Geek University

Evolua seu lado geek!

www.geekuniversity.com.br

Entendendo sobre a segurança de APIs

Entendendo sobre a segurança de APIs

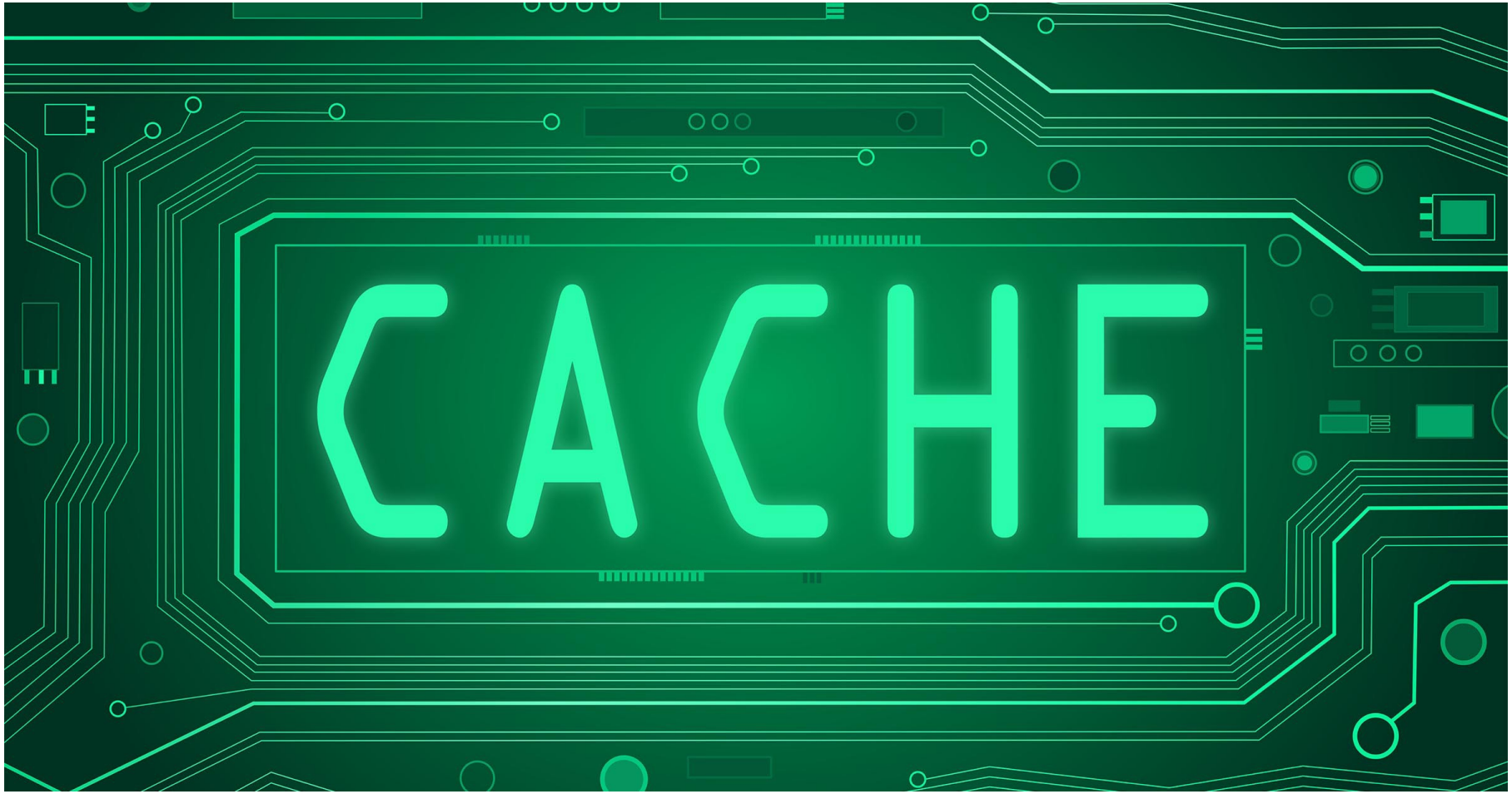
Uma **API** que não consegue suprir a demanda é tão ruim quanto não ter nenhuma **API**.

Os usuários não irão ficar aguardando até que os servidores coloquem sua **API** de volta no ar. Se eles tiverem opções eles irão atrás destas opções.

E você não quer. Você quer conquistar mais e mais clientes até dominar o mundo.

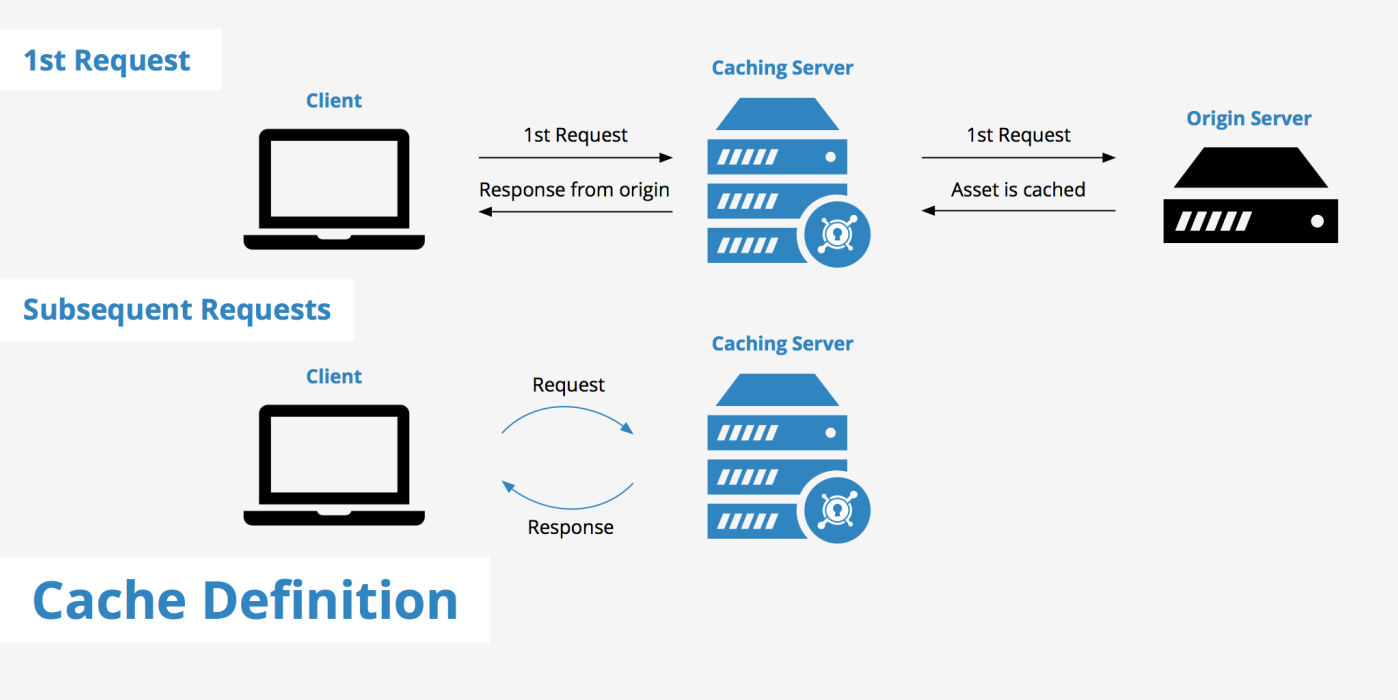
Entendendo sobre a segurança de APIs

O primeiro passo para manter sua **API** disponível para os clientes é fazendo uso de **cache**.



Entendendo sobre a segurança de APIs

Fazendo uso de cache, por exemplo um cliente pode fazer uma requisição e os dados vem direto do servidor que processou e enviou a resposta. Um segundo cliente faz uma nova requisição mas agora os dados podem ser pegos direto do cache. Mais rápido e eficiente.



Ferramentas como o **Redis** ou o Memcache podem ser utilizados para isso.

Entendendo sobre a segurança de APIs

Nem todo o cache do mundo irá te salvar se sua **API** for inundada de requisições.

Imagine que você tenha um servidor com capacidade de 1000 requisições por segundo.

Para a grande maioria das aplicações este número é mais que suficiente. Mas lembre-se, sua intenção é **dominar o mundo!**

Então numa noite de sexta-feira já quase hora do fim do expediente, o servidor recebe 1001 requisições no mesmo segundo e cai.

Isso não deveria acontecer.

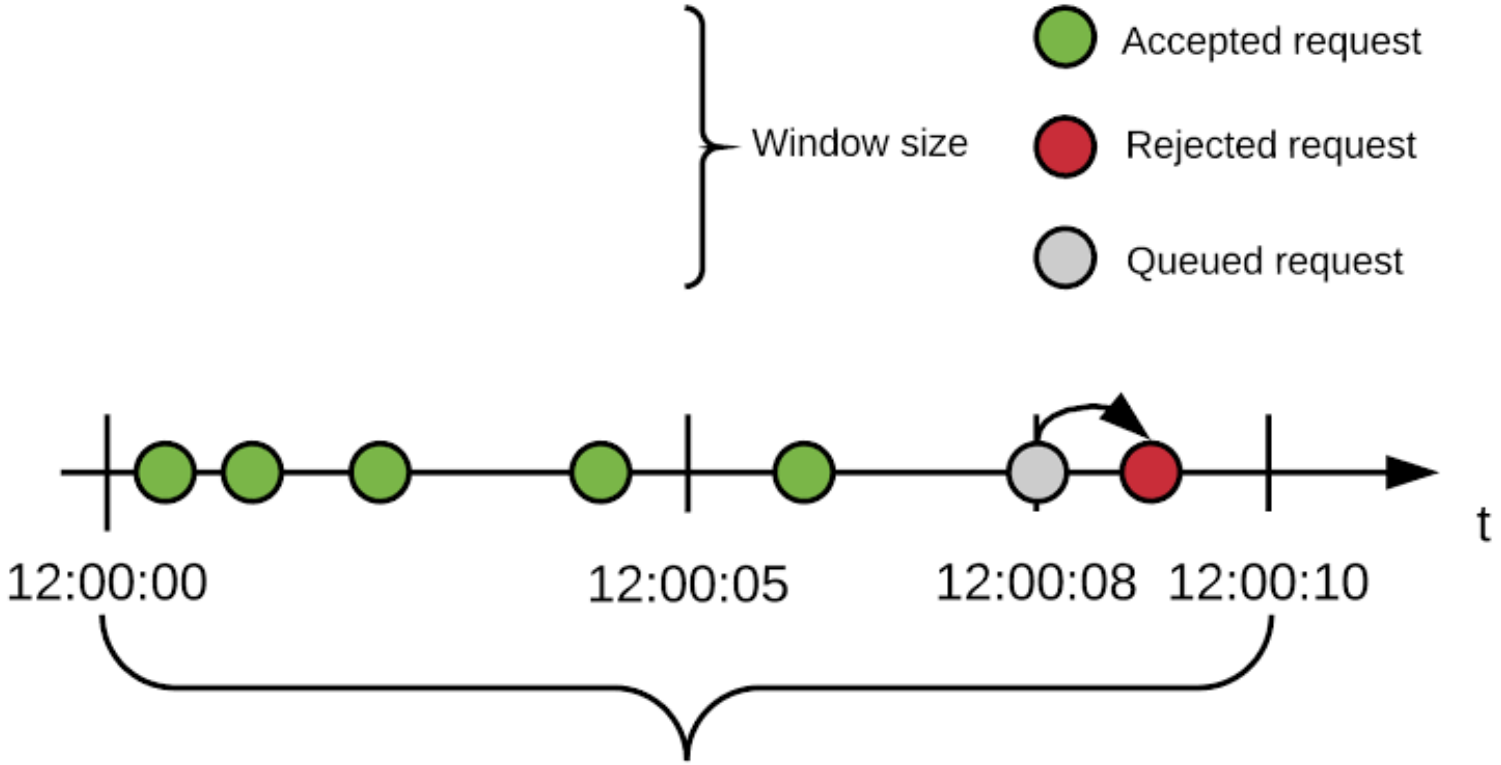
O seu servidor não poderia estar recebendo mais requisições que o suportado.

Você como desenvolvedor pode/deve adicionar limite de requisições.

Isso pode ser feito inclusive na venda de serviços, muito comum atualmente, na qual um cliente contrata 50, 100, 500 ou 1000 requisições por mês para acessar seus serviços.

Entendendo sobre a segurança de APIs

Neste exemplo temos uma janela de 5 requisições a cada 10 segundos. Ou seja, o cliente só pode fazer 5 requisições no espaço de tempo de 10 segundos. Neste exemplo ainda o cliente tentou fazer uma sexta requisição que foi rejeitada.



Entendendo sobre a segurança de APIs

Um último passo para entender tudo que envolve a segurança de **APIs** está a **autenticação** e **autorização**.

É muito difícil você conseguir limitar clientes a acessarem suas **APIs** se você não sabe quem são seus clientes ou quem está tentando fazer acesso à sua **API**.

Lembre-se, o **HTTP** é **Stateless** (não guarda estado)

Como os clientes irão conseguir uma conta de acesso cada a você de acordo com as regras do seu sistema.

Mas existem algumas formas de manusear a autenticação de clientes em **APIs REST**.

Entendendo sobre a segurança de APIs

A forma mais comum de autenticação de clientes em **APIs REST** é fazendo uso de **Tokens**

De forma geral, o token é uma chave criptográfica que identifica o cliente. Ou seja, quando o cliente cria uma conta na sua aplicação ele recebe uma chave (**Public Key**) e através desta chave (**token**) ele envia no cabeçalho ou no corpo da requisição para realizar a **autenticação**.

Como as requisições **HTTP** são **Stateless** este token de autenticação sempre é enviado.

Entendendo sobre a segurança de APIs

Por fim deve ficar claro as diferenças entre **Autenticação** e **Autorização**.

Authorization

What you can do

Authentication

Who you are

Geek University

Evolua seu lado geek!

www.geekuniversity.com.br