THE TREVOR PROJECT

Utilizing ArgoCD & Kubernetes for Suicide Prevention for LGBTQ+ Youth

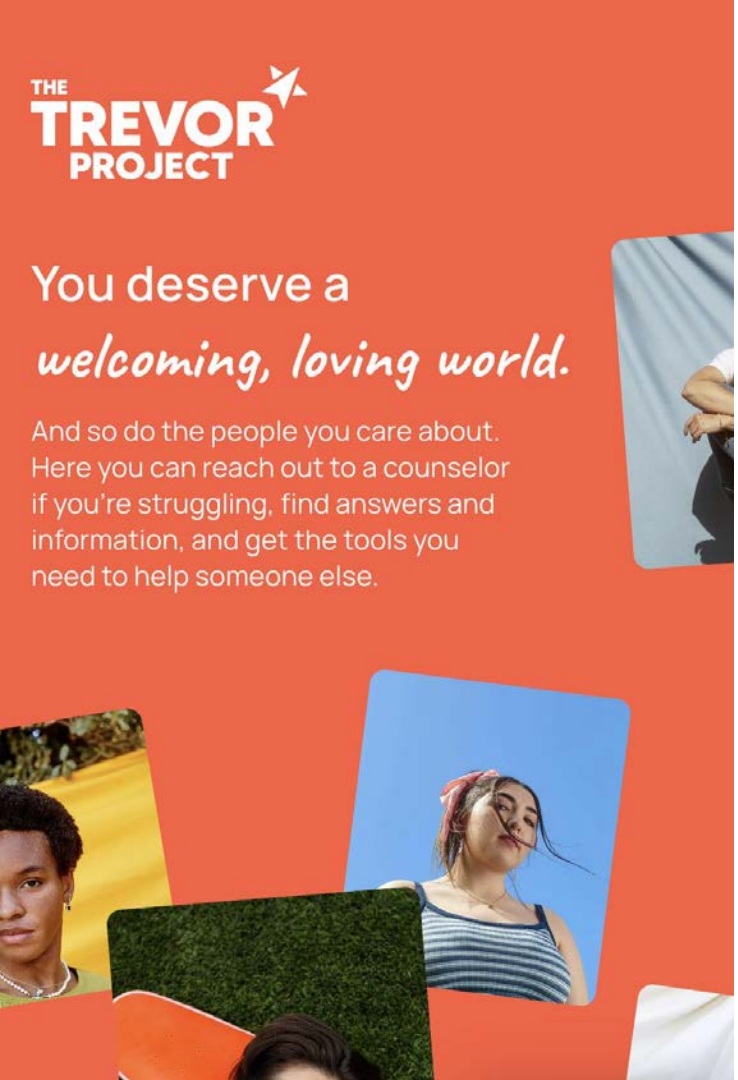
The Trevor Project

alfredo.pizana@thetrevorproject.org

in/alfredopizana

paul.pham@thetrevorproject.org

in/pham-paul

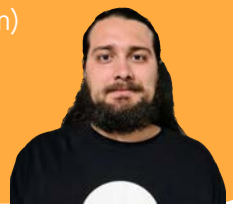


Services

- Crisis Services
- Advocacy
- Research
- TrevorSpace
- Education & Public Awareness

About Us

Alfredo Pizaña
Tech Lead
(he/him)



Paul Pham
Engineering Manager
(he/him)

Agenda

1. Project Context
2. Architecture
3. Challenges & Lessons Learned
4. Impact

01.

Project Context

Project Context

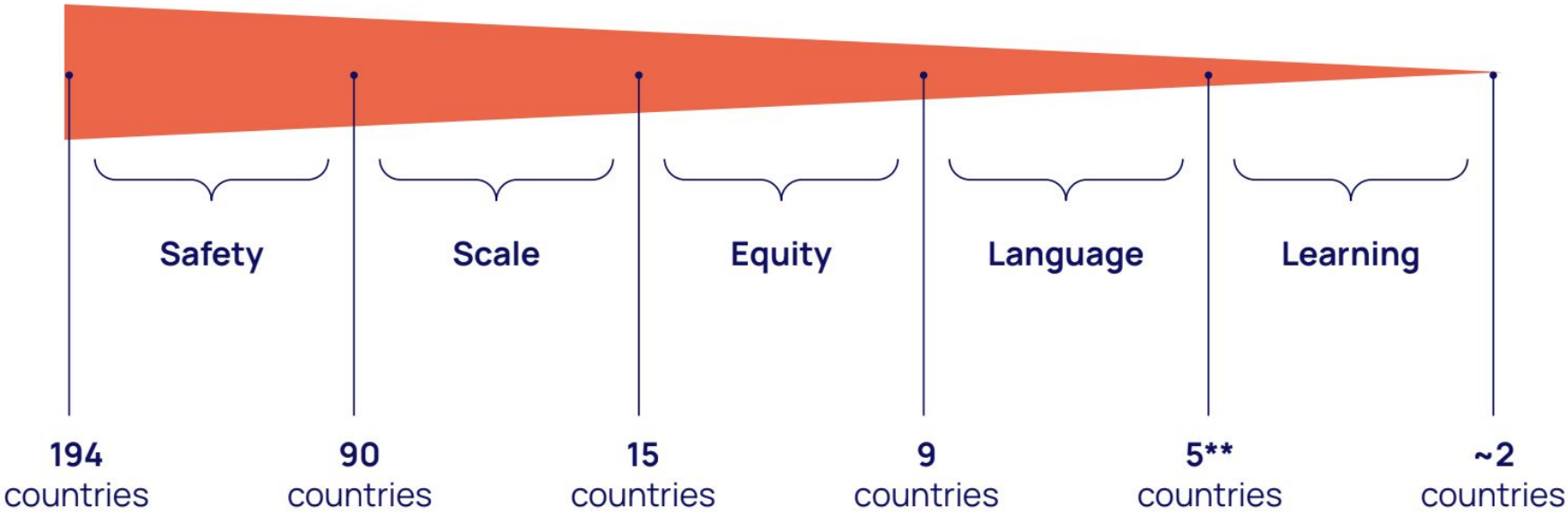
> The Trevor Project's mission is to end suicide among LGBTQ+ young people

Project Context

Aug 11, 1998

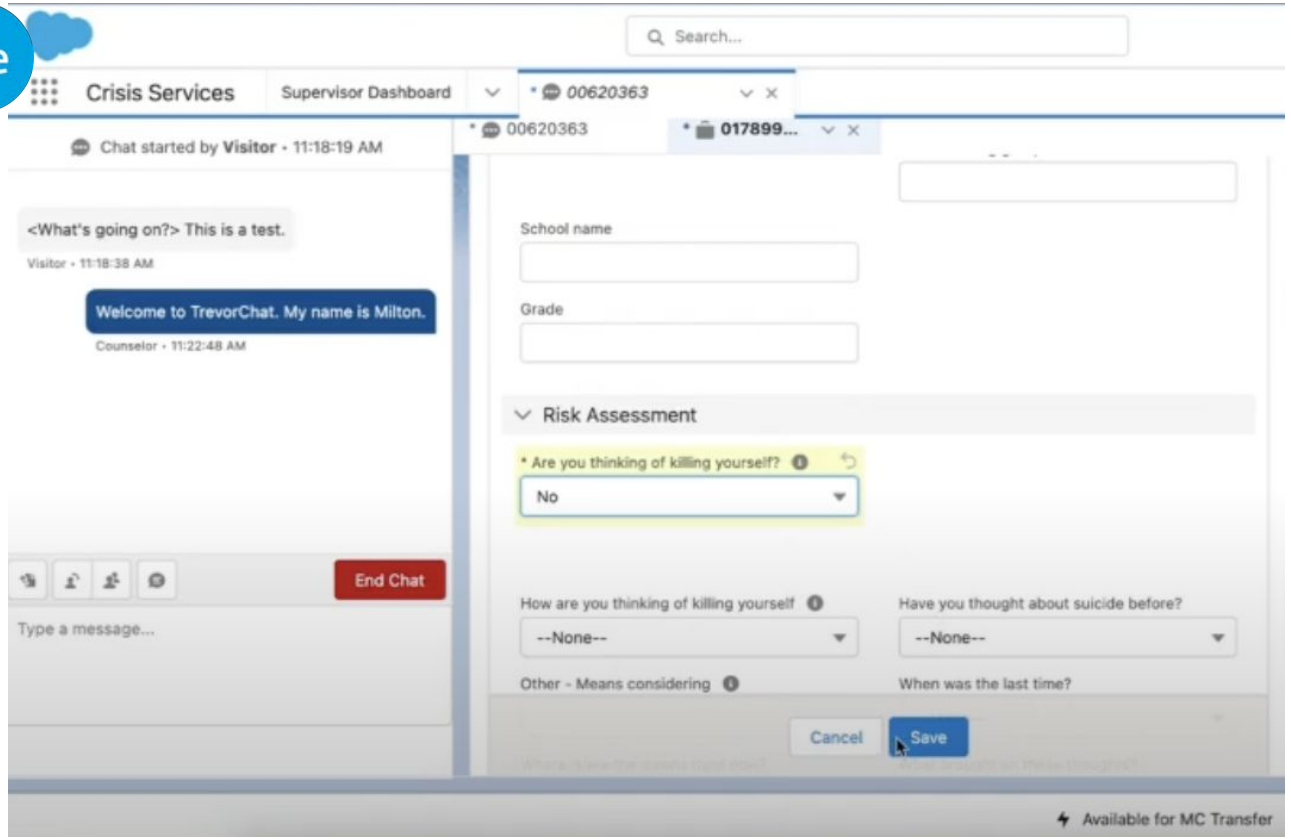
THE TREVOR PROJECT

Project Context

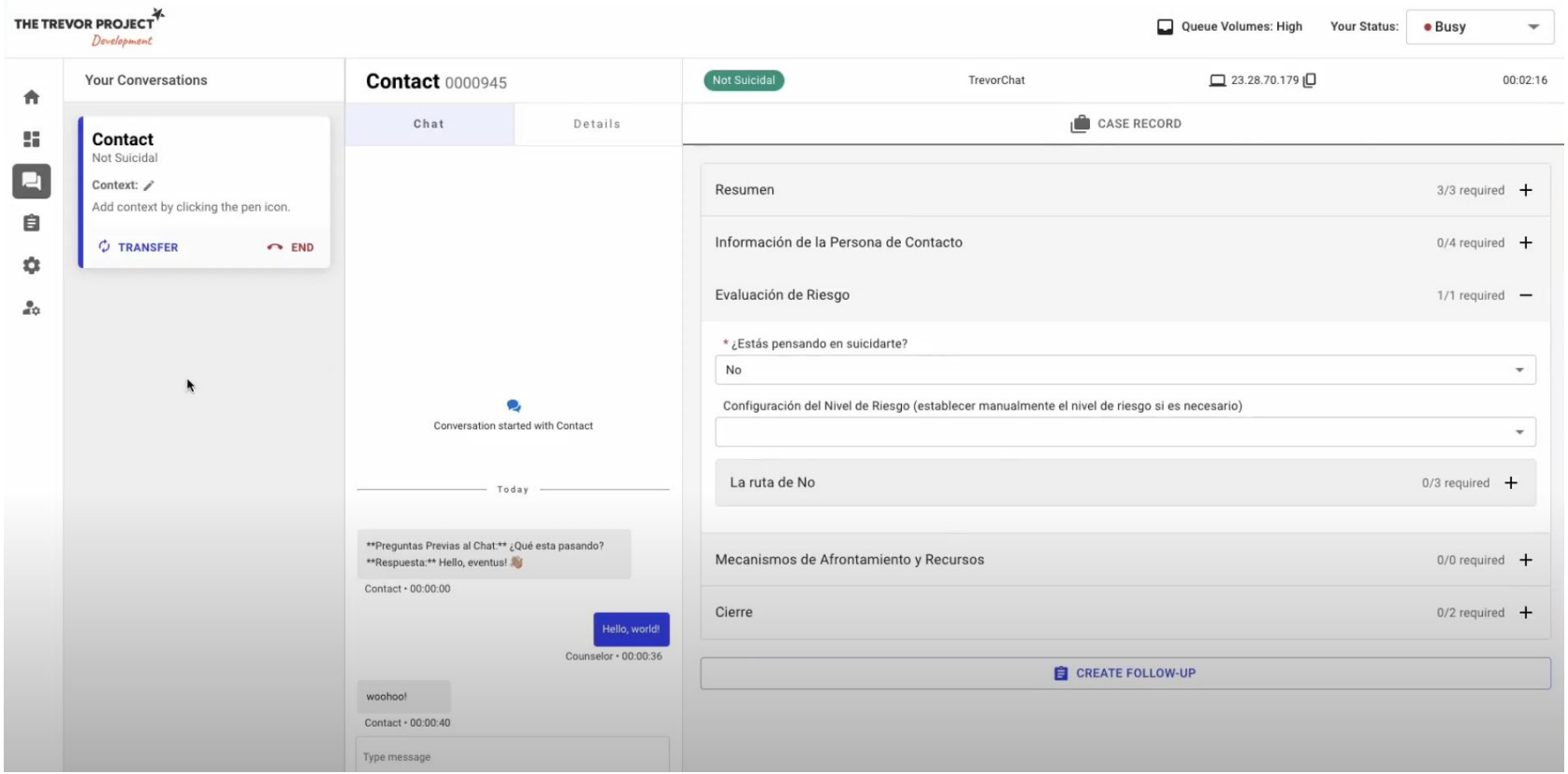


Project Context

Project Context



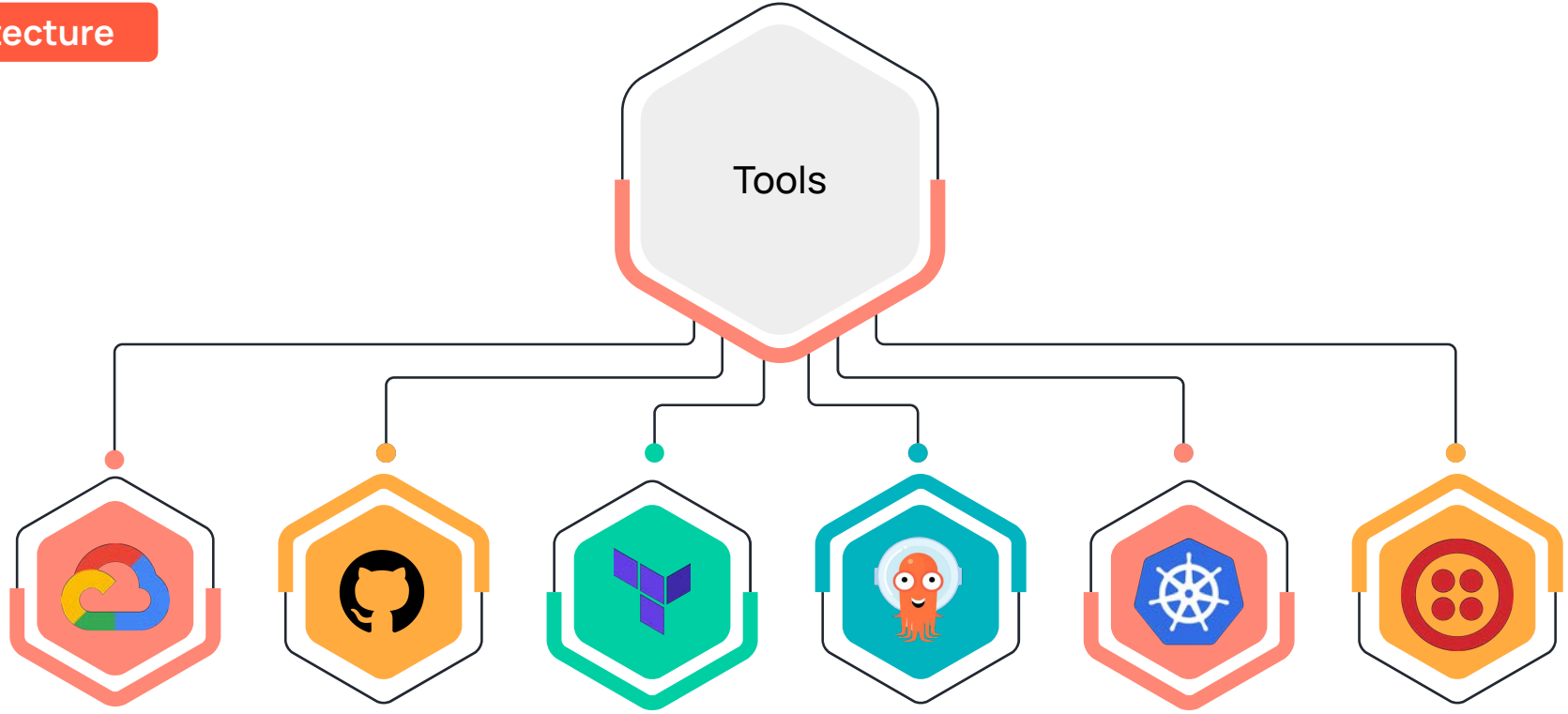
Project Context



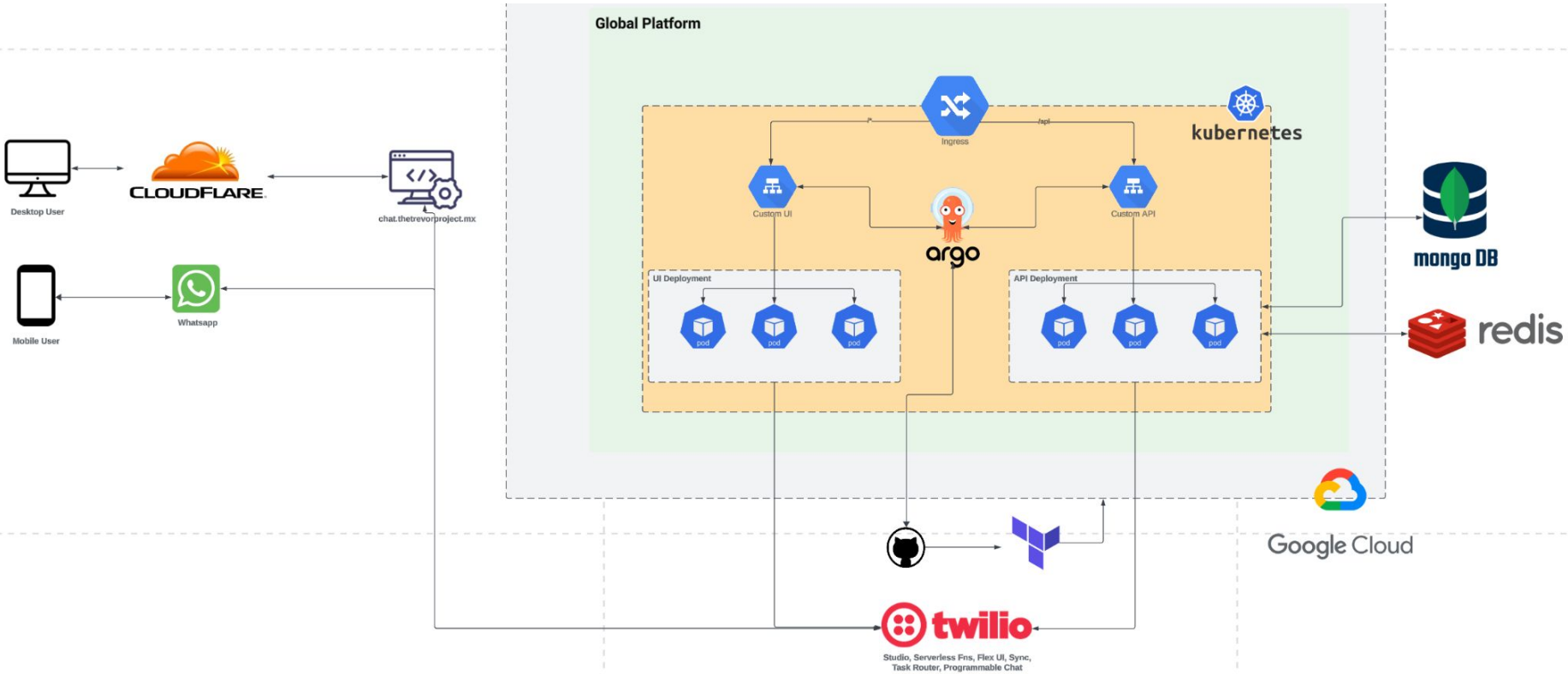
02.

Architecture

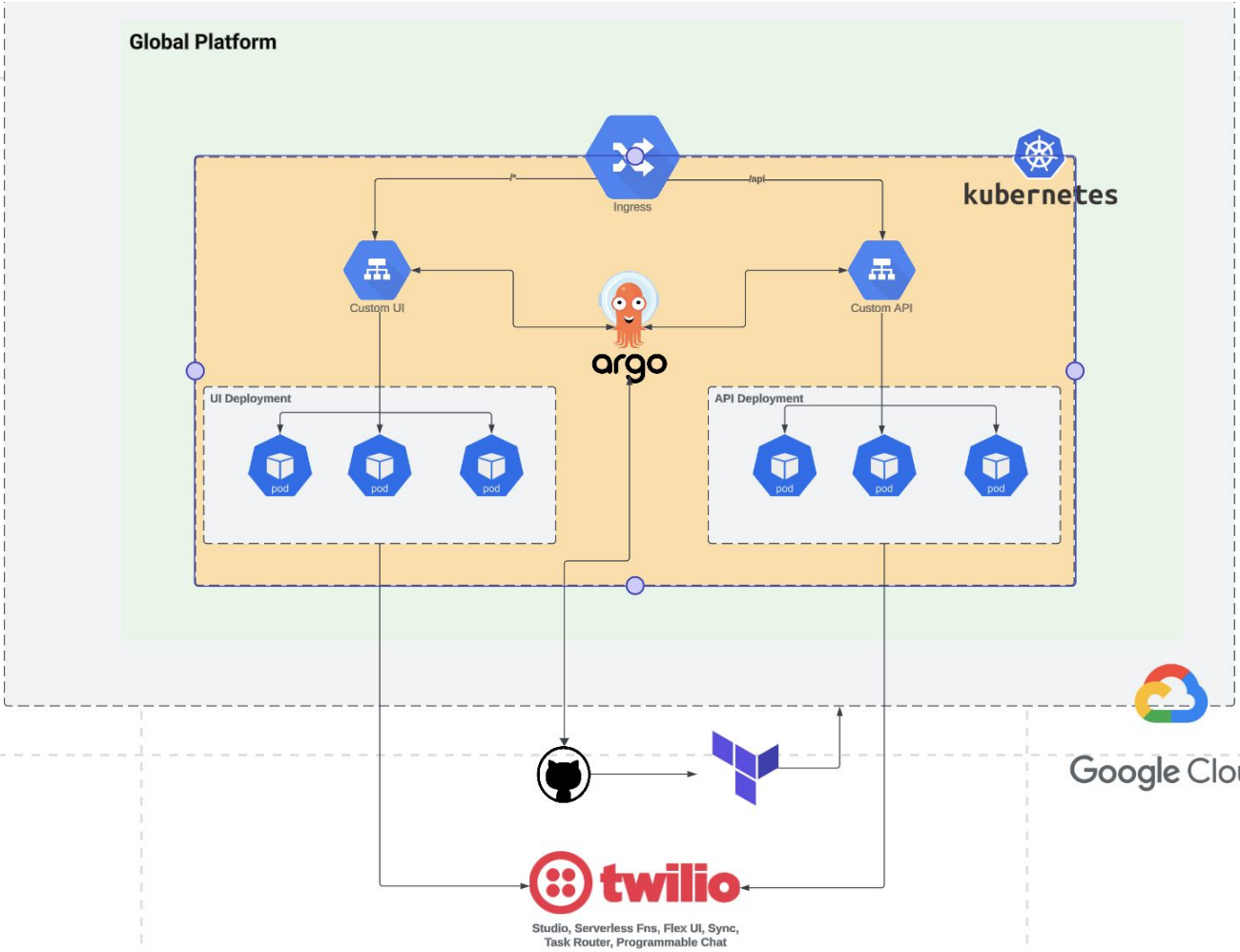
Architecture



Architecture



Architecture



K8s Using Terraform

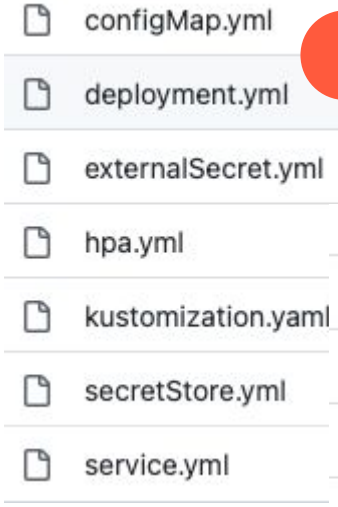
```
module "gke" {
  for_each      = var.gke_clusters
  source        = "./modules/gke/autopilot"
  project_id    = var.project_id
  ip_cidr_range = each.value["ip_cidr_range"]
  # network_suffix  = var.network_suffix
  environment       = var.environment
  business_prefix   = var.business_prefix
  region            = var.region
  network_id        = module.network.yes.vpc_id
  network_name      = module.network.yes.vpc_name
  infra_environment = var.infra_environment

  depends_on = [
    module.network
  ]
}
```

Apps Deployment - Kustomize

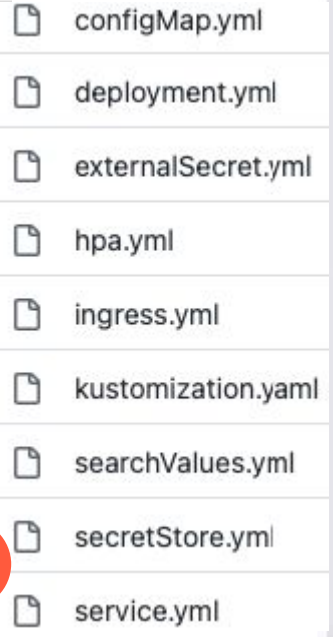
- **Goal**: Define a method to deploy Apps in K8s
- **Setup**: Utilize kustomization file, which is a YAML specification of a Kubernetes Resource Model (KRM) object called a *Kustomization*. A kustomization describes how to generate or transform other KRM objects.

```
app1/
  kustomization.yaml
    | resources:
    | - ../base
    | patches:
    | - patch1.yaml
  patch1.yaml

app2/
  kustomization.yaml
    | resources:
    | - ../base
    | patches:
    | - patch2.yaml
  patch2.yaml

base/
  kustomization.yaml
    | resources:
    | - deployment.yaml
    | - configMap.yaml
  deployment.yaml
  configMap.yaml
```

Apps Deployment - Kustomize



API



UI

Kustomize Example

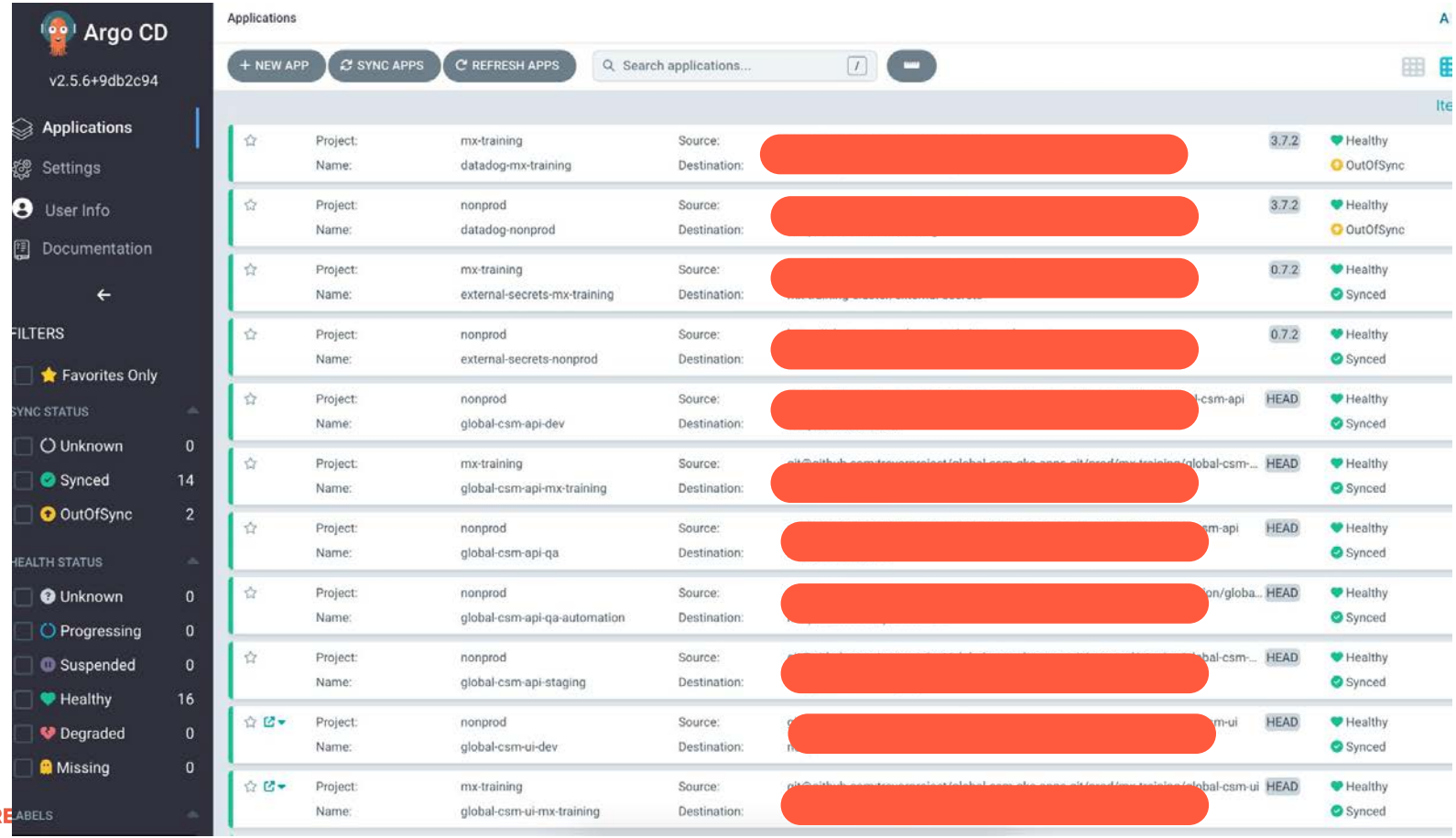
```yaml
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
namespace: dev

resources:
 - ../../../base/global-csm-api

configMapGenerator:
- name: global-csm-api-config
  envs:
    - configMap.yml
images:
  - name: us-docker.pkg.dev/trev
    newTag: latest

patchesStrategicMerge:
- externalSecret.yml

patches:
- target:
     kind: Deployment
     name: global-csm-api-deployment
  patch: |-
    - op: replace
      path: /spec/replicas
      value: 1
```

ArgoCD

Automates the deployment of the desired application states in the specified target environments. Application deployments can track updates to branches, tags, or pinned to a specific version of manifests at a Git commit.

- Argo CD follows the **GitOps** pattern of using Git repositories as the source of truth for defining the desired application state
- Kubernetes manifests can be specified in several ways:
  - Kustomize applications
  - Helm charts
  - Jsonnet files
  - Plain directory of YAML/json manifests

ArgoCD

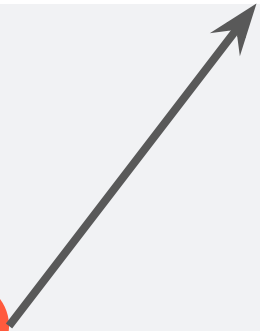

Argo CD
v2.5.6+9db2c94
Applications
Settings
User Info
Documentation
FILTERS
Favorites Only
SYNC STATUS
Unknown 0
Synced 14
OutOfSync 2
HEALTH STATUS
Unknown 0
Progressing 0
Suspended 0
Healthy 16
Degraded 0
Missing 0
LABELS
Applications
+ NEW APP
SYNC APPS
REFRESH APPS
Search applications...
Project: mx-training Name: datadog-mx-training Source: Destination: 3.7.2 Healthy OutOfSync
Project: nonprod Name: datadog-nonprod Source: Destination: 3.7.2 Healthy OutOfSync
Project: mx-training Name: external-secrets-mx-training Source: Destination: 0.7.2 Healthy Synced
Project: nonprod Name: external-secrets-nonprod Source: Destination: 0.7.2 Healthy Synced
Project: nonprod Name: global-csm-api-dev Source: Destination: HEAD Healthy Synced
Project: mx-training Name: global-csm-api-mx-training Source: Destination: HEAD Healthy Synced
Project: nonprod Name: global-csm-api-qa Source: Destination: HEAD Healthy Synced
Project: nonprod Name: global-csm-api-qa-automation Source: Destination: HEAD Healthy Synced
Project: nonprod Name: global-csm-api-staging Source: Destination: HEAD Healthy Synced
Project: nonprod Name: global-csm-ui-dev Source: Destination: HEAD Healthy Synced
Project: mx-training Name: global-csm-ui-mx-training Source: Destination: HEAD Healthy Synced

ArgoCD

```
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: external-secrets-mx-training
  namespace: argo
spec:
  project: mx-training
  destination:
    server: "https:/
    namespace: external-secrets
  source:
    chart: external-secrets
    repoURL: https://charts.external-secrets.io
    targetRevision: 0.7.2
    helm:
      releaseName: external-secrets
     #valueFiles:
     # - external-secrets-values.yaml
```

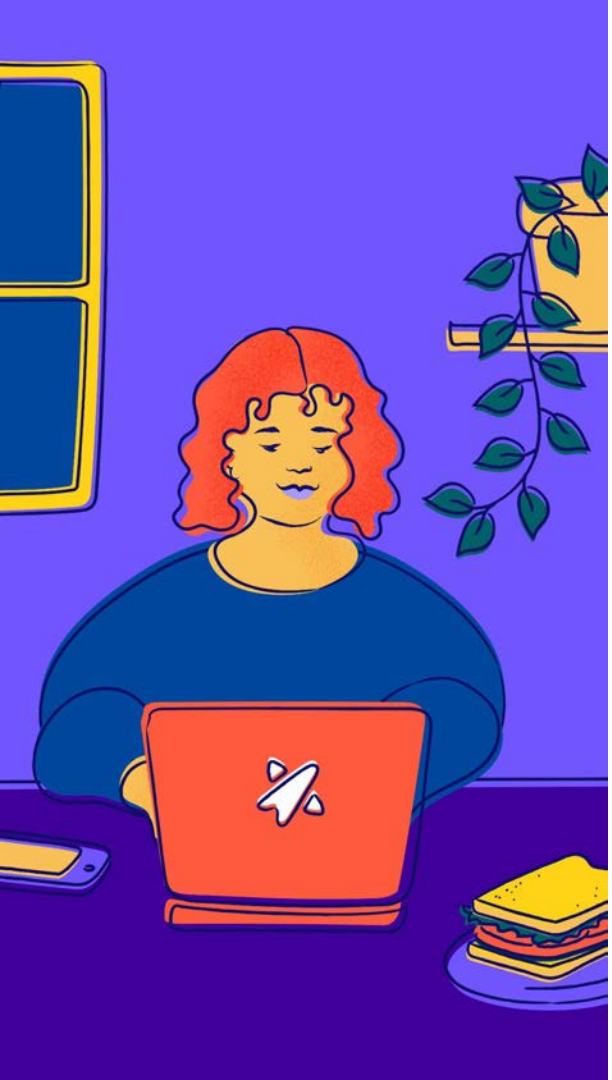
```
kubectl apply -f external-secret-app.yaml
```

Deploy App by K8 Resource



03.

Challenges & Lessons Learned



Challenges

1. Strict launch date by the business
2. Lack of knowledge on build open-source technologies
3. Multiple code repositories to manage (10)

Lessons Learned

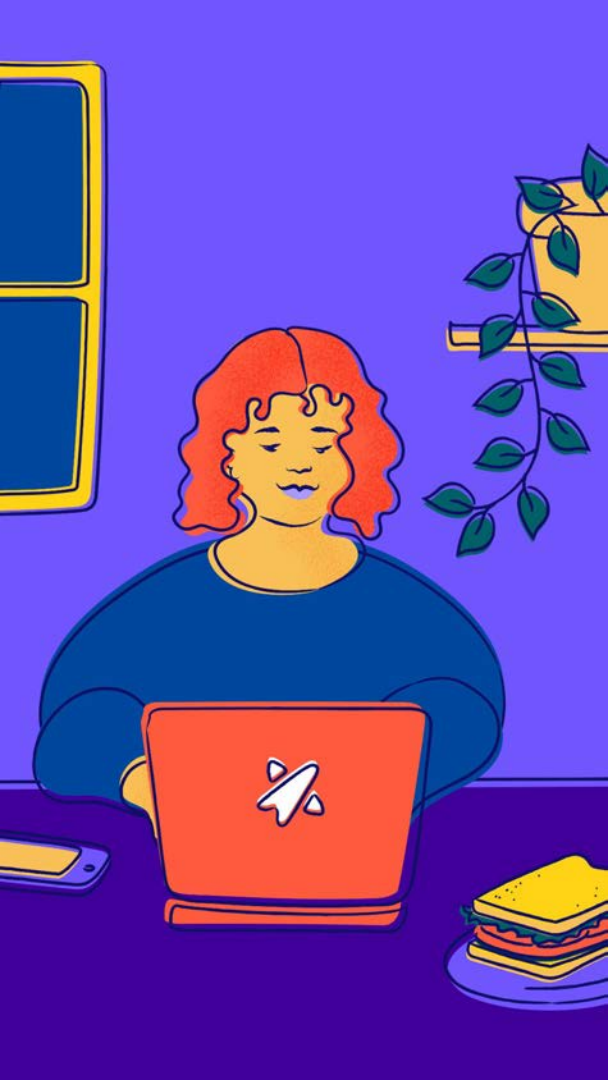
1. Upskilling our engineers, product managers, and other team members on new technologies beforehand

2. Add more observability to ensure systems are up and running

3. Tracking costs and using auto-scaling to reduce computing costs

04.

Impact



Impact of the K8s Platform

1. **With the help of specialized counselors, we have supported 20,000+ young people in Mexico**

2. **Our systems have sustainably scale to support over 1500 chats per month**

THE TREVOR PROJECT

Thank You!

Please feel free to ask questions and connect with us!

alfredo.pizana@thetrevorproject.org

in/alfredopizana

paul.pham@thetrevorproject.org

in/pham-paul