Architecture Overview – Networking – Vehicle data processing

1. Vehicle sends measurements to Vehicle API service (provided by Idneo – data preprocessing +Redis DB)
    - transmission is received by antenna
    - Transmission enters data(IPv4/IPv6) network
    - Data is forwarded to the closest Kubernetes Edge node using (i) Anycast IP address, shared by all Edge Nodes, and (ii) nodePort defined in NodePort service
    - Edge Node receives the data, checks the IP table in kube-proxy, and forwards it to local Pod A as the service is configured with `externalTrafficPolicy: Local`
        - If there is no local Pod for Vehicle API service, the packet is dropped
2. Vehicle API service forwards the data to the ML prediction service(Kserve inference)
    - Data is forwarded to Pod B as Kserve inference service is configured with `internalTrafficPolicy: Local`
        - If there is no local Pod for Kserve inference service, the packet is dropped
    - or ML prediction service pod periodically queries local Redis DB in Vehicle API service – as Vehicle API service is also configured with `internalTrafficPolicy: Local`
        - If there is no local Pod for Vehicle API service, the queries/requests are dropped
3. Kserve inference service pushes the data into Prometheus
4. Grafana periodically queries data from Prometheus

Service name	Service type	Note
Vehicle API	NodePort	externalTrafficPolicy: Local internalTrafficPolicy: Local Externally Accessible through Anycast/NodeIP:nodePort
Kserve Inference	ClusterIP	internalTrafficPolicy: Local
Kepler	ClusterIP	
Prometheus	ClusterIP	
Kserve	ClusterIP	
Kubeflow	ClusterIP	GUI accessible through Ingress
Minio Storage	ClusterIP	
Grafana	ClusterIP	GUI accessible through Ingress