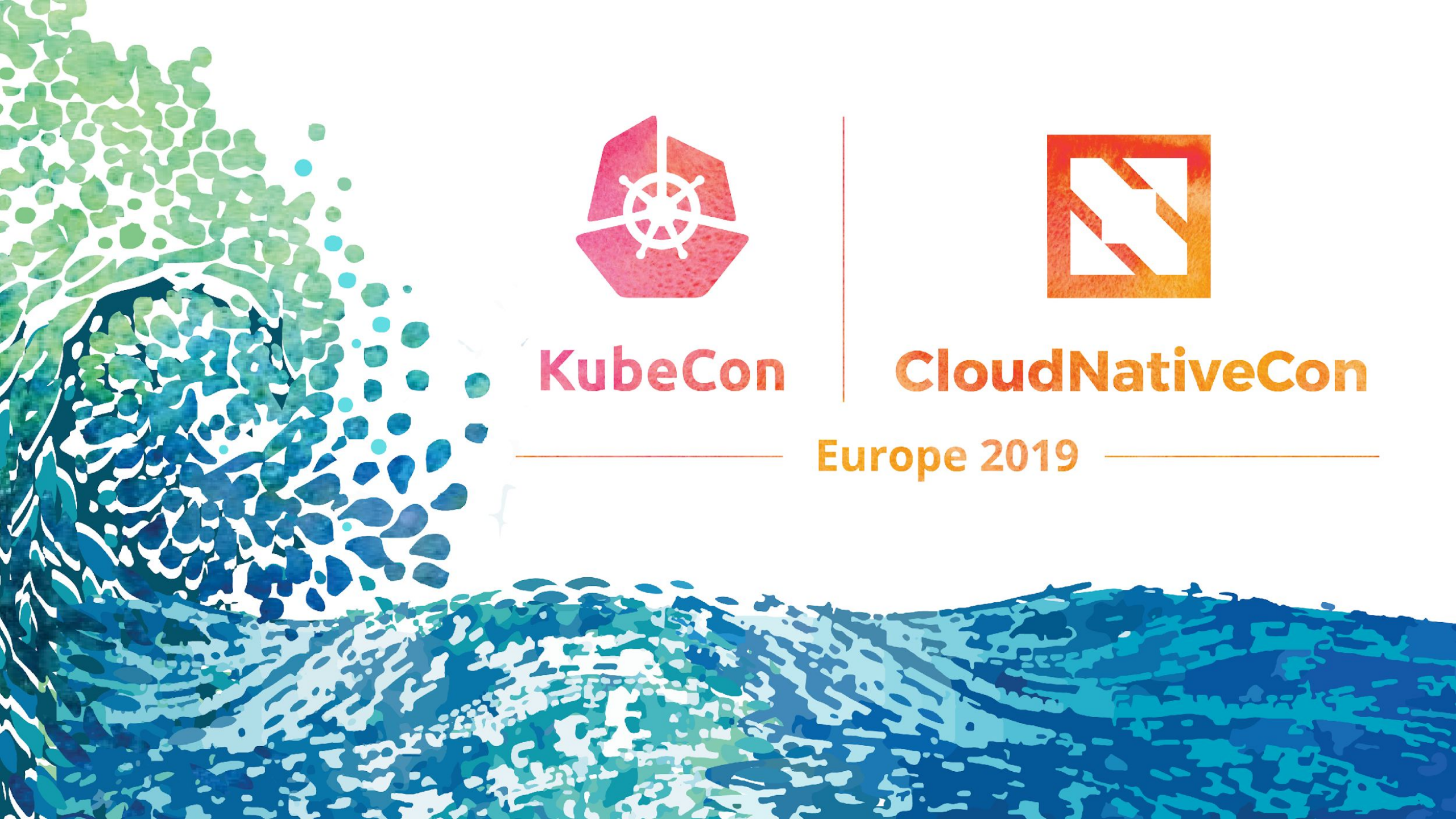
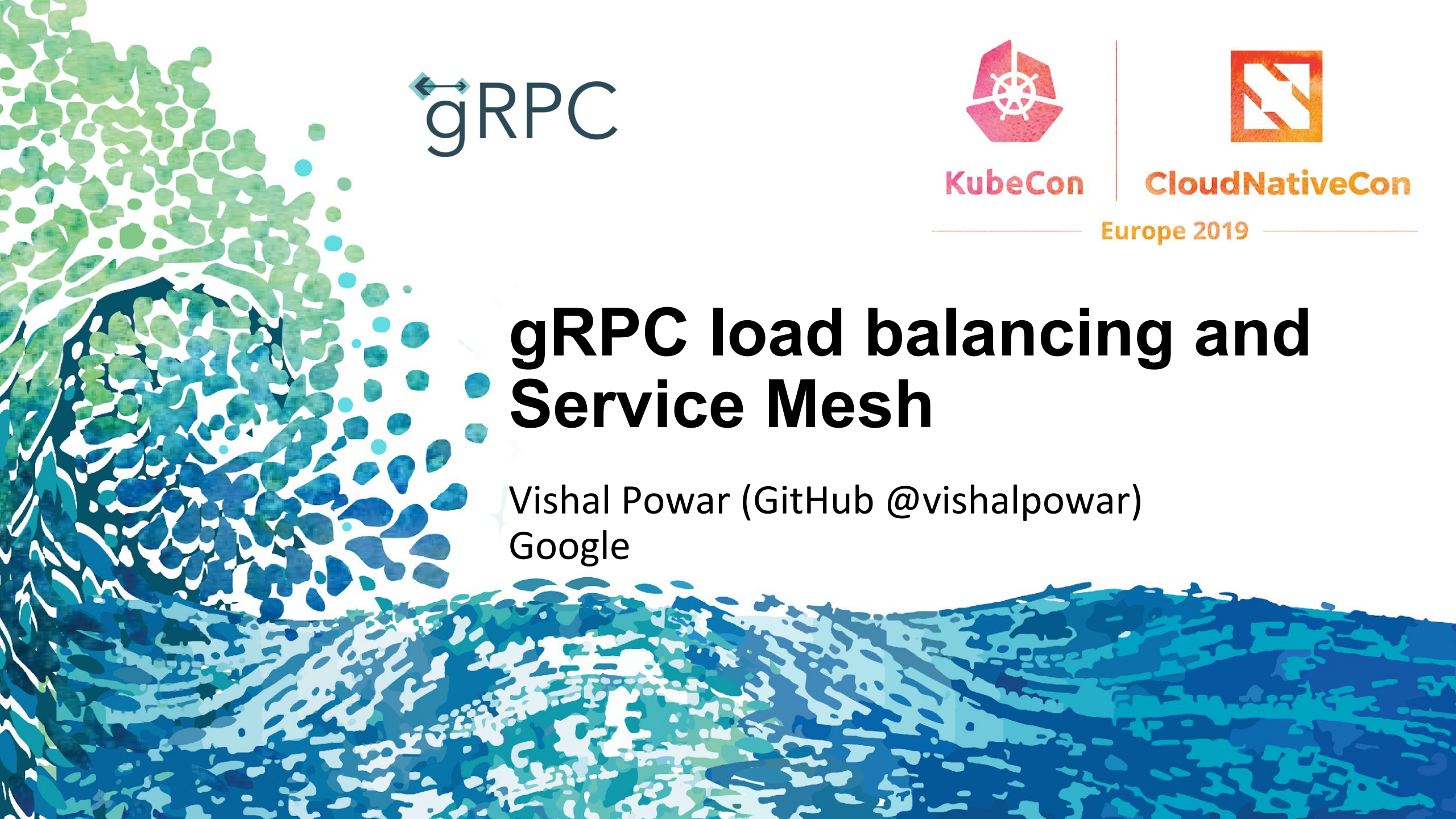

KubeCon
CloudNativeCon
Europe 2019



KubeCon | CloudNativeCon Europe 2019

gRPC load balancing and Service Mesh

Vishal Powar (GitHub @vishalpowar)
Google

Agenda

- Load balancing in gRPC
- Centralized balancing
- Load balancing at scale
- Service mesh
- Demo

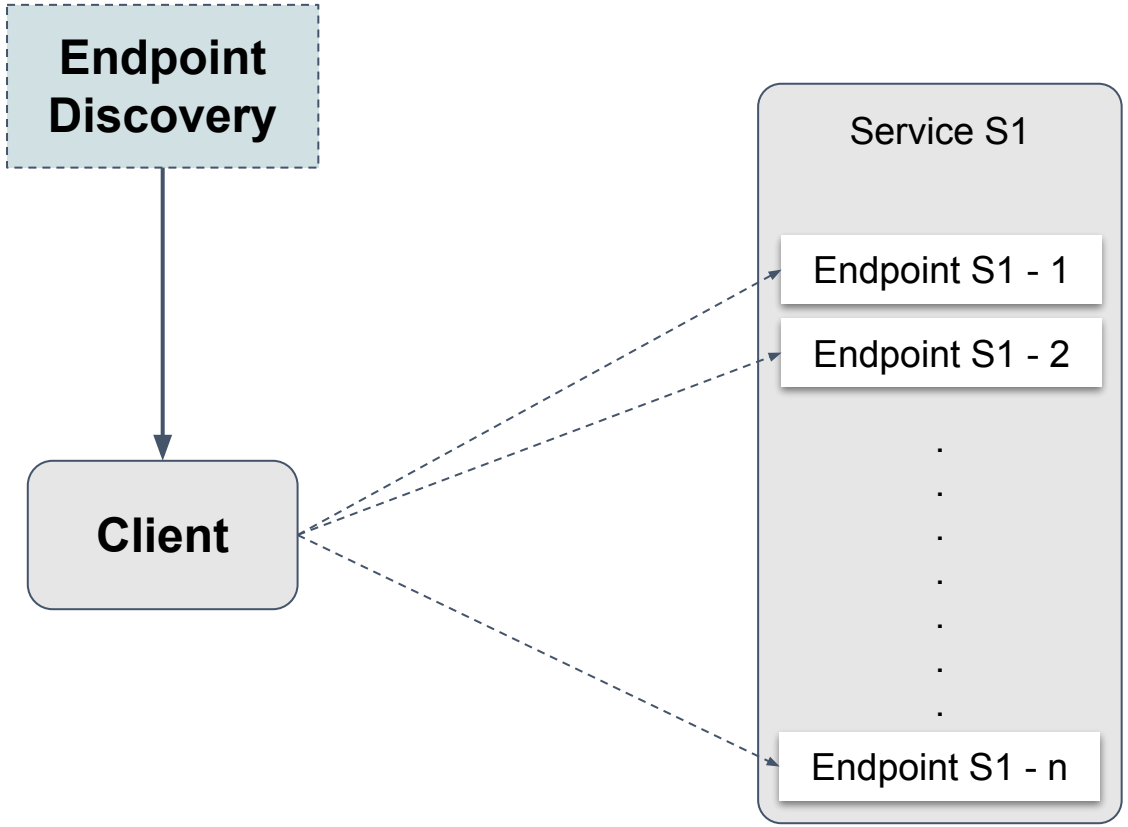
Why load balancing?

- Load balancing is a mechanism to
  - Improve throughput of a service.
  - Improve service availability and reliability.

- Load balancing should help client pick service endpoints
  - Based on client requirements (latency, #connections).
  - Based on endpoint requirements (isolation, #connections).

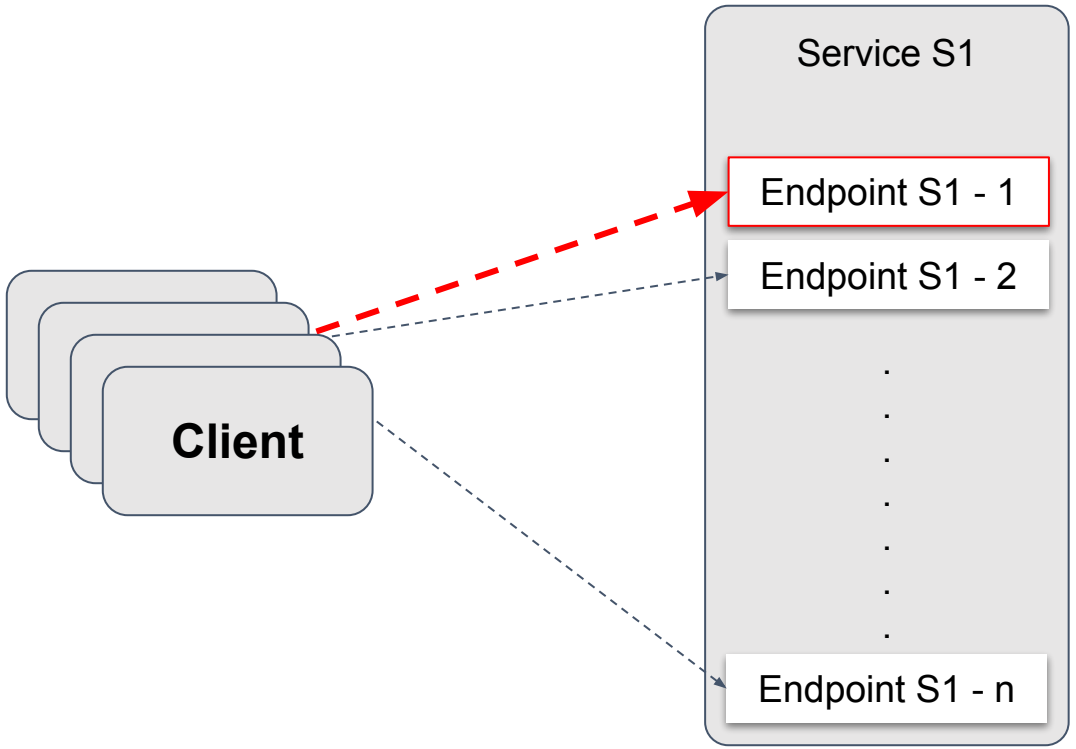
Client-side load balancing (gRPC)

- Client decides which endpoints to connect and send request
  - e.g
    - Round Robin
    - Pick-First
- Client can also connect to subset of endpoints.

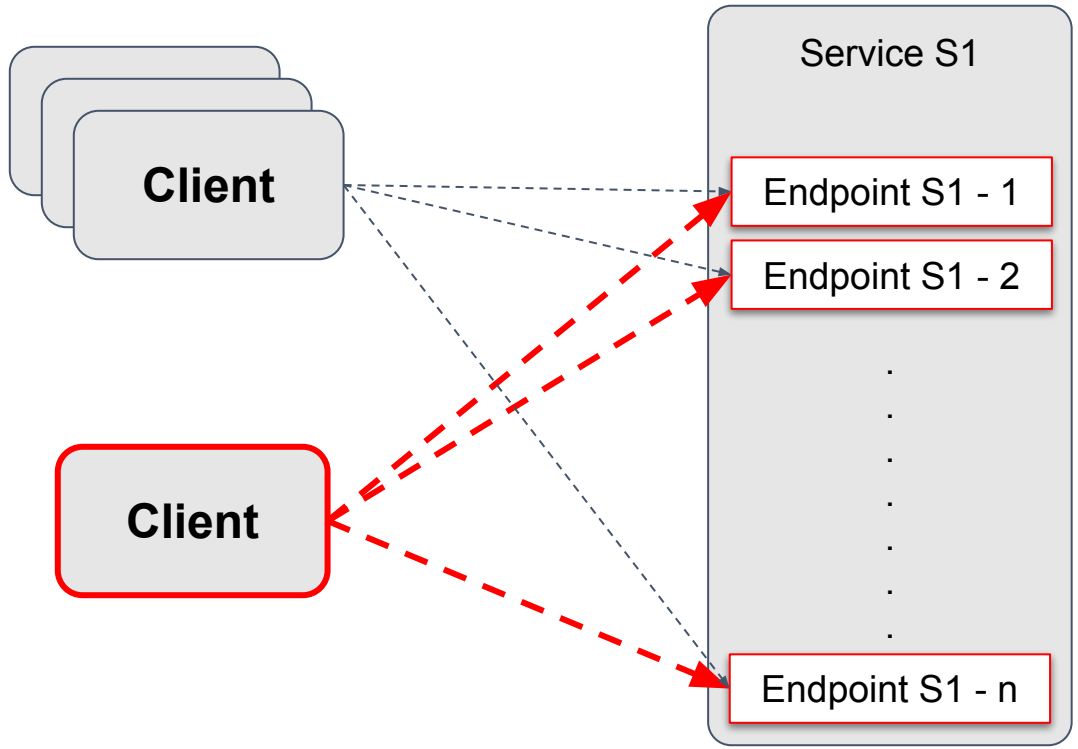


Client-side load balancing (gRPC)

- Pick first : Server overload

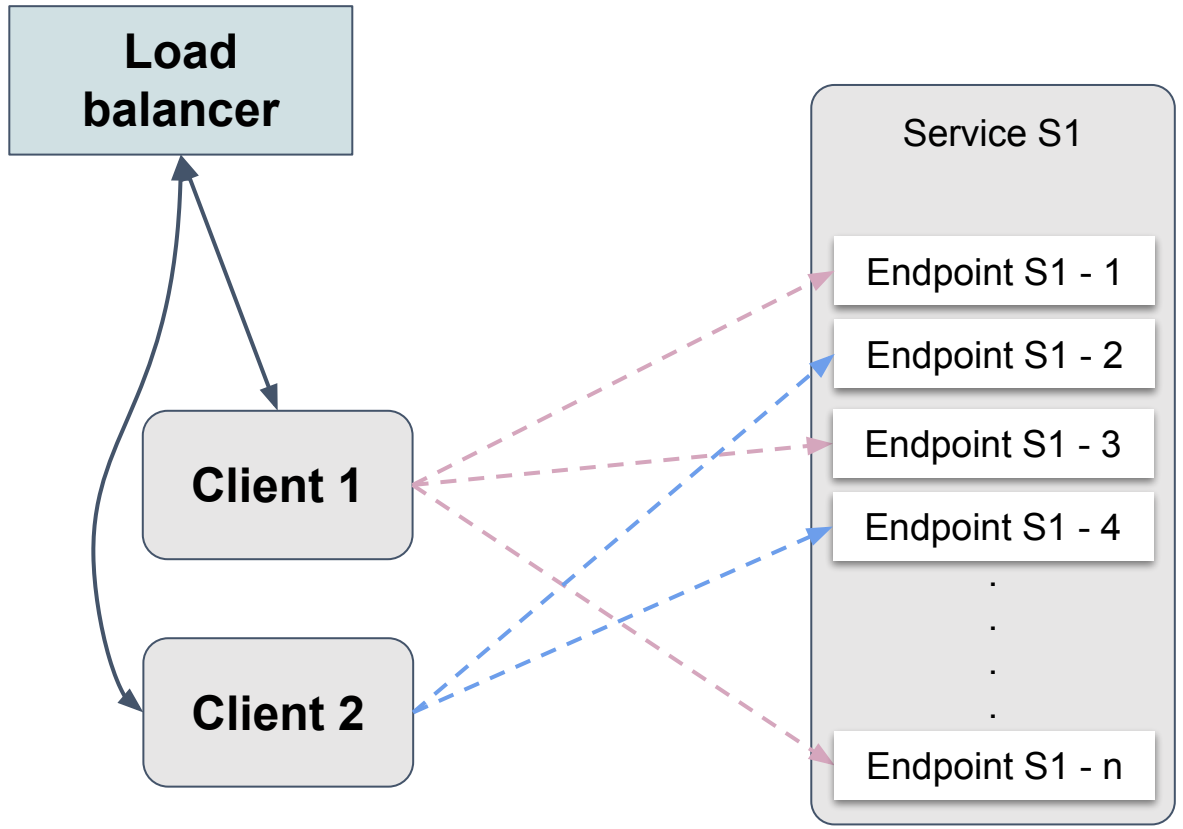


- Round robin : No isolation



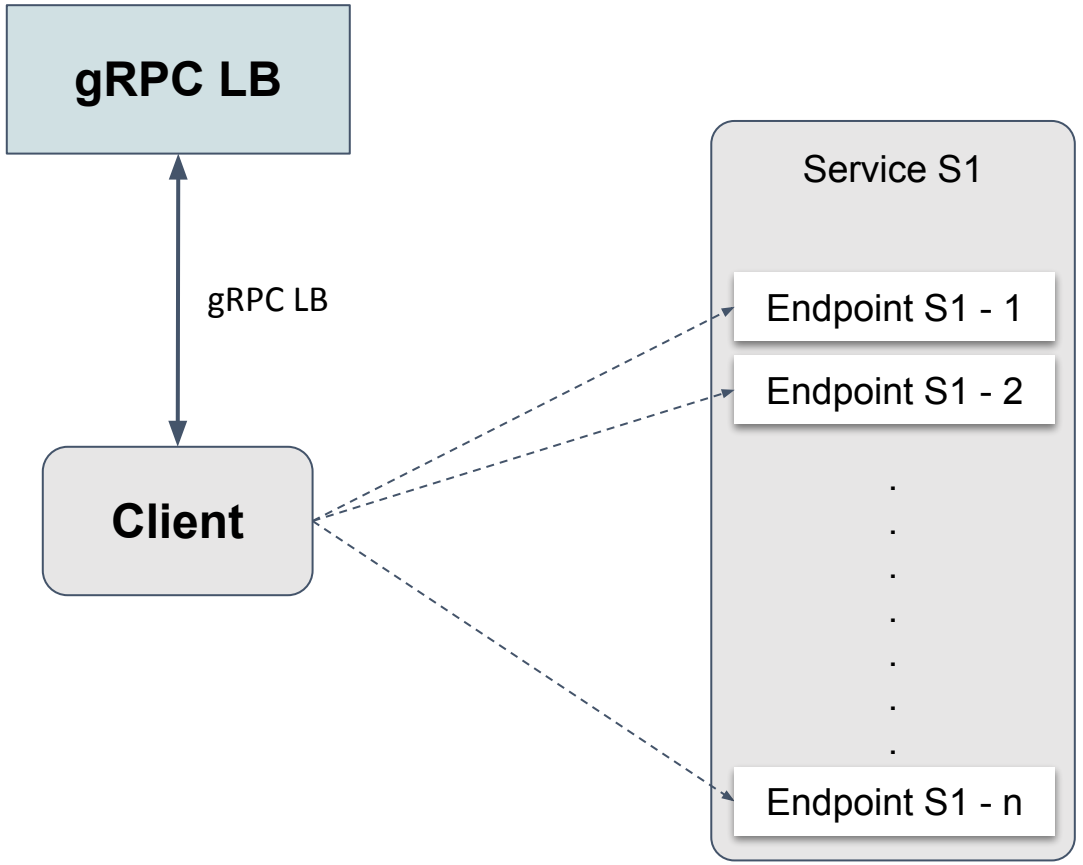
Centralized load balancing

- Take global decision to protect endpoints from client's local decision.



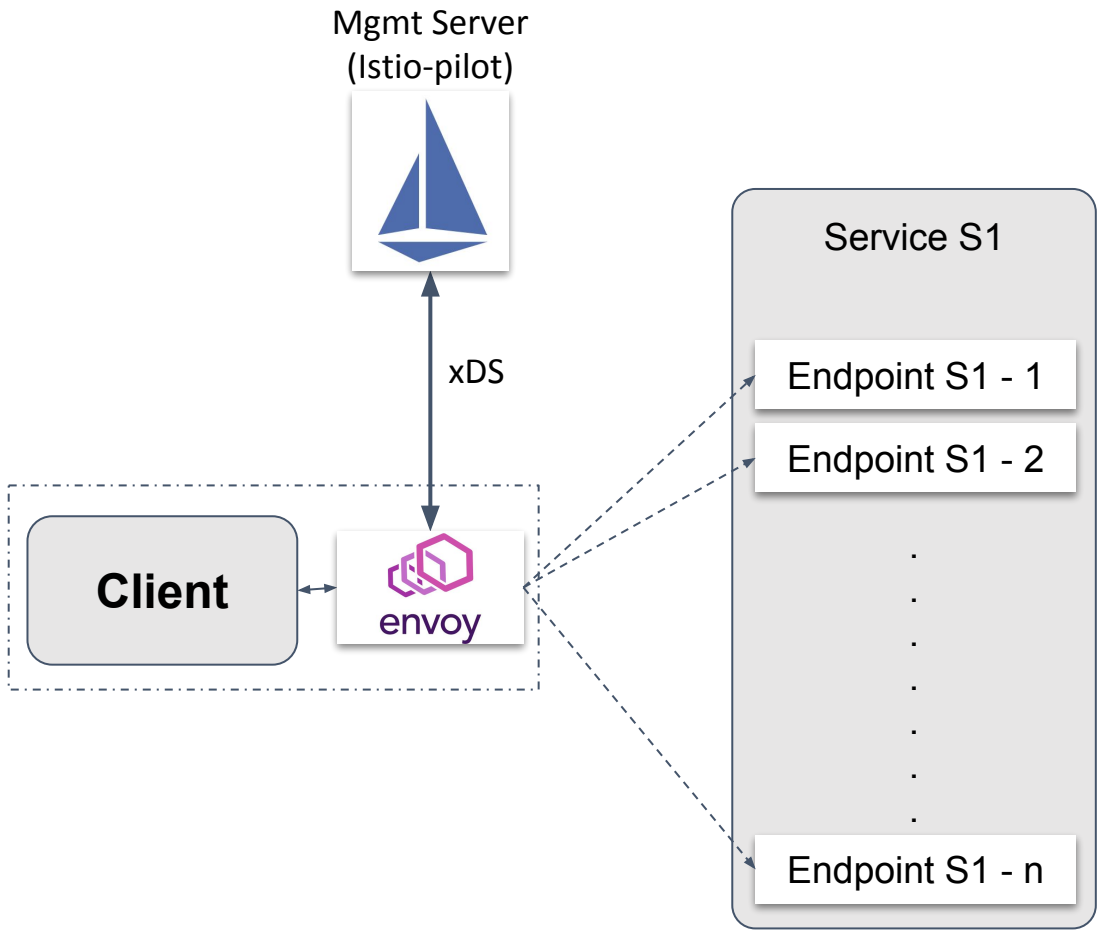
gRPC LB

- Look-aside load balancing with gRPC LB protocol.
- Balancer provides list of endpoints to use.
  - The list of endpoints encodes weight information.



Proxy load balancing (envoy)

- Middle proxy or sidecar deployment.
- xDS server provides endpoints and load balancing configuration for envoy.

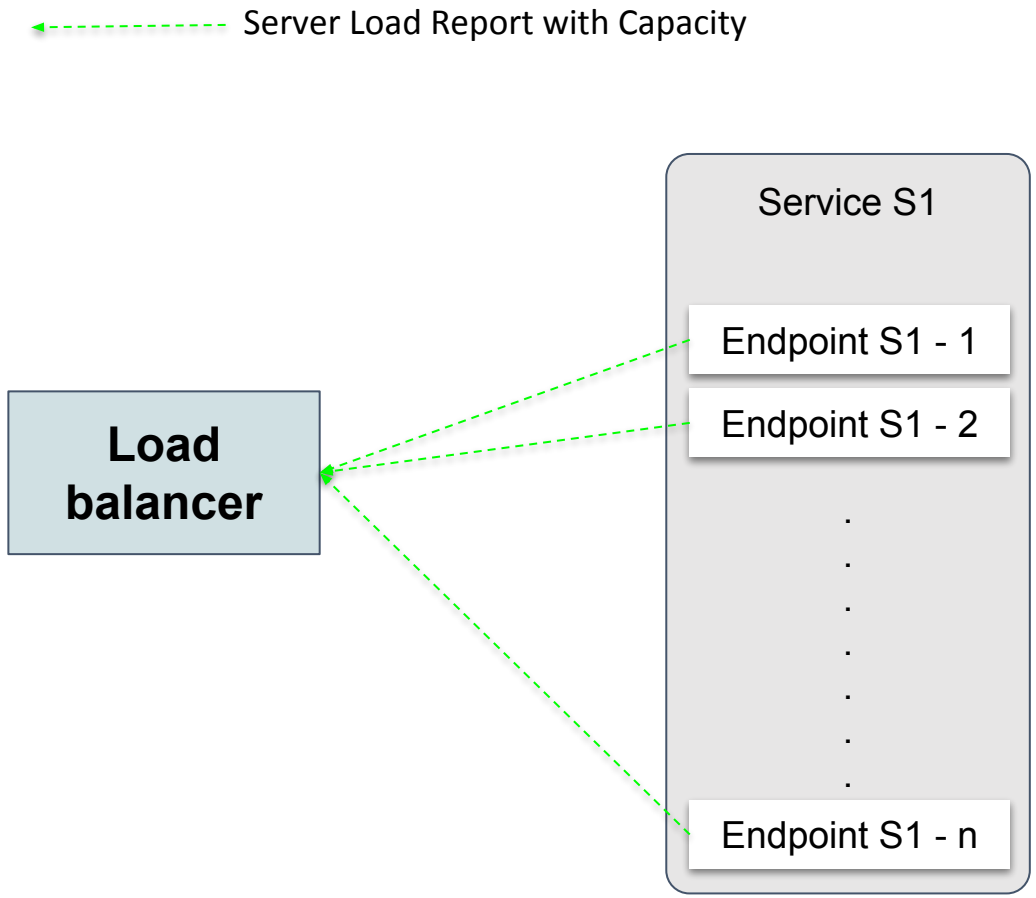


Deployment considerations

- Made up of heterogeneous endpoints.
  - e.g. capacity, location
- Individual endpoint’s health and capacity can change anytime.
  - e.g. service upgrade, hardware failure.
- Central load balancer needs to know above information to make better decisions.

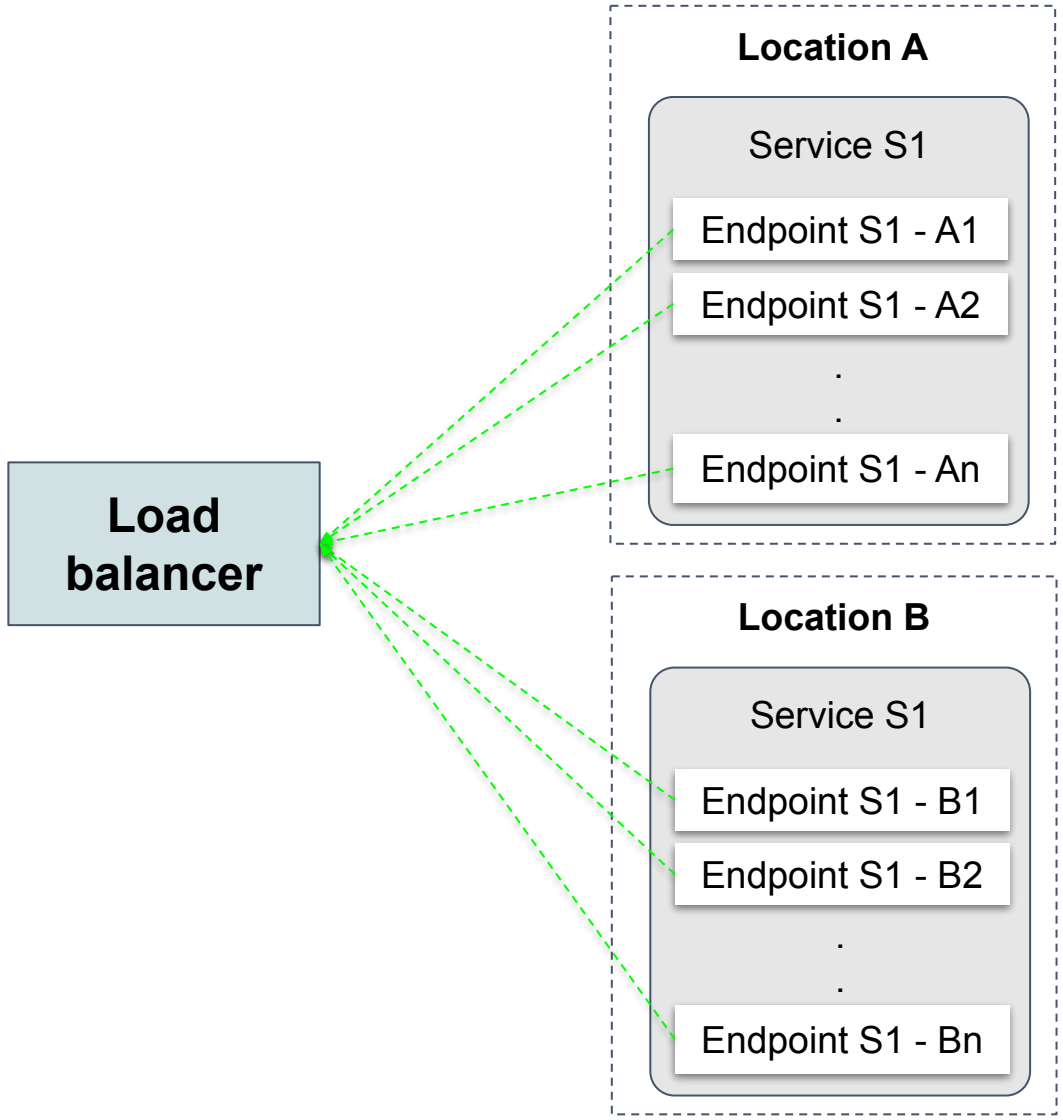
Informed balancing - Capacity

- Health can be determined by connecting to the endpoint.
- Capacity is service specific, and can be configured or reported by endpoints.
  - e.g compute, memory.



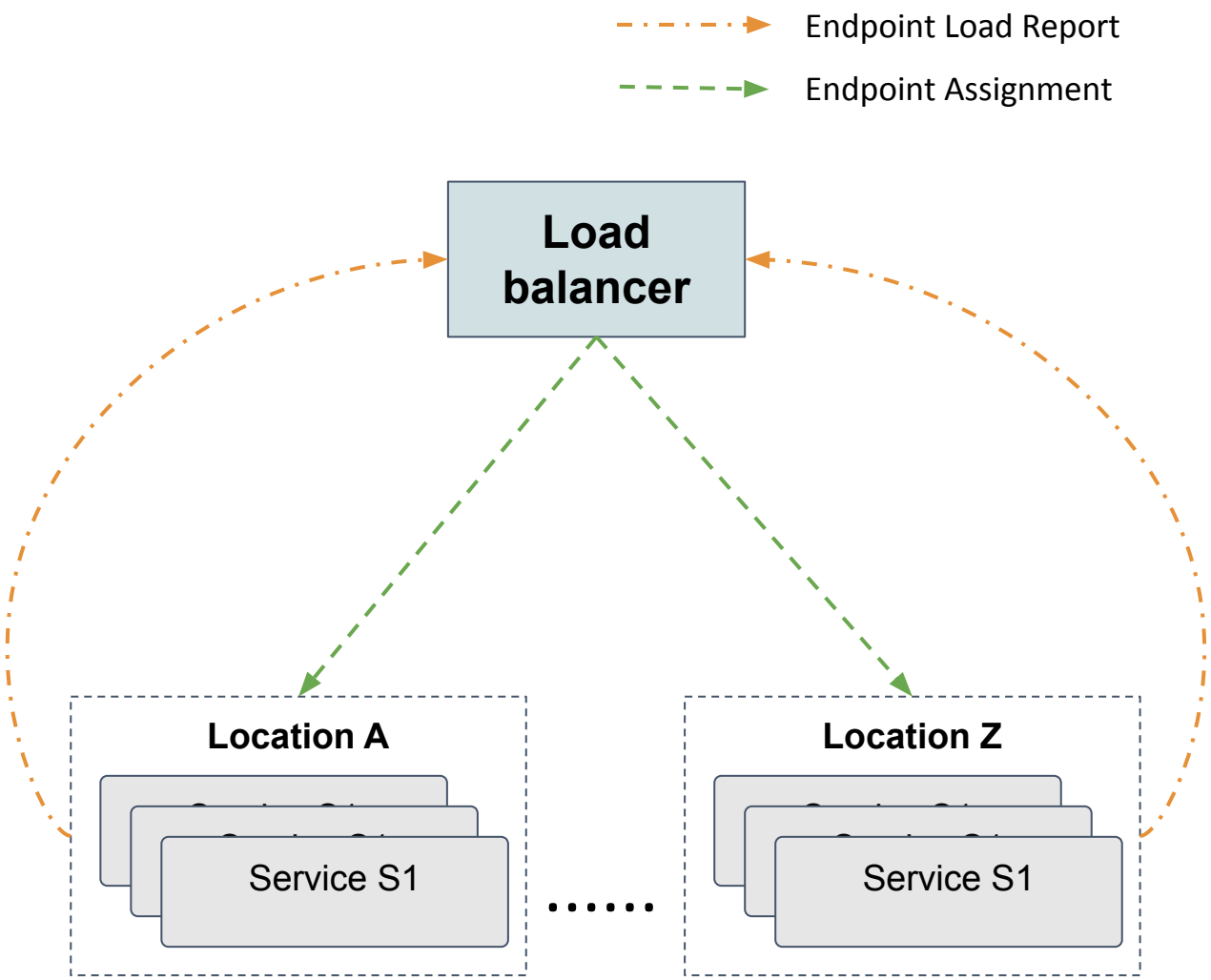
Informed balancing - Locality

- Routing requests closer to the client has advantages.
- Both endpoints and clients locality needs to be known by the load balancer.
- Also, “Locality capacity” can be used for balancing decisions.



Load balancing at scale

- Collecting information and making global decisions for each client is expensive.
- Change propagation to clients is slower (~ seconds).

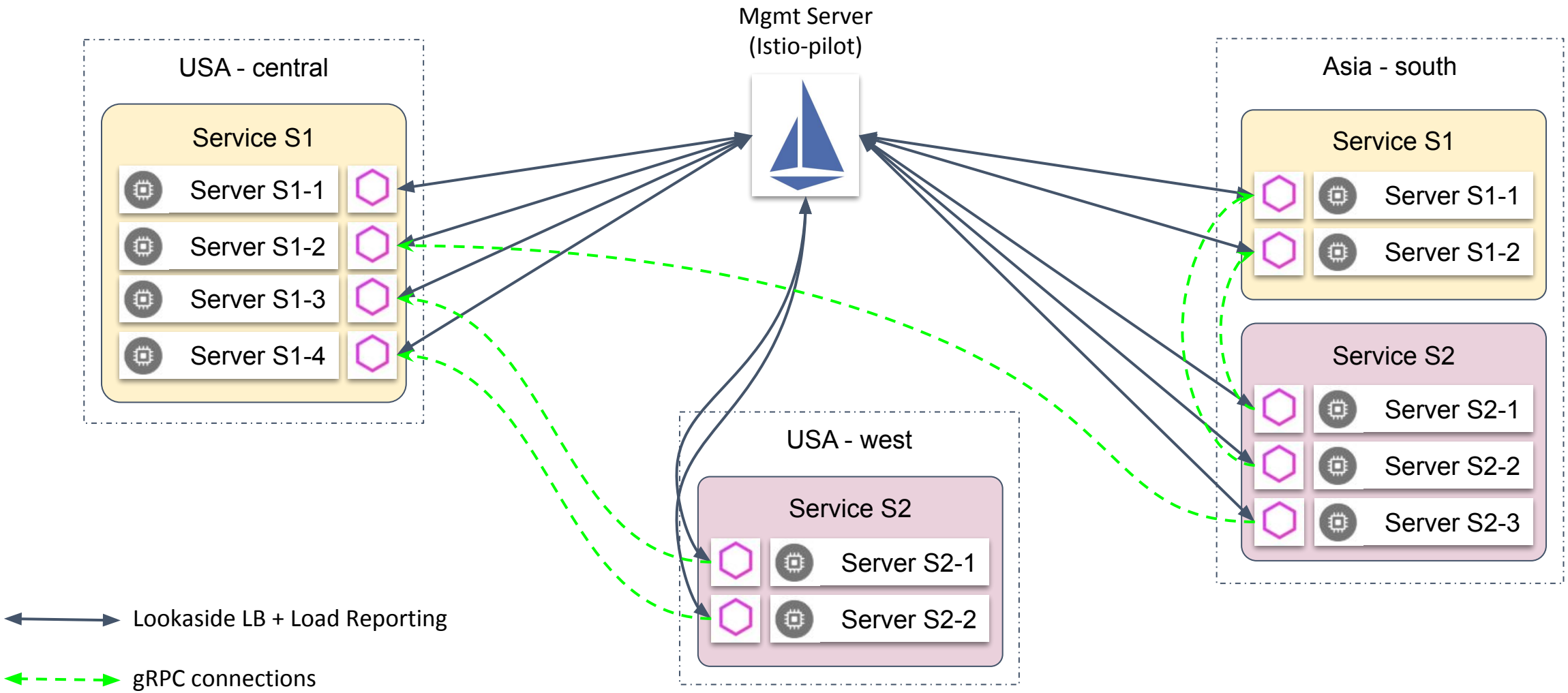


Load balancing at scale

- Take global decisions based on locality capacity and proximity.
  - consider client load on each locality.
  - provide enough information for clients to react quickly.

- Have clients take local decisions based on most recent information.

Service mesh and load balancing

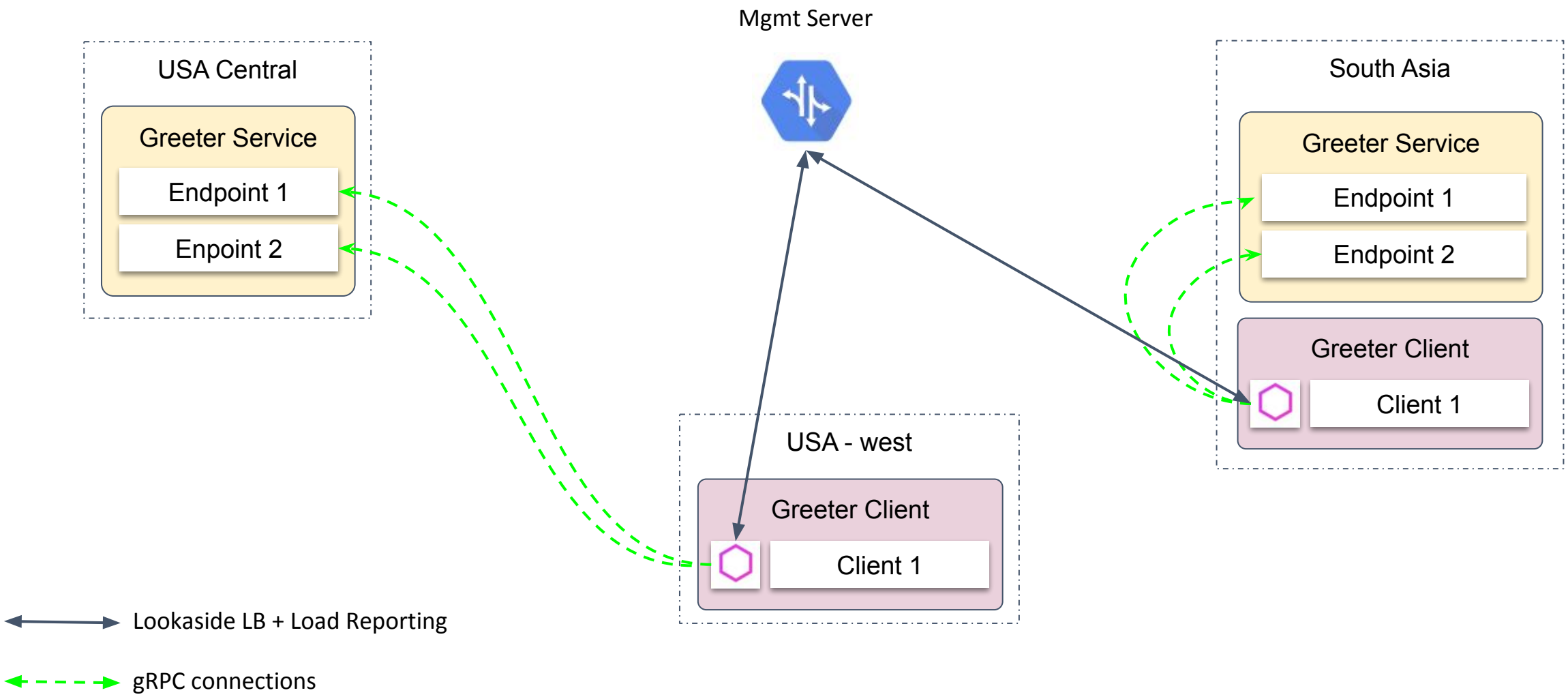


gRPC in service mesh

- xDS provides construct to achieve Load balancing flexibility.

- Information and controls are available for clients side balancing.
  - Weights at Locality and Endpoints
  - Proximity information at locality.

Demo


Mgmt Server
USA Central
Greeter Service
Endpoint 1
Enpoint 2
South Asia
Greeter Service
Endpoint 1
Endpoint 2
Greeter Client
Client 1
USA - west
Greeter Client
Client 1
Lookaside LB + Load Reporting
gRPC connections

Thank You

- Demo scripts and steps
  - https://github.com/vishalpowar/grpc-global-loadbalancing
- gRPC load balancing (kubecon-18)
  - https://github.com/jtattermusch/grpc-loadbalancing-kubernetes-examples