

# Connecting Engineering Data with

The logo for Kiwi, featuring the word "Kiwi" in a white, sans-serif font. The letters are bold and modern, with the 'i's having distinct dots. The logo is centered within a dark blue rectangular area that has a slightly lighter blue gradient, making it stand out from the overall dark blue background of the slide.

Kiwi

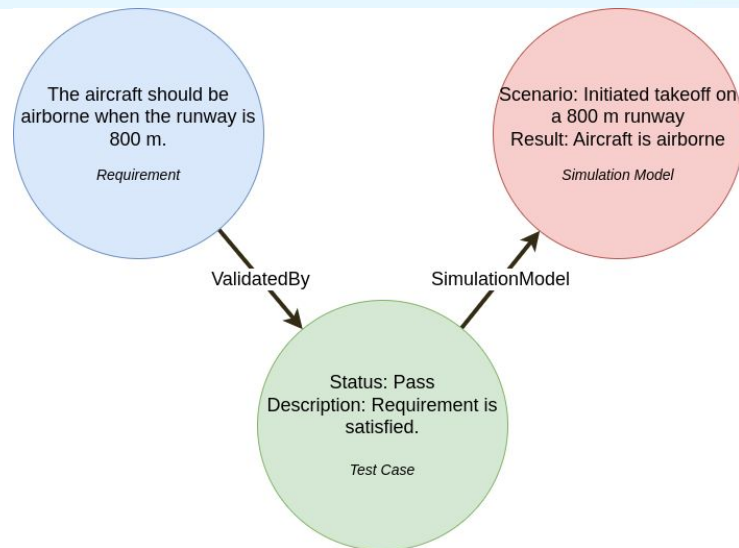
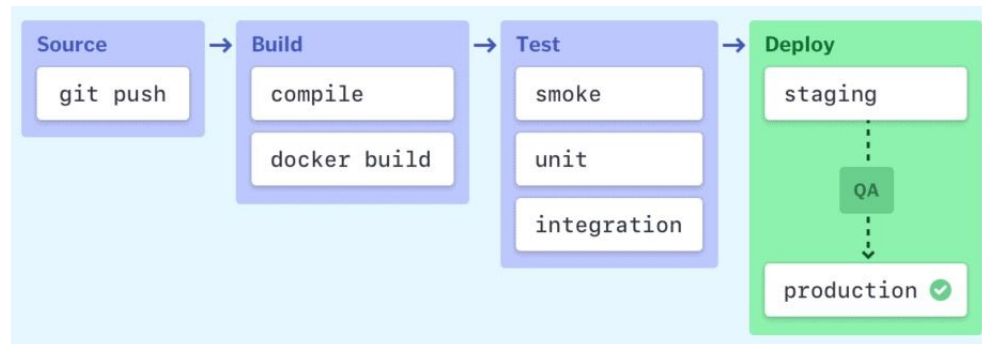
# About Me!

- Born in the Himalayan Valley
- I love math and writing code
- MSc Computer Science graduate from Trinity College Dublin
- Kubernetes Certified Application Developer
- At Koneksys since April 2021:
  - Built ML pipelines on Kubeflow
  - Built data transformation pipeline between 2 non-compatible cloud storage platforms
  - Main developer behind Kiwi



# Why Connect Data

- In multidisciplinary engineering contexts, to build non-linear CI/CD pipelines for
  - Increased Automation
  - Faster Error Detection
  - Feedback Loops (Have all test cases passed after change in requirement?)

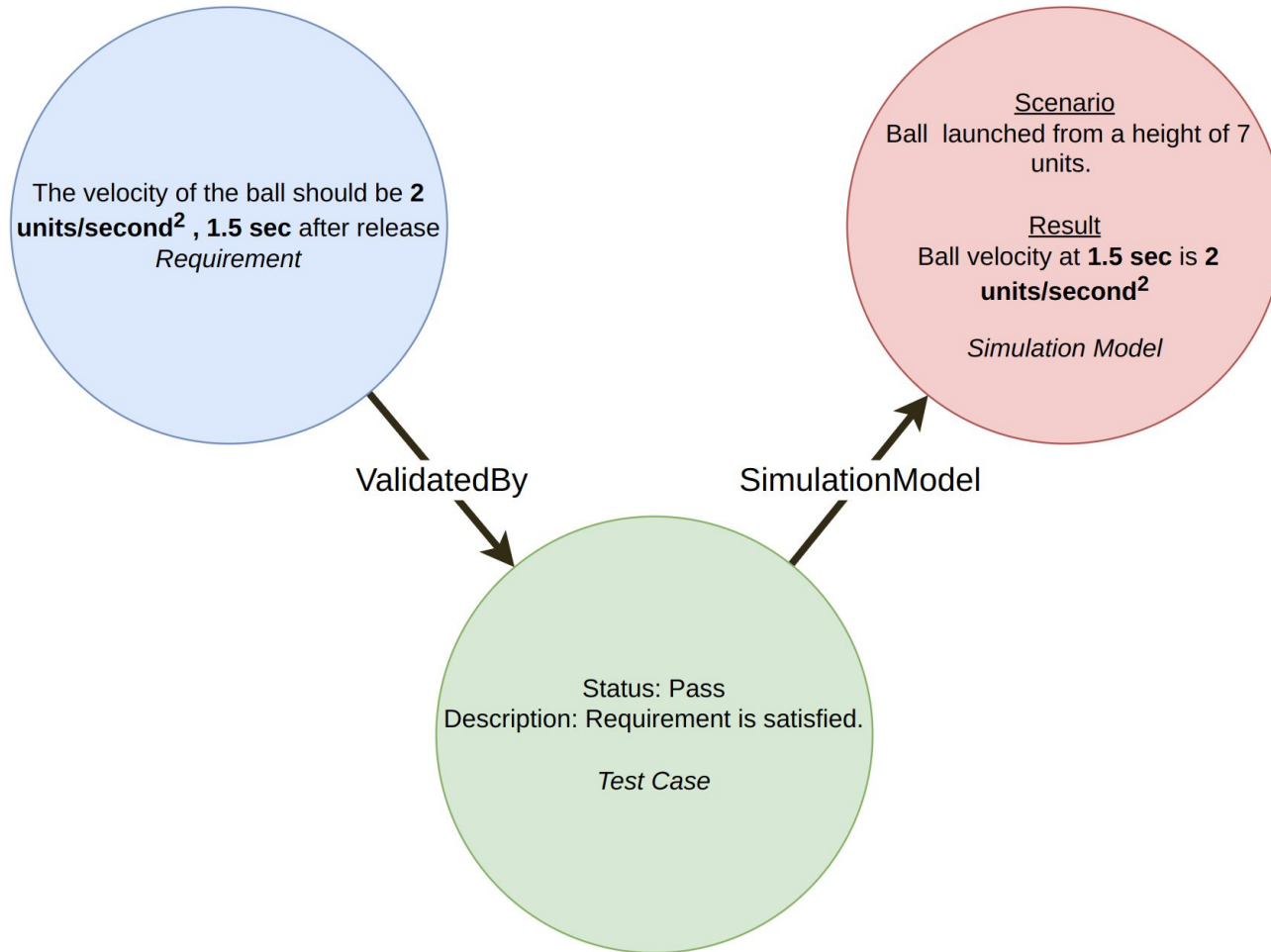


# Solution: Kiwi

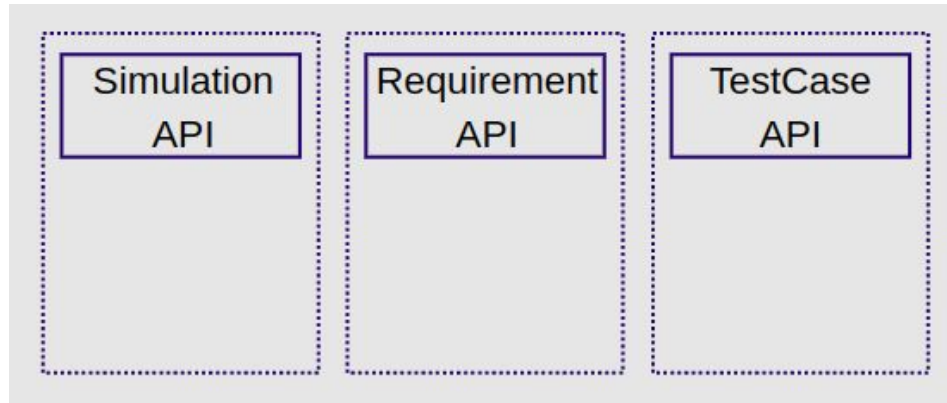
- Kiwi is an application that allows linking **resources** from different **REST APIs**, without any modification to the REST APIs.
- Inspired by **Istio's service mesh architecture**, kiwi is brought to life by **Kubernetes**.



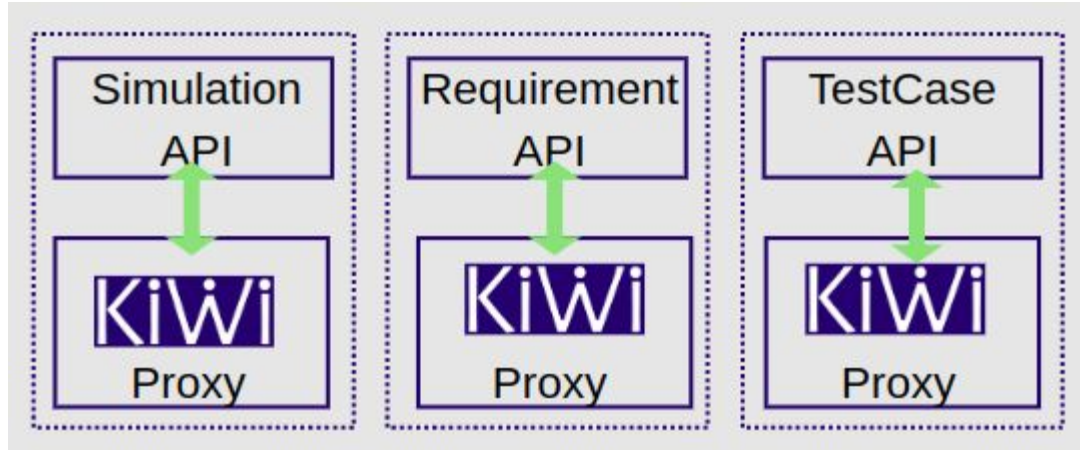
# Demo

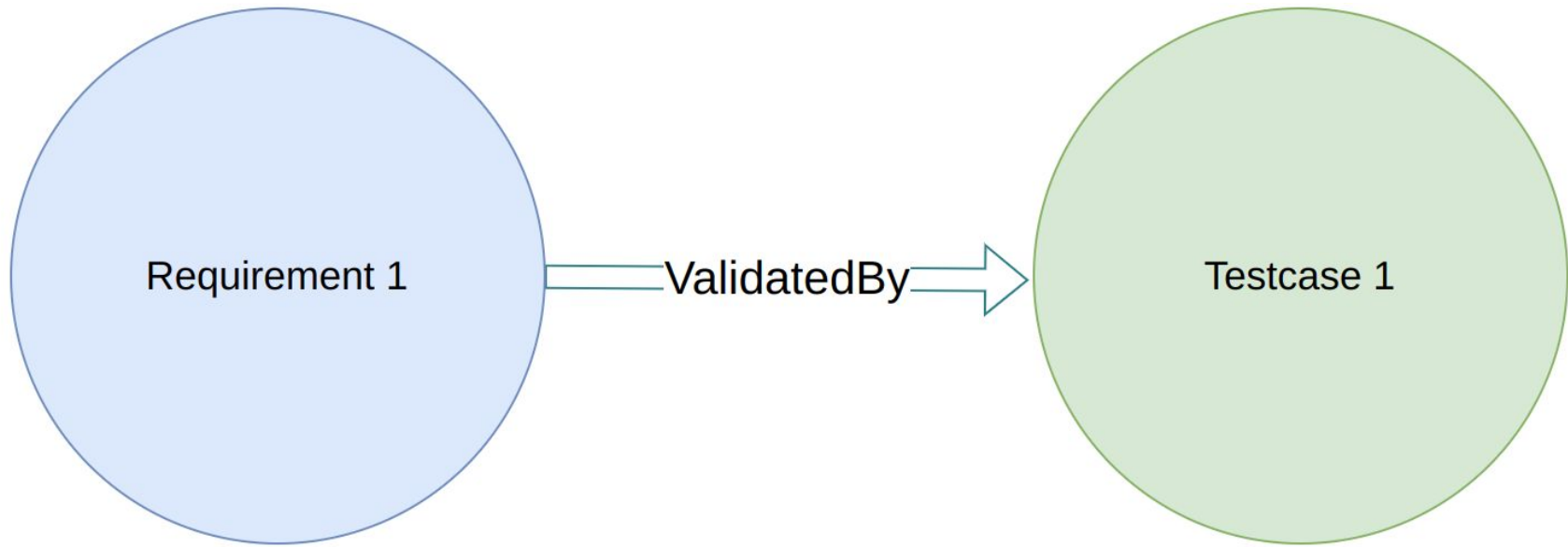


# Our APIs

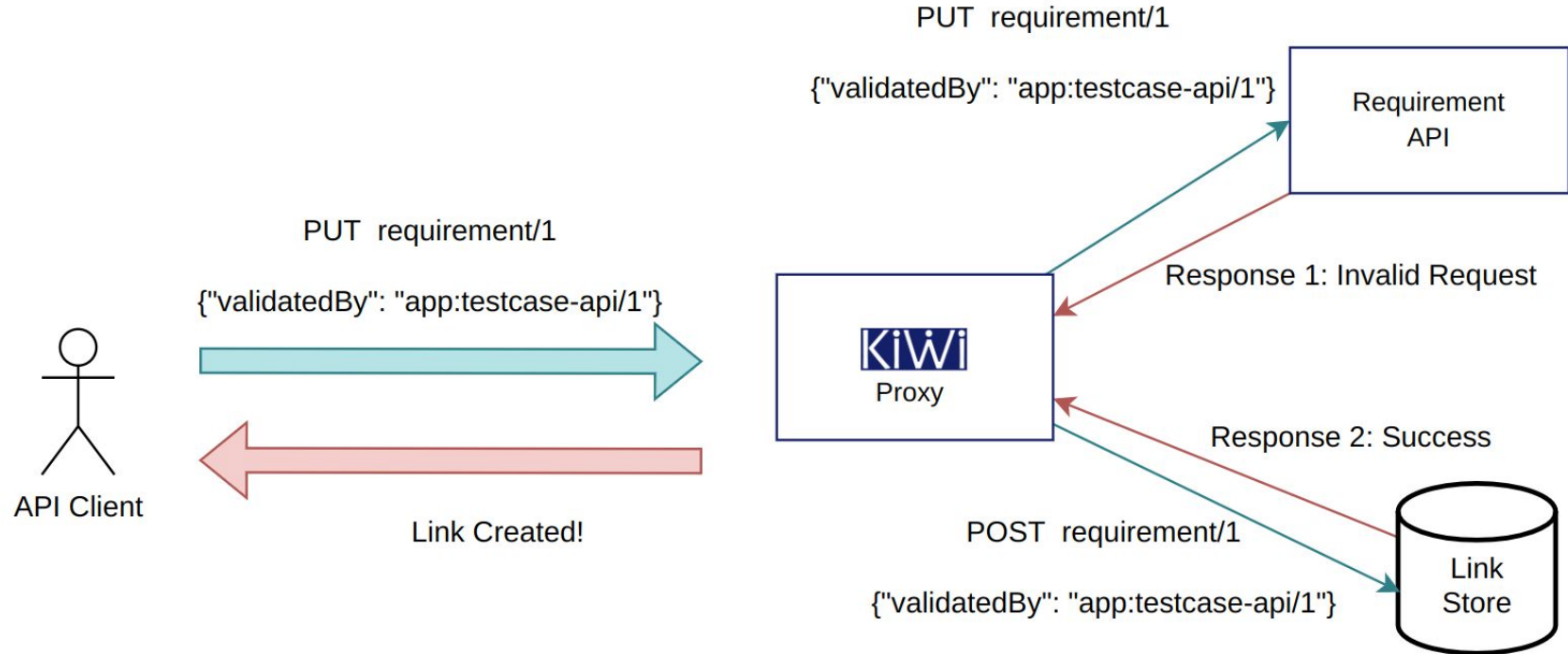


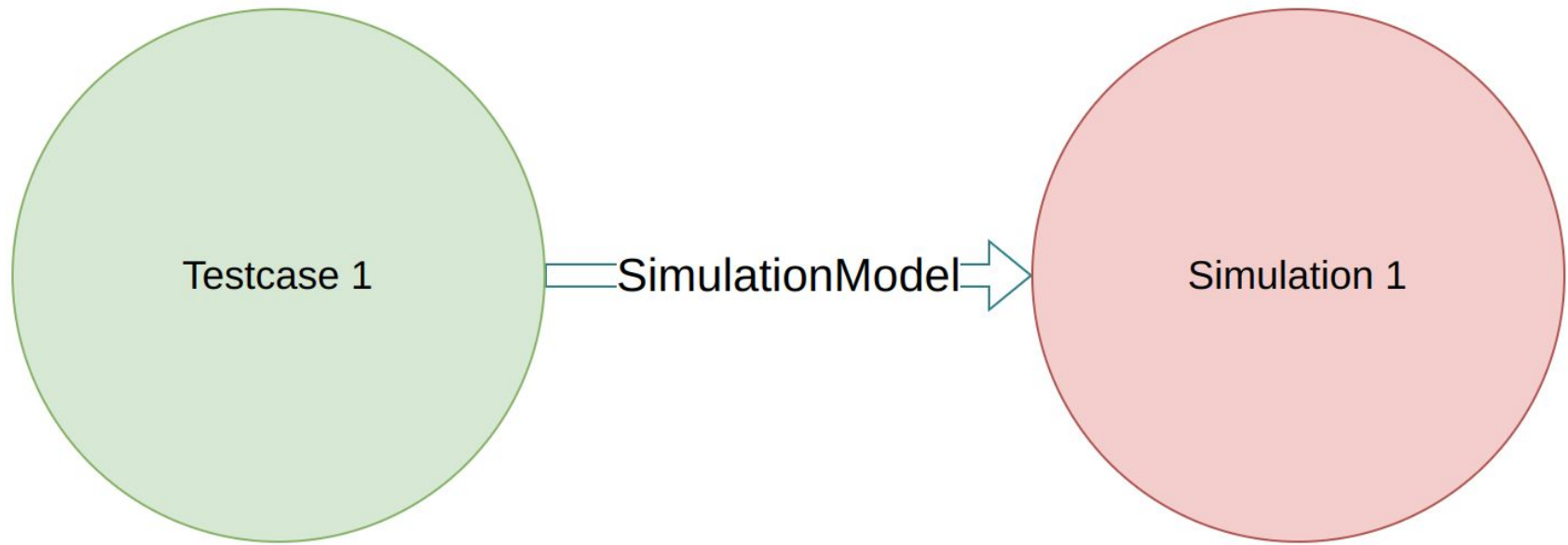
# Kiwi proxy



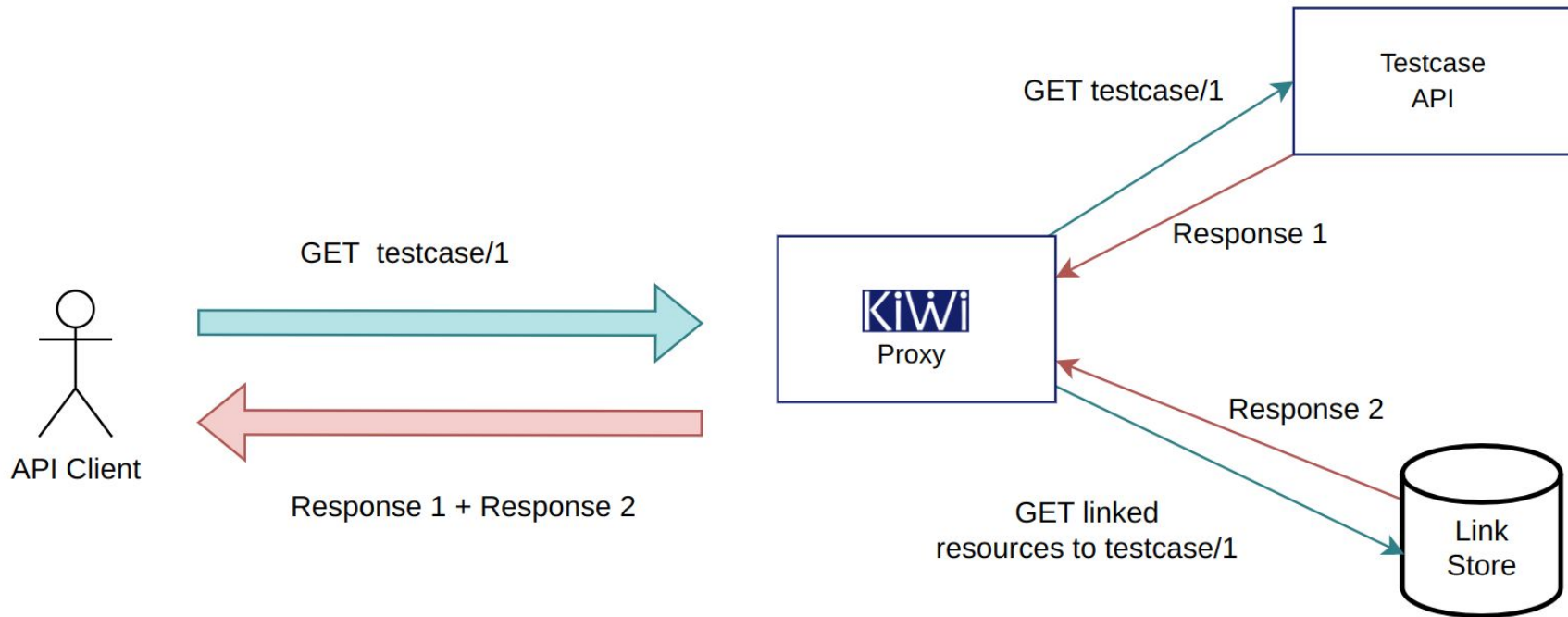


# Kiwi proxy - Link Creation





# Kiwi proxy - Link Discovery



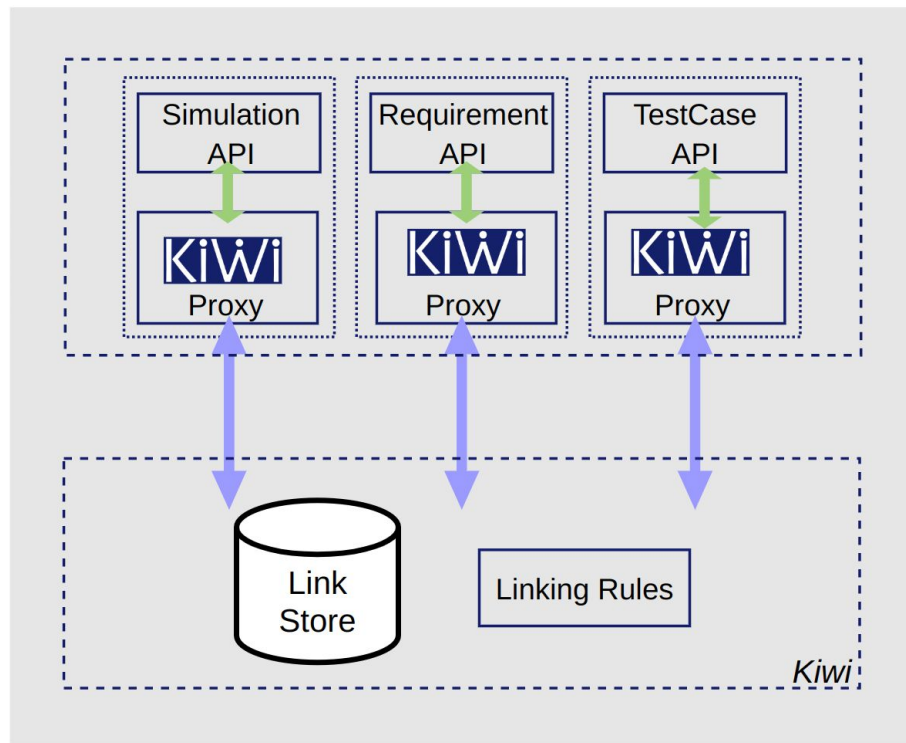
# **Change in Requirement**

# Limitations of Connecting REST API resources

- REST APIs conform to specific schemas not allowing the addition of new links to existing resources.
- Questions to answer:
  - Can we have centrally defined linking rules to avoid chaos?
  - Can we have a centralized link store?
  - How to achieve centralized configuration management?

# Kiwi Control Plane

```
{
  "app:requirement-api": [
    {
      "link_type": "validatedBy",
      "link_target": "app:testcase-api",
      "multiplicity": "N"
    }
  ],
  "app:testcase-api": [
    {
      "link_type": "simulationModel",
      "link_target": "app:simulation-api",
      "multiplicity": "N"
    }
  ]
}
```



# Kiwi

- Kiwi is an application that allows linking **resources** from different **REST APIs**, without any modification to the REST APIs.
- Inspired by **Istio's service mesh architecture**, kiwi is brought to life by **Kubernetes**.

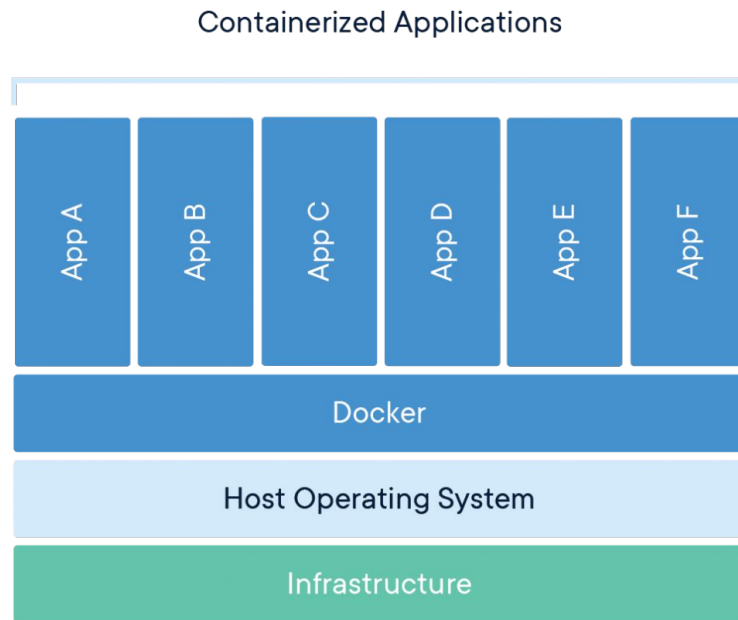


# Kubernetes

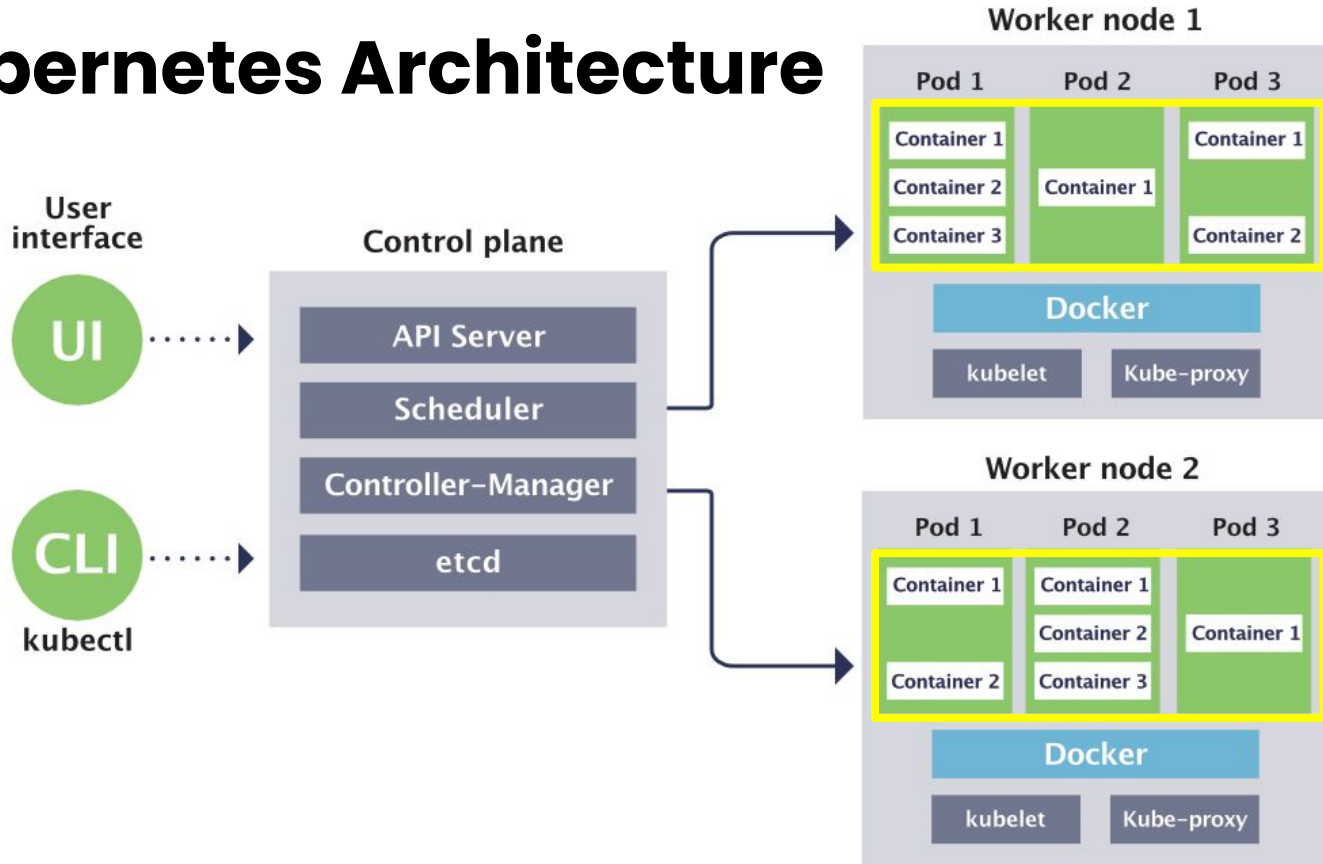
# Kubernetes – platform to deploy containerized applications.

## What are containerized applications?

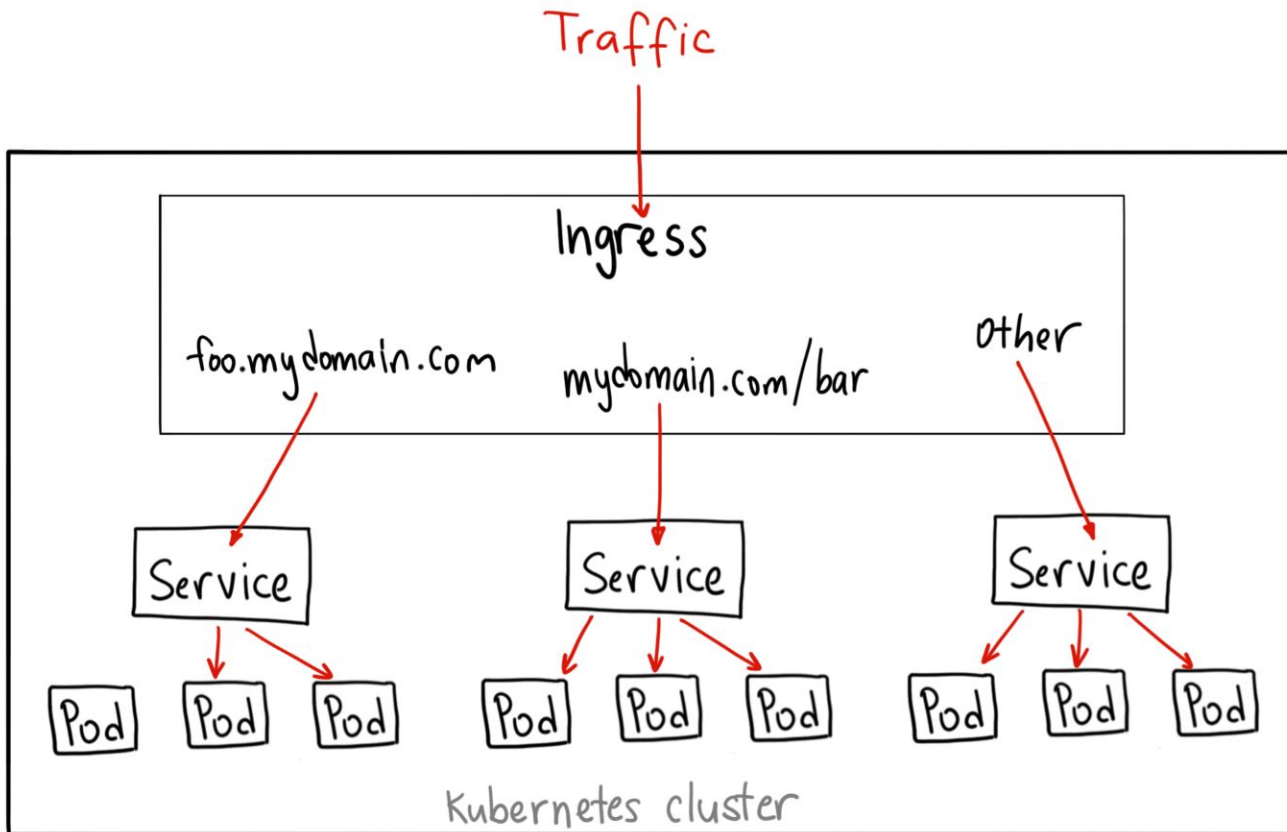
“ A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another ”



# Kubernetes Architecture



# Exposing containers outside their pods.



# Managing Services

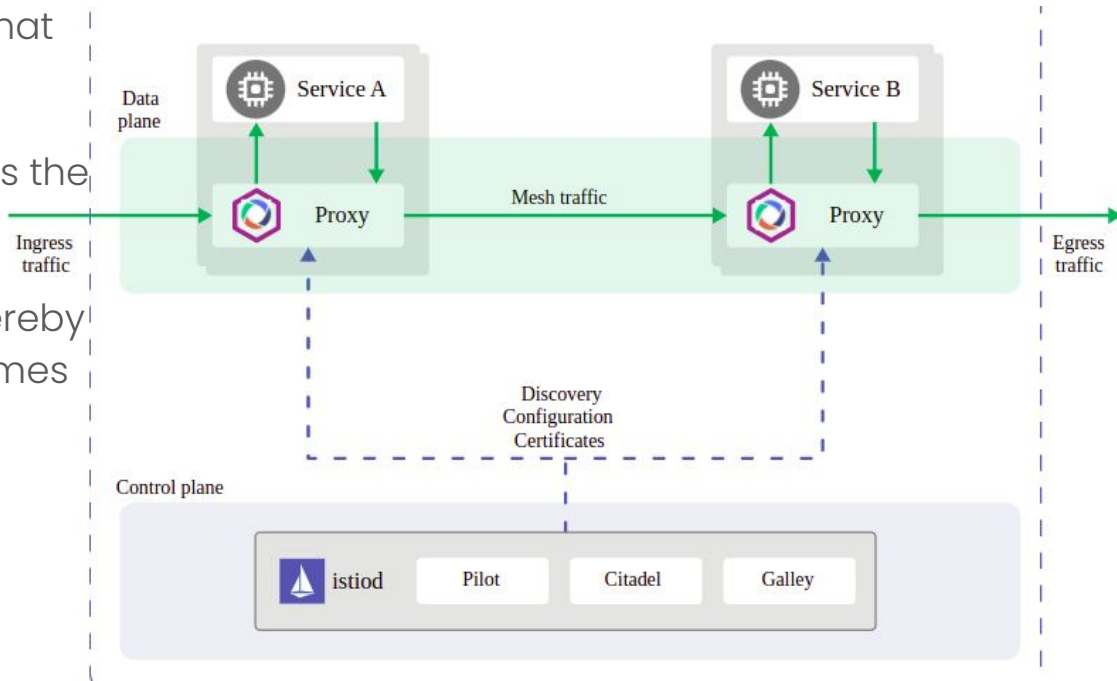
## Istio

# Istio Architecture

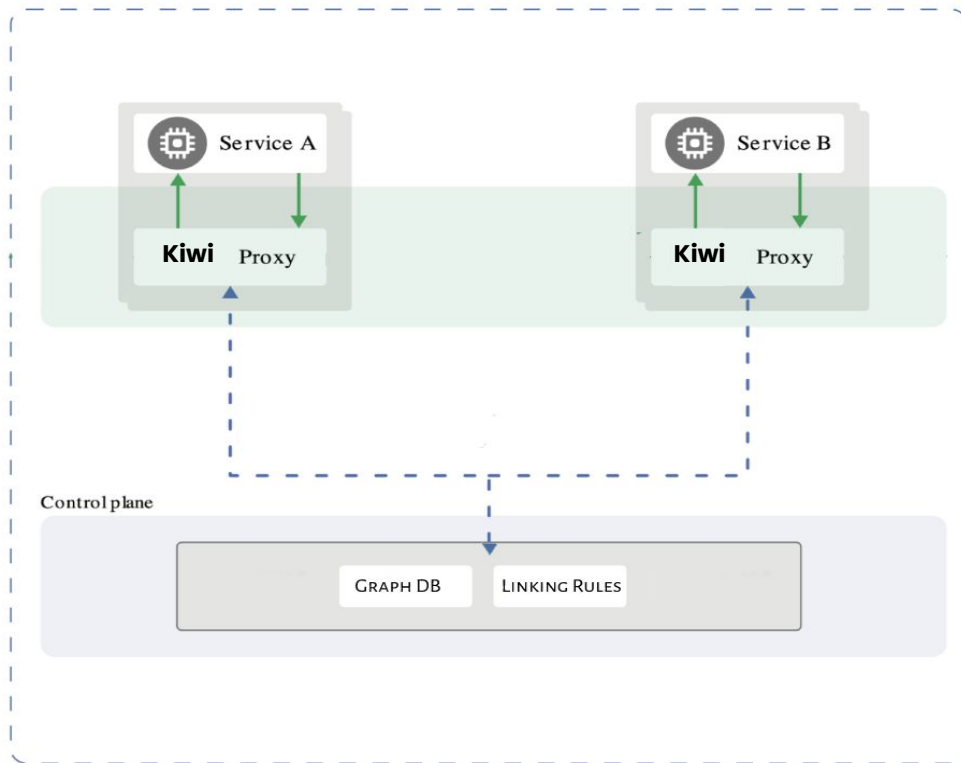
Data Plane: Envoy Proxies for services that intercept network traffic.

Control Plane: Configures and Manages the envoy proxies.

Traffic management and routing is thereby taken over by envoy proxies and becomes centralized.



# Kiwi Architecture



# Next Steps

- As kiwi employs the automatic sidecar deployment in Kubernetes, it can only be used for REST APIs hosted on a Kubernetes cluster. Explore ways in which Kiwi can be employed outside of Kubernetes.
- **All** REST APIs are hosted on the same Kubernetes cluster. Can we link resource across different clusters?

# Summary

- Kiwi connects resources in different REST APIs **without** any modification to the resources in the REST APIs.
- Kiwi uses sidecar proxies to achieve linking resources.
- Kiwi manages link store and linking configuration centrally via the control plane. It is heavily inspired by Isitio's service mesh architecture.
- Kiwi can only be used if all REST APIs reside on the same kubernetes cluster.

# Thanks!

**[hemlata.sharma@koneksys.com](mailto:hemlata.sharma@koneksys.com)**