Functional Requirements
allowRequest(request):boolean

NonFunctional Requirements
Low Latency (make decision as soon as possible)
Accurate (as we can get)
Scalable (Supports arbitrary large number of hosts in the cluster)
-HIgh Availability (not necessary, do not throttle if do not know)

Design

Data Structure & Algorithms

- Token Bucket Algorithm
- Leaky bucket Algorithm
- Sliding Window Algorithm
- Fixed Window Algorithm

Protocols

- Gossip
- Raft

Message Broadcasting (RateLimiters talk to each other to pass currently utilized tokens)

1) Full Mesh
2) Gossip Protocol
3) Using Distributed Cache (**Redis**)
4) Coordination Service (Leader Election: Paxos, Raft) - High Maintenance, Sophisticated

Communication Protocol

1) TCP (Accurate)
2) UDP (Faster)

Hosting

1) Library
    a) Faster
    b) Resillient to interprocess call failures
2) **Process**
    a) Programming Language Agnostic
    b) Uses it’s own memory space
    c) Easier to deal with service teams paranoia