Tiny URL

Functional Requirements	Non Functional Requirements
Given a long url create a short url	Very low latency
Given a short url redirect to a long url	Very high availability

Tiny URL

API Design

REST API

1. POST: /create-url
    - Params: long-url
    - Status code: 201 Created
2. Get: /{short-url}
    - Status code: 301 Permanent Redirect

Tiny URL

API Design

REST API

1. POST: /create-url
   - Params: long-url
   - Status code: 201 Created
2. Get: /{short-url}
   - Status code: 301 Permanent Redirect

Schema

long-url	string
short-url	string
created-at	timestamp

Tiny URL

Key question: How long should the URL be?

Tiny URL

Key question: How long should the URL be?

1. Need to know the scale of the application?

Tiny URL

Key question: How long should the URL be?

1. Need to know the scale of the application?

- Example: 1,000 URLs generated per second

1,000 * 60 * 60 * 24 * 365 = 31.5 billion urls created each year

Tiny URL

Key question: How long should the URL be?

1. Need to know the scale of the application?

   - Example: 1,000 URLs generated per second

1,000 * 60 * 60 * 24 * 365 = 31.5 billion urls created each year

10 to 1 read to write requests means 300 billion reads per year

Tiny URL

Key question: How long should the URL be?

1. Need to know the scale of the application?

- Example: 1,000 URLs generated per second

1,000 * 60 * 60 * 24 * 365 = 31.5 billion urls created each year

10 to 1 read to write requests means 300 billion reads per year

What characters can we use?

Tiny URL

Key question: How long should the URL be?

1. Need to know the scale of the application?

- Example: 1,000 URLs generated per second

1,000 * 60 * 60 * 24 * 365 = 31.5 billion urls created each year

10 to 1 read to write requests means 300 billion reads per year

What characters can we use?

Alphanumeric:

• a - z	26
• A - Z	26
• 0 - 9	10
	62 characters

Tiny URL

Key question: How long should the URL be?

1. Need to know the scale of the application?

- Example: 1,000 URLs generated per second

1,000 * 60 * 60 * 24 * 365 = 31.5 billion urls created each year

10 to 1 read to write requests means 300 billion reads per year

What characters can we use?

Alphanumeric:

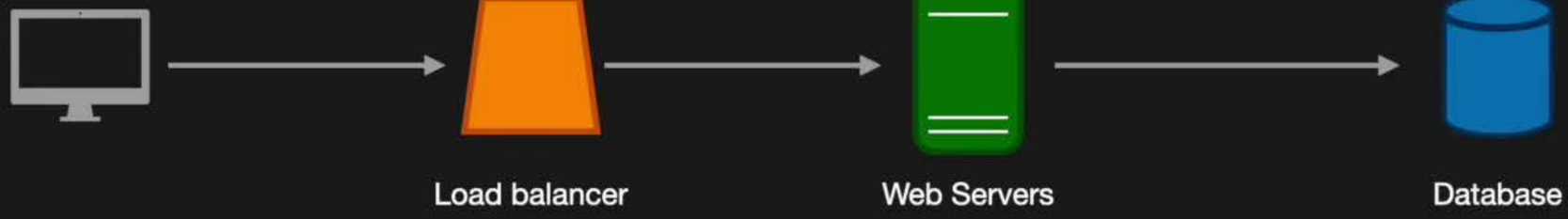
• a - z	26
• A - Z	26
• 0 - 9	10
	62 characters

Unique short urls

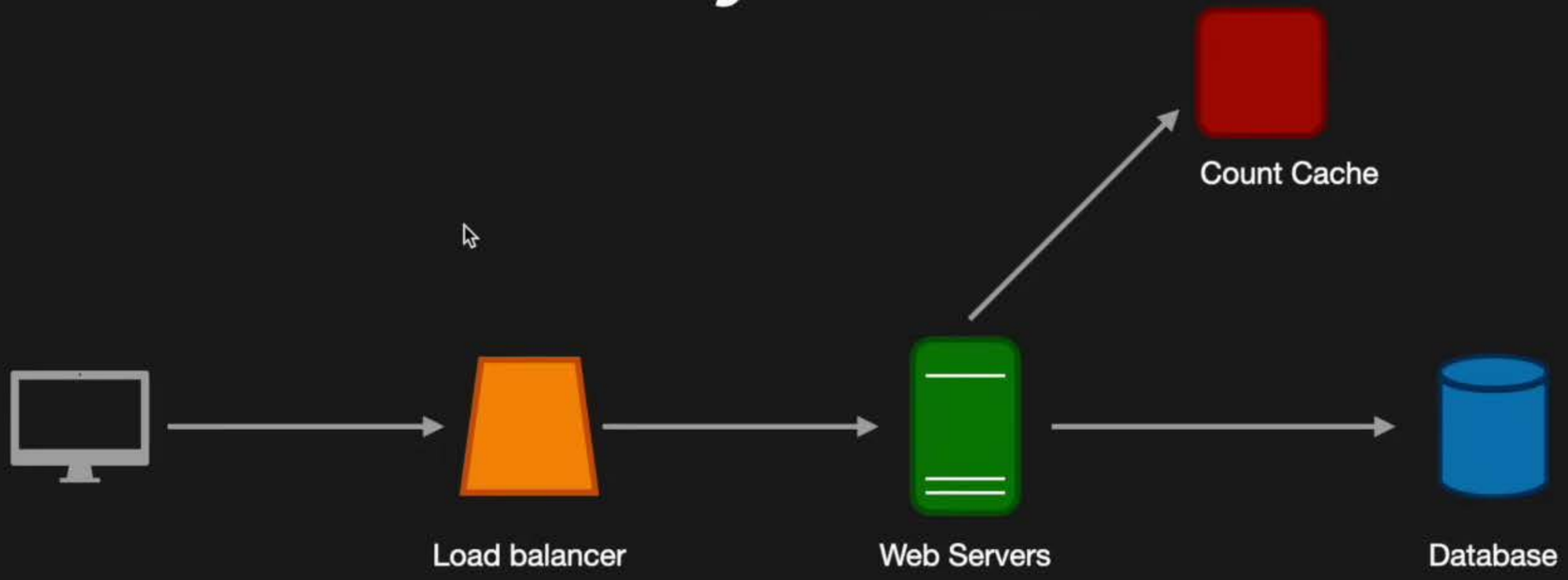
1^{62}	62
2^{62}	3,844
...	
6^{62}	56 billion
7^{62}	3.5 trillion

7 characters

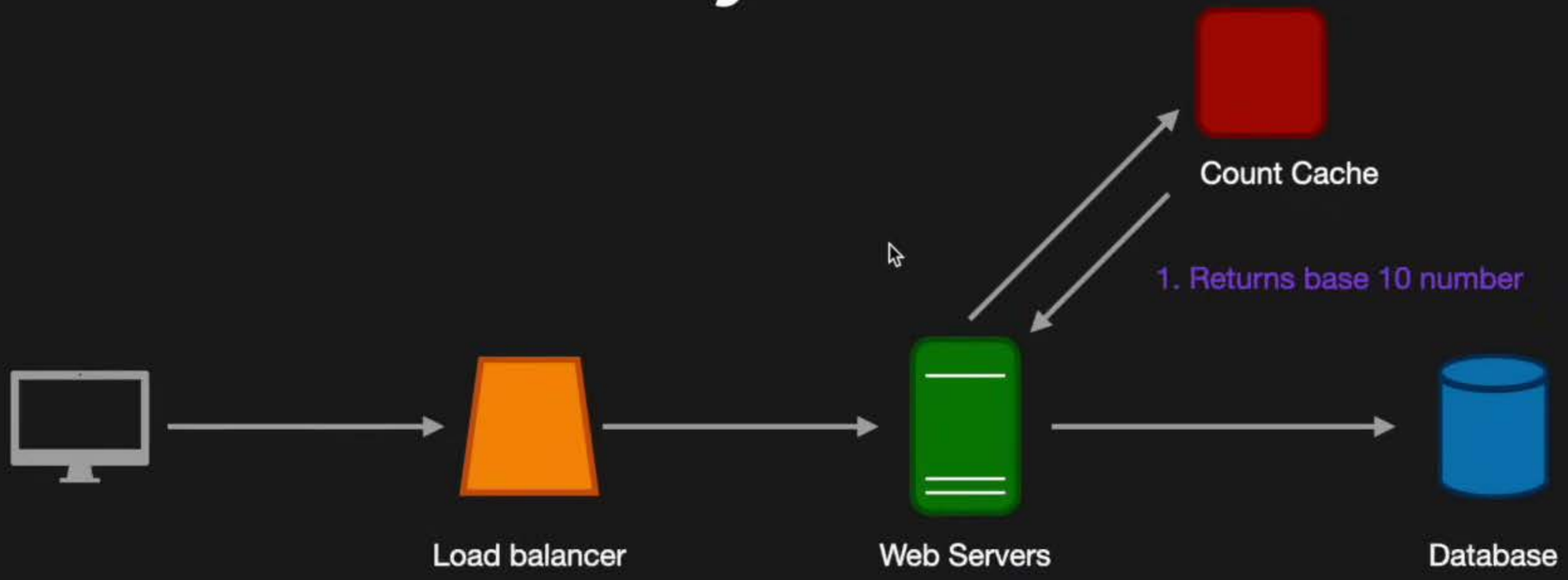
Tiny URL



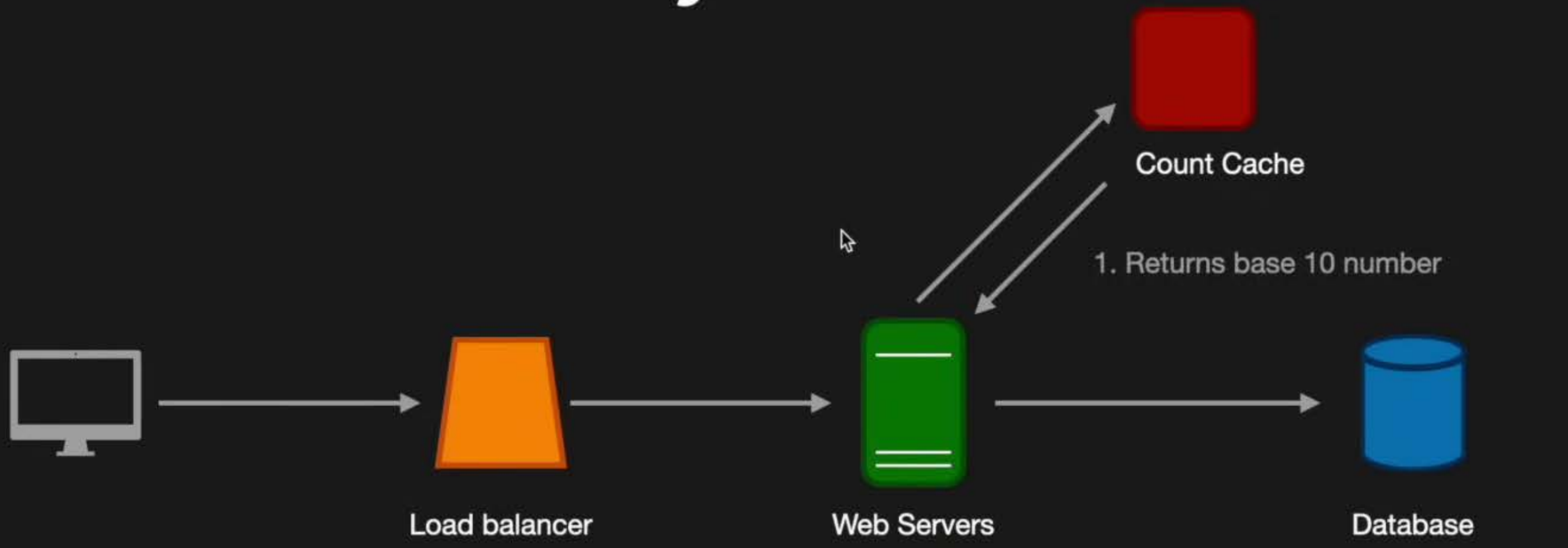
Tiny URL



Tiny URL

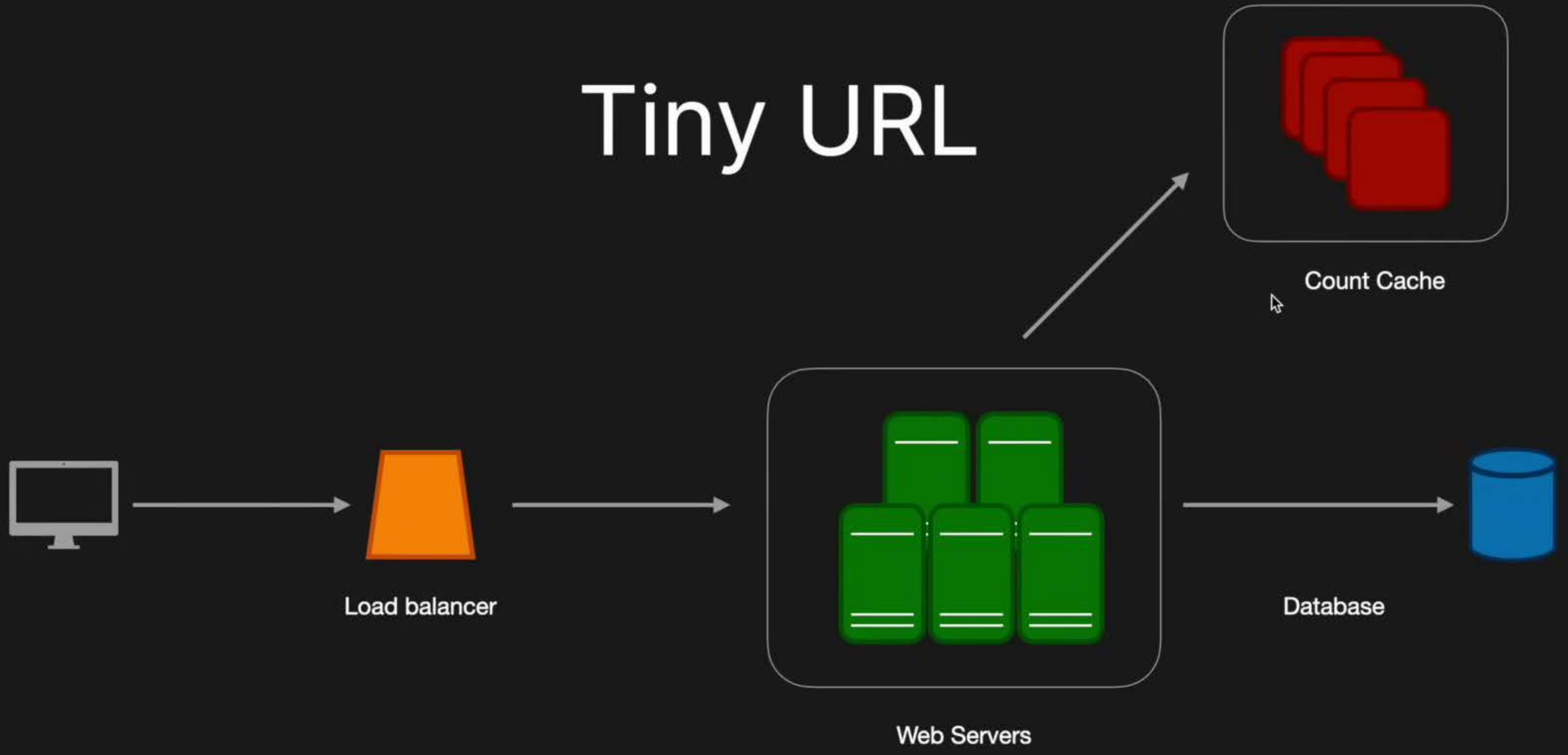

Count Cache
1. Returns base 10 number
Load balancer
Web Servers
Database

Tiny URL

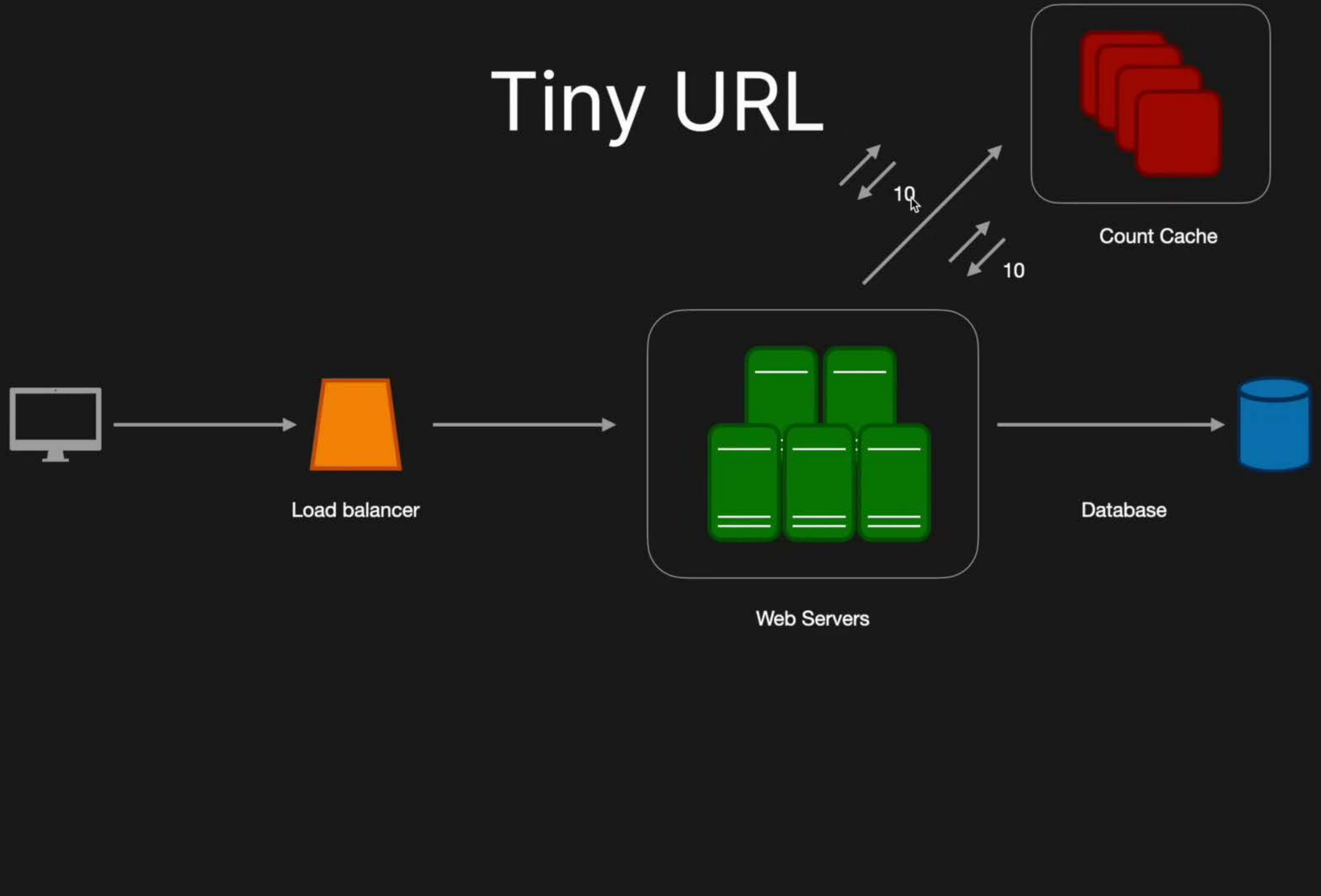


2. Converts to base 62 which will be used as the short url

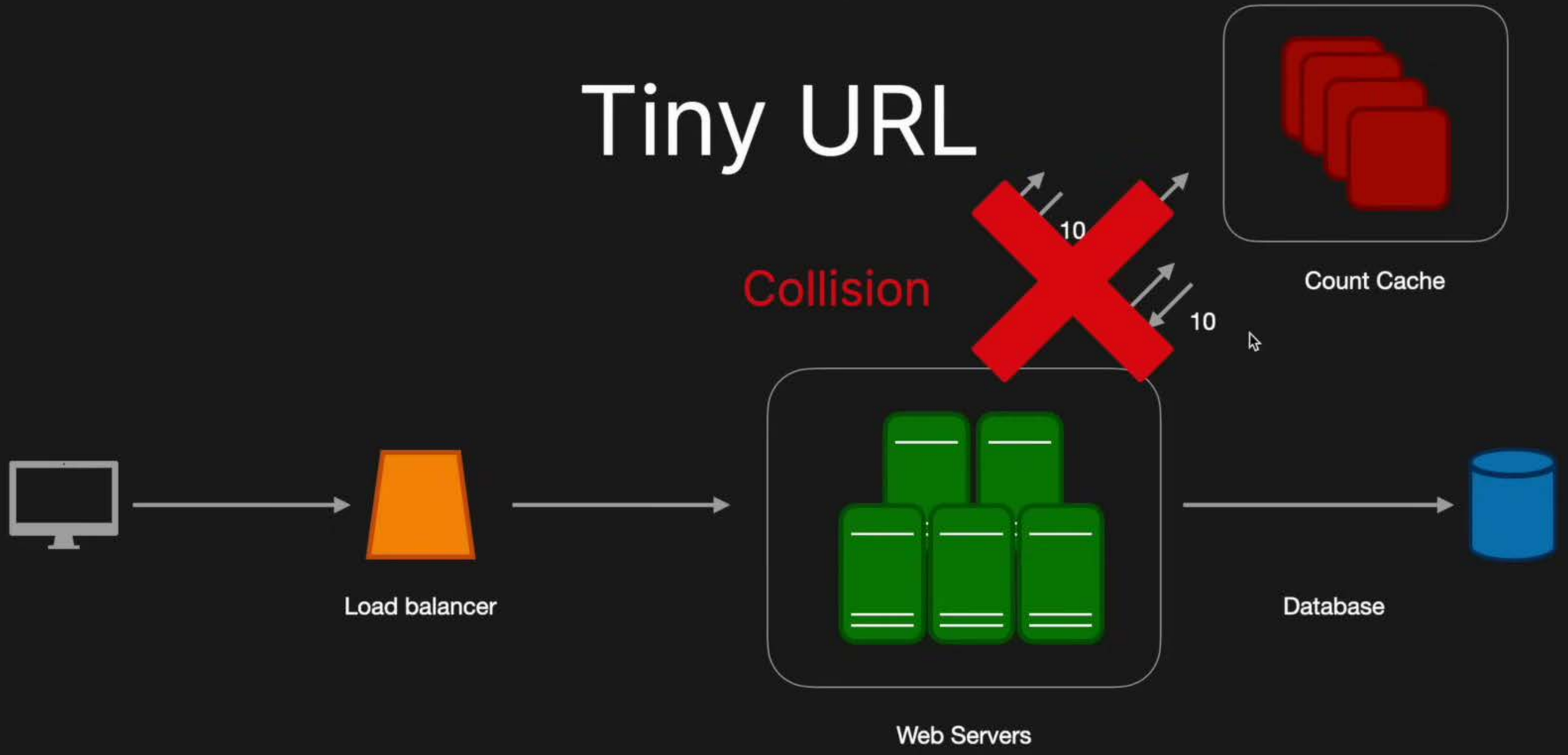
Tiny URL



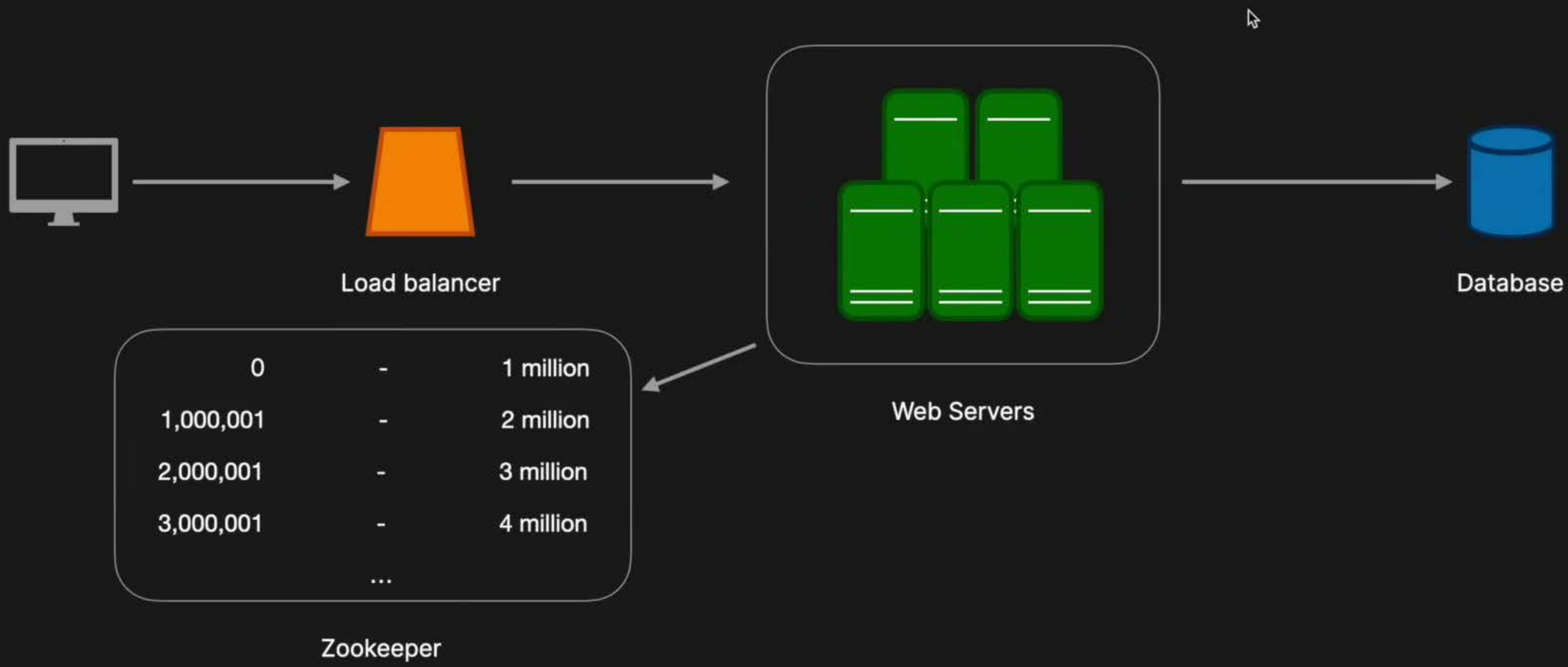
Tiny URL



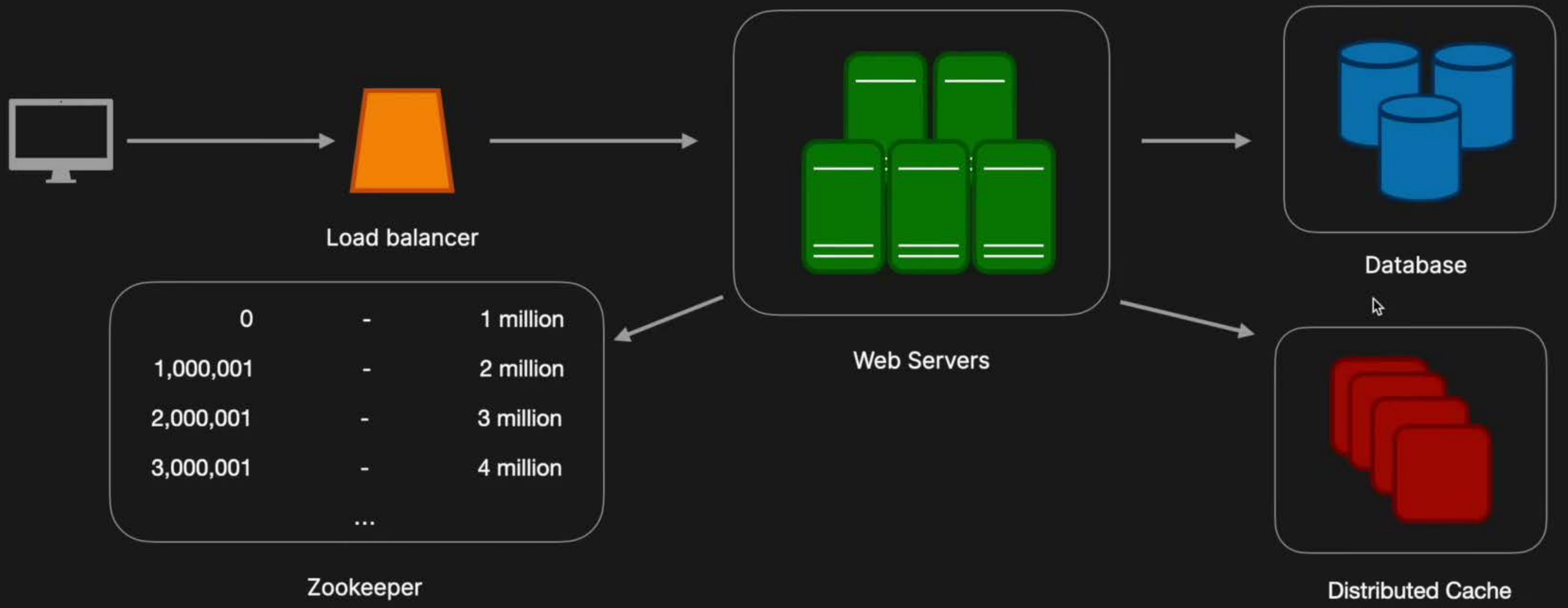
Tiny URL


Collision
10
10
Count Cache
Load balancer
Web Servers
Database

Tiny URL


Load balancer
Web Servers
Database
0 - 1 million
1,000,001 - 2 million
2,000,001 - 3 million
3,000,001 - 4 million
...
Zookeeper

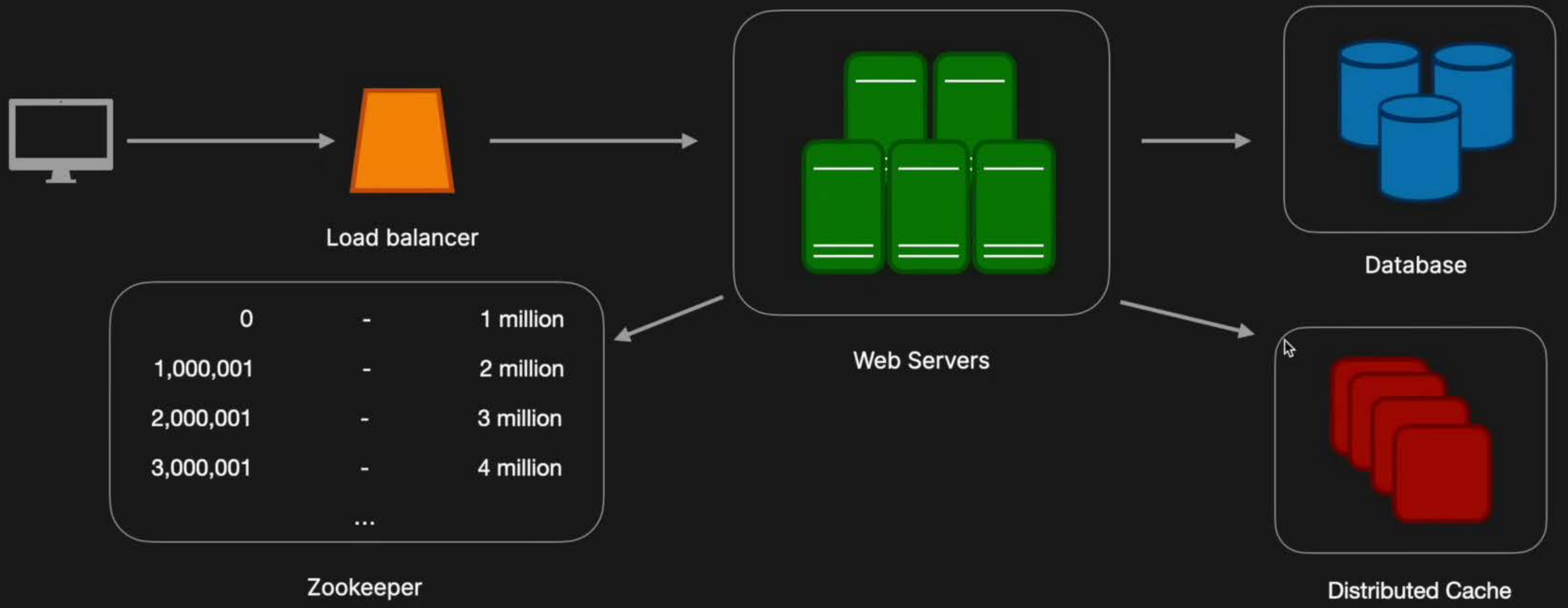
Tiny URL


Load balancer
Web Servers
Database
0 - 1 million
1,000,001 - 2 million
2,000,001 - 3 million
3,000,001 - 4 million
...
Zookeeper
Distributed Cache

Tiny URL

Post Request

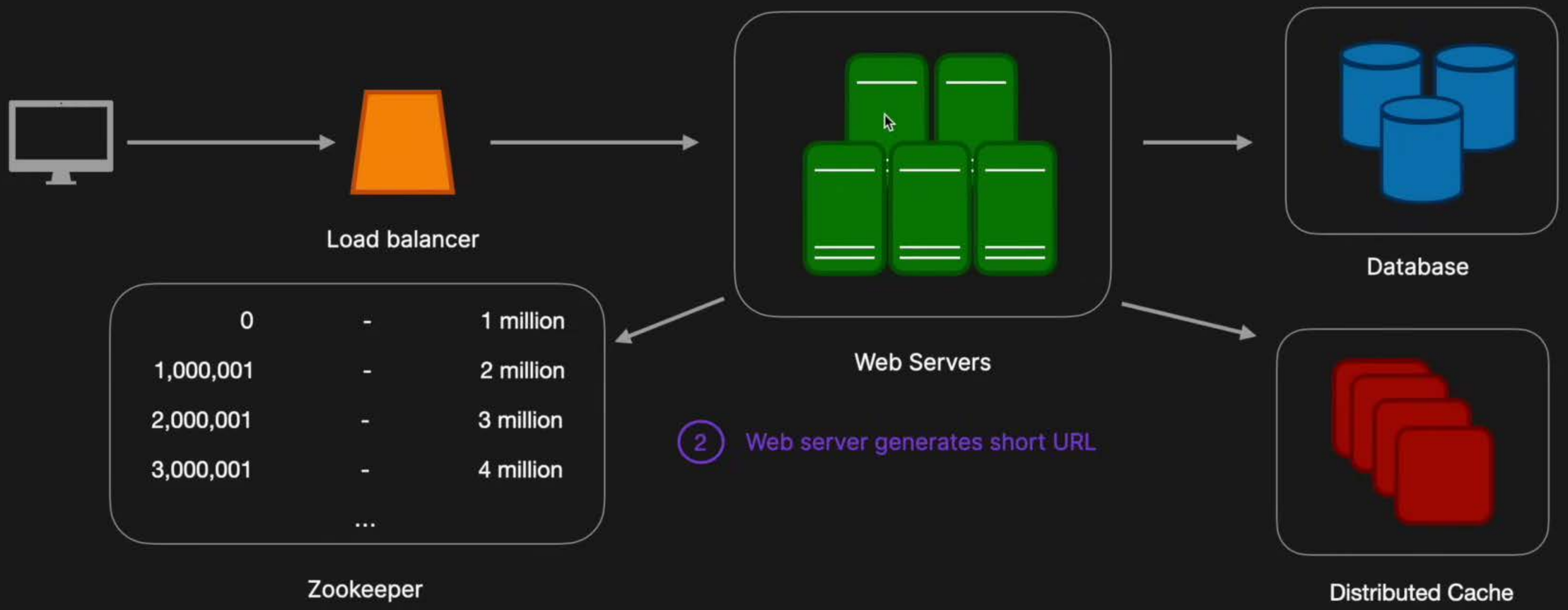
1 Post request: /create-url



Tiny URL

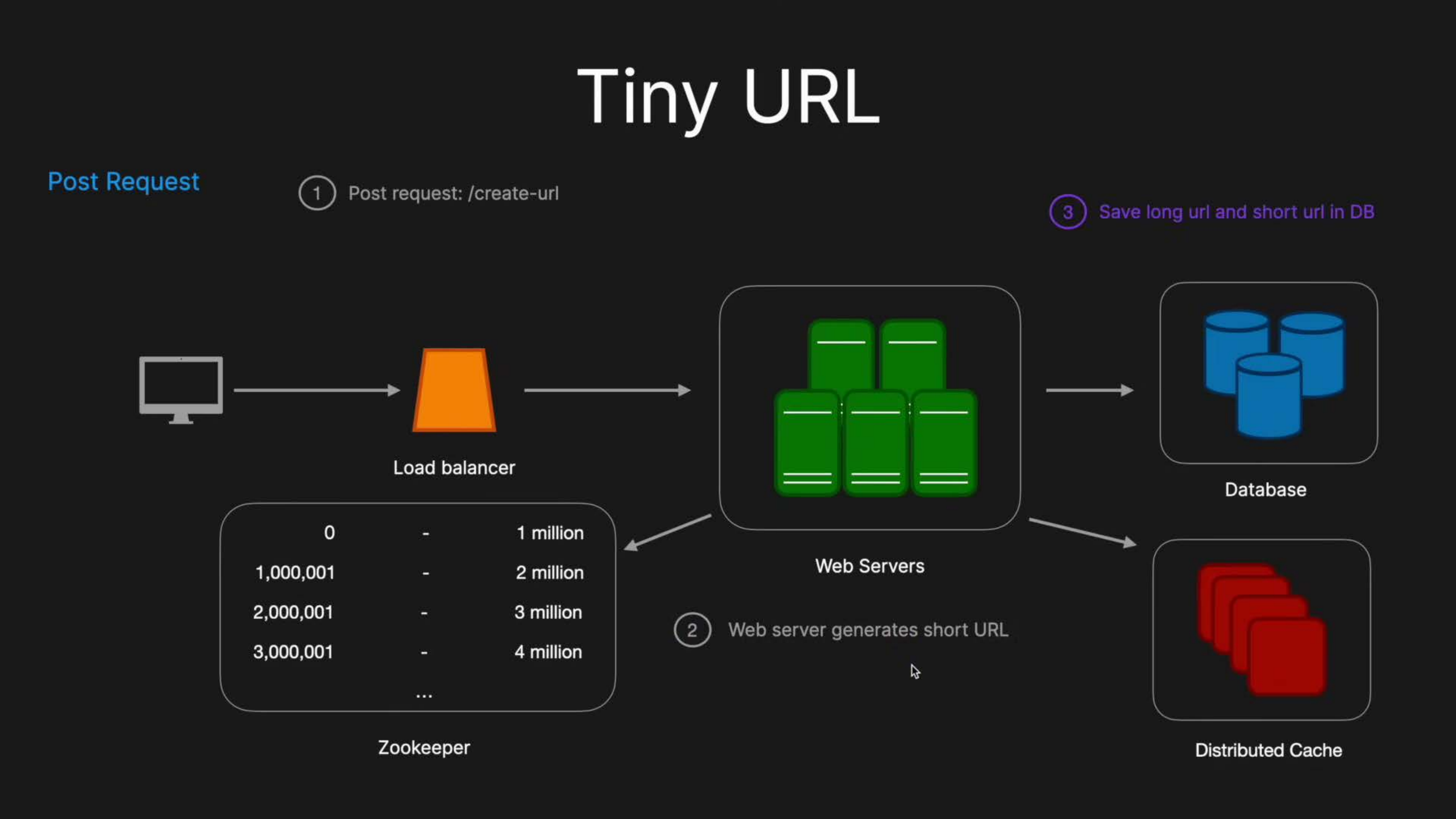
Post Request

1 Post request: /create-url



2 Web server generates short URL

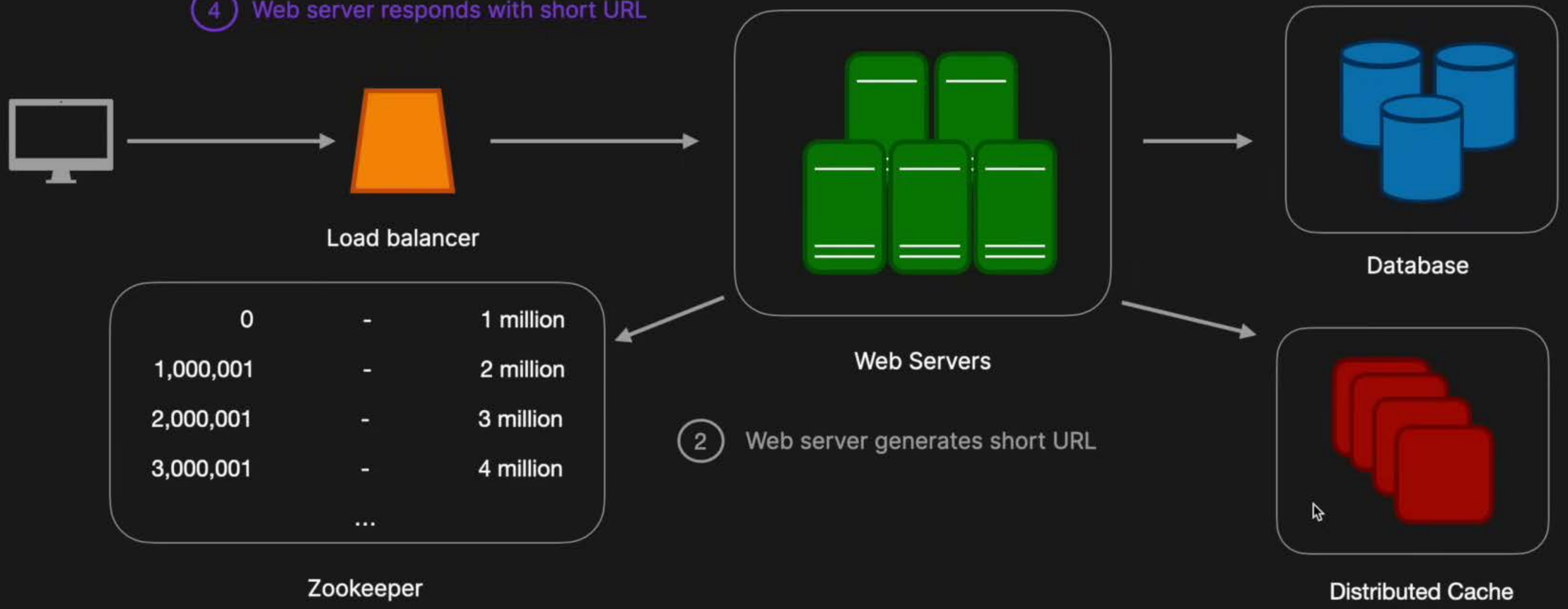
Tiny URL


Post Request
1
Post request: /create-url
3
Save long url and short url in DB
Load balancer
Web Servers
Database
0 - 1 million
1,000,001 - 2 million
2,000,001 - 3 million
3,000,001 - 4 million
...
2
Web server generates short URL
Zookeeper
Distributed Cache

Tiny URL

Post Request

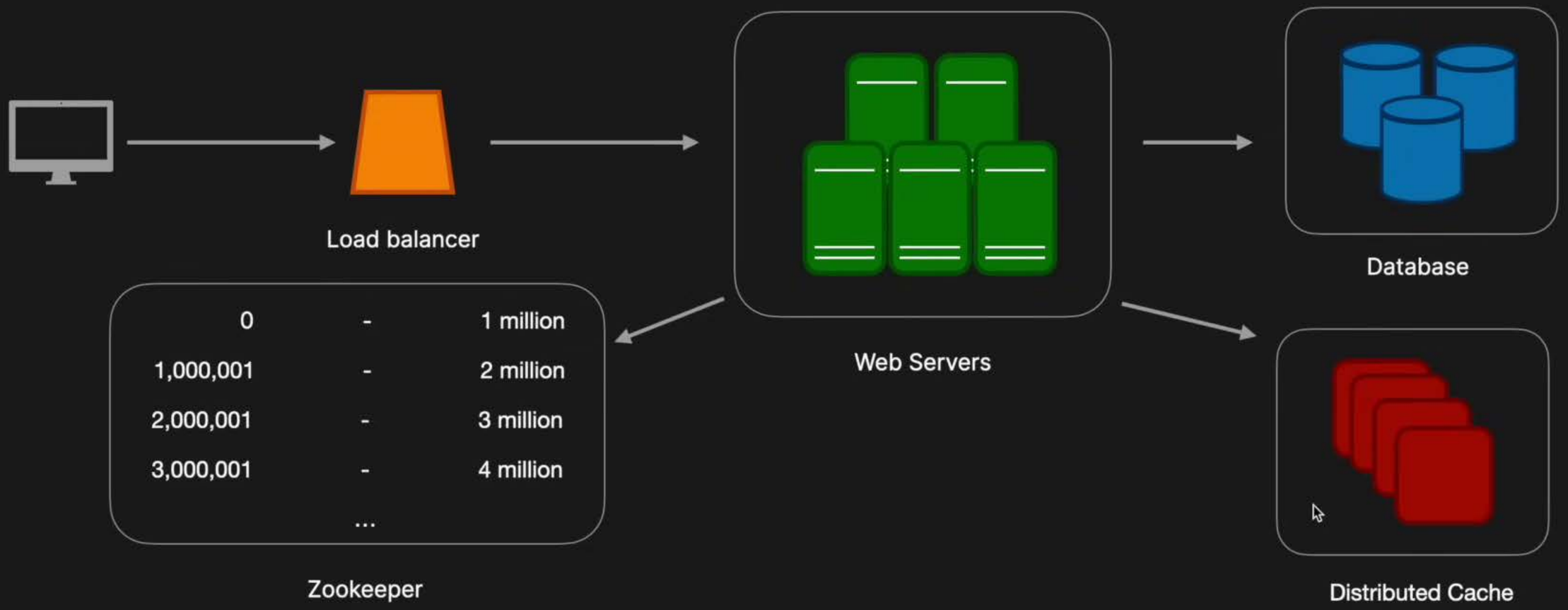
1. Post request: /create-url
2. Web server generates short URL
3. Save long url and short url in DB
4. Web server responds with short URL



Tiny URL

Get Request

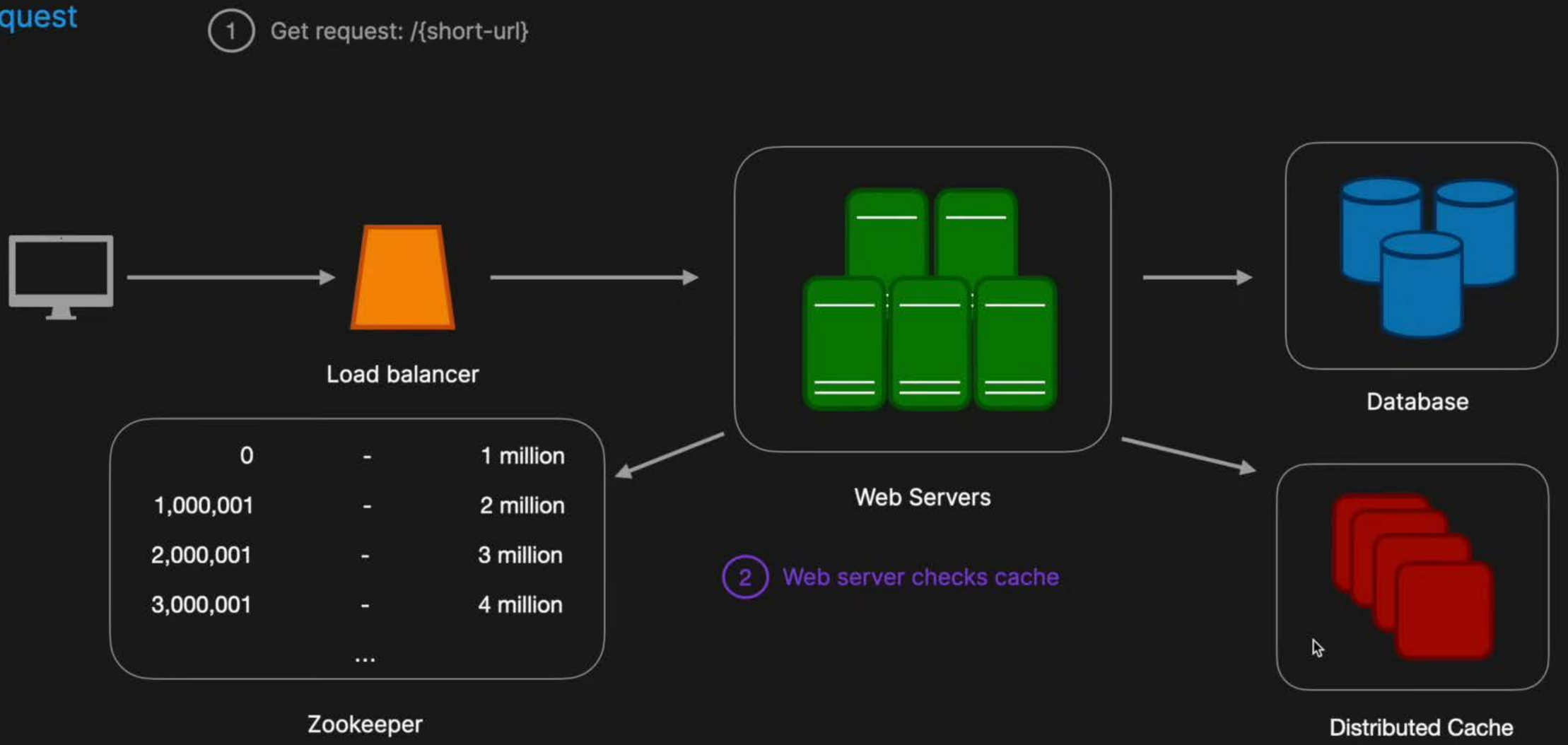
1 Get request: /{short-url}



Tiny URL

Get Request

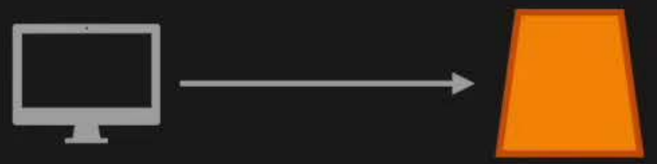
1. Get request: /{short-url}



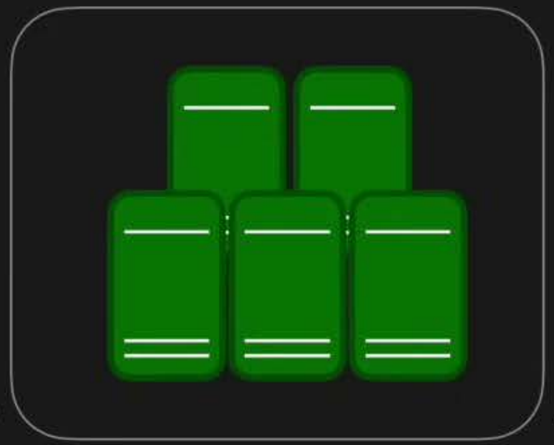
Tiny URL

Get Request

1. Get request: /{short-url}



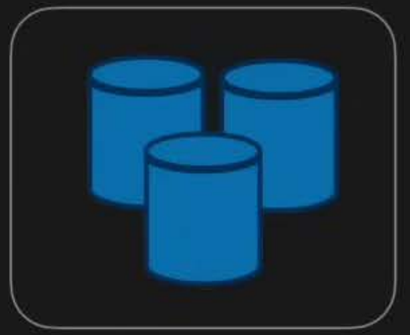
Load balancer



Web Servers

2. Web server checks cache

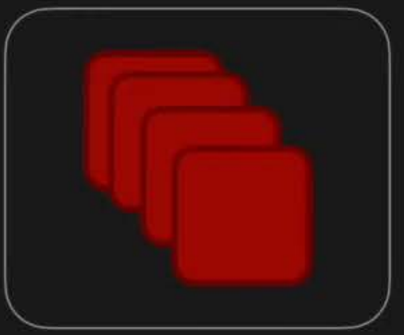
3. Web sever retrieves long url from DB



Database

0	-	1 million
1,000,001	-	2 million
2,000,001	-	3 million
3,000,001	-	4 million
	...	

Zookeeper

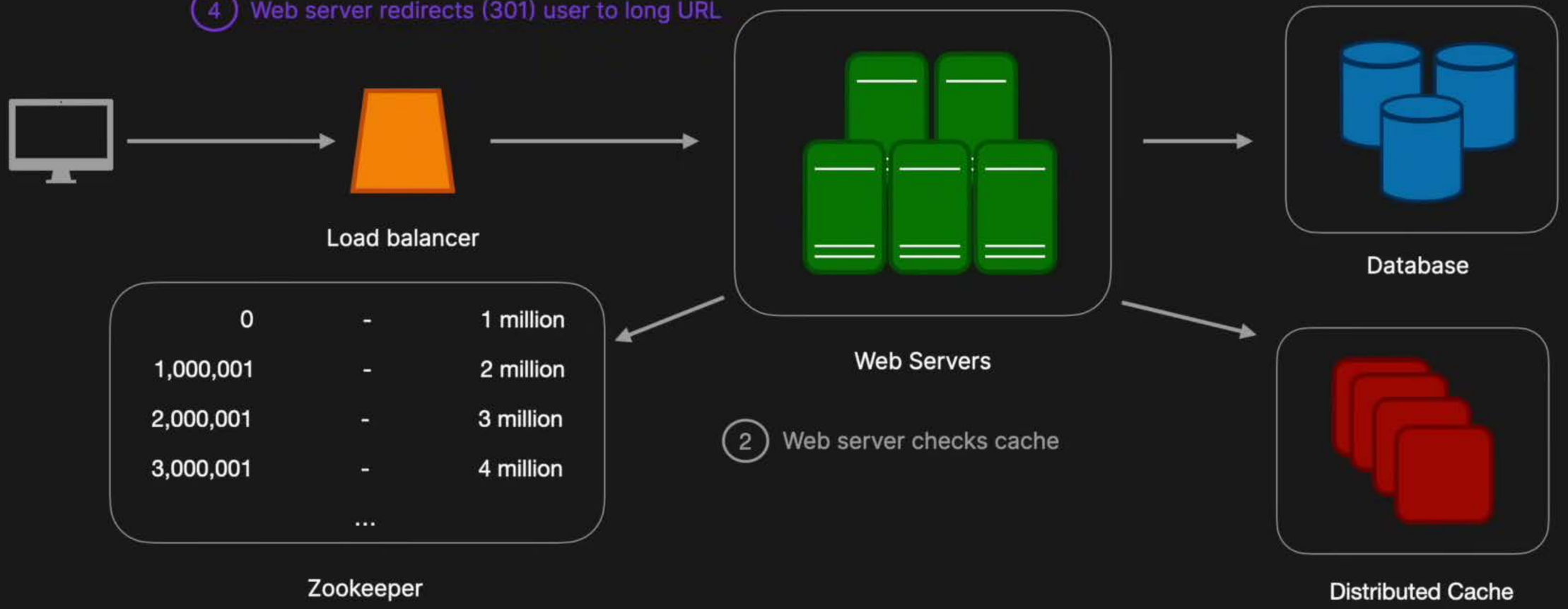


Distributed Cache

Tiny URL

Get Request

1. Get request: /{short-url}
2. Web server checks cache
3. Web sever retrieves long url from DB
4. Web server redirects (301) user to long URL



Tiny URL

Additional talking points

Tiny URL

Additional talking points

1. Analytics

- Counts for each URL to determine which short URLS to cache
- IP address to store location information to determine where to locate caches etc.

Tiny URL

Additional talking points

1. Analytics

- Counts for each URL to determine which short URLS to cache
- IP address to store location information to determine where to locate caches etc.

2. Rate limiting

- Prevent DDoS attacks by malicious users

Tiny URL

Additional talking points

1. Analytics

- Counts for each URL to determine which short URLS to cache
- IP address to store location information to determine where to locate caches etc.

2. Rate limiting

- Prevent DDoS attacks by malicious users

3. Security considerations

- Add random suffix to the short url to prevent hackers predicting urls