Consul

Consul overview

Microservices

Consul for Service Discovery

Terraform integration

Consul overview

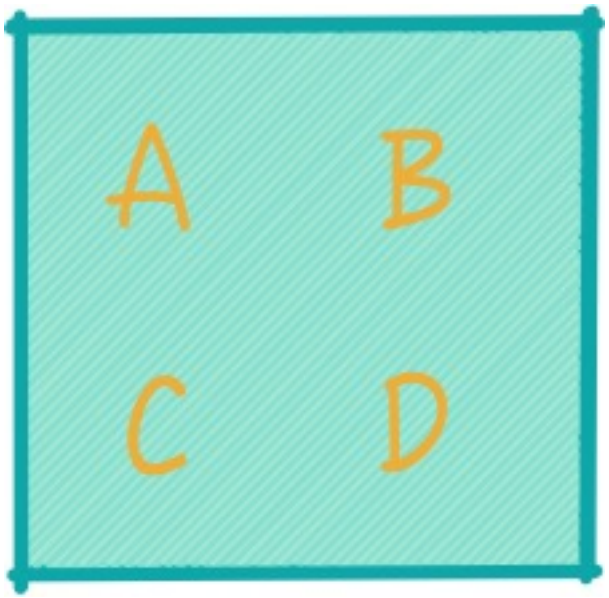
Consul overview

Microservices

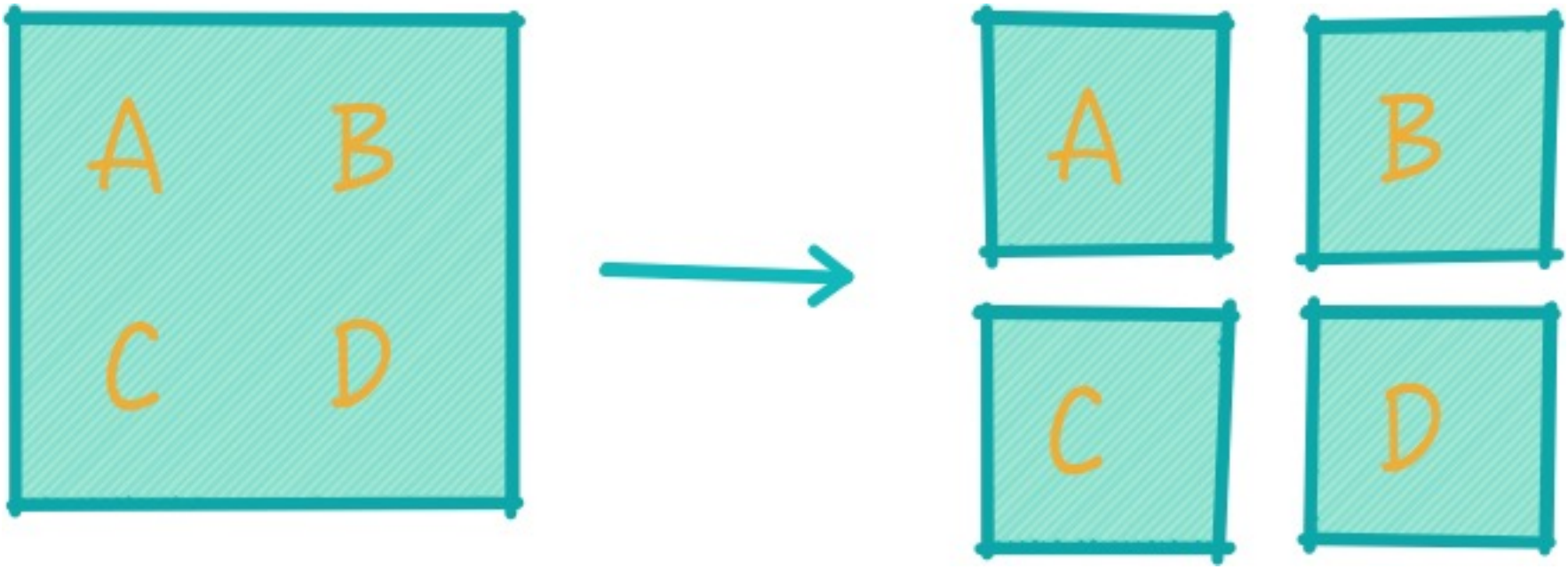
Consul for Service Discovery

Terraform integration

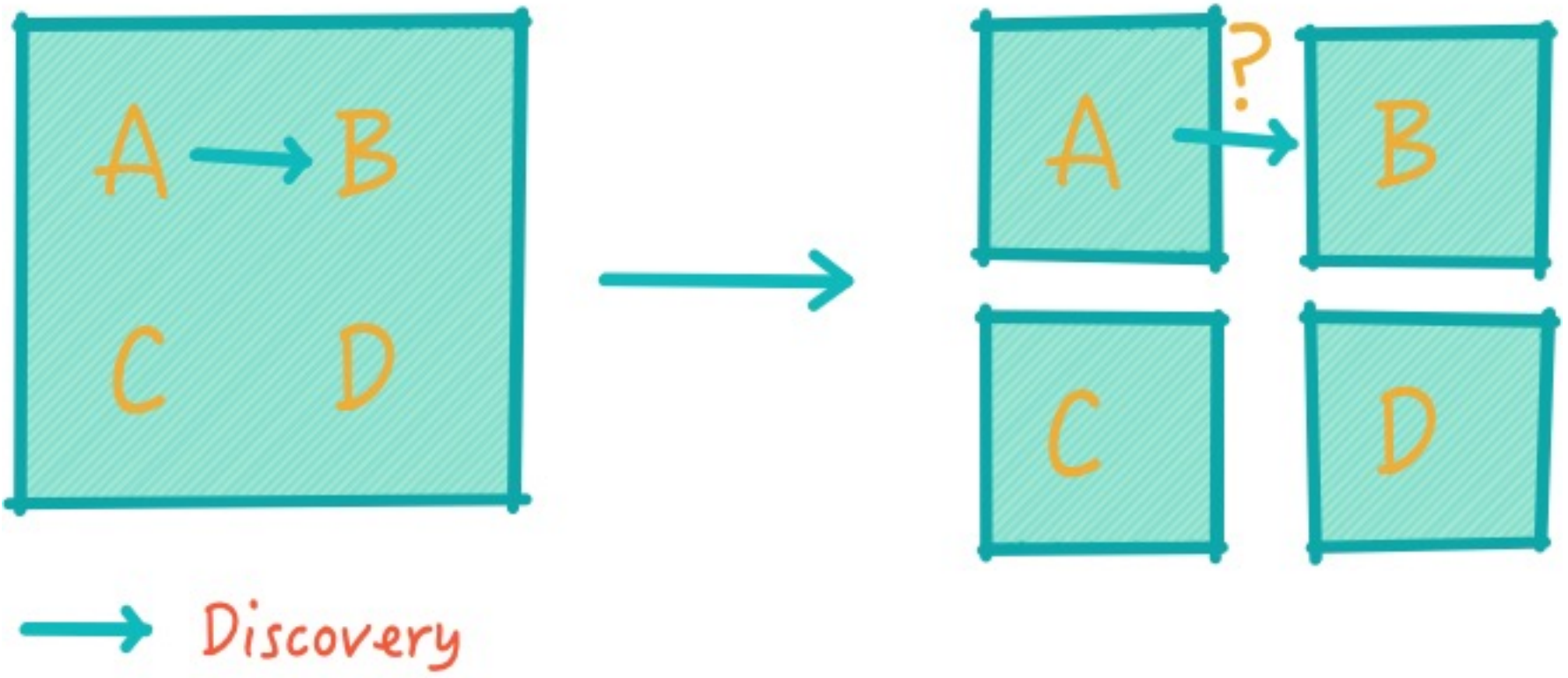
Monolith



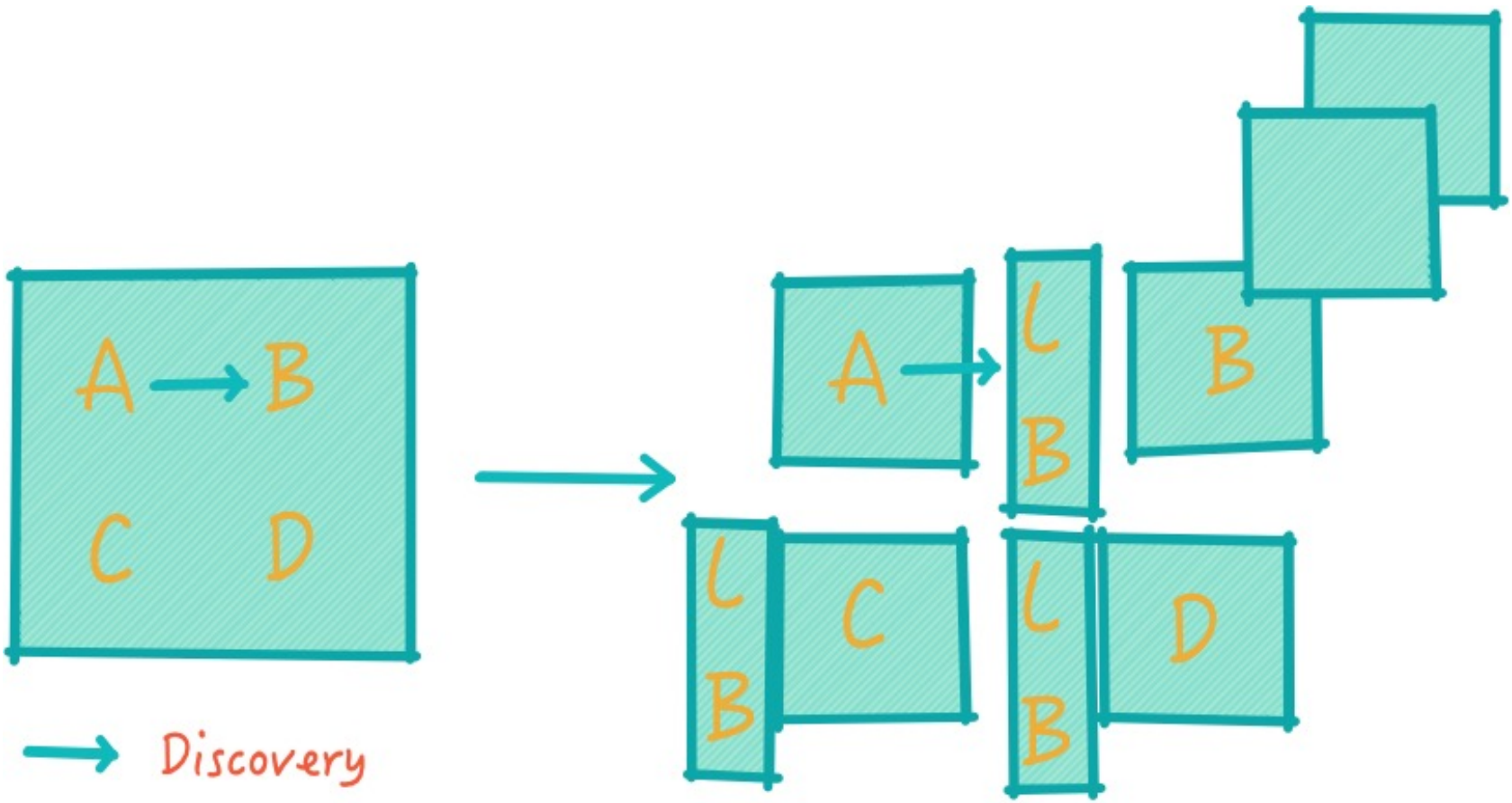
Microservices



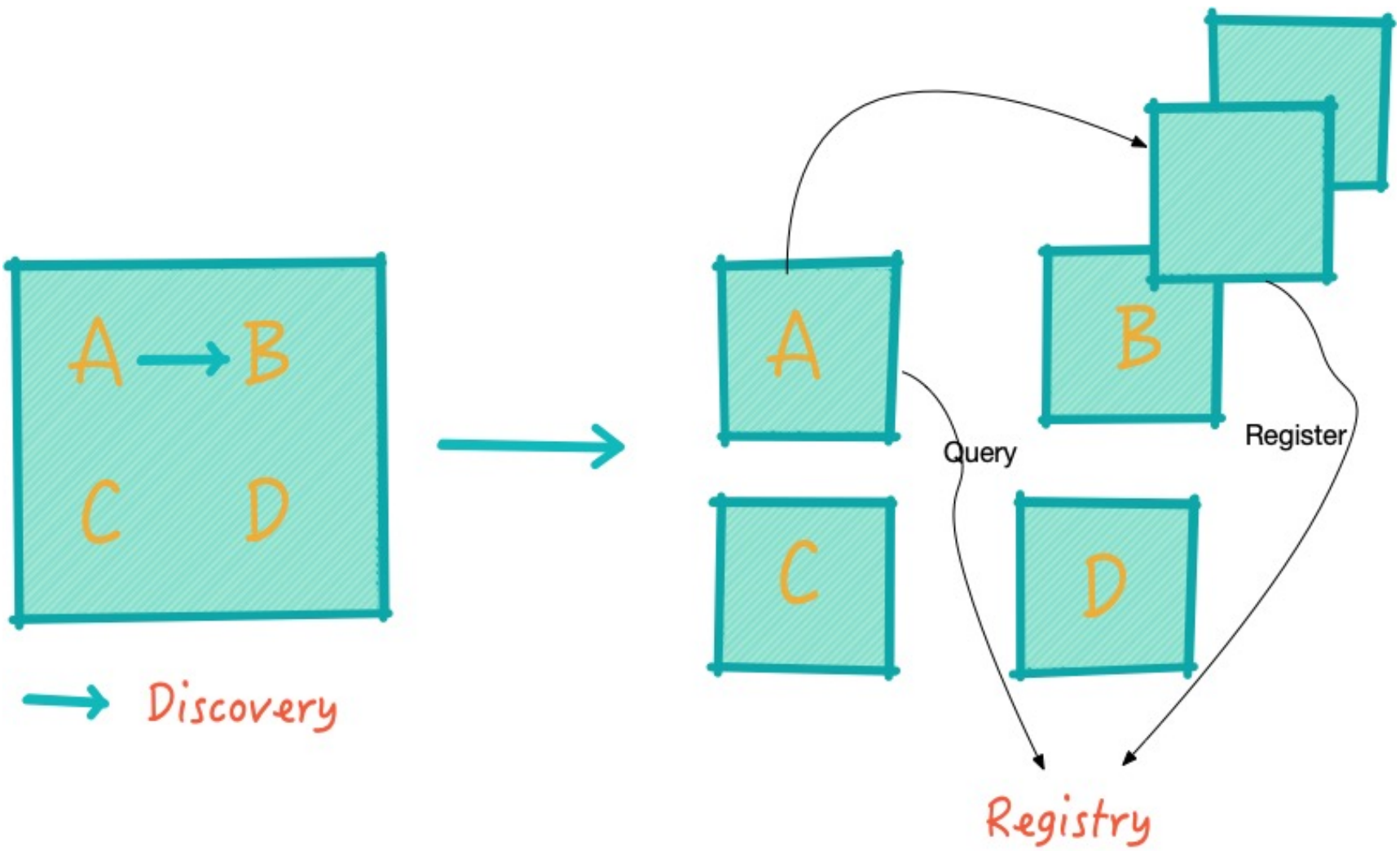
Discovery



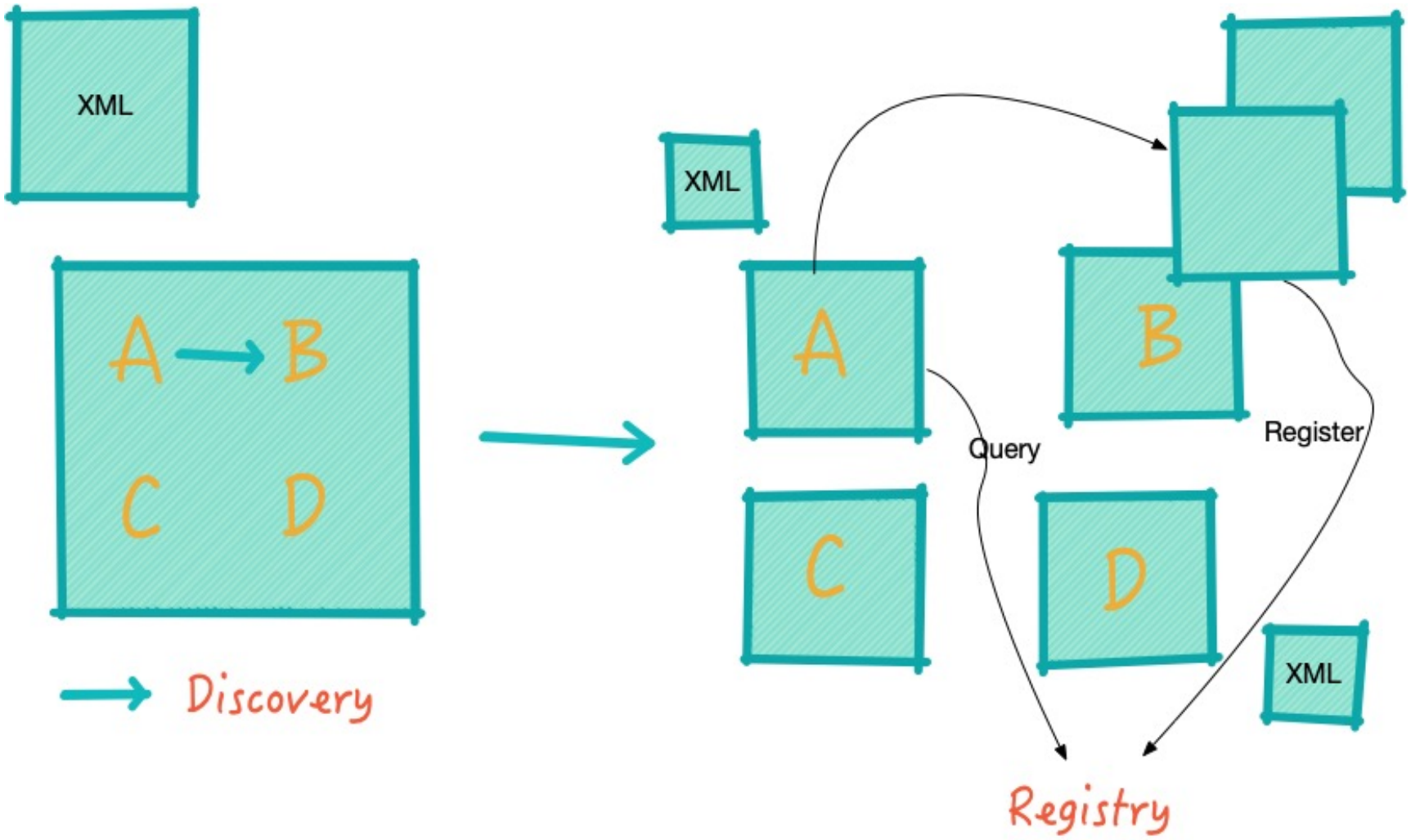
Load balancers?



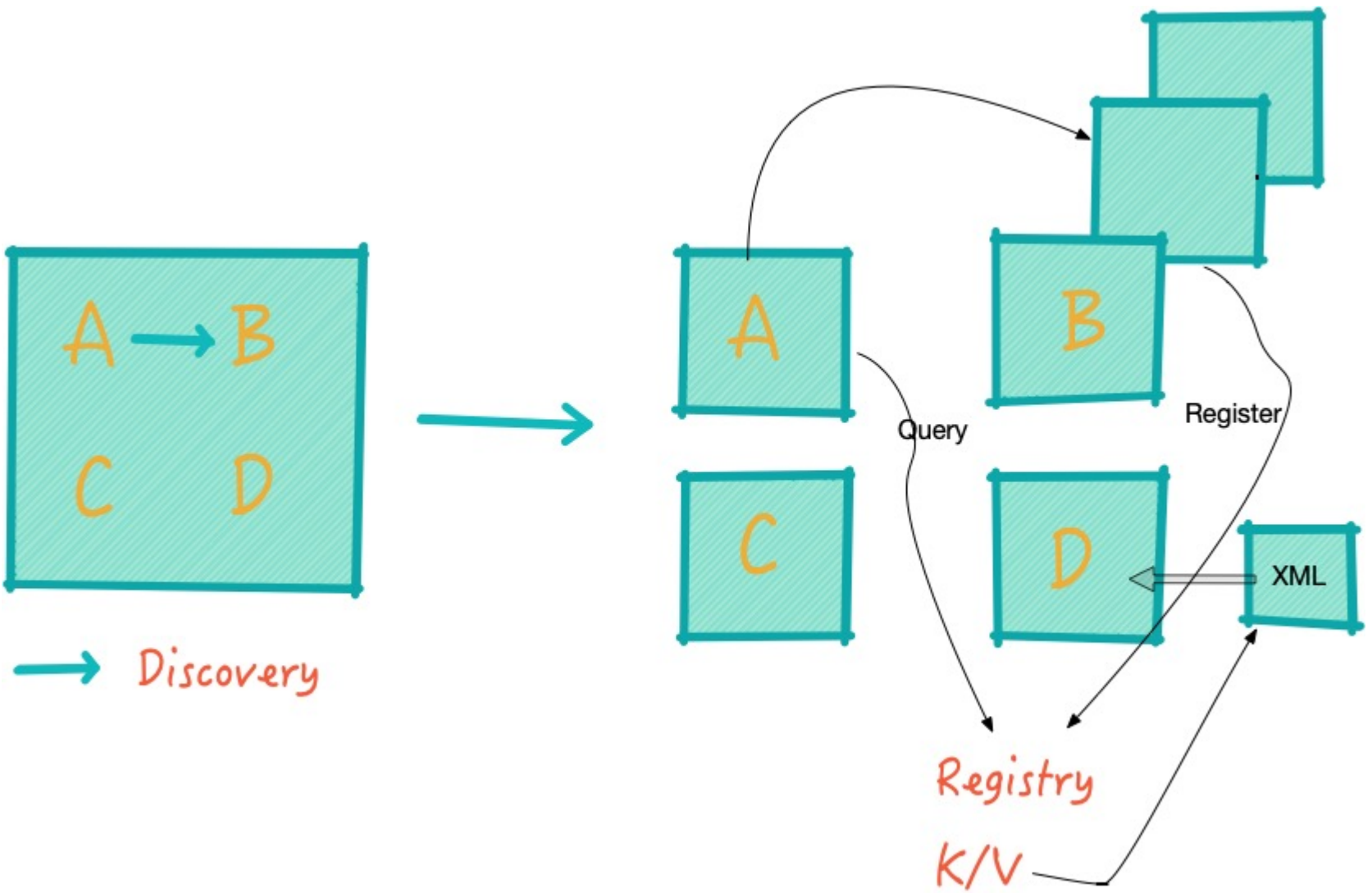
Registry



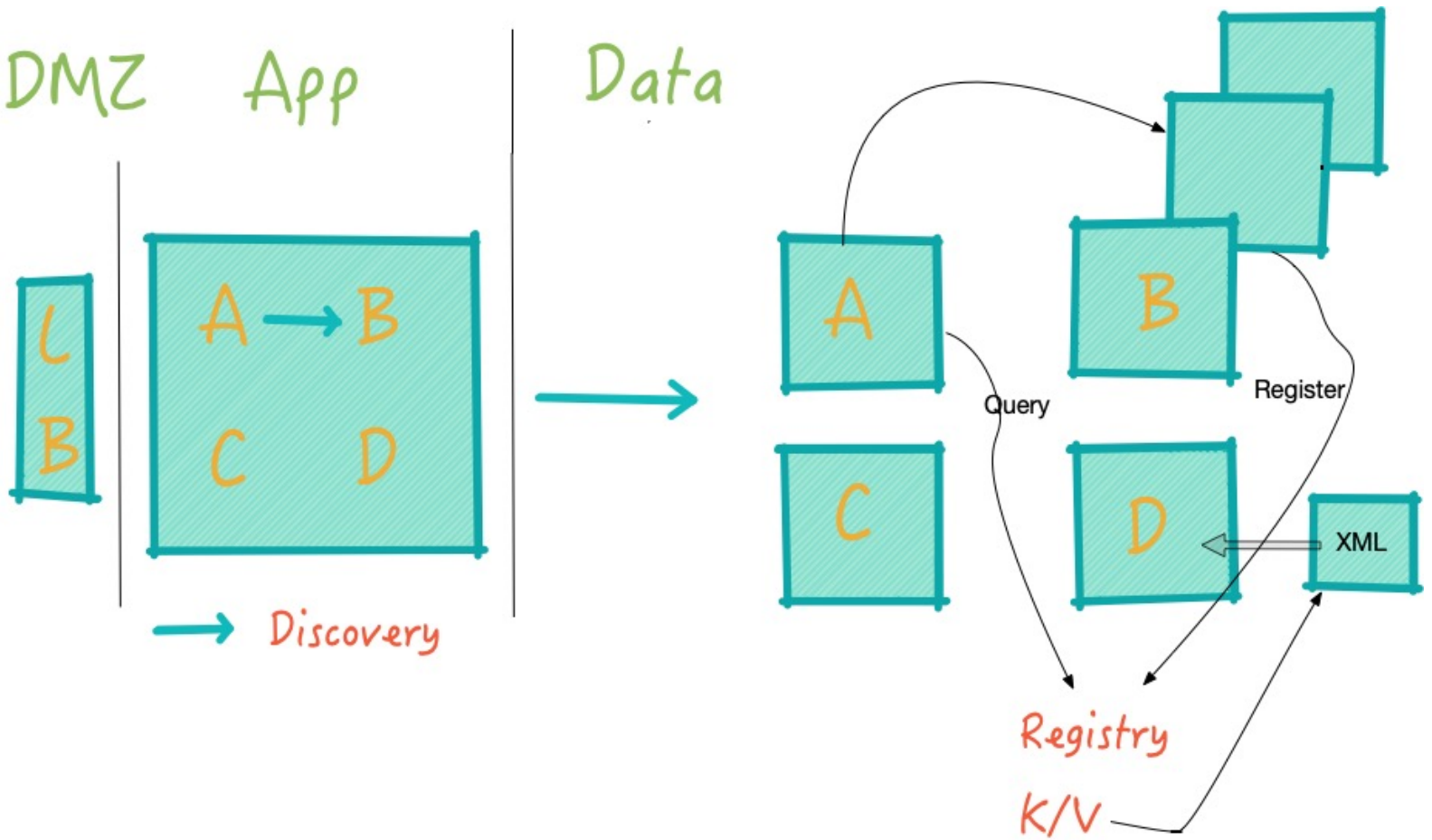
Configuration



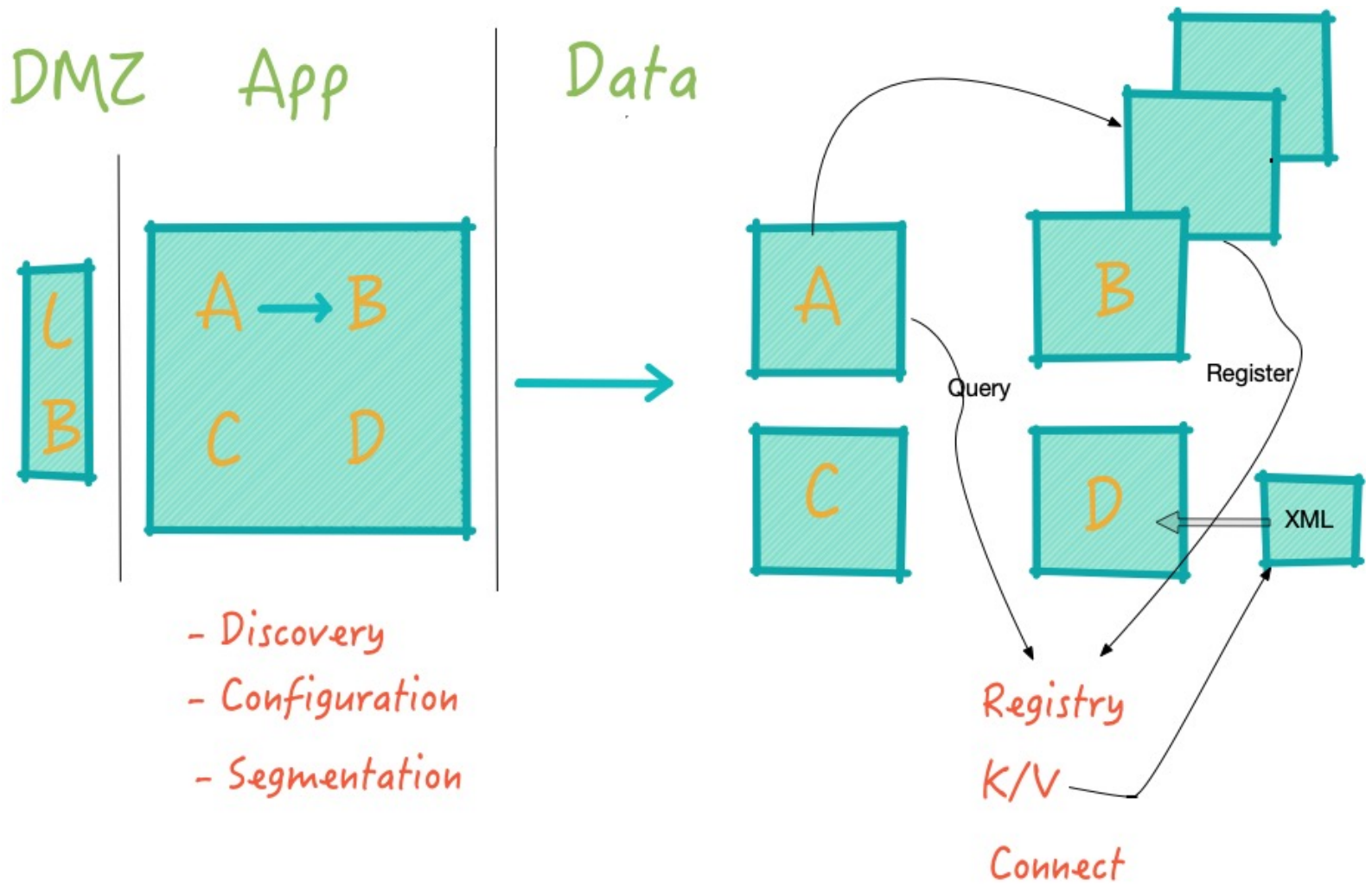
Configuration in Consul



Zones



Segmentation



Service graph

Service Graph

A->B

C->D

Web->DB

50*5???

Identity

Service Graph

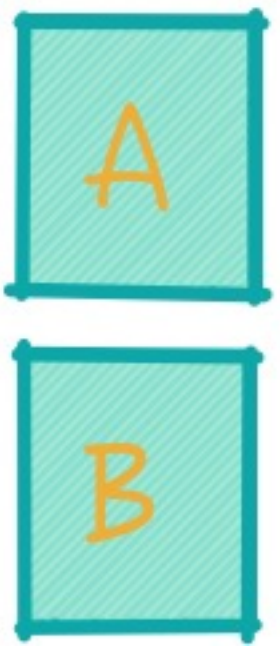
A->B

C->D

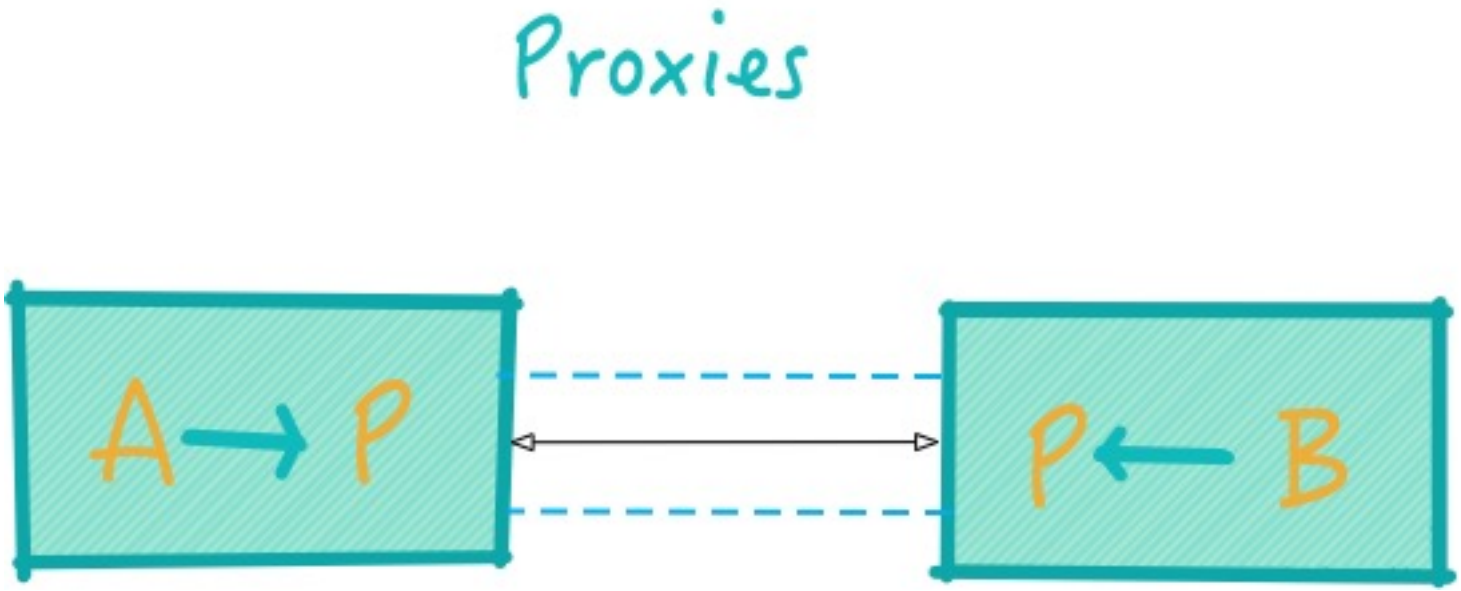
Web->DB

50*5???

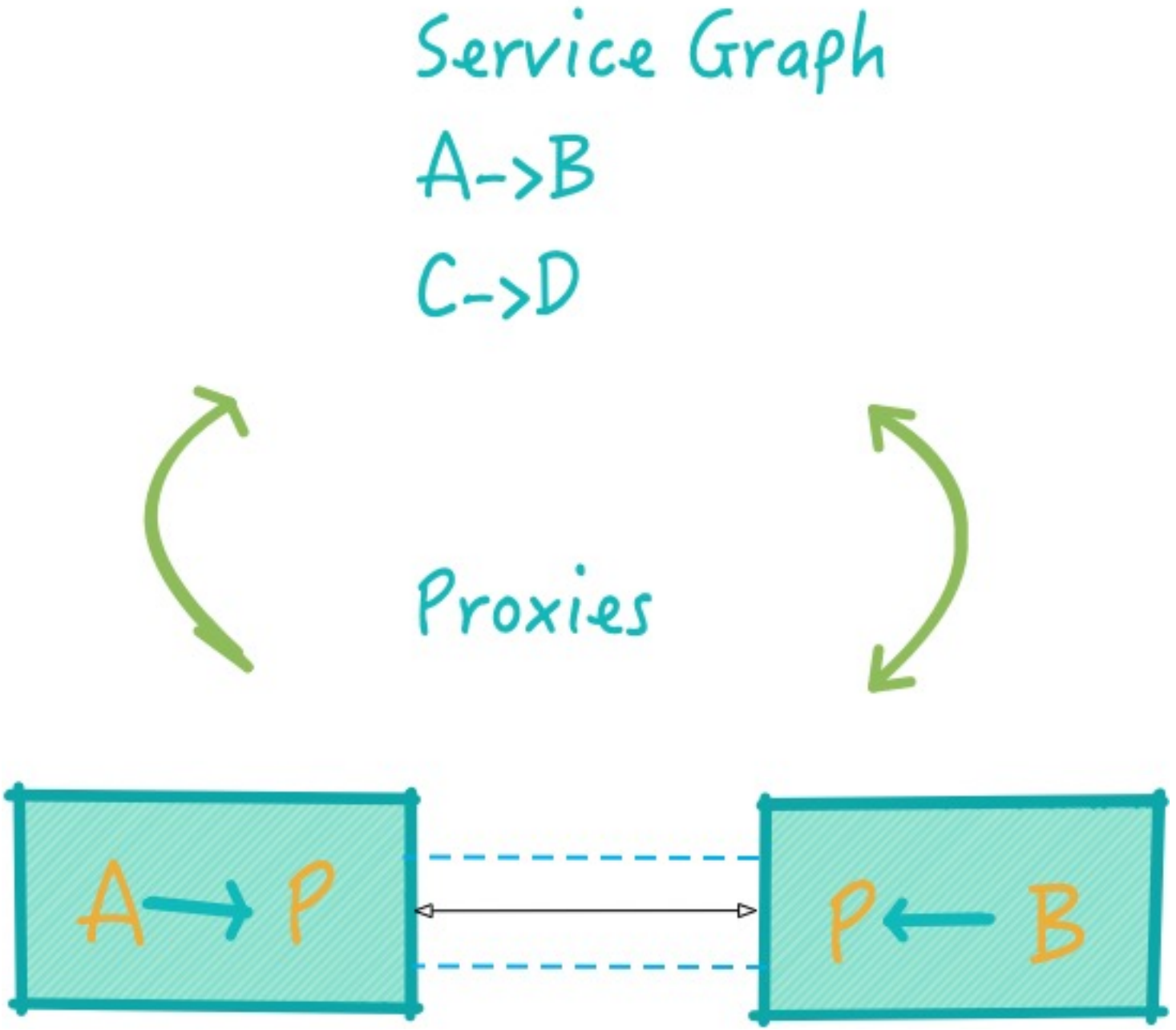
Cert Auth



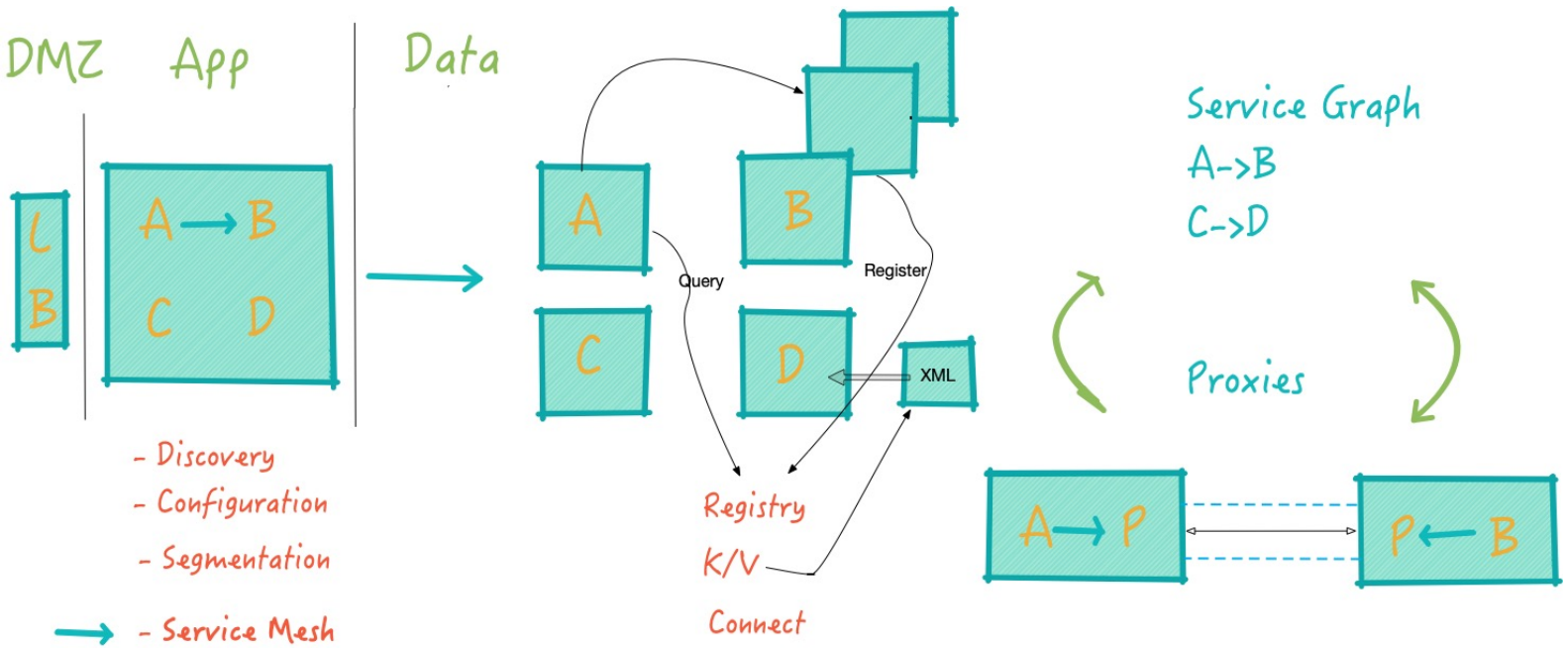
Proxies



Proxies and Service Graph



Service mesh



Lab: Consul install

- Please do lab 'lab31'
- https://github.com/elephantscale/vault-consul-labs-answers/tree/main/lab31

Consul for Service Discovery

Consul overview

Microservices

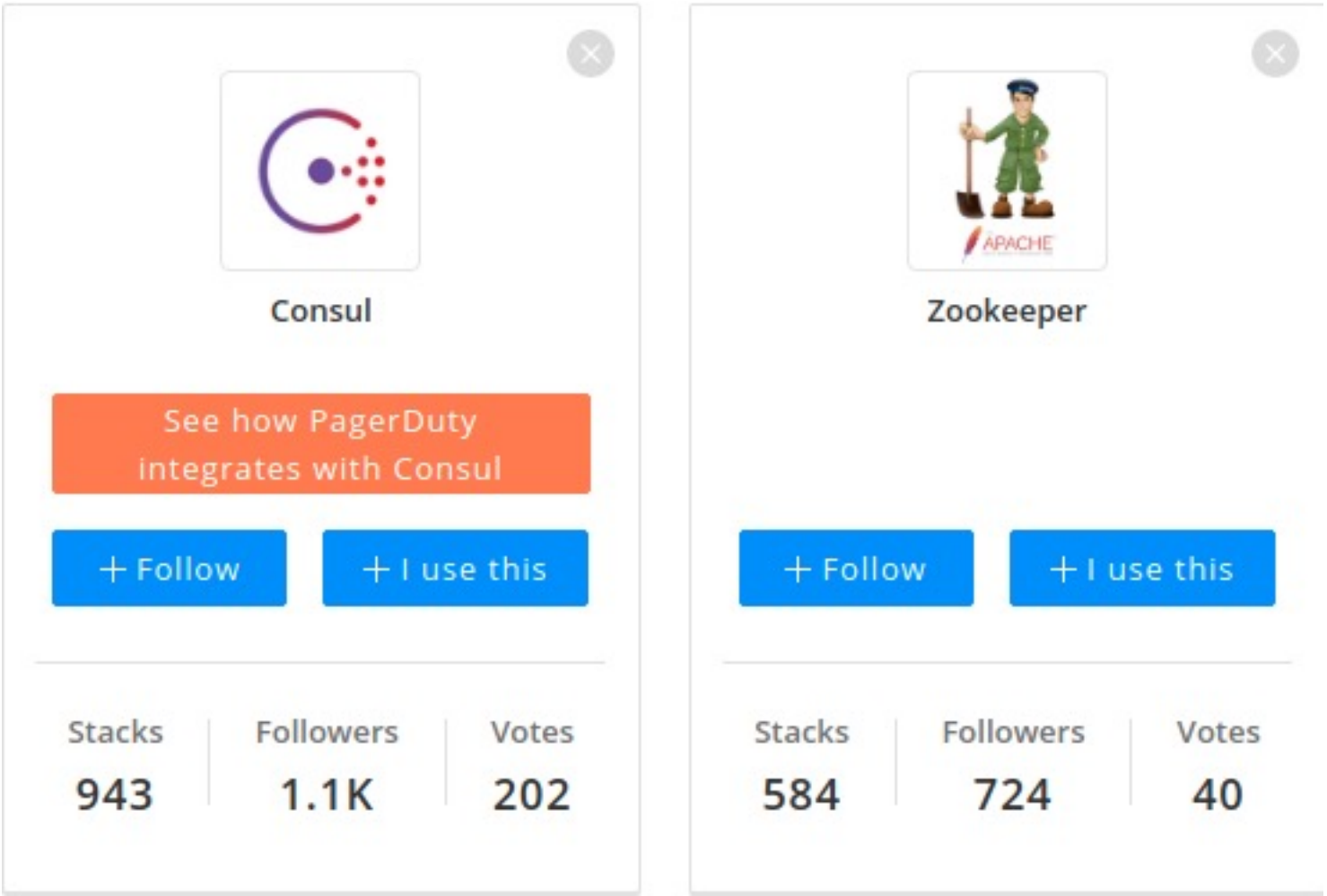
Consul for Service Discovery

Terraform integration

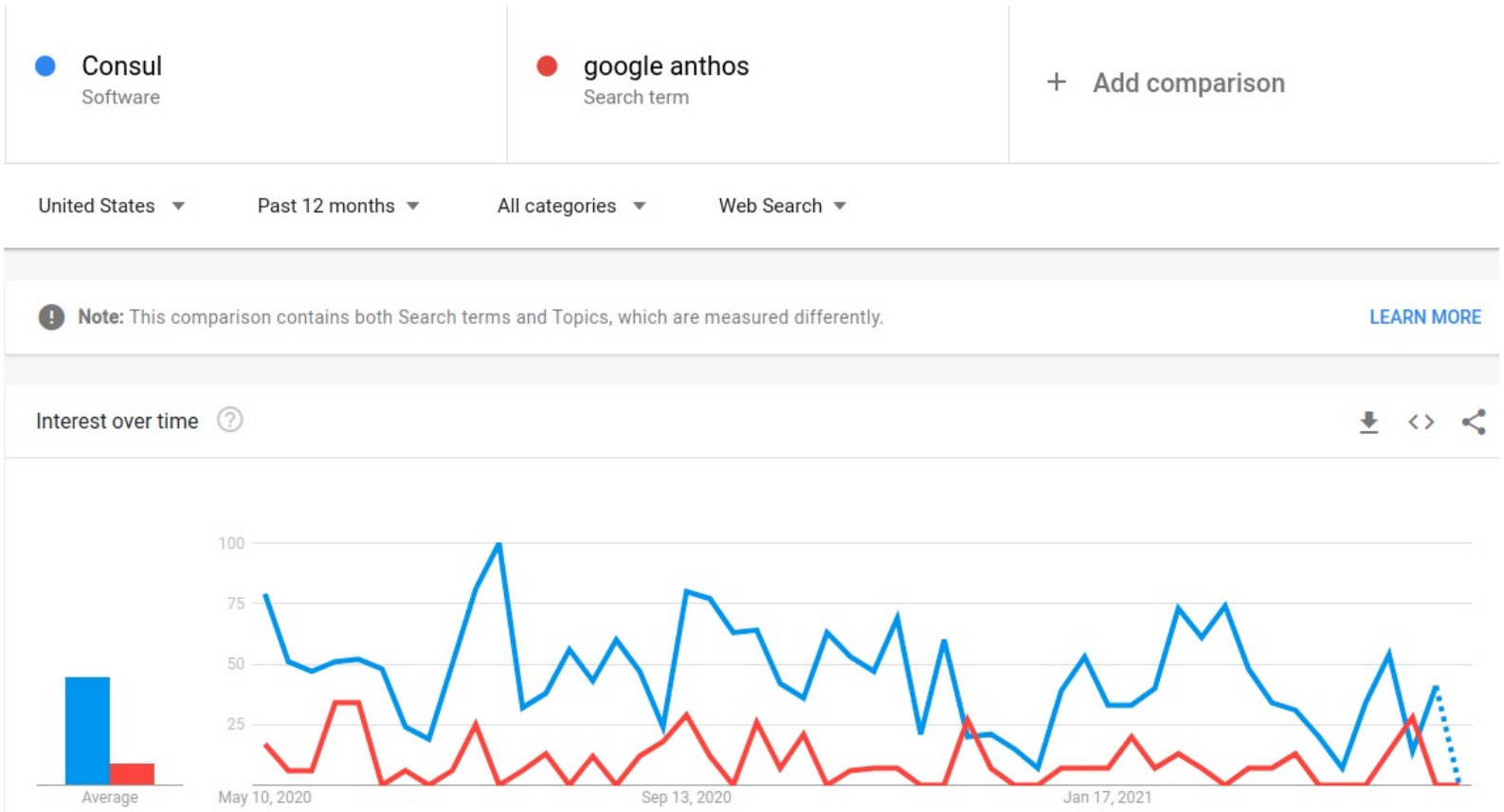
Why service discovery?

- Applications are running on dynamic infrastructure
- Need to determine services are running and how they can be accessed
- HashiCorp provides a solution, and it is Consul
- How?
  - applications and services register with Consul
  - service registry is populated
  - connect to critical services throughout the organization

Consul vs Zookeeper



Consul vs Google Anthos



Consul is from HashiCorp

- Consul is a solution provided by HashiCorp
- tight integration with Consul to simplifies access to Vault
- Users and applications can query Consul using either DNS or the API
- Consul responds to the query with the information needed to establish communication with the active Vault node
- Faster and smarter than a load balancer

Consul and Vault cluster

- Consul can also simplify the connectivity to standby nodes within the cluster
- Especially when running Vault Enterprise with performance standby nodes
- Performance standby nodes can service Vault read operations and help Vault operators scale those read operations within a cluster

Registering the Vault Service with Consul

- When you deploy a Vault cluster with Consul backend -
  - Vault service is automatically registered with the Consul service registry by default
- Best practice
  - If Consul service discovery is desired, a second Consul cluster should be deployed to manage this functionality

How to register

- The *service_registration* stanza is added to the Vault configuration
- This configuration includes
  - the address and port of the Consul cluster
  - the Consul ACL token to permit Consul access (if Consul ACLs are enabled)
  - other configurable parameters such as service_tags.

service_registration stanza example

```
service_registration "consul" {
   address           = "consul.example.com:8500"
   token             = "a1b2-c3d4e5f-6a7b-8c9d"
   service_tags      = "vault, production, us-east-1"
}
```

Connecting to Vault using Consul

- In Consul, all Vault cluster nodes are registered as the Vault service
- To use Consul to discover the active node in a cluster, a client can query Consul with the DNS name

```
1 active.vault.service.consul
```

Performance standby node(s)

- In Vault Enterprise
  - To discover the performance standby node(s)
  - client can query Consul for the DNS name

```
performance-standby.vault.service.consul
```

Terraform integration

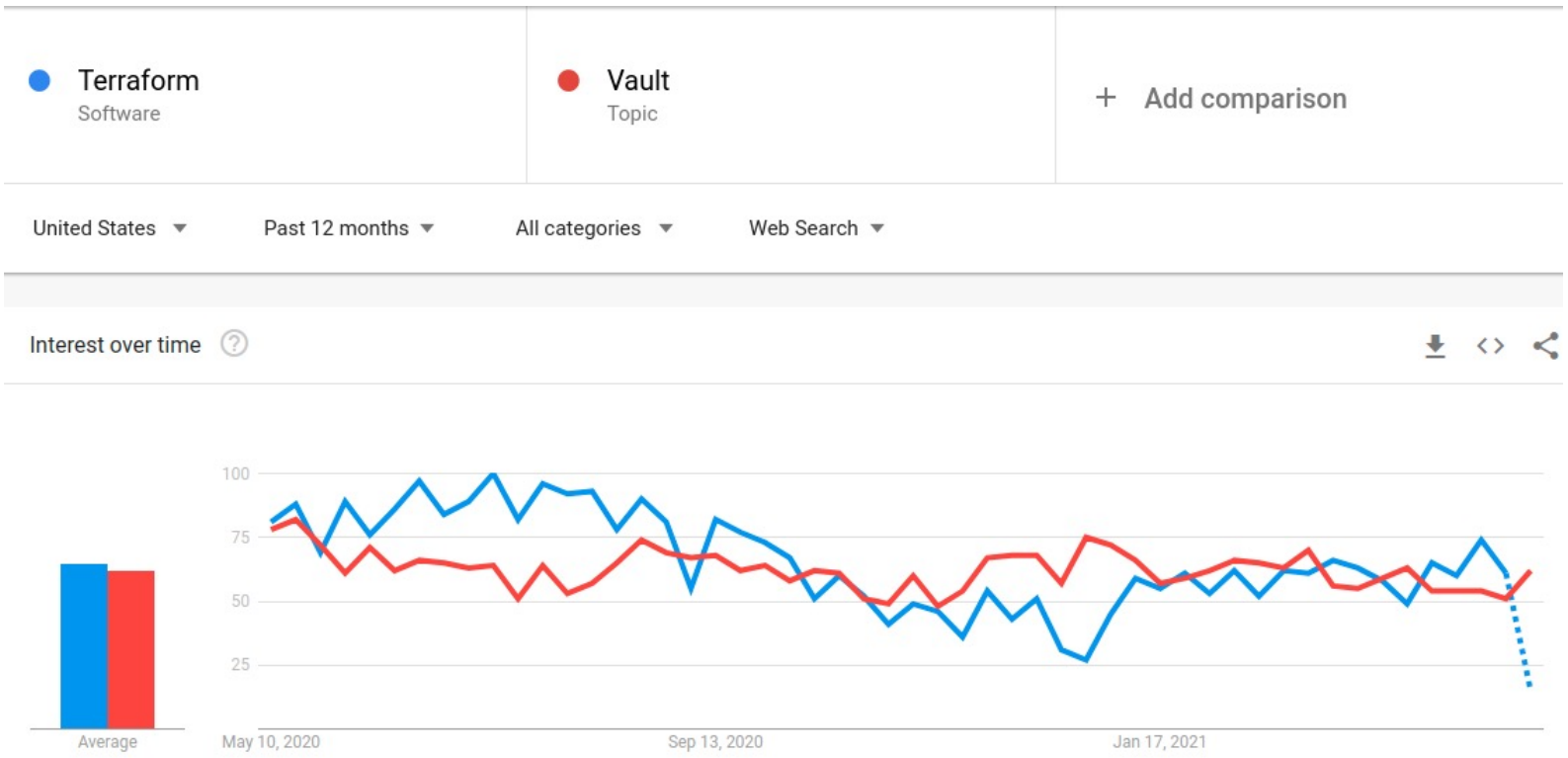
Consul overview

Microservices

Consul for Service Discovery

Terraform integration

Terraform and Vault



Vault Infrastructure with Terraform

- Terraform and Vault often go together
- Your deployment may have custom Terraform modules
- If so, it may make sense to store the modules in the Terraform pubic registry

Vault Architecture

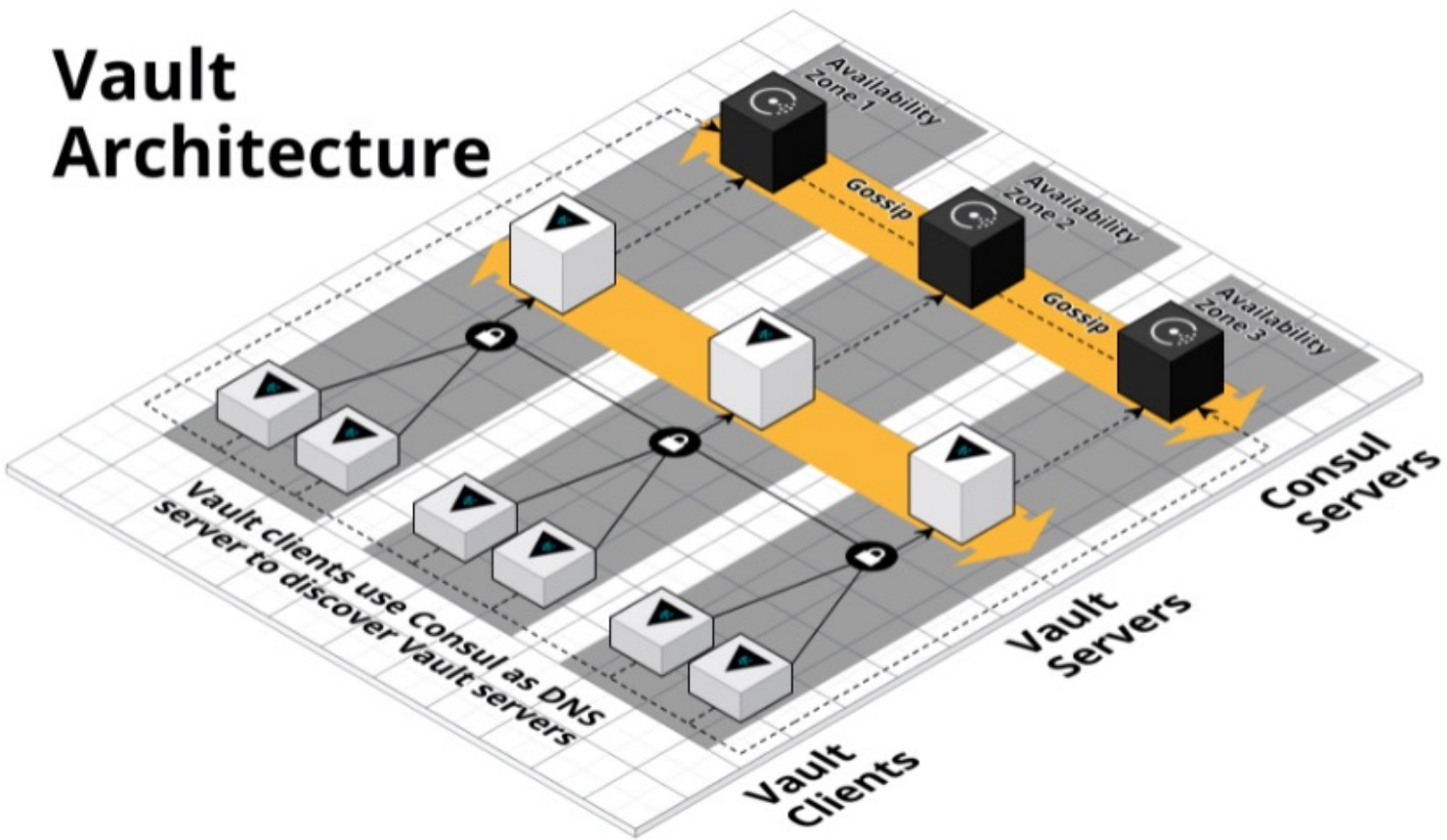
- Components to go in the Terraform configuration files:
  - Consul nodes
  - Vault nodes
  - load balancers
  - security groups
  - DNS records
  - network connectivity

Repeatability

- Repeatability if one the benefits of deploying Vault with Terraform
- Other benefits
  - performance or
  - disaster recovery
  - replicated clusters *high availability

Terraform configurations for Vault

- Amazon Web Services (https://github.com/hashicorp/terraform-aws-vault)
- Microsoft Azure (https://github.com/hashicorp/terraform-azurerm-vault)
- Google Cloud (https://github.com/terraform-google-modules/terraform-google-vault)



Configure Vault Provider and Credentials

- Similar to other Terraform providers:
  - declared within the Terraform configuration file
  - Sensitive information should be provided using environment variables
- Example
  - Terraform authenticates with Vault using a token
  - Vault address and token should be provided as an environment variable
  - VAULT_ADDR - the IP address or hostname of the targeted Vault cluster
  - VAULT_TOKEN, which is used to authenticate to Vault

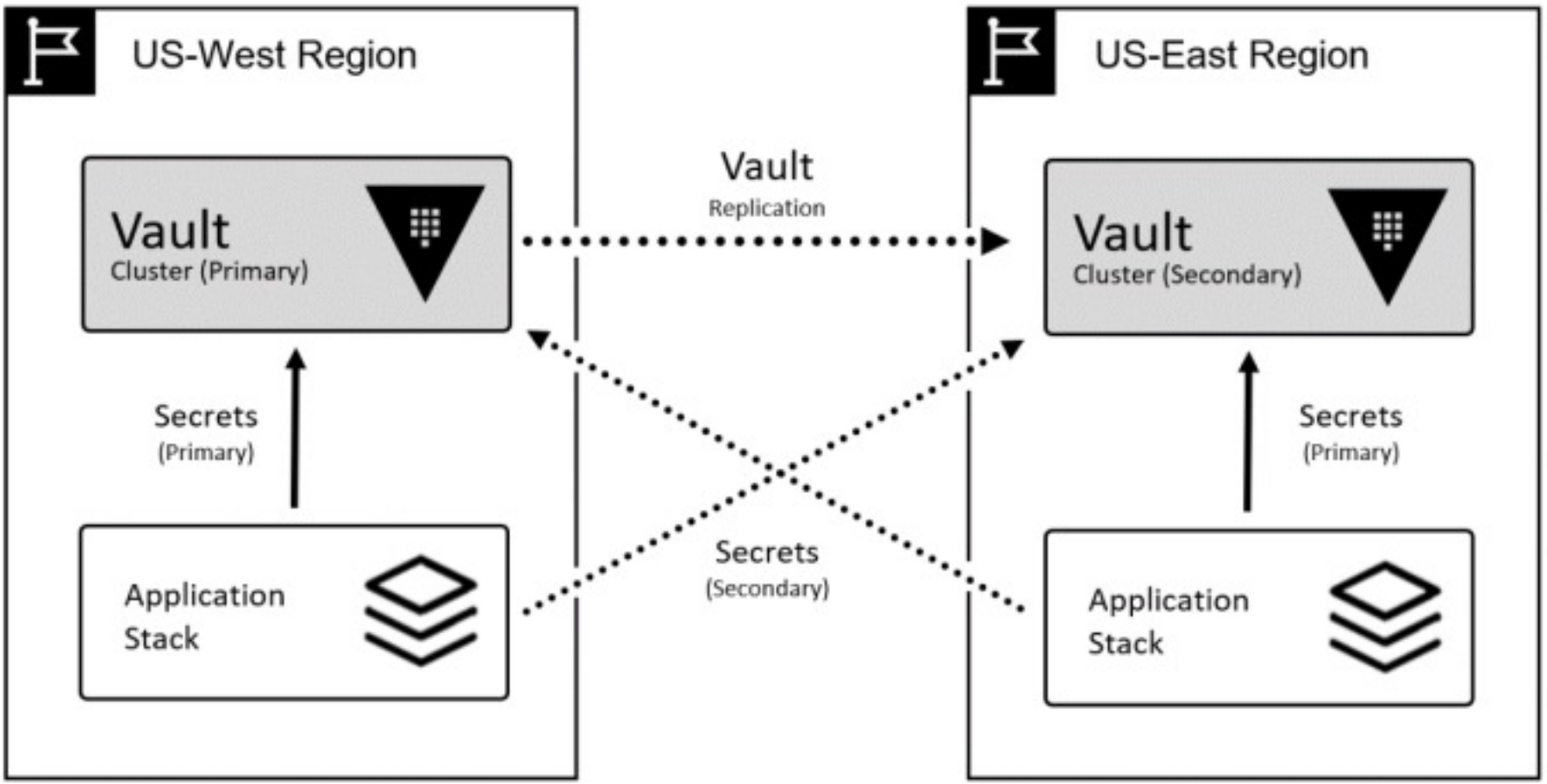
Consuming Secrets with Terraform

- Rather than provide Terraform with static credentials
- Terraform can integrate with Vault to retrieve or generate credentials before applying the configuration
- For example,
  - Terraform can retrieve temporary AWS credentials from Vault to deploy EC2 instances
  - After the lease has expired, Vault automatically revokes the AWS credentials.

Coding for Reliability

- Additional automated processes
  - automated processes can ensure applications can access the vault service
- Example
  - a team is responsible for managing a shared, multi-tenant Vault service for hundreds of internal teams
  - Vault environment was a large, multi-cluster Vault deployment
  - Each onboarded application was configured to attach to a single, local cluster.
- Solution
  - native disaster recovery options built into Vault, or
  - a load balancer in front of the Vault clusters, or even better
  - use the AppRole auth method since both RoleIDs and SecretIDs are replicated across all clusters

Reliability solution



More with Consul

- Now that we know the basics, what is next? Perfect your skills in the following areas:
  - Consul Service Mesh
  - Create a datacenter with Consul Docker containers as the agents
  - Learn how to deploy Consul on Kubernetes
  - Deploy HashiCorp Consul Service on Azure
  - Secure Nomad jobs with Consul service mesh