Secure and Scalable Webhooks

Marvin Collins Hosea

SENIOR GOLANG ENGINEER | SRE | TEAM LEAD

@MARVIN_HOSEA

What We Will Discuss

- Introduction
- What is Webhooks
- When To Use Webhooks
- Security Concerns
- Webhooks Security Approach
- Webhooks Scalability
- Last Words
- Questions

Introduction on Webhooks

□ Personal experience developing webhooks

□ Webhooks integrations are becoming increasingly popular in applications.

□ Webhooks can be a useful tool to consider for application developers.

□ Many people may not be familiar with webhooks and their potential benefits.

□ Google trends data shows a growing interest in webhooks over the past five years.

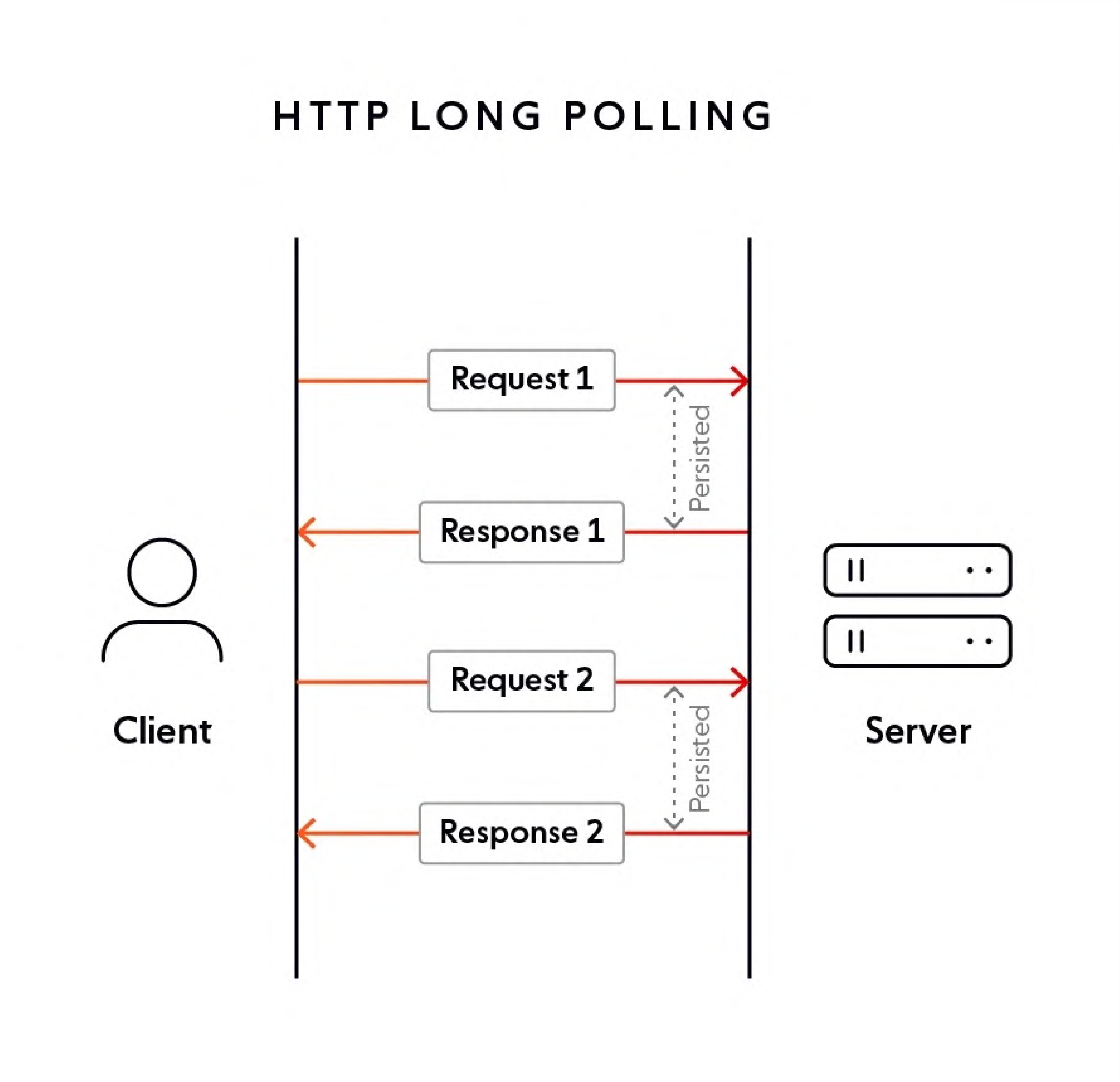
What is Webhook?

"Reverse API"

Webhooks

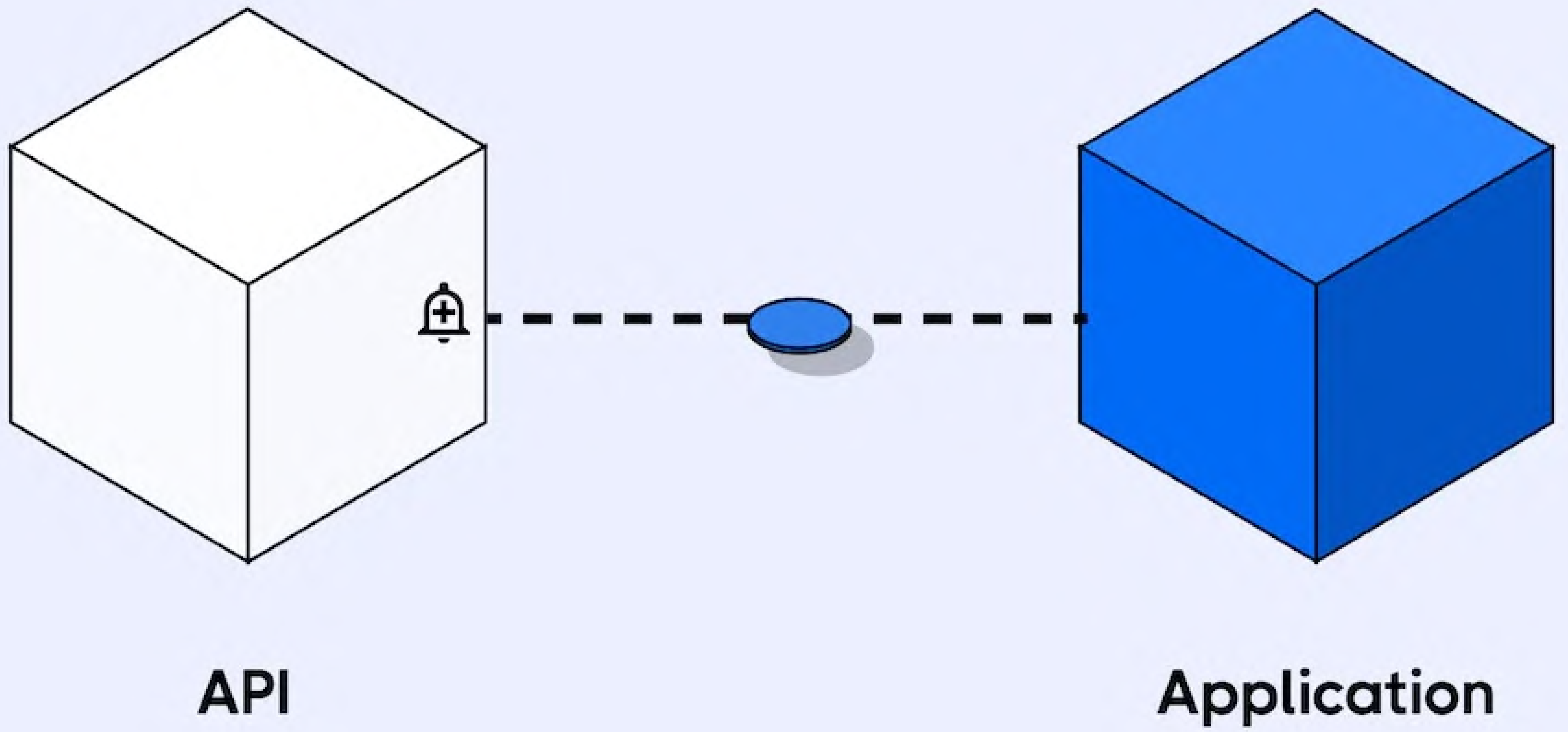
- The term "webhook" was coined by Jeff Lindsay in 2007.
- A web hook is a URL that is created by the application developer (client) to receive information from the API provider (server) without the need for polling.
- The format used is usually JSON and the request is done as a HTTP POST request.
- Webhooks comes with security tradeoffs
- Webhooks were not built to be secure out-of-the-box
- Polling

What is Polling?

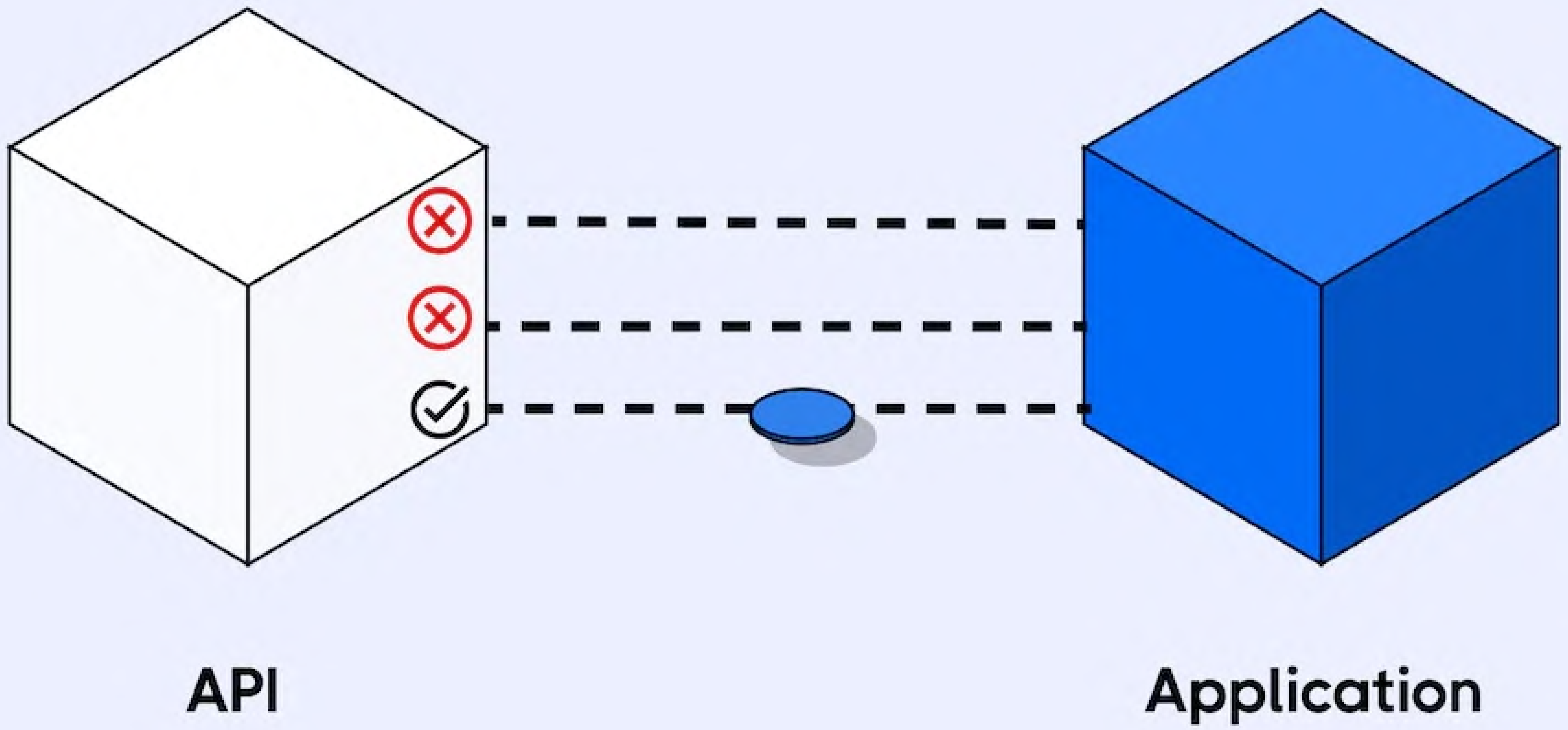


New Events

Webhooks



Polling



</>

When to use web hooks

Webhooks Usecases

- To eliminate the need for polling.
- Automate data transfer on event.
- Integration
- Triggers and actions.

Security Concerns

Webhook communication mechanism lacks a native way to identify the source or destination of a webhook.

This means that a webhook producer cannot verify that it is sending its webhook to the correct destination, and the webhook consumer cannot verify that it is receiving its webhook from the expected source.

This vulnerability allows anyone to act as a webhook producer or receiver, and potentially send malicious webhooks to a webhook consumer, thereby compromising the receiving application.

Webhooks Security

Security On Setup or During Runtime

One time verification

Verification Token (Shared Secret)

Hash Based Message Authentication Code

A code snippet

```
request.Header.Add("Content-Type", "application/json") request.Header.Add("Conf42-Signature",
getSignature) request.Header.Add("User-Agent", "UserAgent-http-client/1.1")
```

Source: css-tricks.com

IP whitelisting

Mutual TLS (Transport Layer Security)

What Is The Best Approach

?

Hashed Based(timestamp) With Dataless Notification

Webhooks Scalability

Optimize your webhook payload

Implement load balancing

Use a message broker

Implement caching

Monitor performance

Parting shots

WORDS FROM A WISE MAN

</>

thank you, asante

@MARVIN_HOSEA