shortened Links

1. HTML Processor parses the HTML.
2. Identifies all hyperlinks.
3. Replaces original URLs with shortened URLs using the URL Shortener Service.
4. Stores the mapping of original to shortened URLs.

From Link Click to Analytics Capture and Redirection

1. User clicks on a shortened URL.
2. URL Shortener Service captures the click event (timestamp, user agent, IP address).
3. Redirects the user to the original URL.
4. Stores the click event in the Analytics Storage.

Scalability Considerations

Scaling Strategies

Component Scaling Approach HTML Processing Horizontal scaling with auto-scaling group URL Shortener Sharded Redis cluster + Read replicas Analytics Kafka-based event streaming + Batch processing

Handling Growth

- Use a distributed storage system for URLs and analytics data.
- Implement load balancing for the URL Shortener Service.

Fault Tolerance and Monitoring

- Implement retry mechanisms for failed requests.
- Use monitoring tools (e.g., Prometheus, Grafana) to track system health.

- Disaster Recovery: Multi-region replication with 15-minute RPO

Implementation Steps

gantt title Implementation Phases dateFormat YYYY-MM-DD section Core Services

```
 HTML Processor        :active, p1, 2025-03-17, 14d
URL Shortener         : p2, after p1, 21d
Analytics Pipeline    : p3, after p2, 28d
section Testing
E2E Testing           : p4, after p3, 14d
```

Development Phases

1. **Phase 1**: Implement HTML Processor.
2. **Phase 2**: Implement URL Shortener Service.
3. **Phase 3**: Implement Analytics Capture and Reporting.
4. **Phase 4**: Integrate all components and perform end-to-end testing.

Testing Strategies

- Unit tests for individual components.
- Load testing to ensure scalability.

Deployment Checklist

- Load balancer configuration
- Auto-scaling policies
- Database replication setup
- Monitoring alerts configuration
- Rate limiting enabled
- TLS termination configured

Maintenance Plan

Daily: Check system health metrics

Weekly: Audit security rules

Monthly: Review scaling thresholds

Quarterly: Full disaster recovery drill