다중 클러스터 연결의 혁신 N.E.W.S :

Istio 기반 Gloo Platform으로 손쉽게 관리하는 서비스 메시와 API 게이트웨이

OSC Korea 남궁성환 이사

목차

OSC 소개 Overview

- OSC Motto & OSC Asia Group
  - Opensource(O), Security(S), Content_Delivery(C) 기반을 Motto로 아태지역 중심의 ICT Business Platform Group
- 오픈소스 기반의 전문 Knowledge Service 및 신기술 컨설팅, 설계, 구축, 개발, 운영, 자문 서비스
  - 한국 리눅스 재단(비영리) 활동을 통해, 오픈소스 기반의 다양한 기술 트렌드, 전문성 및 생태계 파트너 보유
  - 시대를 주도하는 다양한 신기술 Knowledge/경험과 폭넓은 글로벌 네트워크를 통해 지속적인 신규 솔루션 및 서비스 확보

오픈소스 기반의 MSA(Microservice)

- 오픈소스와 클라우드 네이티브 기술로, 벤더 Lock-in 과 특정 클라우드 제공업체 (AWS, Azure, etc)의 Lock-in 방지 및 애자일한 CI/CD, DevOps, MSA 설계, 개발, 구축, 유지보수, 자문 서비스

글로벌 솔루션 인큐베이션

- 오픈소스 기반의 글로벌 신기술 솔루션 /서비스 (SaaS, PaaS, etc)에 대한 한국 지사 대행, 총판, 전략적 파트너십

클라우드&컨테이너 매니지드 서비스 (OpenMSP)

- Cloud Native MSP: 주요 컨테이너 플랫폼(FlyingCube, ZCP, TACO)기반 구축 시스템 대상 쿠버네티스 유지보수 서비스

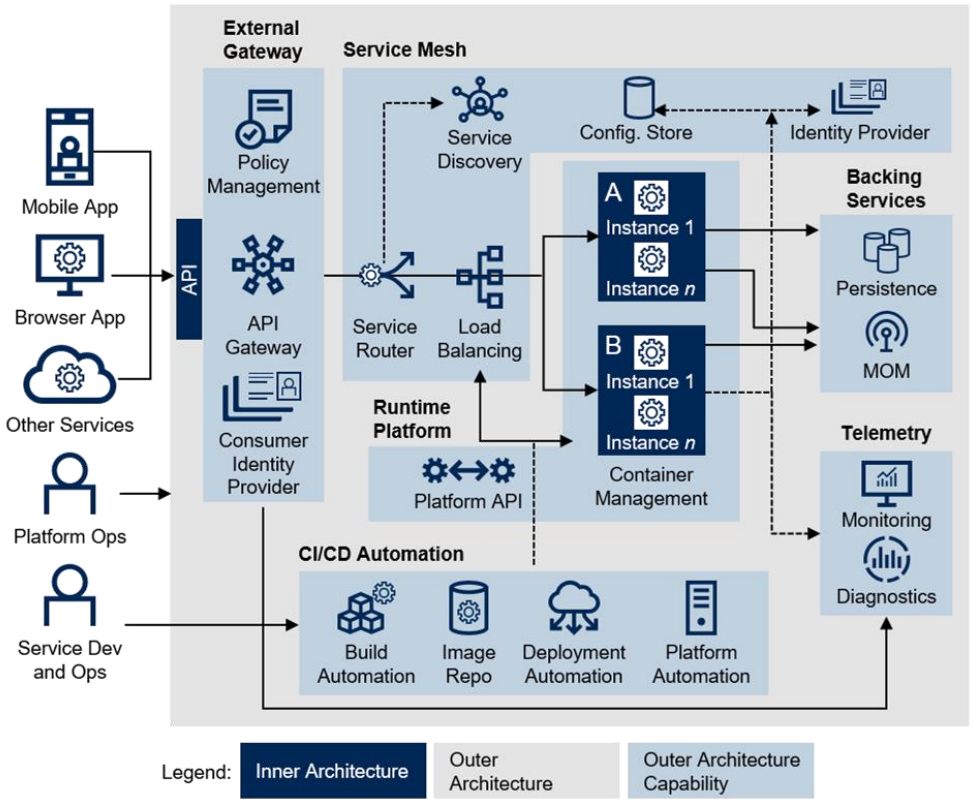
오픈소스 공인교육

- 리눅스 재단 공인 교육파트너 : 쿠버네티스 중심의 어드민, 개발자 온사이트 교육 제공
  - CKA: Certified Kubernetes Administrator 시험 연계 과정
  - CKAD: Certified Kubernetes Application Developer 시험 연계 과정
  - CKS : Certified Kubernetes Security Specialist 시험 연계 과정

100% Opensource기반의 OpenMSA™ - 계속

- OpenMSA™ 소개
  - OpenMSA™는 MSA 공통 서비스와 Outer Architecture(DevOps, GitOps, FinOps* 등)에 대해 순수 오픈소스로 구성한 표준 MSA 프레임워크로, 산업별/도메인별 최적의 MSA 환경 제공

Microservice Architecture 구성 가이드 by Gartner



ID: 353896

100% 오픈소스 기반으로 구체화한 서비스 프레임워크

애자일한, 맞춤형 MSA 환경 구성 OpenMSA

OpenMSA™ Outer Architecture 구성

Solo.io 소개 Overview

2017 설립
설립자 Idit Levine(이딧 르빈)

본사: 매사추세츠 캠브리지
글로벌 지사 운영중

어플리케이션 네트워킹, 서비스 메시, 모던 API
게이트웨이 기술 리더로 구성

Open-Core, “엔터프라이즈” 구독 모델

Growing fast with happy customers

500+%
bookings growth y/y

98%+
renewal rate

Well Funded

$171.5M
venture financing

$1 Billion
valuation

Altimeter Redpoint true Ventures

산업계가 인정한 리더

Gartner COOL VENDOR

SD Times Open-Source Project of the Week

TechCrunch Solo.io reaches $1B valuation

주요 고객사

SAP, Macquarie, American Express, T-Mobile, Informatica, Constant Contact, USAA, Chick-fil-A, snyk, Booking.com, ADP, Adobe, Schneider Electric, Delta Dental, CARFAX, BMW Group, Vonage, ING

Gloo Plaform

Gloo Plaform

대표적으로 Gloo Gateway, Gloo Mesh, Gloo Network 제품을 통해 다양한 플랫폼 관리 및 모니터링 도구 제공

GlooGateway

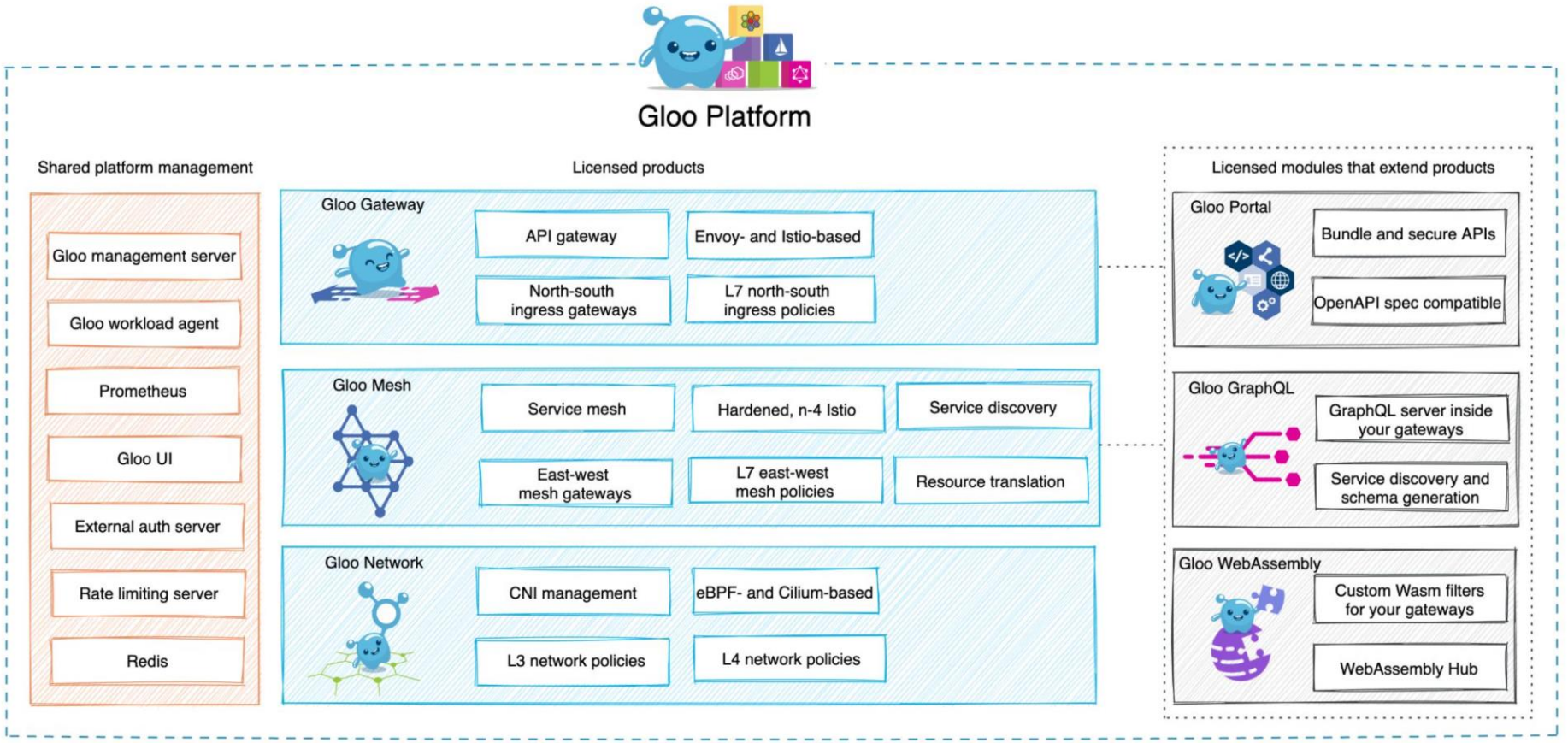
API gateway와 Ingress 기능 모두 제공

GlooMesh

서비스 메시 기능 및 관리 제공

GlooNetwork

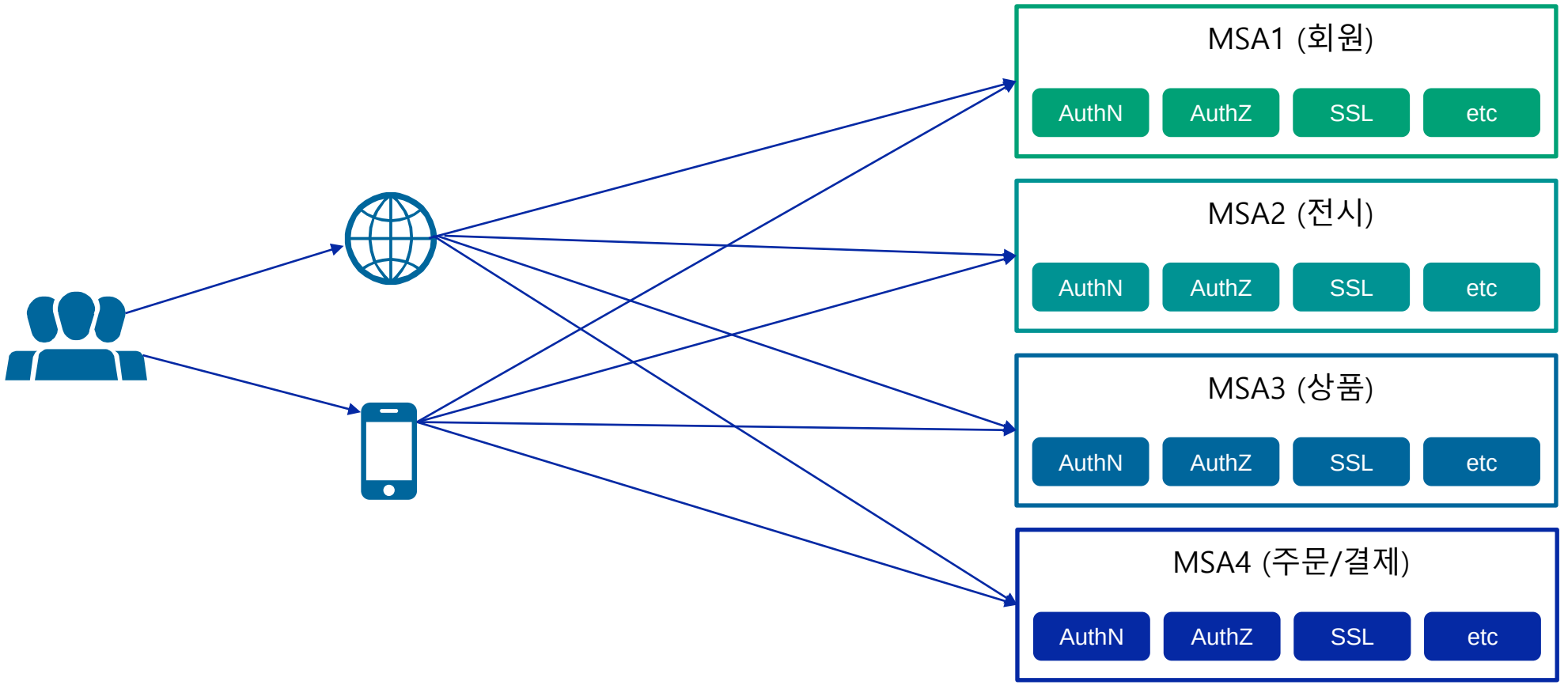
Kubernetes 클러스터를 위한 강력한 Cilium CNI 제공



Gloo Gateway Journey

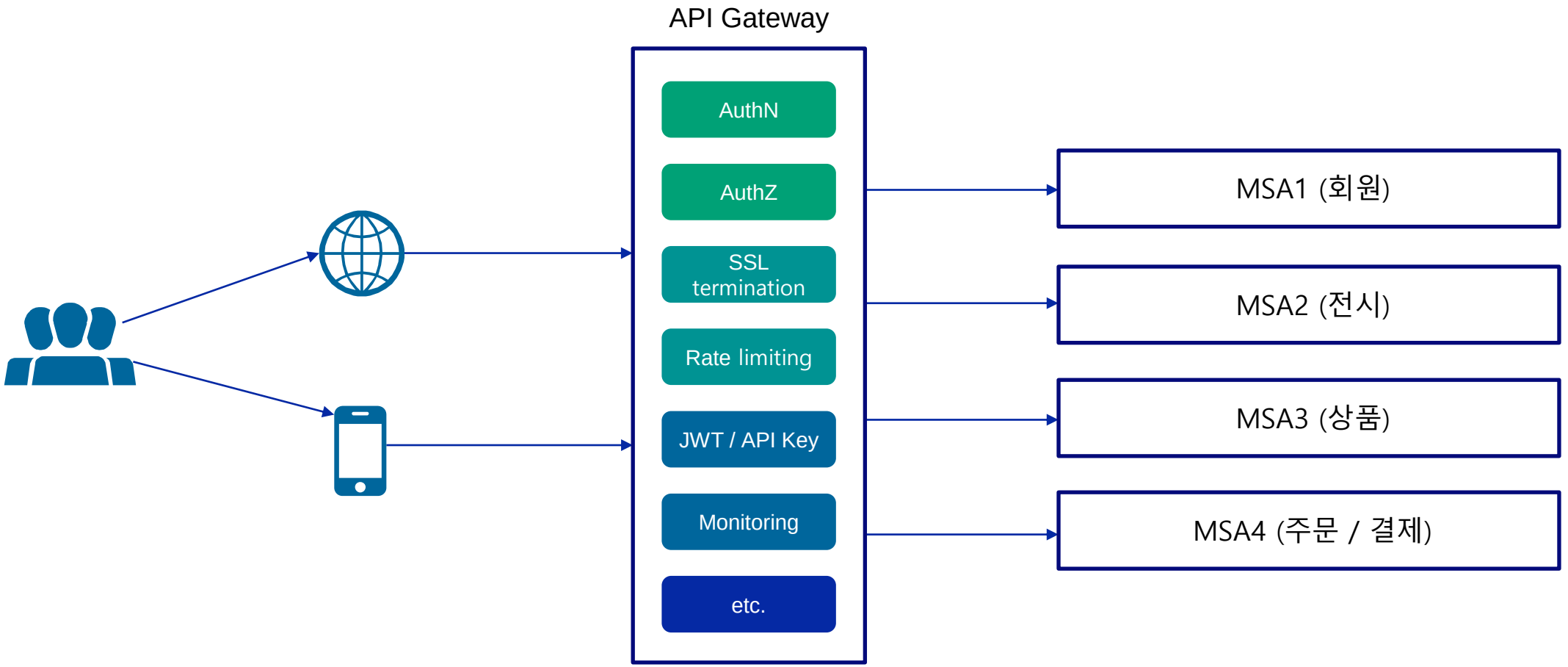
API Gateway는 왜 중요할까? (1/3)

MSA 서비스들에는 모두 각각 여러 보안 모듈 구현이 필수적으로 필요함



API Gateway는 왜 중요할까? (2/3)

MSA 서비스들에서 구현할 보안 모듈들을 API Gateway에서 대신 구현 및 적용 가능



API Gateway는 왜 중요할까? (3/3)

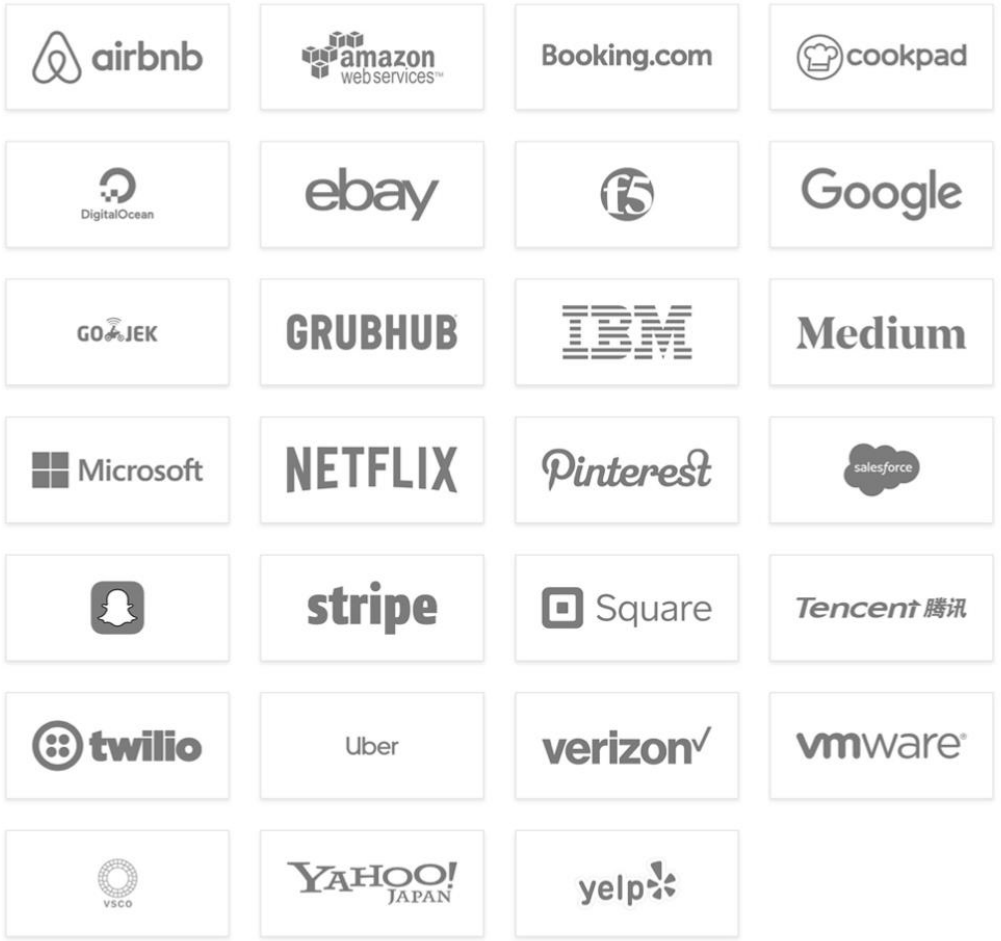
MSA 서비스 환경에서 API Gateway 구성으로 다양한 혜택 존재

혜택	설명
Non-Business Logic은 gateway 대신 구현	- MSA에 구현할 보안 적용을 API Gateway에서 대신 구현 및 적용 - 모든 MSA 대한 요청은 API gateway 보안 장벽을 통과해야 함
클라이언트 변경없이 다양한 Protocol 연동 가능	- HTTP , HTTS , SOAP 등 서로 다른 Protocol 서비스에 대한 클라이언트 변경없이 연동 가능 - 인터넷 친화적인 Protocol를 지원하지 않는 서비스에 엑세스 가능
향상된 클라이언트 성능	- 클라이언트는 API gateway에 하나의 요청으로 multi MSA 서비스의 정보를 묶어서 클라이언트로 전송 - 클라이언트의 수행하는 요청 수를 크게 줄이고 요청 대기시간과 UI 복잡성을 줄임
외부 공격으로부터 MSA 보호	- API gateway 는 전체 MSA Architecture에서 동일한 보안 조치 시행토록 보안 장벽 역할 - Ex) 모든 클라이언트에 대한 인증 및 권한 부여를 시행하고 HTTPS 연결만 허용
라우팅 및 로드 밸런싱	- API gateway는 모든 MSA 라우터 역할 - Ex) 로드 밸런싱, 특정 MSA로 라우팅 , Mirroring, A/B 테스트 등
MSA 요청의 모니터링	- 모든 클라이언트 요청이 API 게이트웨이로 전송되기 때문에 이를 소스로 사용하여 들어오는 요청을 모니터링 - Ex) 요청 수, 성공 또는 실패한 요청 수, 요청 대기 시간 , MSA 간의 의 요청 경로 모니터링
MSA 응답 캐싱으로 응답시간 개선	- API Gateway 에서 MSA의 응답을 캐시 하도록 하여 요청 대시 시간과 응답 시간을 크게 개선

API Infrastructure 환경에서의 Envoy Proxy 역할

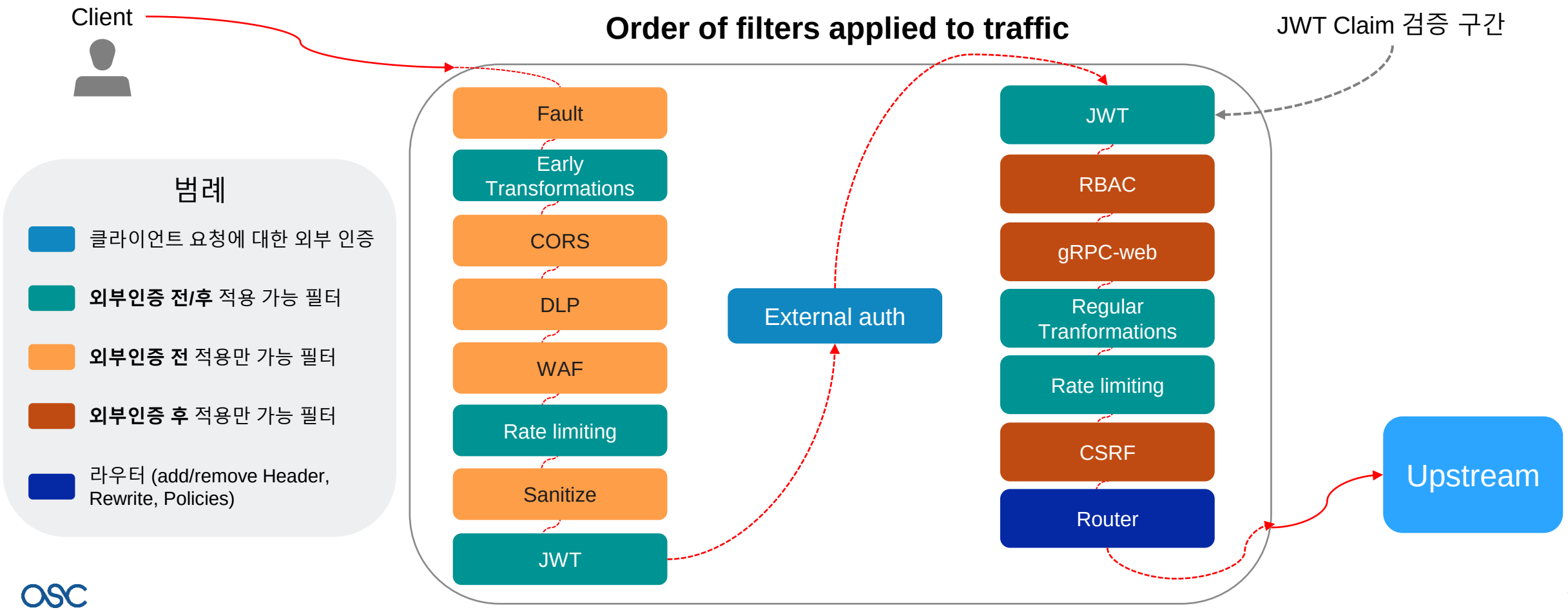
Envoy Proxy 점점 Cloud 및 Infrastructure 각광받고 있으며 Gloo Gateway는 Envoy Proxy 기반의 아키텍처 설계화

- 중립적 재단인 CNCF의 프로젝트
- 대규모 활발한 커뮤니티 생태계 형성
- API 기반의 동적 구성
- 높은 확장성
- 다양한 L7 filters 지원
  (HTTP/1, HTTP/2, gRPC, redis, mysql, Kafka, etc)
- 다양한 배포 옵션 지원



API Gateway Filter Chain 아키텍처

API Gateway에는 다양한 필터들이 필요하며 또한 순차적인 체인 연결로 구성 되어야 함
Gloo API Gateway는 이러한 Filter Chain 아키텍처로 구성되어 있음



Gloo Gateway와 Istio Ingress Gateway의 비교

Gloo Gateway는 Istio Ingress 기능 뿐만 아니라 API Gateway 기능을 함께 제공

Feature	Gloo Gateway	Istio Ingress
CORS(Cross-origin resource sharing) 지원	☑	☑
TLS 종료	☑	☑
Header 조작	☑	☑
Retries, redirects, timeouts, fault injection, 이상치 감지	☑	☑
미러링	☑	☑
요청 및 응답 변환	☑	✗
연합(Federation)	☑	✗
고급 rate limiting	☑	✗
WAF 및 DLP를 포함한 고급 보안	☑	✗
OIDC, OPA, API 키 및 LDAP에 대한 고급 외부 인증	☑	✗
요청 및 응답 SOAP 변환	☑	✗
직접 응답 및 경로 위임과 같은 고급 트래픽 라우팅 및 shaping	☑	✗
고급 지역 기반 및 다중 클러스터 라우팅	☑	✗

Gloo Gateway Features

Gloo Gateway Enterprise 제품에는 보다 더 추가적인,유용한 기능을 제공

CONNET

- All Workload Types
- Auto Service Discovery
- HTTP Routing
- TCP Proxy
- gRPC Web
- CORS
- Kubernetes Services
- Consul Service
- Serverless Functions
- Configuration Validation
- Request / Response Transformation
- Service Mesh integration

ENTERPRISE

- Developer Portal

SECURE

- TLS
- Hashicorp Vault Secrets
- Let’s Encrypt
- Custom Authentication(DIY)

ENTERPRISE

- Data Loss Prevention
- Web App Firewall(WAF)
- Basic Authentication
- API Key
- JSON Web Token(JWT)
- LDAP Support
- OAuth / OIDC
- Open Policy Agent

CONTROL

- Admin Dashboard(Read Only)
- Role Delegation
- Access Logging & Usage Stats
- Prometheus and Grafana
- Tracing
- Circuit Breaking
- Retries
- Timeouts
- Traffic Shifting
- Traffic Shadowing
- Rate Limiting(DIY)

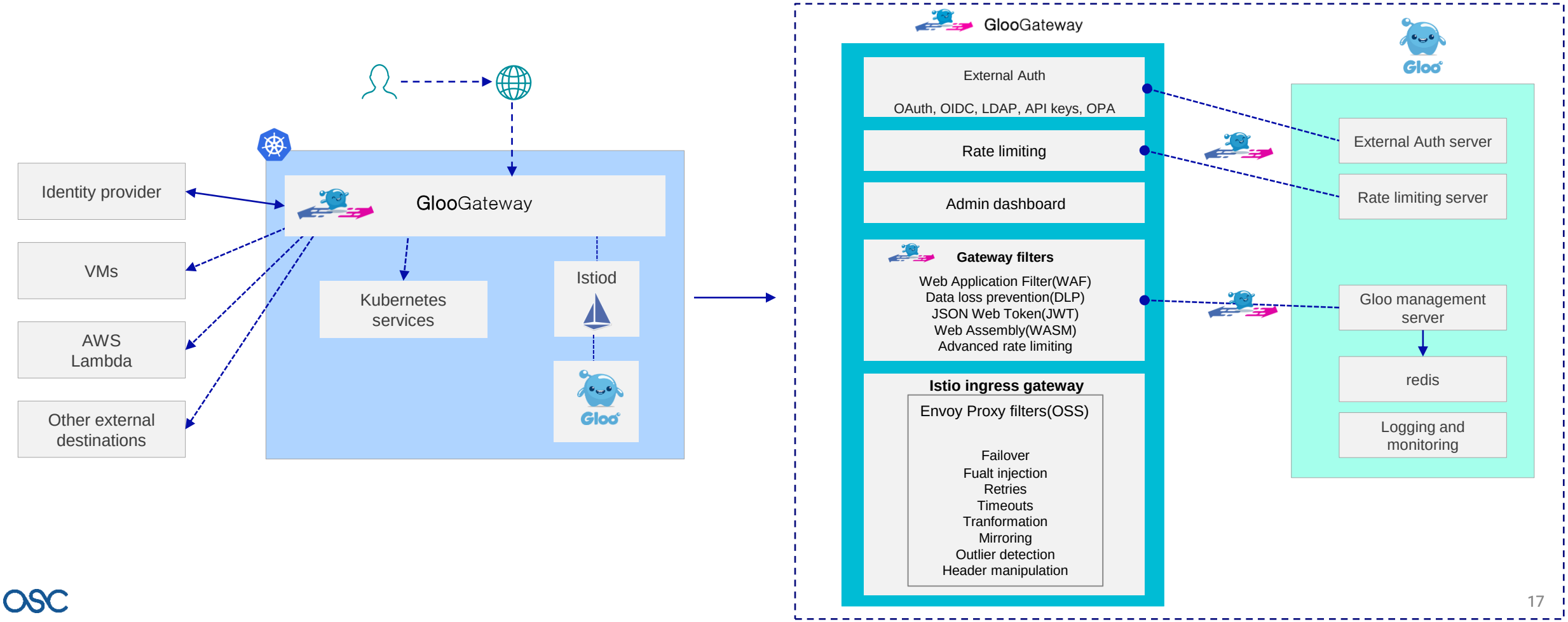
ENTERPRISE

- Admin Dashboard(Full Access)
- Advanced Rate Limiting

Gloo Gateway 아키텍처

- Gloo Gateway 수신 요청을 Kubernetes 클러스터의 서비스 또는 외부 서비스로 라우팅 구성 가능
- Gloo 플렛폼 아키텍처를 기반으로 다양한 트래픽 정책 잠금을 해제하여

클러스터에 들어오는 요청을 보호, 제어 및 모니터링 함

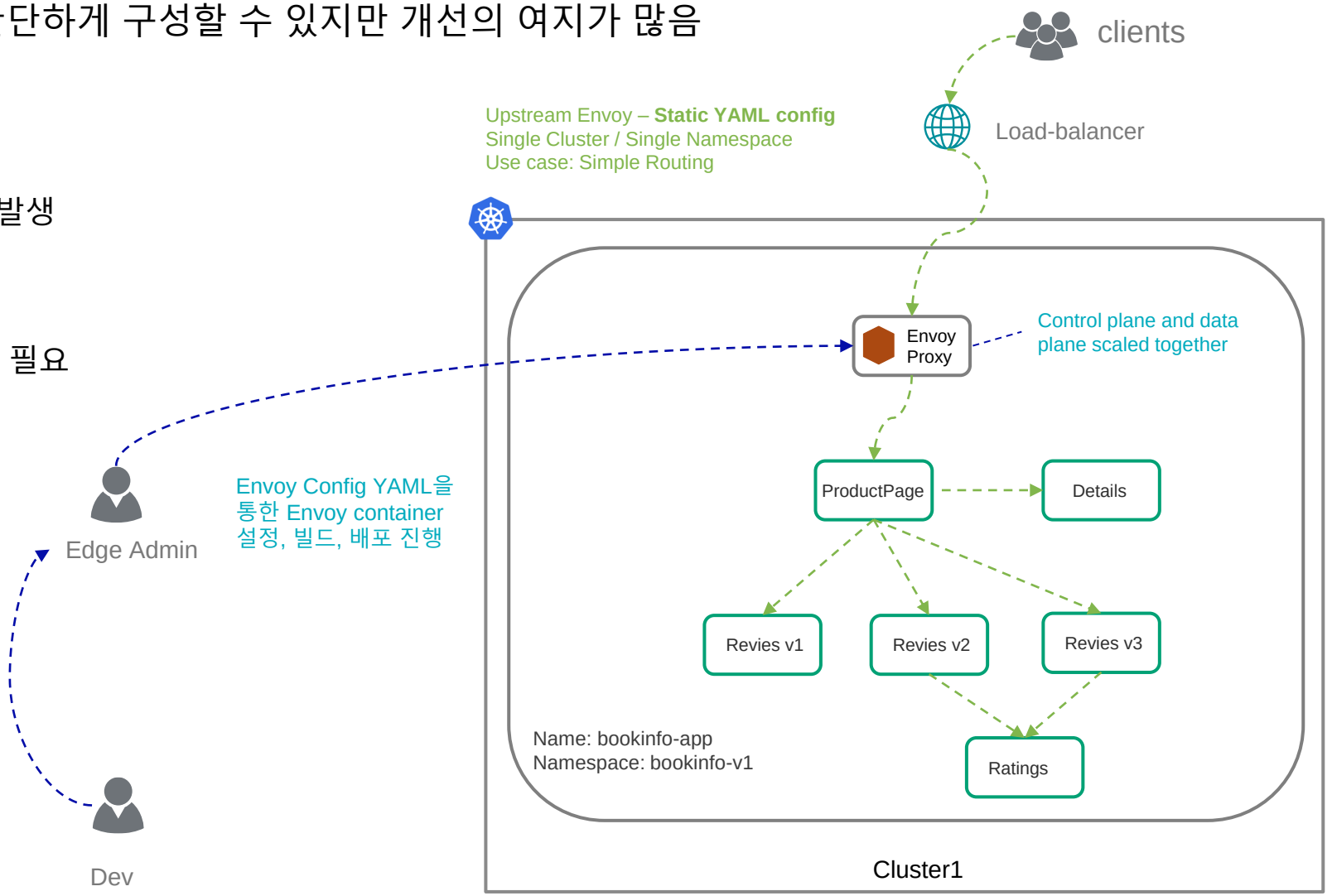


K8S 환경에서 Envoy Proxy로만 구성하는 경우

Envoy Proxy로만 구성하는 경우 간단하게 구성할 수 있지만 개선의 여지가 많음

단점

- 수동 구성으로 인한 조직 병목 현상 발생
- 확장성의 비효율
- 어플리케이션(App) 마다 Proxy 구성 필요

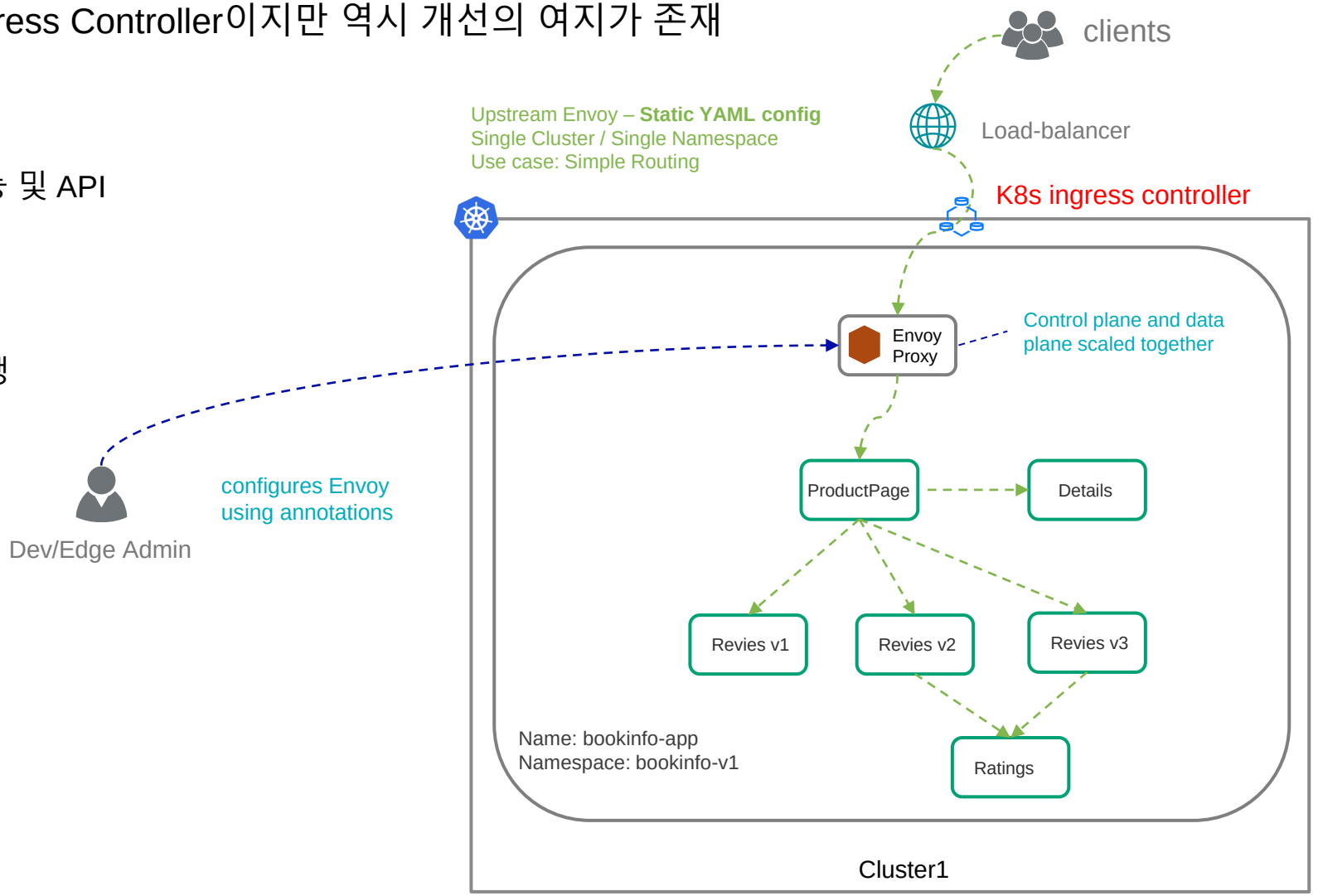


K8S의 Ingress Controller 적용(1/2)

일반적으로 가장 많이 구성하는 Ingress Controller이지만 역시 개선의 여지가 존재

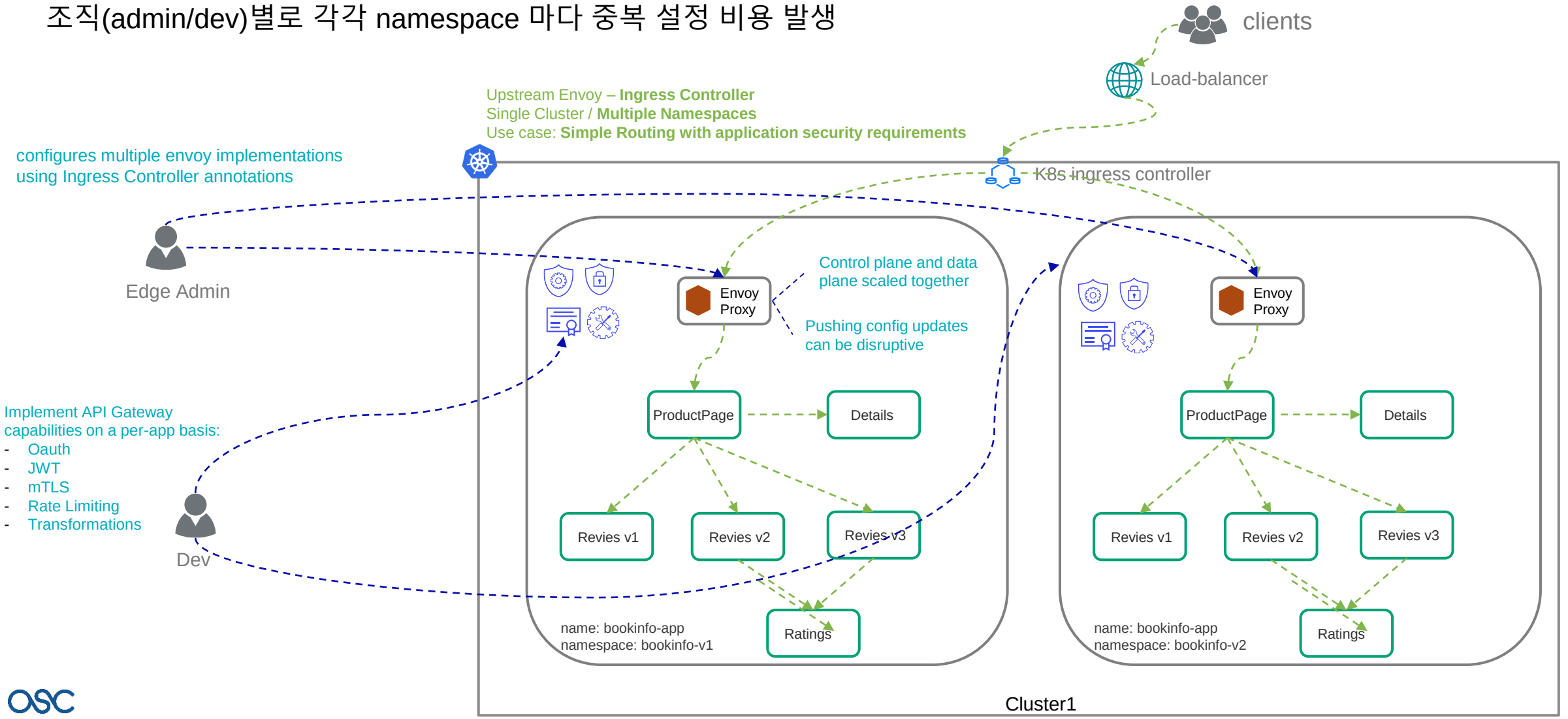
단점

- Ingress API 기능의 한계성(추가 기능 및 API Gateway 기능의 부재)
- 확장성의 비효율
- 개발 또는 운영팀 간의 중복 비용 발생



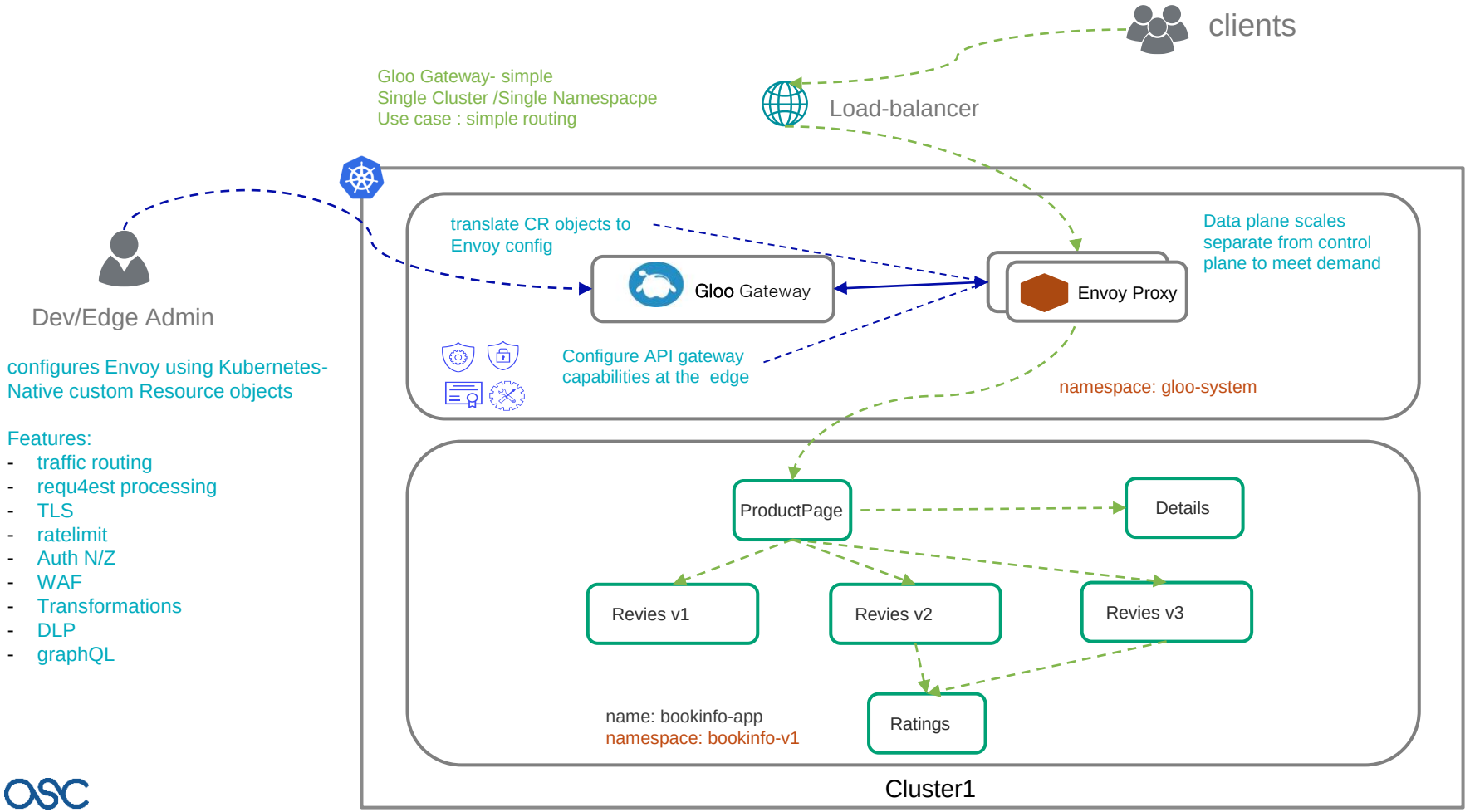
Kubernetes Ingress Controller 적용(2/2)

클러스터내의 multi namespace 요구 시 각 namespace 마다 중복 설정 발생
조직(admin/dev)별로 각각 namespace 마다 중복 설정 비용 발생



Gloo Gateway 구성으로 해결

- Gloo Gateway 설치만으로 API Gateway + Kubernetes Ingress 기능이 모두 제공되어 설치 및 관리의 효율성이 높음
- 컨트롤 plane과 데이터 plane 분리하여 효율적인 확장이 가능해지고 복잡성을 추상화한 간소화된 API를 제공함

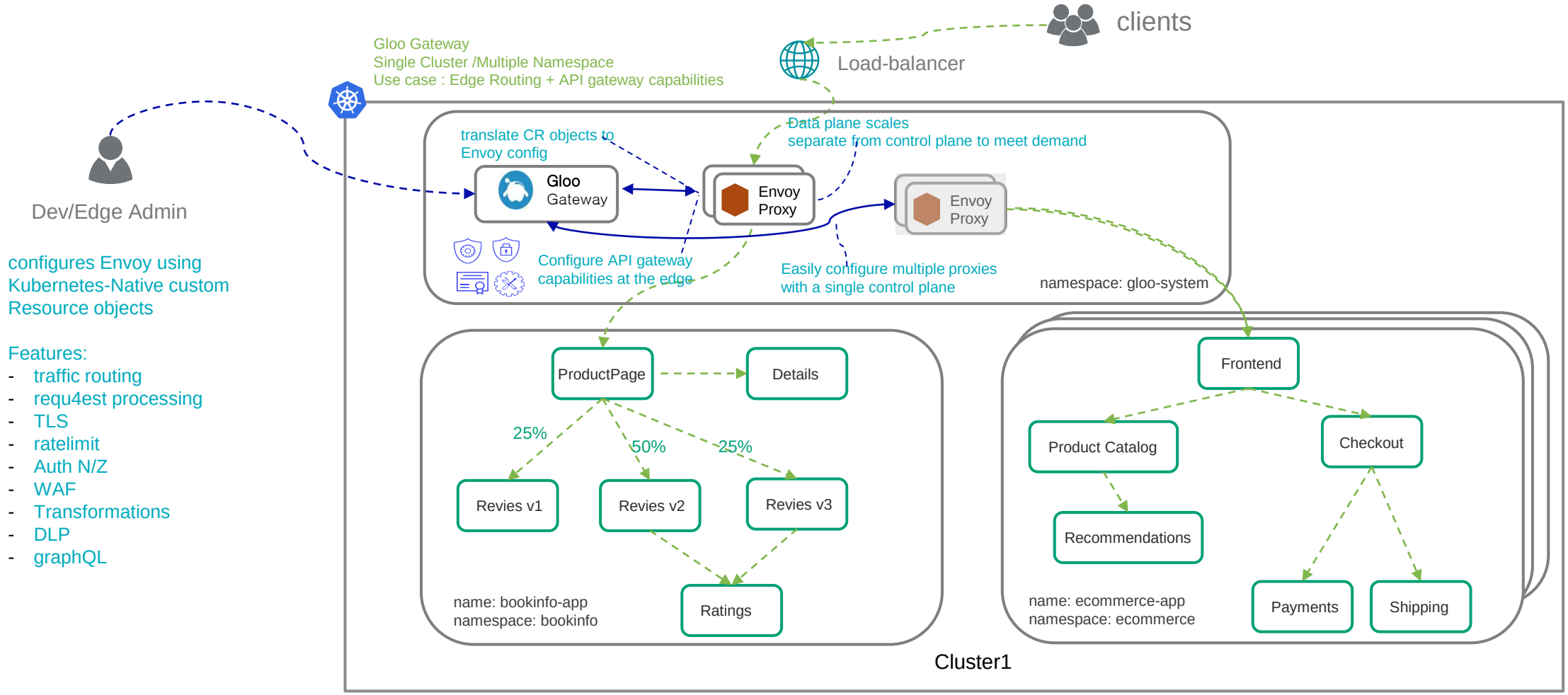


심플한 탈중앙화 API 아키텍처

- 최신 Cloud Native workloads(Envoy)기반 설계
- 효율적 확장을 고려한 컨트롤 plane과 데이터 plane 분리
- GitOps 워크플로에 쉽게 맞출 수 있는 코드로 구성
- 복잡성을 추상화한 간소화된 API
- 다양한 API Gateway Edge 기능 지원 (authN/Z, rate-limiting, WAF, DLP, Transformations, and more)

Gloo Gateway의 효율적인 확장을 고려한 설계

- Gloo Gateway는 Multi namespace 환경의 확장이 대단히 용이함
- 조직(admin/Dev)의 운영 설정도 각 네임스페이스에 설정 없이 Control plane에만 적용으로 가능



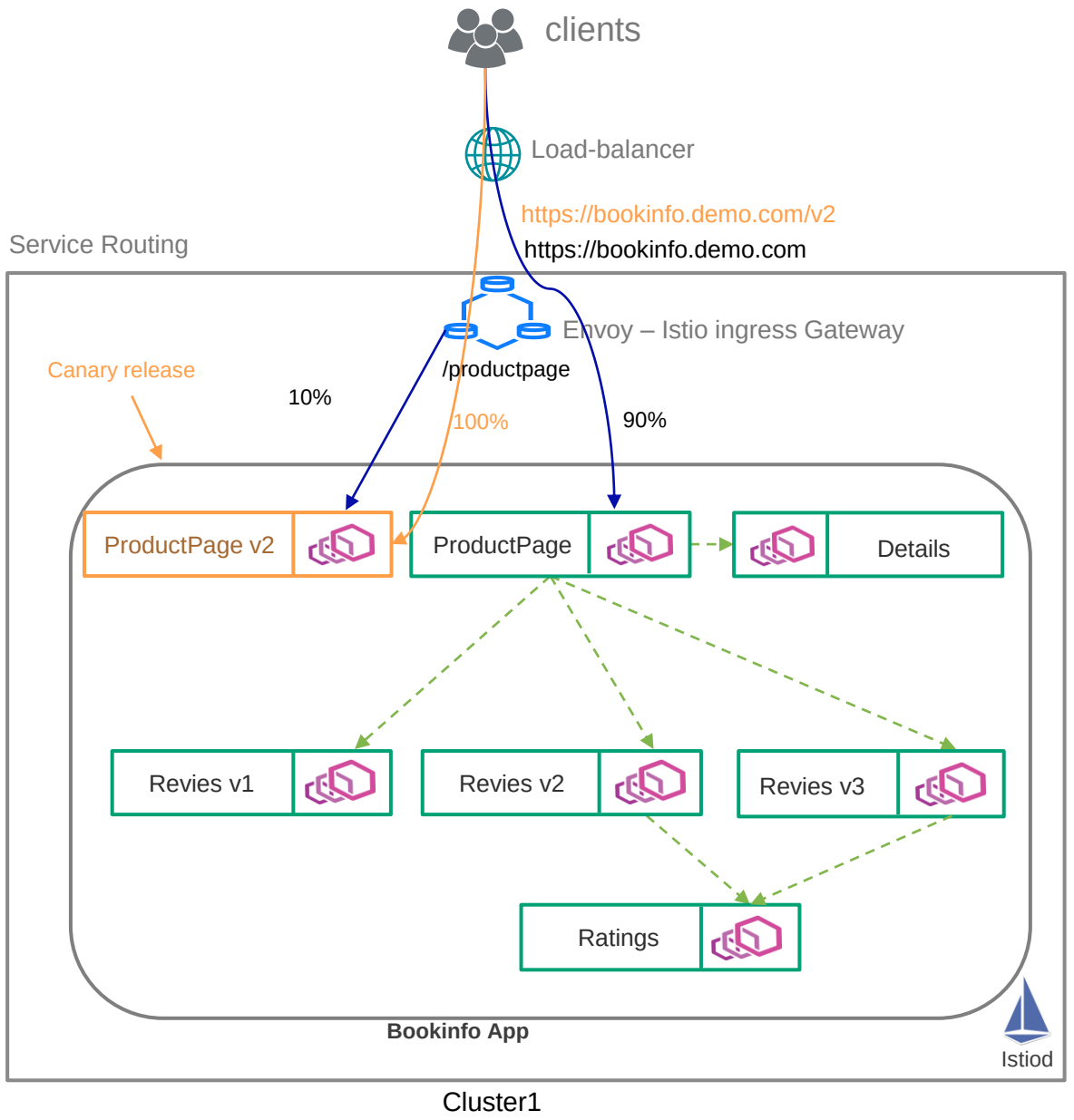
GlooGateway

API Gateway 기능

서비스 Routing

Traffic Manager 차원의 다양한 기능의 서비스 라우팅 제공

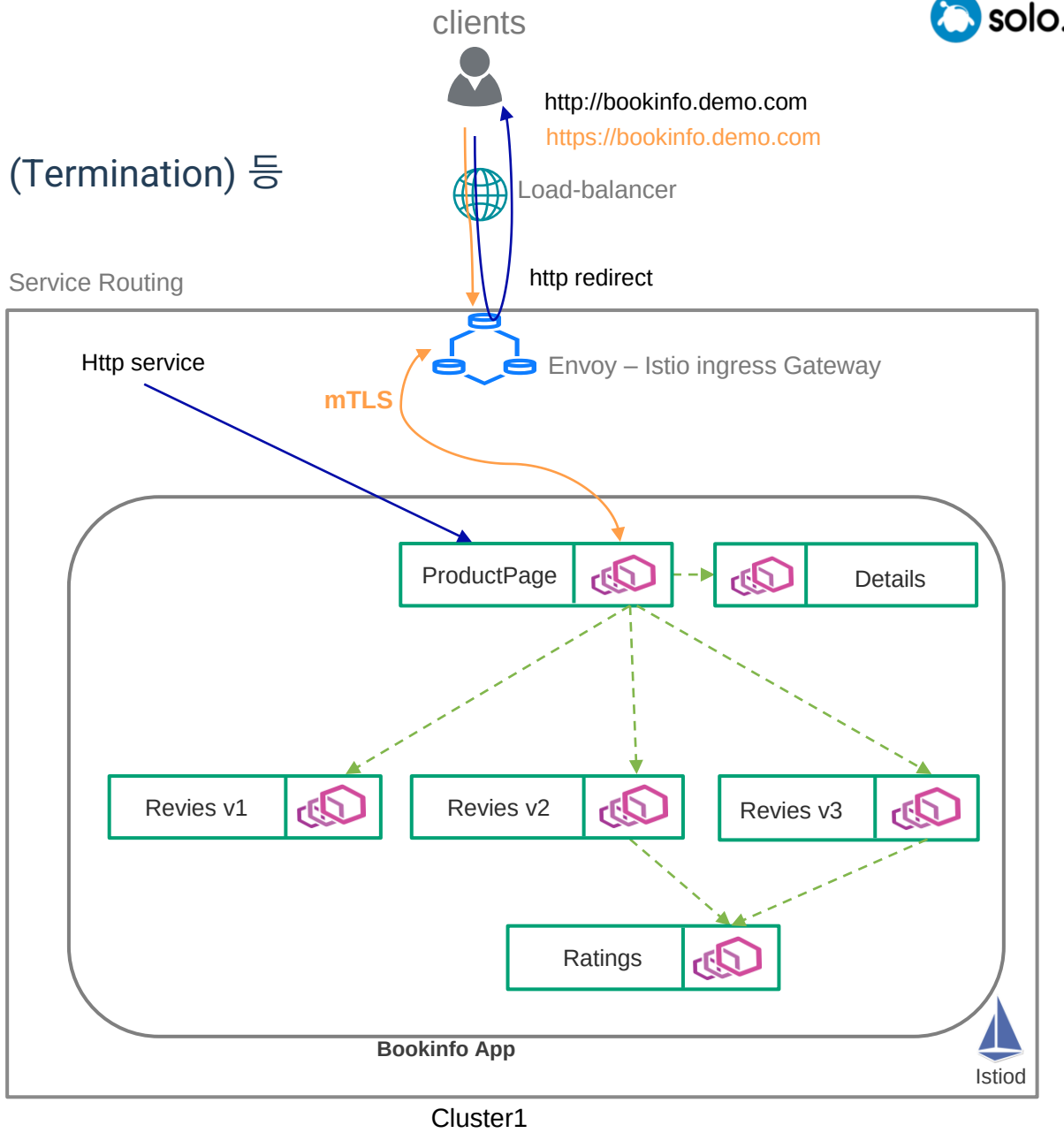
- 다양한 Path 기반의 라우팅 지원
  (exact, prefix, regex matching)
- 헤더 정보 기반의 라우팅 지원
- HTTP 메서드를 기반 Request의 Path 매핑 지원
- Weighted destinations
- Non-Mesh 서비스로 라우팅 지원
  (ex. cloud functions, VM, kubernetes service, etc.)
- Route Delegation 지원



TLS Offloading

클러스터 내부 또는 외부의 TLS암호화 전송 또는 종료 전송 (Termination) 등 다양한 기능 제공

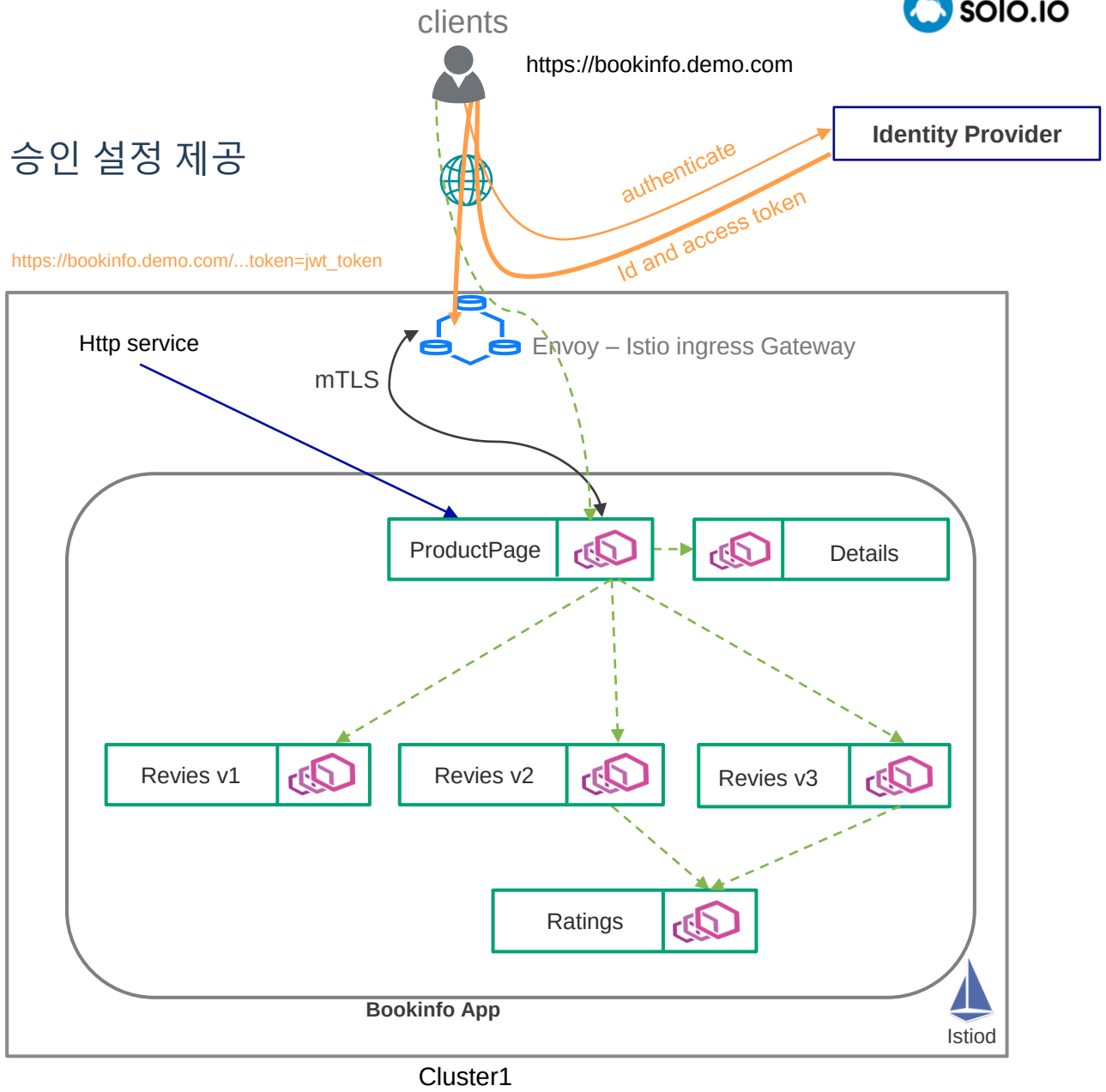
- Downstream(client 요청) TLS offloading 지원
- Upstream의 mTLS end-to-end 암호화, 인증/인가 지원
- 해당되는 경우 Downstream mTLS 지원
- URL별 보안정보 지원



ExtAuth

클러스터의 워크로드를 보호하기 위한 다양한 외부 인증 및 승인 설정 제공

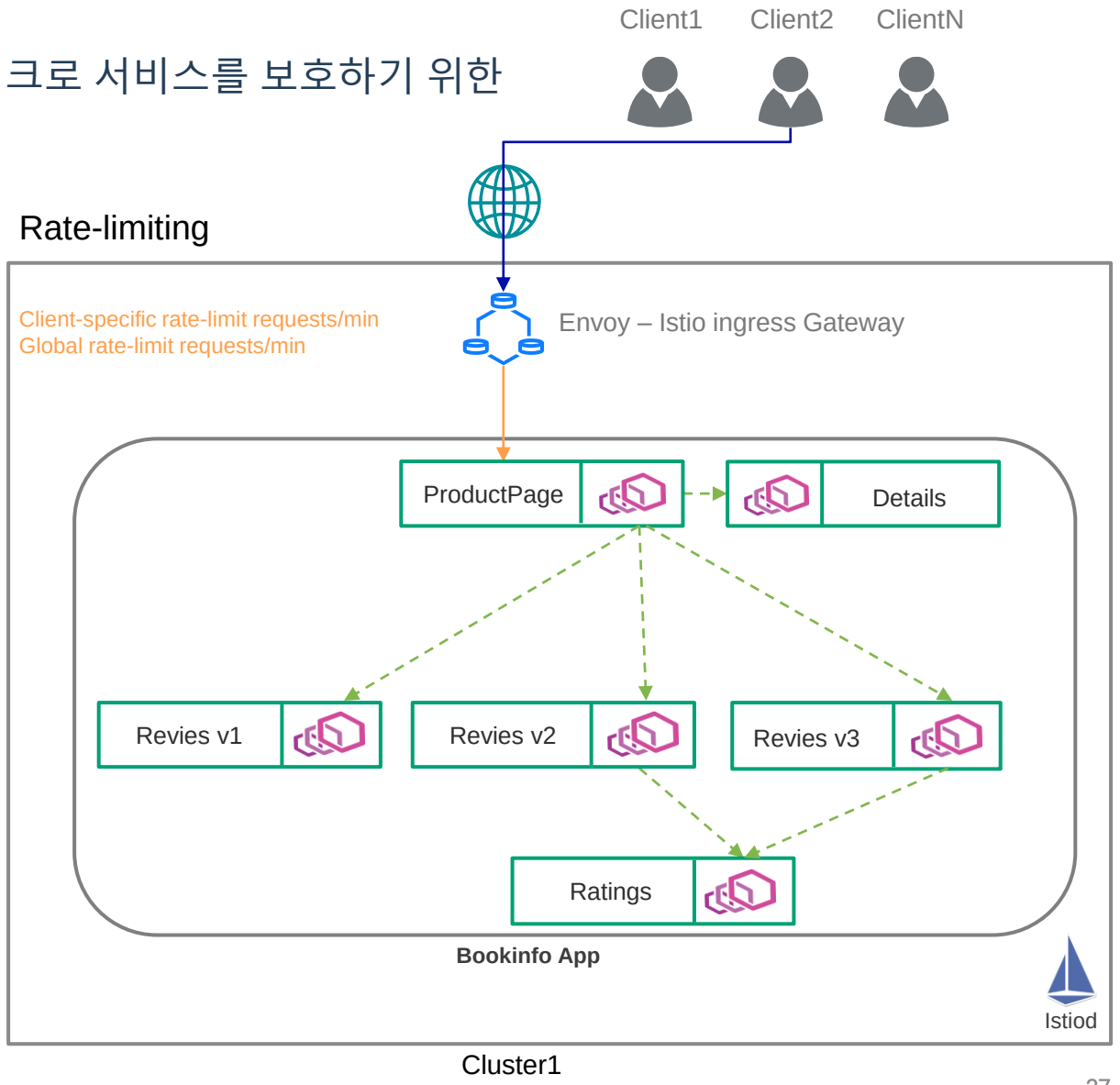
- Basic 인증 : usernames / passwords 지원
- API key : API key 기반의 인증 요청 지원
- Oauth : Oauth 2.0 기반 access token 의 외부 ID 공급자 OIDC (OpenID Connect) 지원
- LDAP : LDAP에 저장된 회원 정보에 대한 인증 지원
- OPA : 보다 세분화된 액세스 제어를 위한 OPA(Open Policy Agent) 정책 지원
- Passthrough : 외부 gRPC 서비스로 인증 요청 지원



Rate-limiting (속도 제한)

외부 클라이언트 요청(North-South)으로부터 back-end 마이크로 서비스를 보호하기 위한 다양한 속도 제한 설정 기능 제공

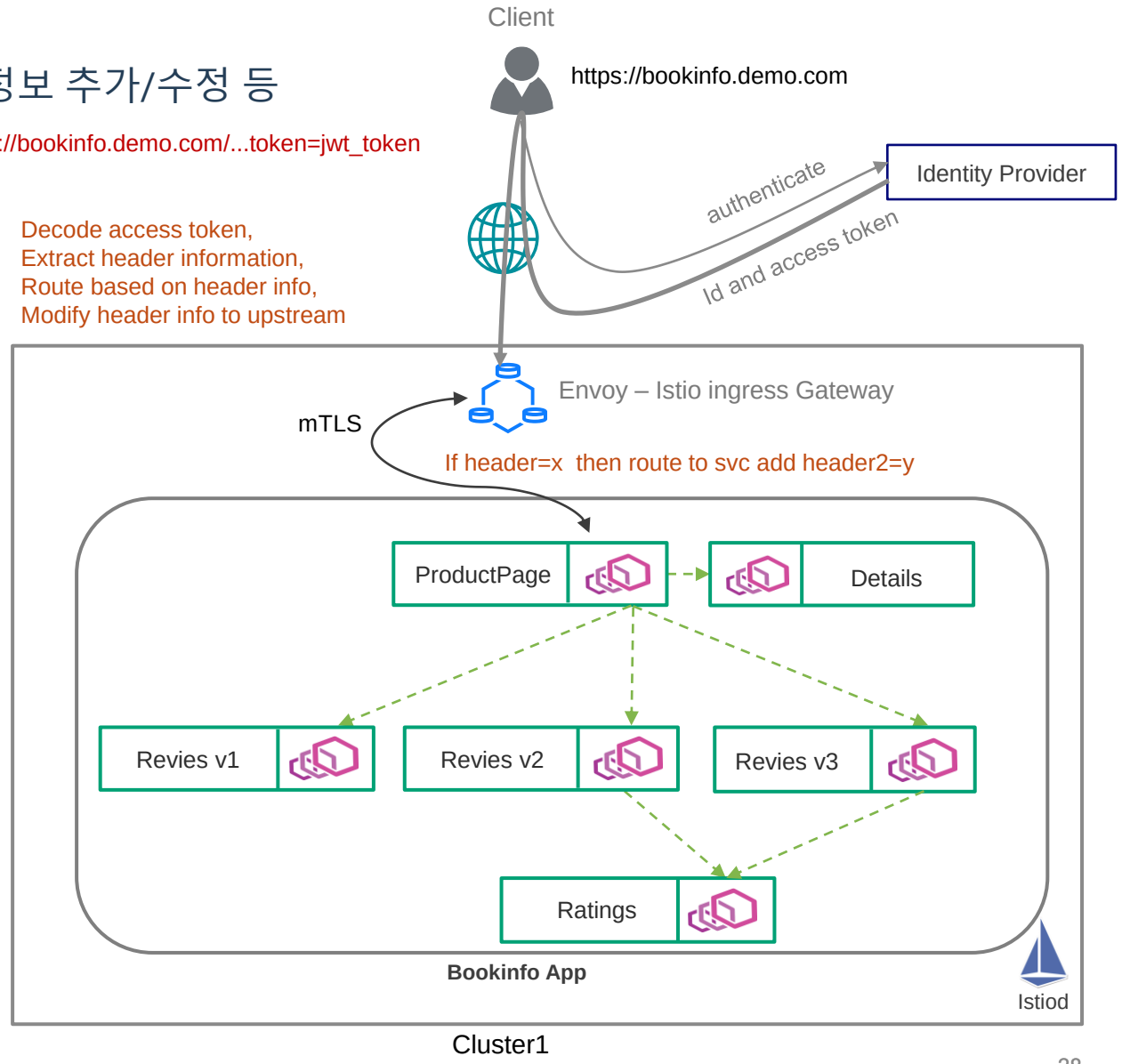
- 요청당 Rate-limiting Parameters 설정 지원
- Application별 global rate-limiting parameters 설정 지원
- 악성 트래픽 속도 제안 설정 지원
- Internal traffic 이벤트에 대한 경계 설정 지원



Transformations

API Gateway에서 request/response 변환, 경로 변환 , 헤더정보 추가/수정 등 다양한 변환 정책 제공

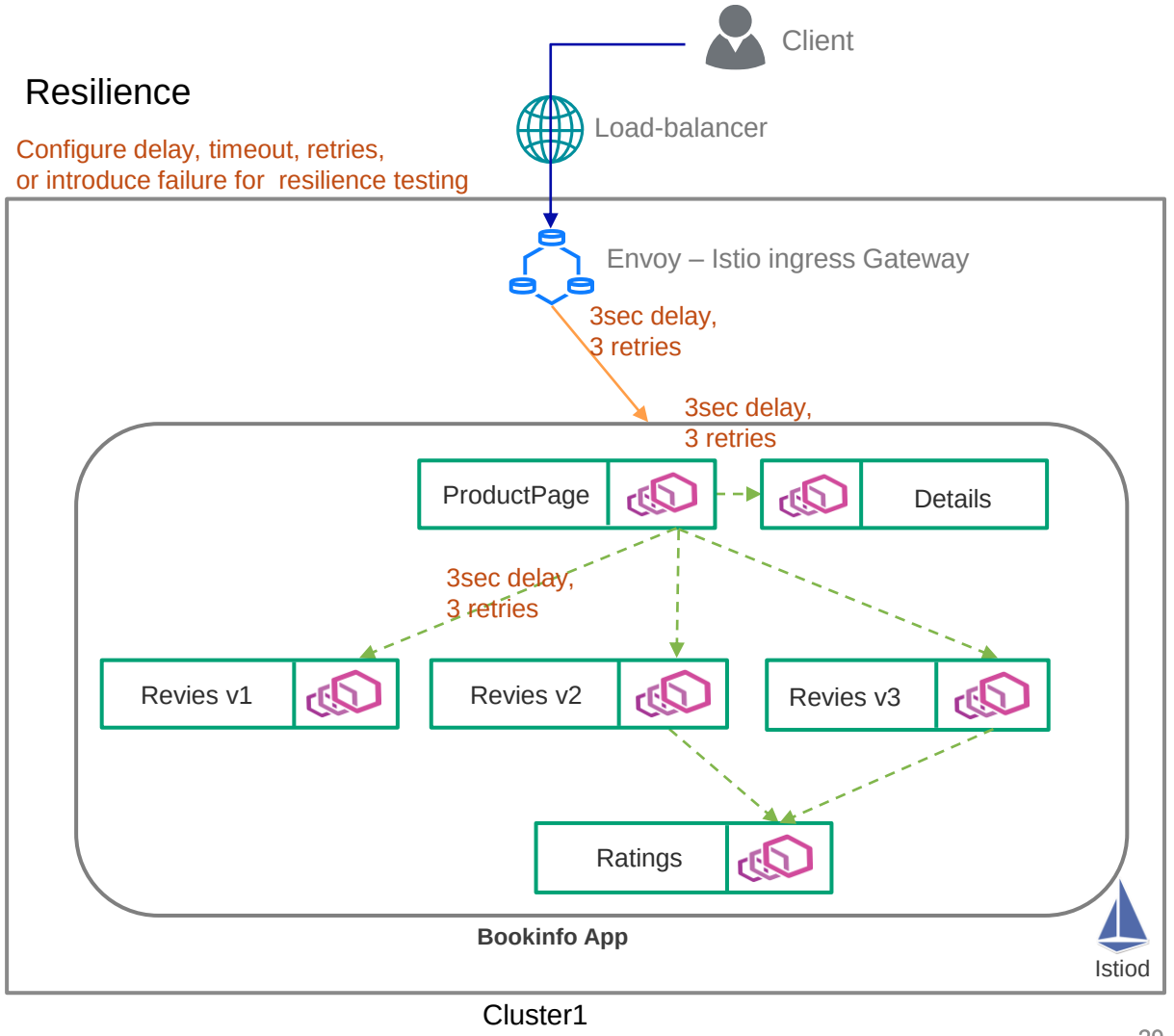
- API Mocking Test에 용이한 Direct Response 기능 지원
- 헤더 정보의 추출, 복사, 수정 및 추가 기능 지원
- Rewrite prefix 지원
- Redirect 지원
- gRPC to REST 지원



Application Resilience

검증된 Istio Service Mesh의 애플리케이션 복원 기능을 바탕으로 다양한 기능 제공

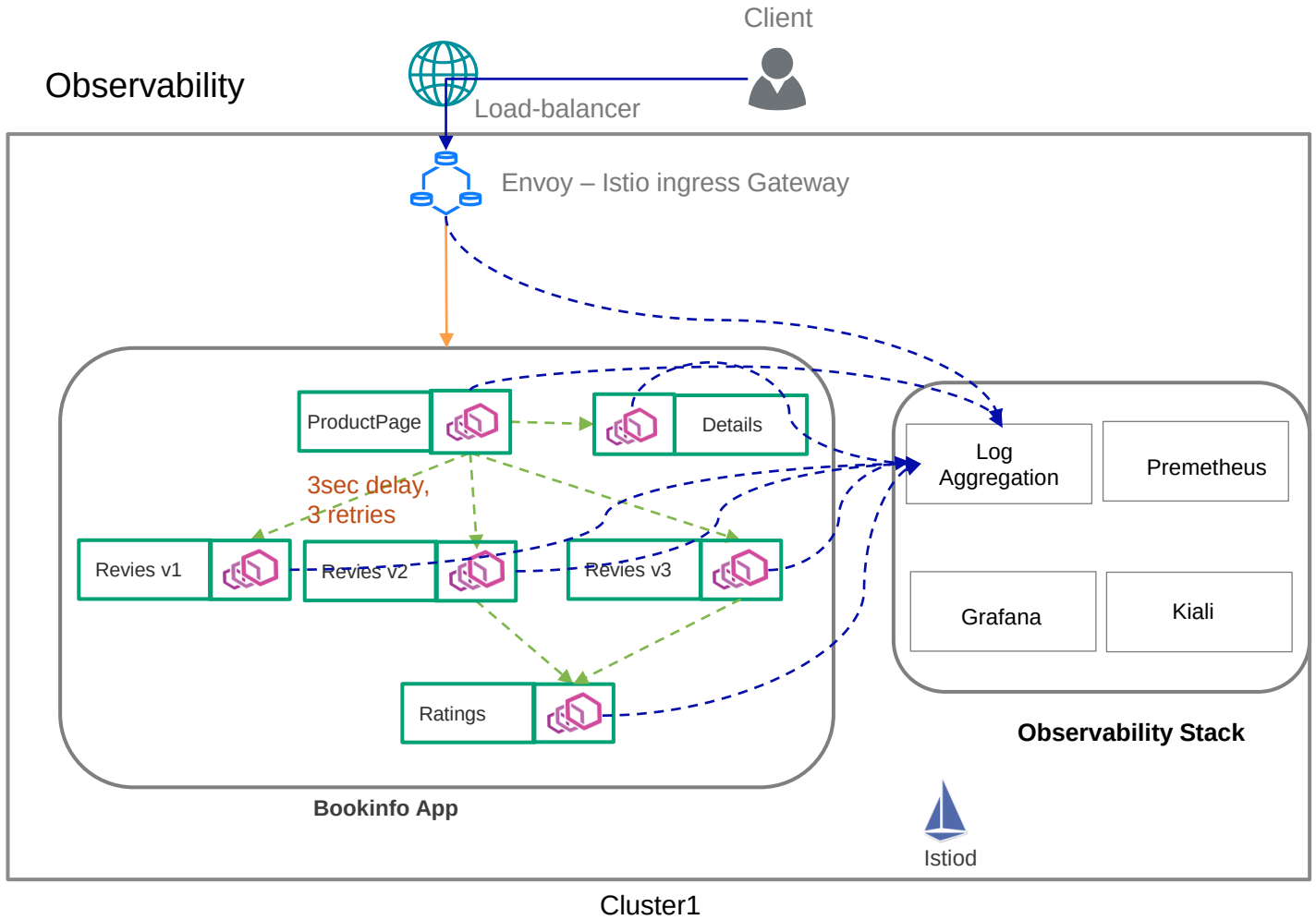
- Envoy 사이드카 수준의 애플리케이션 복원력 설정 지원
- delay, timeout, retries 지원
- Circuit Breakers 구현 지원
- Fault injection 지원



Observability

게이트웨이의 접속로그 및 요청 메트릭 수집 뿐만 아니라 각 App의 수집된 메트릭을 시각화, 분석, 추적 할 수 있는 도구 지원 및 운영 대시보드 제공

- 중앙집중식 로그집계시스템과 연동 지원
- 사용량 메트릭 제공
- Trace Metric (ex. Zipkin, Jaeger) 제공



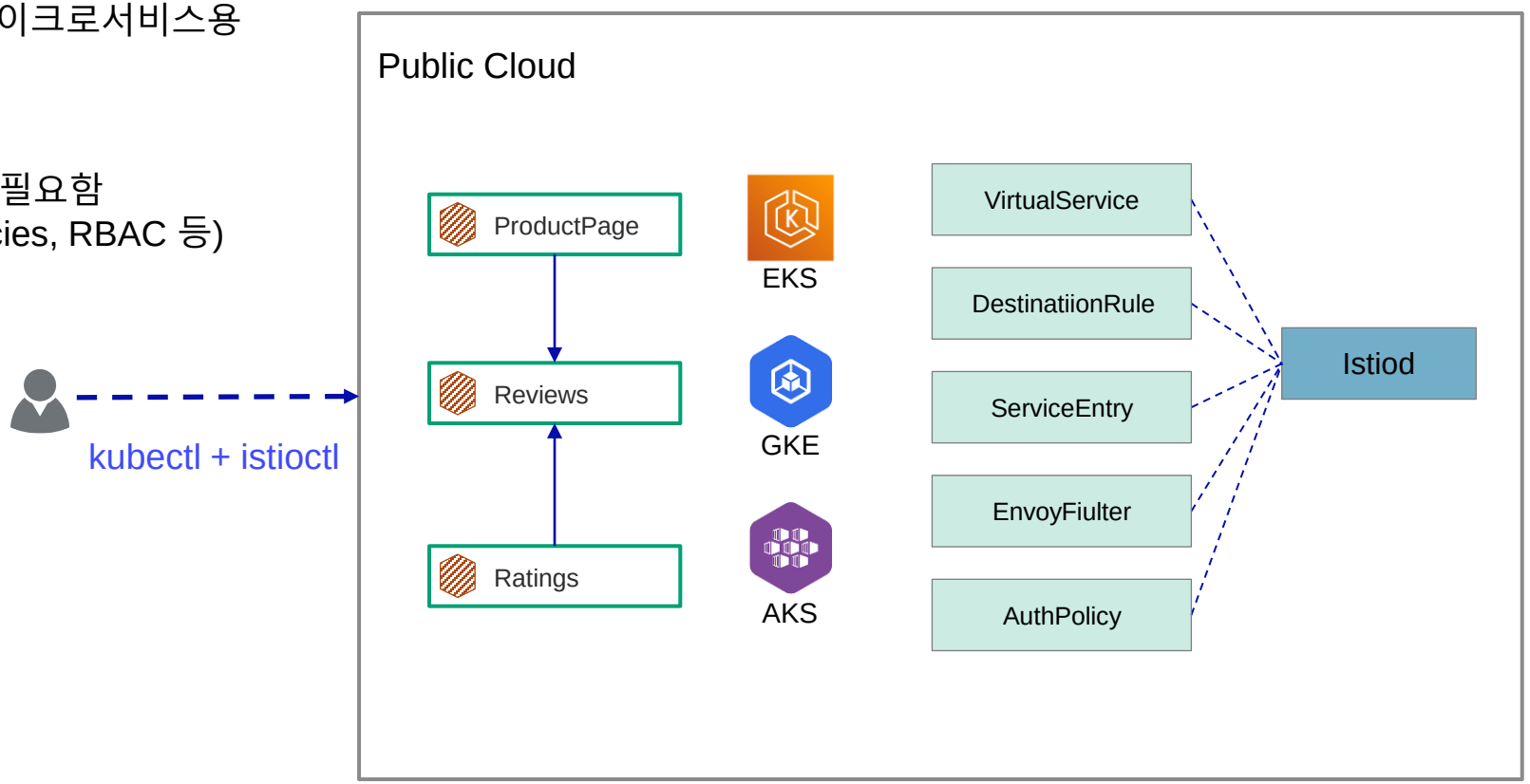
Service Mesh Journey

Single-cluster Istio 접근 방식

Single-cluster 환경에서 Istio로 어플리케이션 배포 및 관리는 용이
하지만 Multi-Clusters 환경에서 클러스터 간의 서비스 검색 및 보안 통신 등 클러스터 운영 관리의 어려움 존재

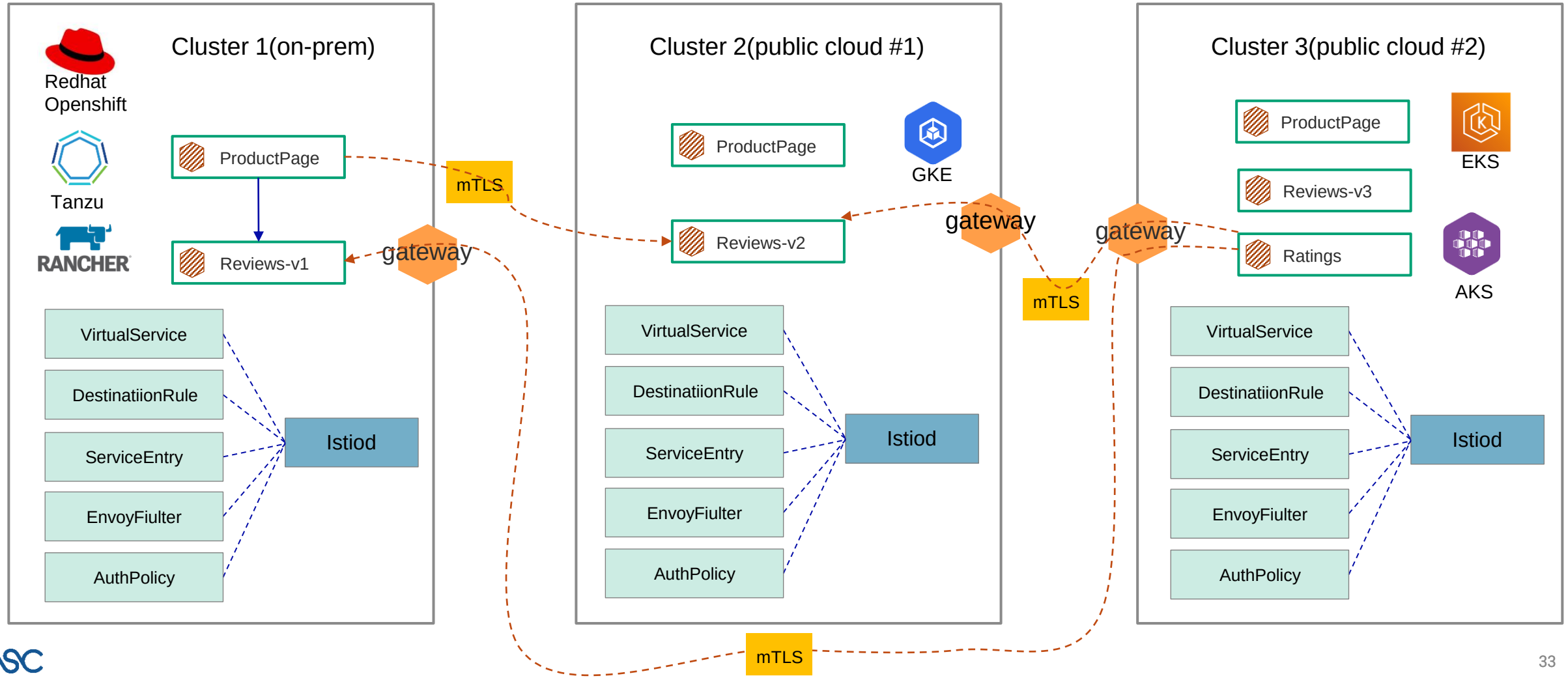
- 간단하고 쉬운 관리성
- Single Cluster 환경에서 소수의 마이크로서비스용 Istio low-level APIs 관리는 용이함
- Multi-Clusters 환경의 도전 과제
  - Multi-Clusters 들 간의 보안 통신 필요함
    (Shared Root Trust, Access Policies, RBAC 등)

Single Cluster 기반에서 소수의 마이크로서비스 관리는 쉽다



Multi-cluster 운영의 복잡성

다양한 Platform 기반의 멀티 클라우드 환경에서는 클러스터들 간의 서비스 검색, 트래픽 셰이핑 및 보안 통신 정책 구성의 어려움 존재



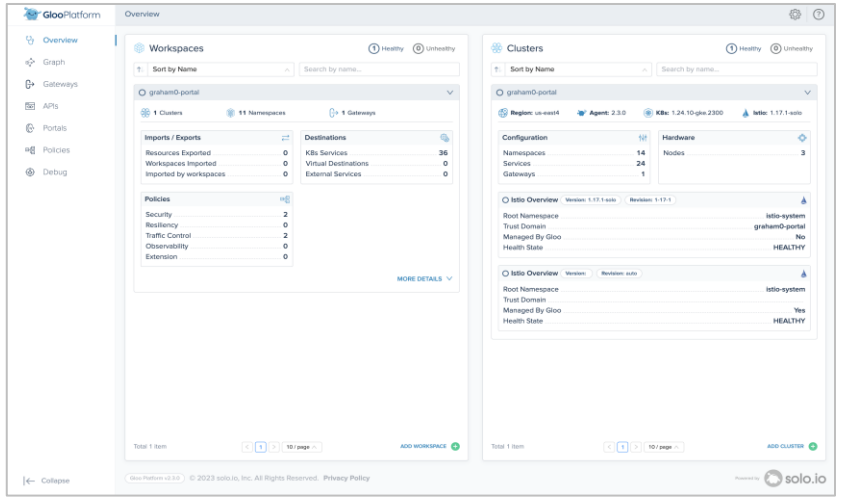
Multi-cluster 운영시 고려사항

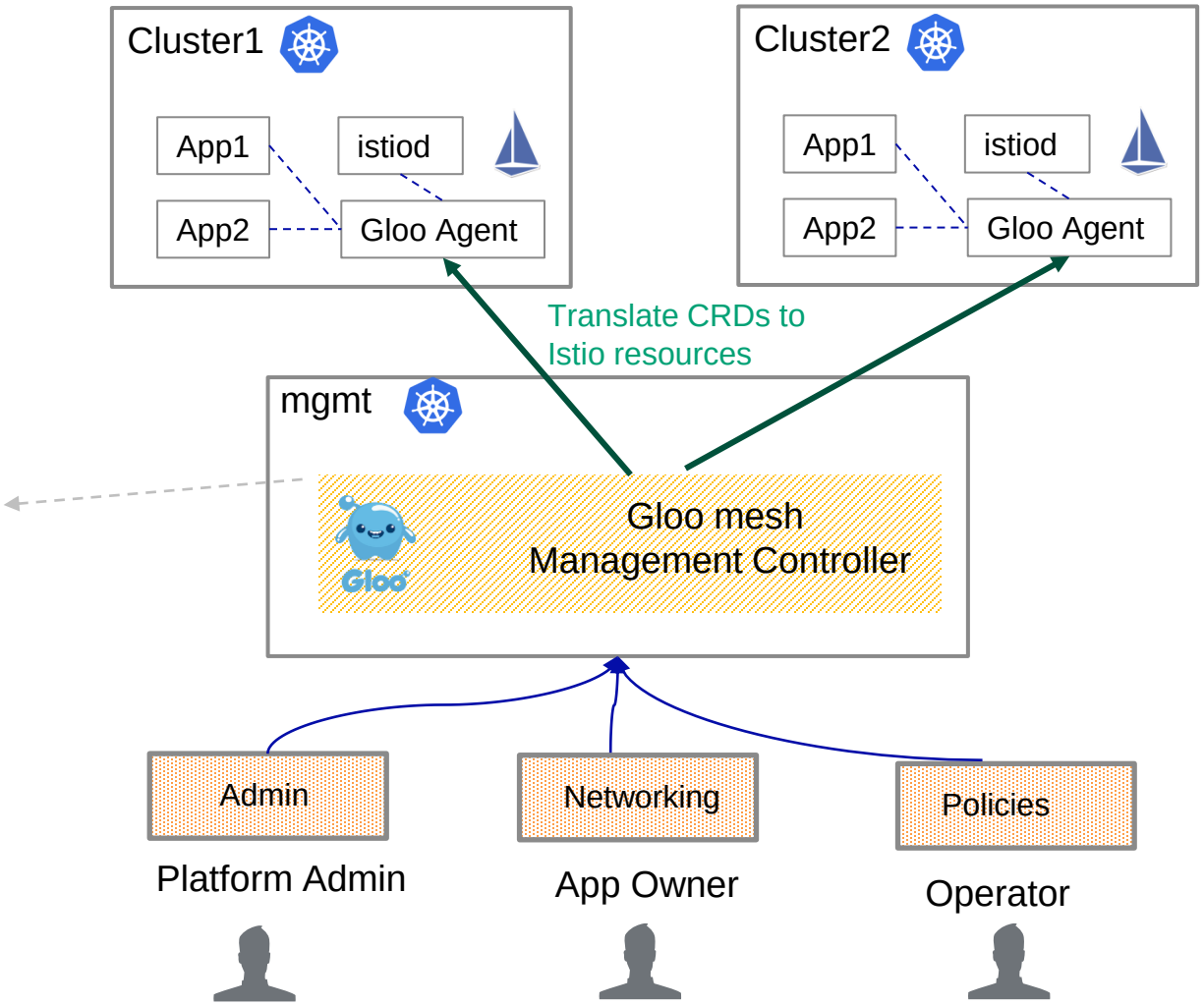
Multi-cluster들 간의 서비스 검색, 보안 통신 외에 추가적으로 더 고려해볼 사항들이 존재

- Multi-cluster 환경의 멀티 테넌시 모델
- Multi-cluster의 RBAC 모델
- Multi-cluster 간의 Zero Trust 시행
- Global observability (중앙 집중식 메트릭 및 액세스 로깅)
- 간소화된 클러스터 간 통신 (using virtual destinations)
- 표준화된 라이프사이클 관리
- 장기적인 N-4 지원
- FIPS 및 ARM 호환 이미지
- 글로벌 워크스페이스 기반 멀티 테넌시

Gloo Mesh Architecture (1/2)

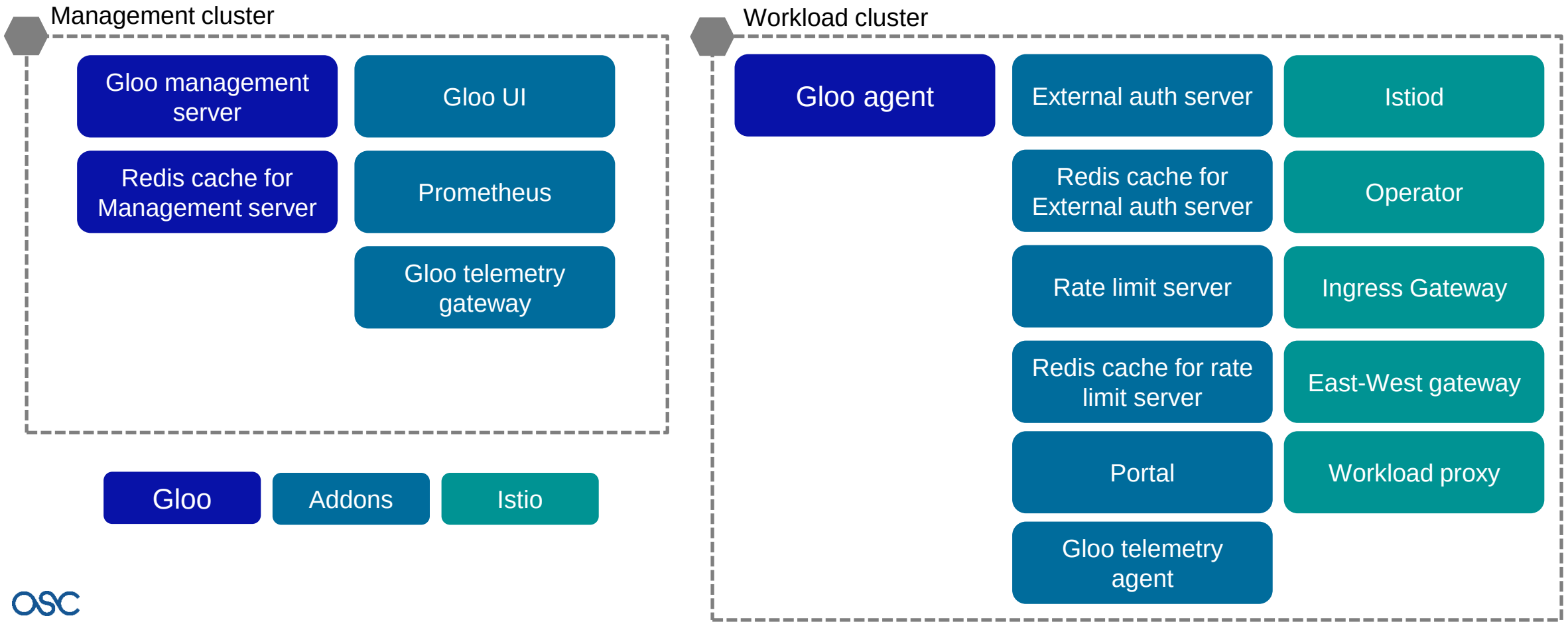
Gloo Mesh는 멀티 클러스터들 환경에 최적화된 아키텍처로 설계되어 있음
클러스터들 간 간소화된 서비스 검색, 보안 통신 ,Zero Trust 및 글로벌 Observability 기능을 제공하여 운영 효율 극대화





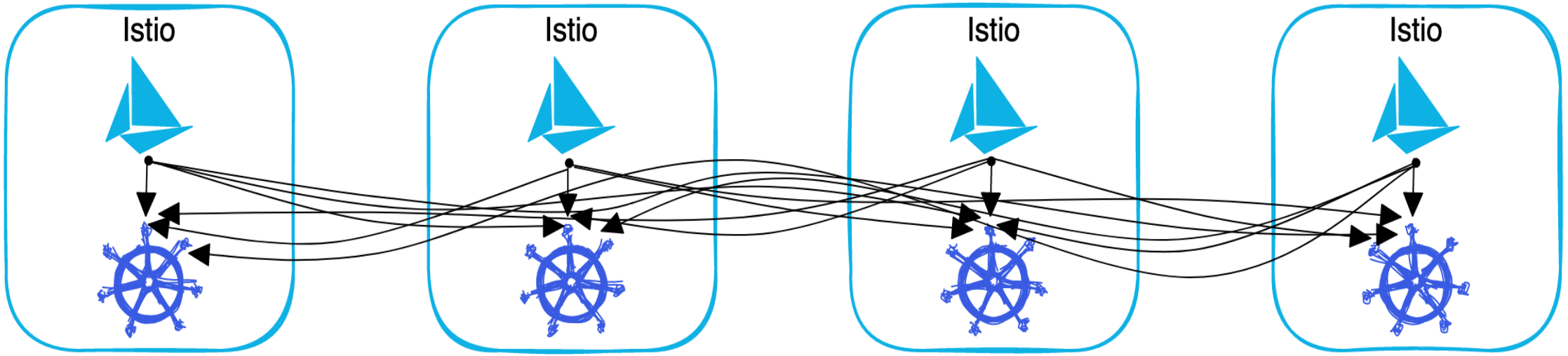
Gloo Mesh Architecture (2/2)

- Gloo Mesh는 관리 클러스터와 워크로드 클러스터로 구성되며 관리 서버와 워크로드 클러스터에 배포되는 Gloo agent로 구성
- 이 아키텍처는 클라우드 ,클러스터 환경에서 앱에 대한 뛰어난 라우팅, 검색, 보안, 확장성 및 모니터링 가능성 기능을 모두 제공



Cross-cluster Istio services 구성

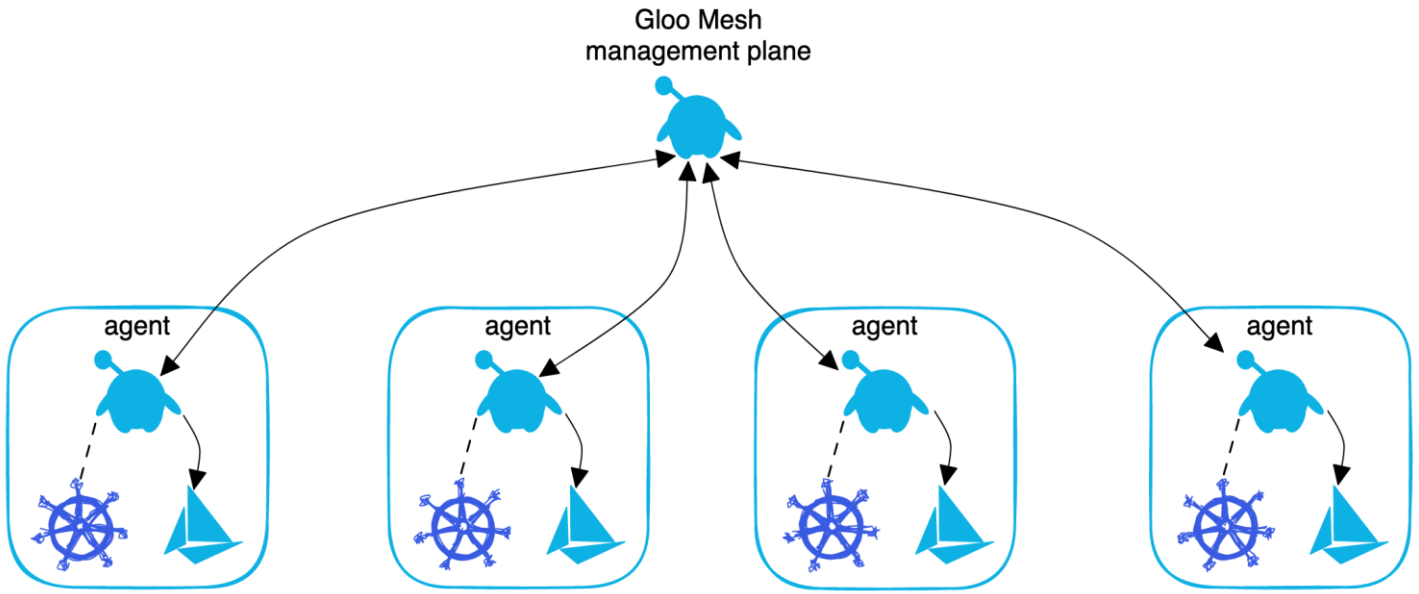
멀티 클러스터에서 Istio 구성은 클러스터들 간 통신문제, istio 컨트롤 플레인 간의 API 접속 문제,
클러스터들 간 복잡한 보안 통신 관리의 어려움 존재



- 도전적인 istio 구성은 cross-cluster 들의 communication problem 발생할 수 있음
- 각 istio 컨트롤 플레인이 각 클러스터의 Kubernetes API서버에 엑세스 할 수 있어야 함
- cross-cluster 간의 security concerns (통신 인증서 관리, mTLS)
- Istio 컨트롤 플레인은 클러스터 중 하나에 연결할 수 없는 경우 시작할 수 없으며 이는 성능에 영향을 미침

Gloo Mesh Istio discovery 단순화

Gloo Mesh는 관리서버- 워크로드 에이전트 방식인 릴레이 아키텍처로 구성되어 있어서
멀티 클러스터 간 Istio discovery 문제를 간단하게 해결함



- Gloo Mesh는 서버-에이전트 방식으로 이러한 문제 해결
- 각 클러스터의 agent는 로컬 Kubernetes API 서버를 감시하고 보안 gRPC 채널을 통해 Gloo Mesh management plane 서버에 정보 전달
- 전달받은 Gloo Mesh는 agent에게 다른 클러스터에서 발견된 워크로드에 해당하는 Istio ServiceEntries를 생성하도록 지시

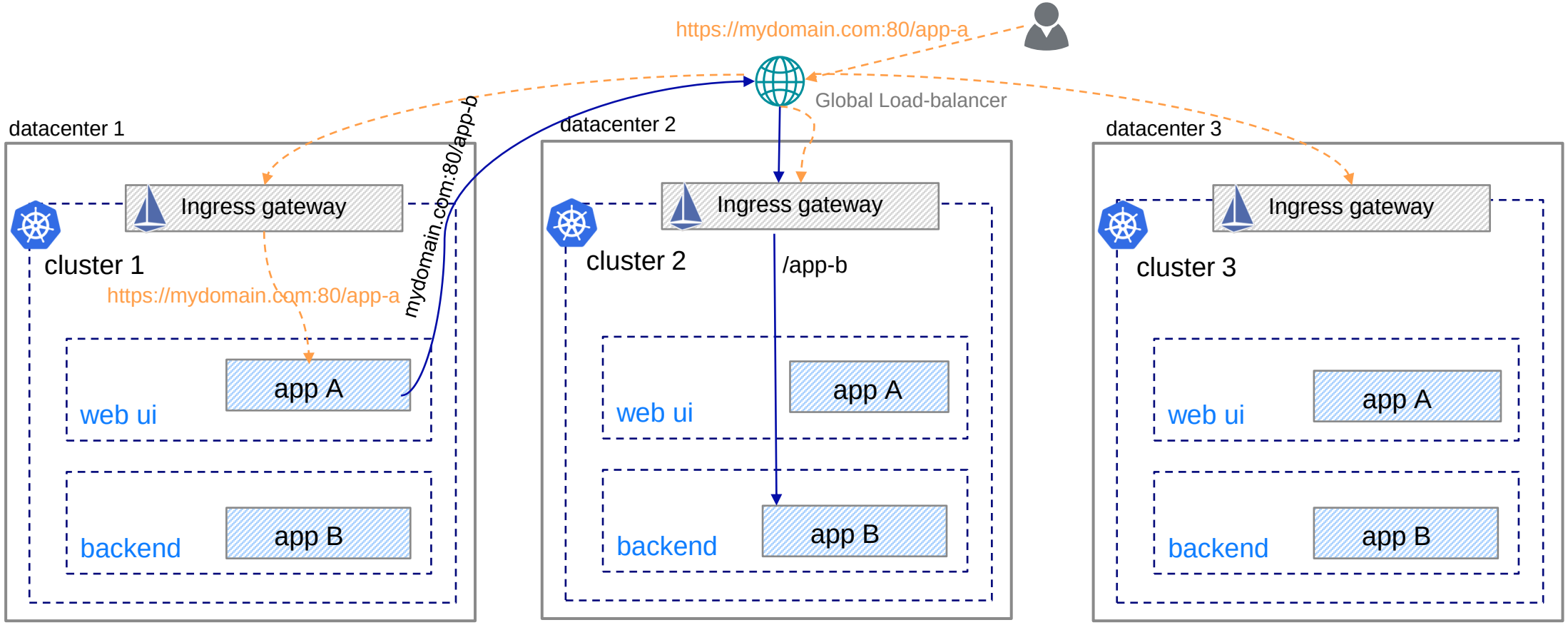
Gloo Mesh Enterprise

Gloo Mesh Enterprise 사용하면 여러 클러스터 및 ServiceMesh에서 서비스를 연결하고 관리할 수 있는
확장 가능한 여러 도구 및 다양한 혜택 제공

BENEFIT	GLOO MESH ENTERPRISE	GLOO MESH OPEN SOURCE	COMMUNITY ISTIO
기능 개발에 대한 업스트림 우선 접근 방식	☑	☑	☑
클러스터 및 서비스 메시 전반에 걸친 설치, 업그레이드 및 관리	☑	☑	✕
보안, 트래픽 라우팅, 변환, 관찰 가능성 등을 위한 고급 기능	☑	✕	✕
n-4 버전에 대한 end-to-end Istio 지원 및 CVE 보안 패치	☑	✕	✕
Distroless 이미지 및 FIPS 규정 준수를 위한 특수 빌드	☑	✕	✕
24x7 및 1 SLA (one-hour Severity) 지원	☑	✕	✕
기능 확장을 위한 GraphQL 및 Portal 모듈 지원	☑	✕	✕
간소화된 멀티테넌시를 위한 작업 공간 지원	☑	✕	✕

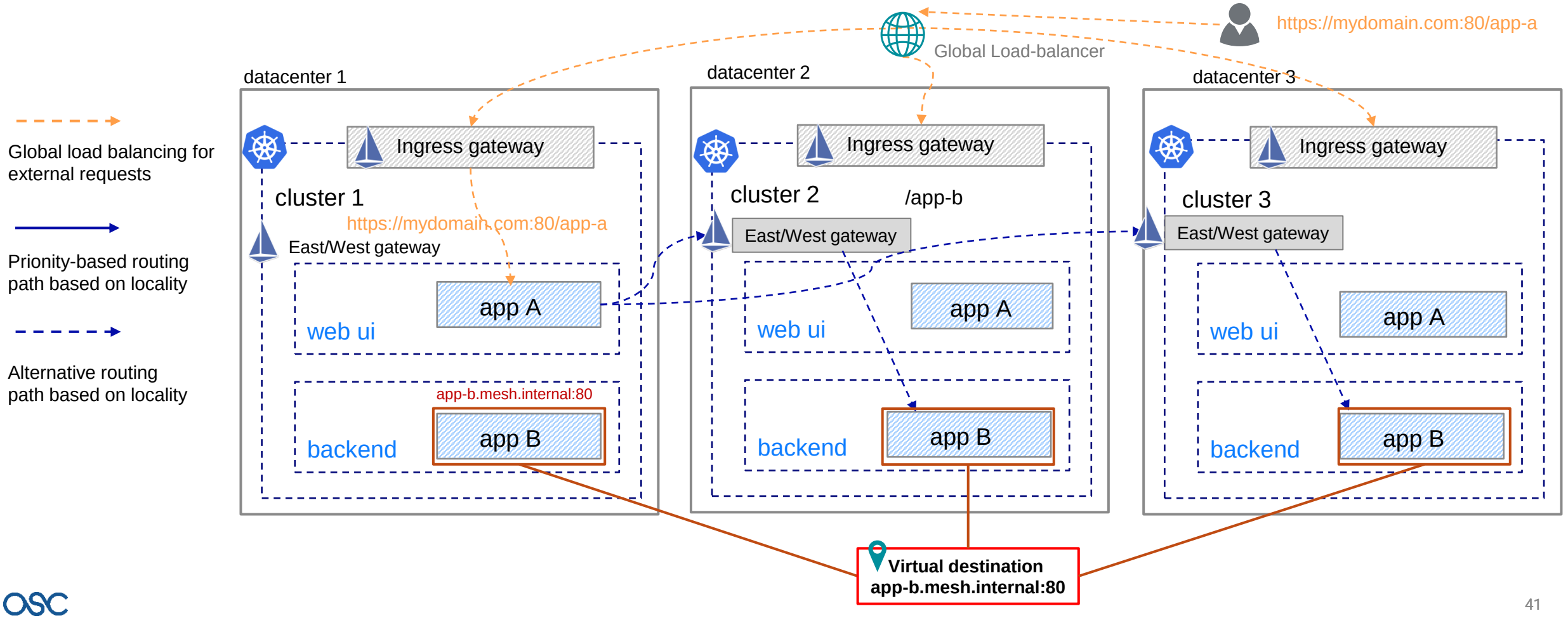
Kubernetes 환경의 Multi-cluster 라우팅 문제점

- 서비스를 멀티 클러스터에 배포하여 서비스 가용성을 보장할 수 있는 운영 환경으로 구축할 경우 서비스 멀티 라우팅 문제 발생
- 클라이언트로 요청 받은 app A는 app B를 호출하는데 같은 클러스터내에 app B가 존재하더라도 가용성을 위해서 밖의 글로벌 로드밸런서로 호출해야 하는 단점 발생
- 글로벌 로드밸런서는 각 클러스터들의 app B가 정상상태인지 여부를 알 수 없이 app B를 호출해야 하는 단점 발생



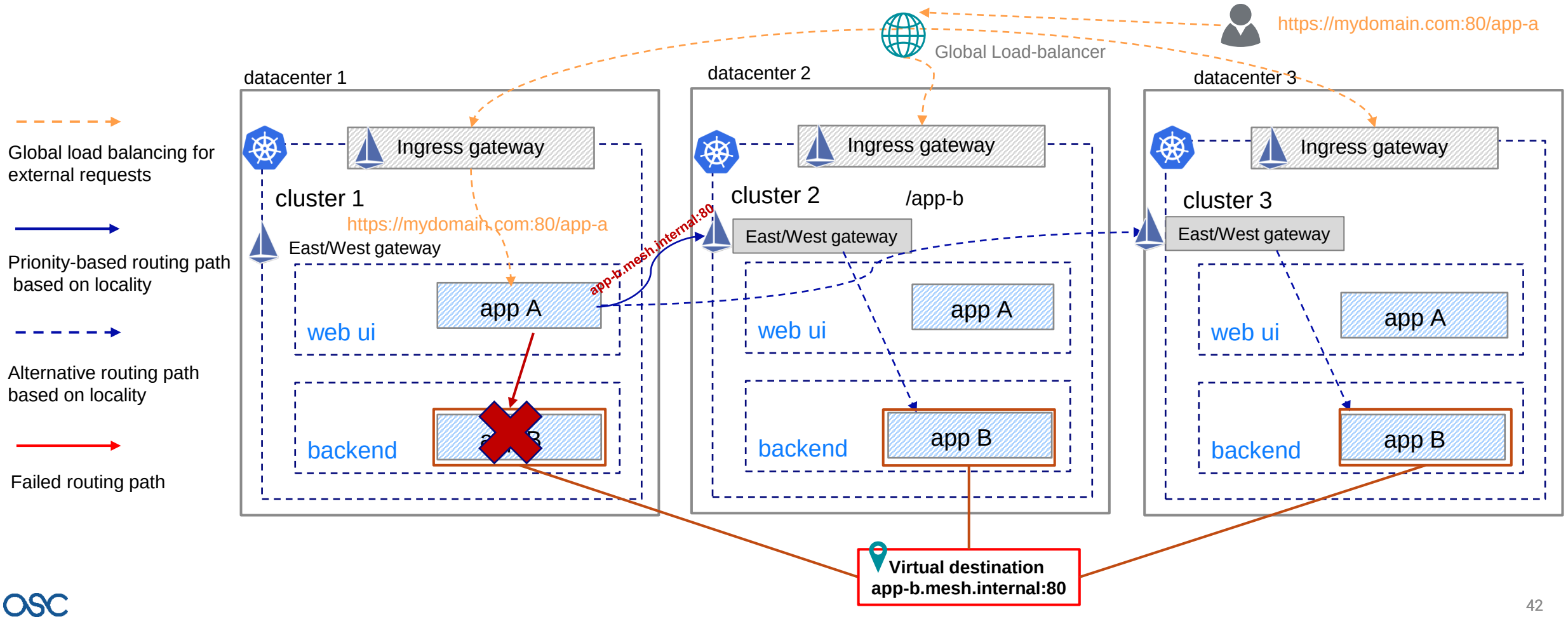
Gloo Mesh 기반의 Multi-cluster 라우팅 (1/3)

- Gloo Mesh는 각 클러스터에 app B가 배포된 위치와 정상 상태를 여부를 관리하므로 알고 있음
- app A의 모든 요청은 app-b.mesh.internal:80 내부망용 Internal URL로 서비스 메시를 떠나지 않고 가장 가까운 app B 인스턴스로 직접 라우팅 하거나 각 클러스터의 east/west gateway를 이용하여 app B를 적절하게 호출할 수 있음



Gloo Mesh 기반의 Multi-cluster 라우팅 (2/3)

- Cluster1의 app B가 장애 발생시 Gloo Mesh는 라우팅 규칙을 자동 업데이트, 가장 가까운 클러스터의 app B에 우선 순위 부여
- 여전히 서비스 메시를 떠나지 않고 Cluster2 의 east/west gateway로 직접 라우팅 되어 정상적으로 app B를 호출



Gloo Mesh 기반의 Multi-cluster 라우팅 (3/3)

Gloo Mesh는 Muti-cluster 라우팅을 위해서 Gloo Mesh Resource인 Virtual Destination, Rotue Table 설정으로 간단하게 구성할 수 있음

Virtual Destination

```
apiVersion: networking.gloo.solo.io/v2
kind: VirtualDestination
metadata:
  name: app-b-vd
  namespace: web-ui
spec:
  hosts:
  - app-b.mesh.internal
  ports:
  - number: 80
    protocol: HTTP
    targetPort:
      number: 80
  services:
  - labels:
      app: app-b
```

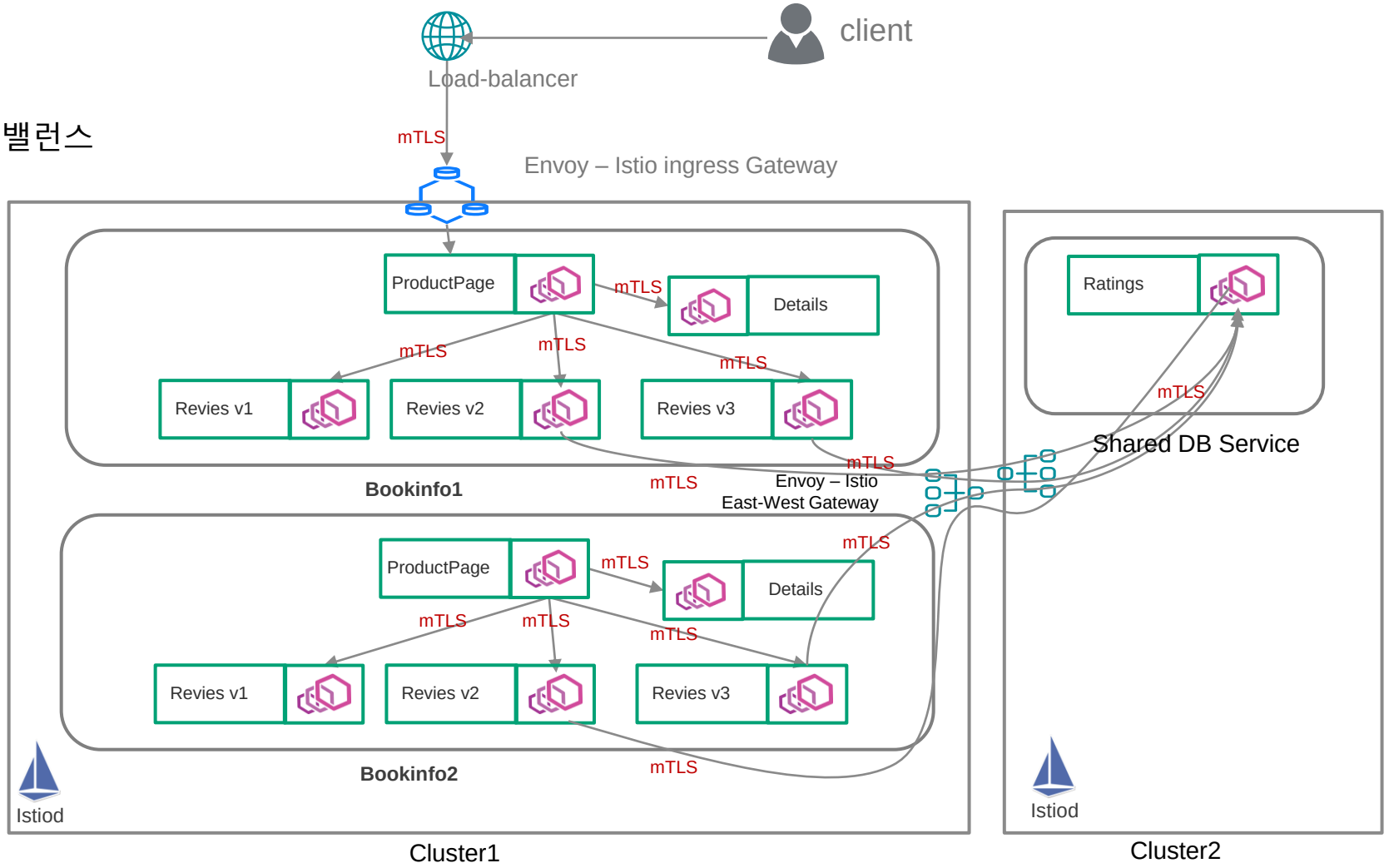
Rotue Table

```
apiVersion: networking.gloo.solo.io/v2
kind: RouteTable
metadata:
  name: app-b-estwest-rt
  namespace:web-ui
spec:
  hosts:
     - app-b.backend.svc.cluster.local'
  http:
    - name: app-b
      matchers:
      - uri:
          prefix: /
      forwardTo:
        destinations:
          - ref:
              name: app-b-vd
              namespace: api
            kind: VIRTUAL_DESTINATION
            port:
              number: 80
      labels:
        route: app-b
```

Gloo Mesh Features

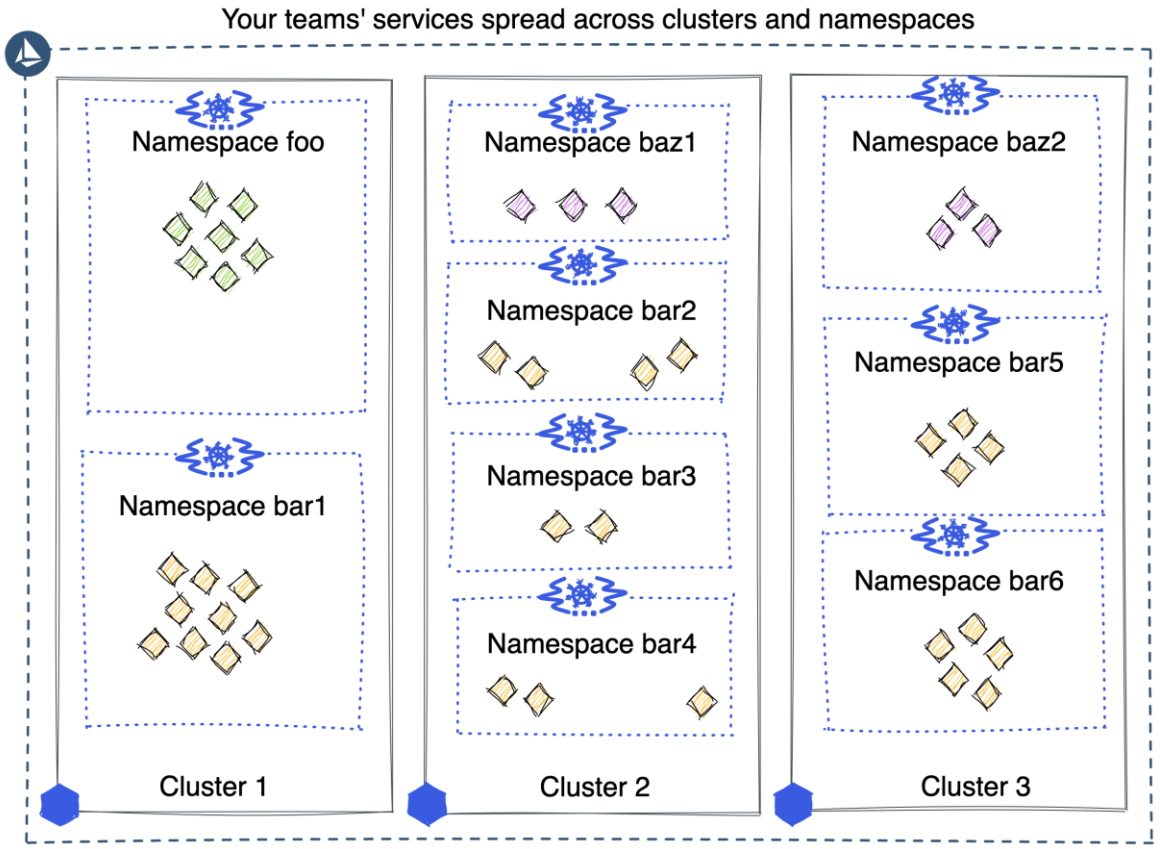
Gloo Mesh는 Gloo gateway 모든 기능을 지원하며 추가적으로 클러스간의 운영 관리 및 mTLS 통신 지원

- Service-to-Service Traffic 라우팅 및 로드밸런스
- End-to-end 인증/인가, 암호화
- 클러스터 간 Shared Root Trust
- Multi-Cluster 라우팅
- Application Resilience
- Failover

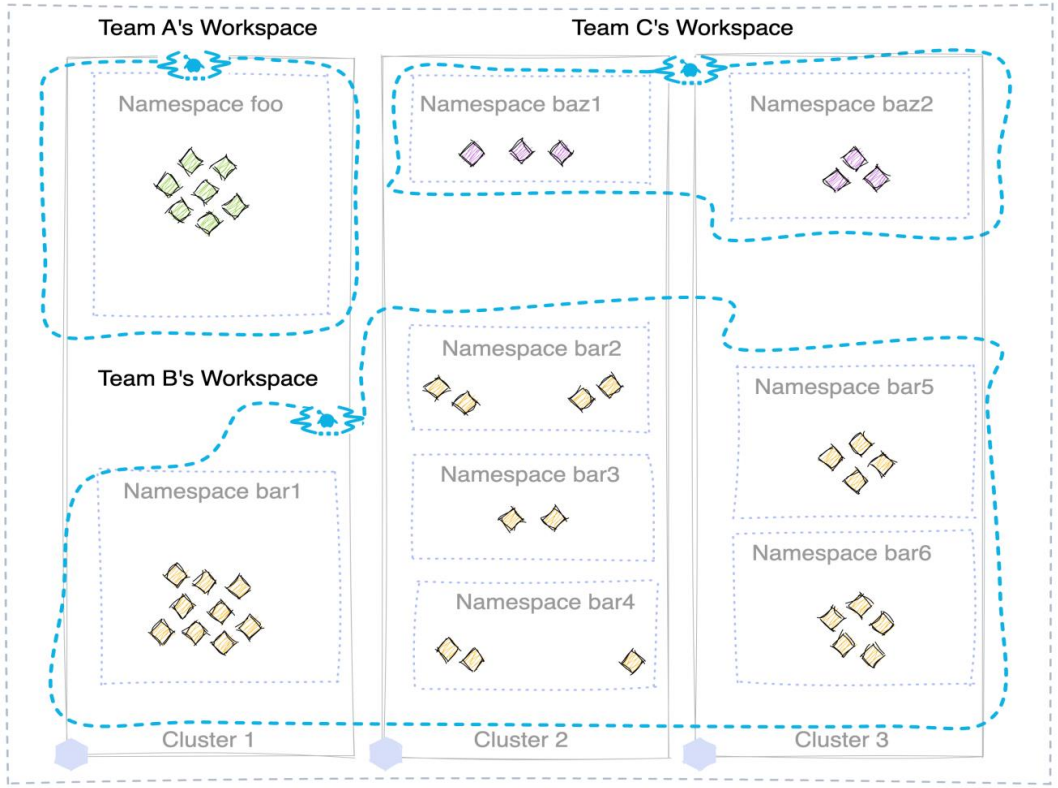


GlooMesh Workspaces

- Gloo Mesh는 Kubernetes 기반 멀티테넌시에 대한 새로운 개념인 Workspace 사용자 지정 리소스 제공
- Gloo Workspace는 클러스터와 네임스페이스에 걸쳐 있는 팀의 모든 리소스를 그룹화하여 설정에 따라 다른 리소스에서 자동으로 사용 가능



클러스터/네임스페이스 기반으로 리소스 분리

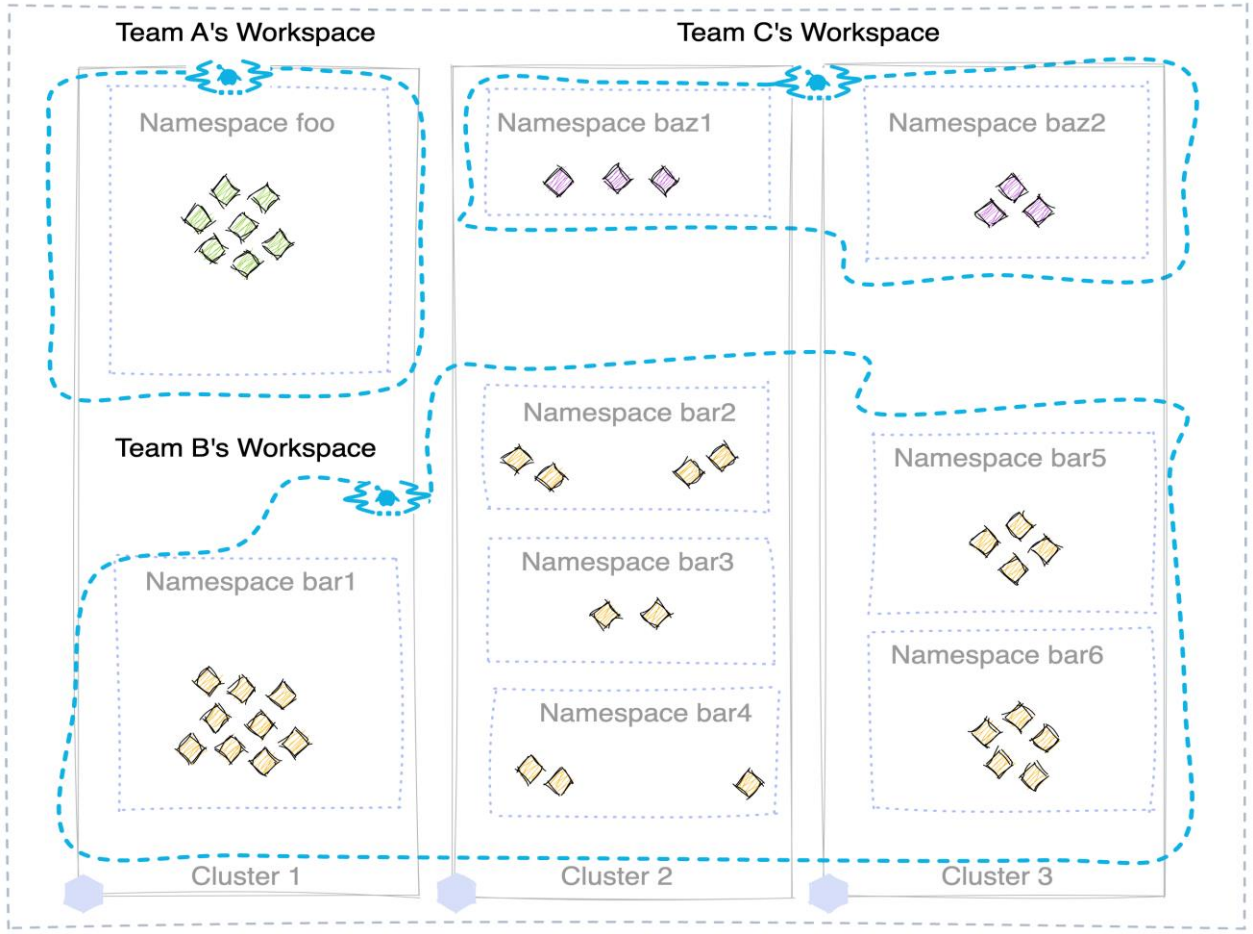


팀 기반의 리소스 분리

GlooMesh Workspaces 기반의 Multi-Tenancy

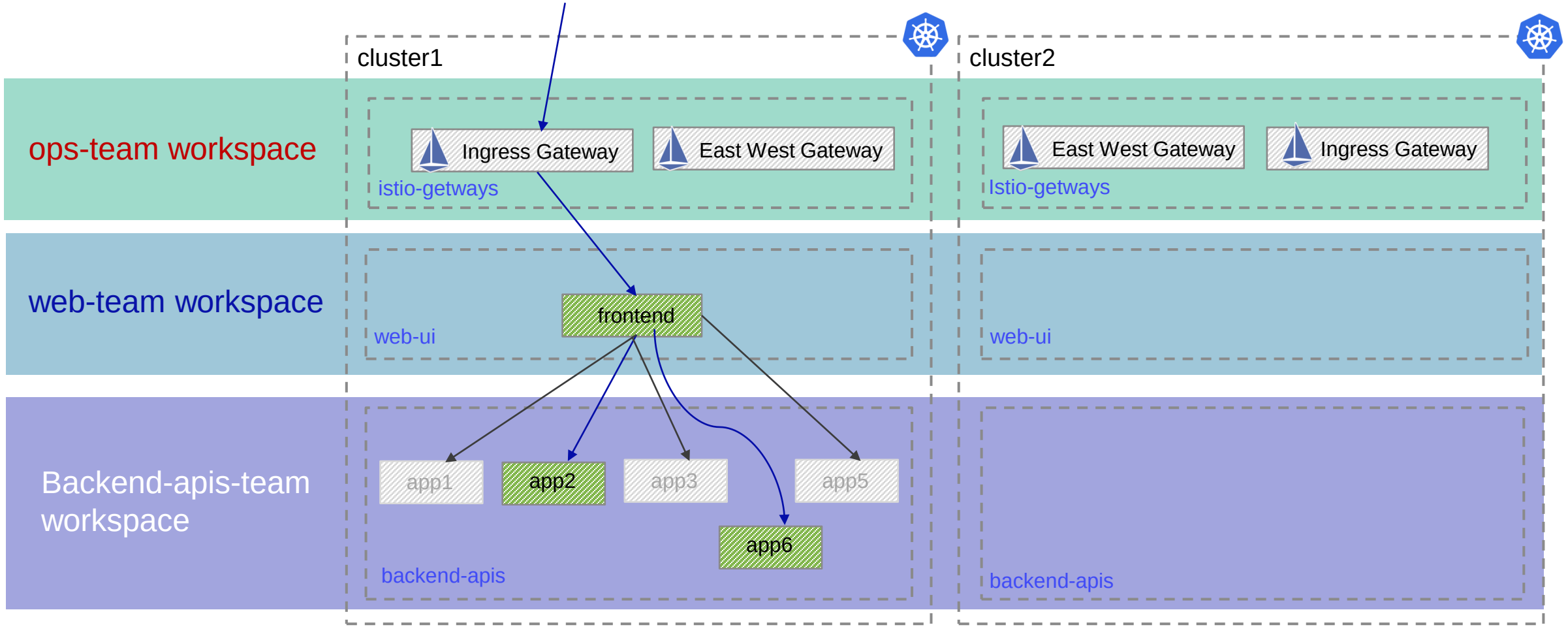
그룹화된 리소스들은 팀(조직) Workspace로 클러스터 별, 네임스페이스별로 관리가 가능한
Multi Tenancy 구성 가능

- TeamA
  - clusters: cluster1
  - namespaces:
    - foo
- Team B
  - clusters: *
  - namespaces:
    - bar1,bar2,bar3,bar4,bar5,bar6
- Team C
  - clusters: *
  - namespaces:
    - baz1,baz2



GlooMesh Workspaces 페르소나(personas)

- Gloo workspace는 팀 또는 개인의 페르소나 책임을 부여하여 클러스터 리소스, 네임스페이스 등 팀 자원 제어 가능
- 다른 workspace 영역과 리소스를 공유하거나 가져오고 내보낼 수 있는 workspace 설정 구성을 제공함으로써 팀 리소스에 대한 구축 및 접근 통제 등 손 쉬운 운영 관리 기능 제공



GlooMesh Workspaces 적용 예제

Gloo Mesh Workspace는 Gloo Mesh 리소스인 Workspace, WorkspaceSettings으로 쉽게 설정 할 수 있음

Workspace

```
apiVersion: admin.gloo.solo.io/v2
kind: Workspace
metadata:
  name: backend-apis-team
  namespace: gloo-mesh
  labels:
    gloo.solo.io/team: 'backend-apis'
spec:
  workloadClusters:
  - name: 'mgmt'
    namespaces:
    - name: backend-apis-team
  - name: '*'
    namespaces:
    - name: backend-apis
```

WorkspaceSettings

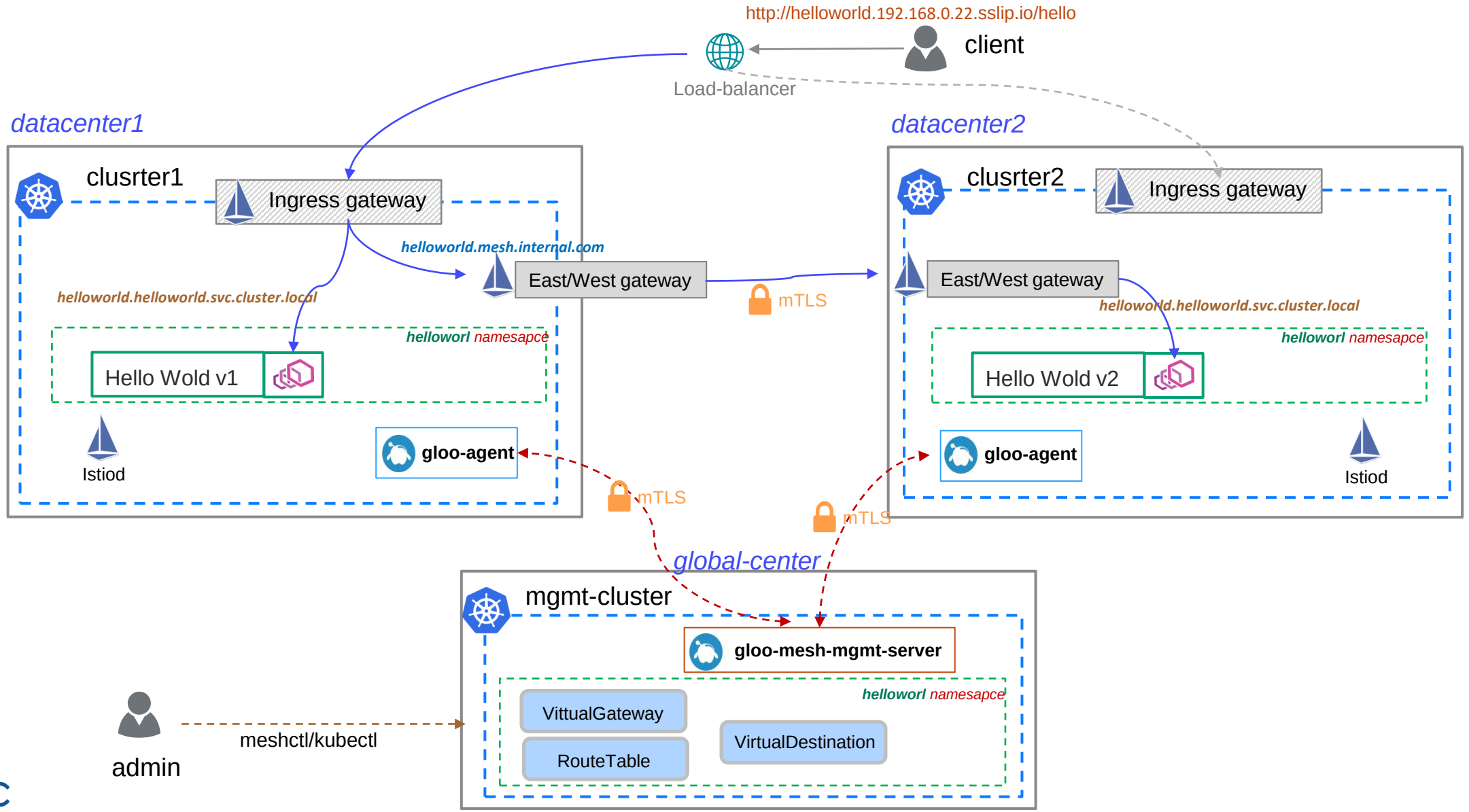
```
apiVersion: admin.gloo.solo.io/v2
kind: WorkspaceSettings
metadata:
  name: web-team
  namespace: web-team-config
spec:
  importFrom:
  - workspaces:
    - name: backend-apis-team
  - workspaces:
    - name: ops-team
  exportTo:
  - workspaces:
    - name: ops-team
  options:
    eastWestGateways:
    - selector:
        labels:
          istio: eastwestgateway
    federation:
      enabled: false
    serviceIsolation:
      enabled: true
      trimProxyConfig: true
```

Service Mesh Demo

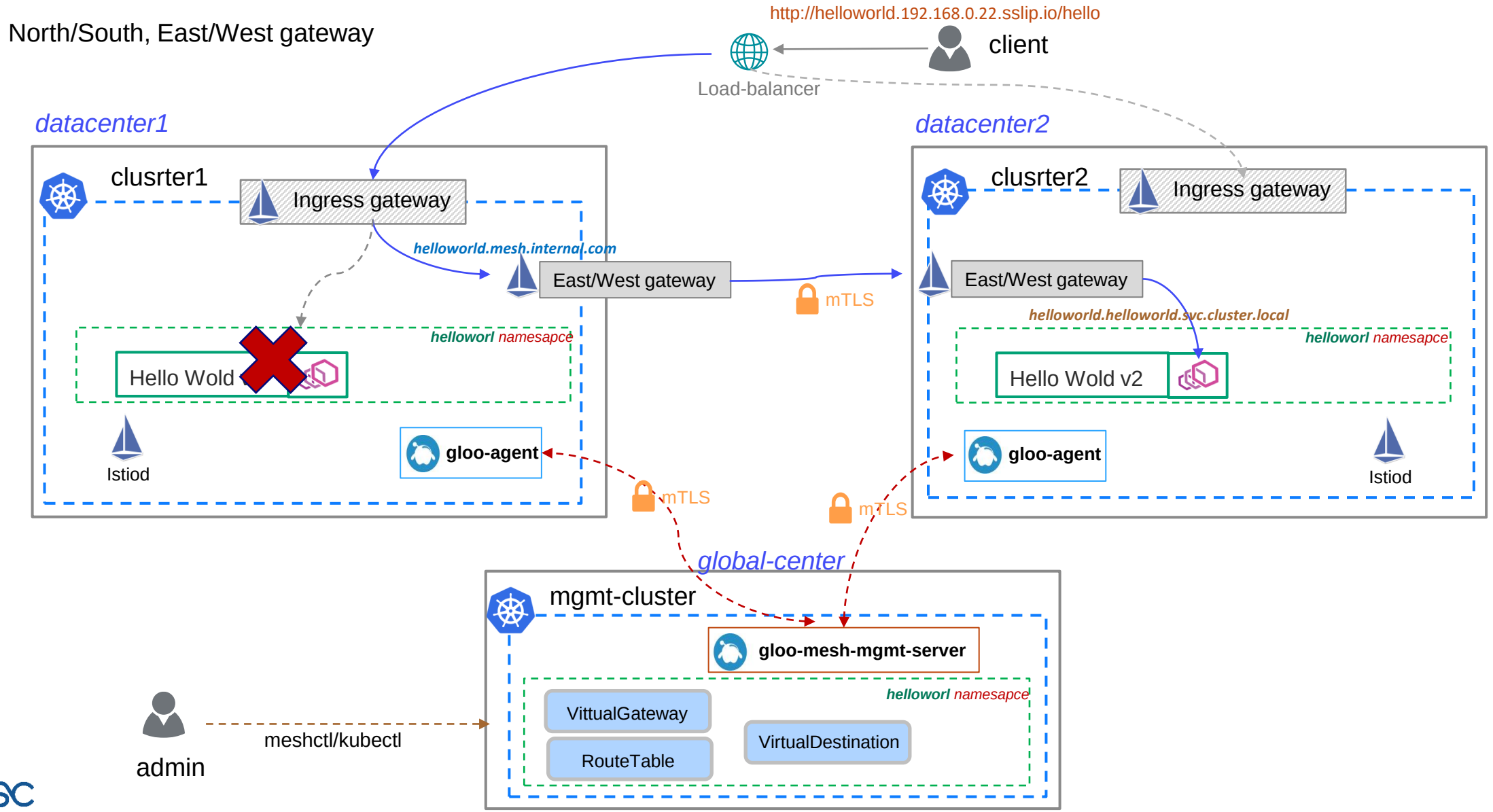
Gloo Mesh Demo 환경

	VM	• ubuntu : 20.04 기반 • 3개의 VM및 클러스터 생성 - mgmt-cluster vm: 192.168.0.21 - cluster1-vm : 192.168.0.22 - cluster2-vm : 192.168.0.23 • 가상의 3개의 datacenter와 cluster - datacenter1 , datacenter2, global-center - mgmt.-cluster, cluster1, cluster2
	KUBERNETES	• K3S : v1.26.1+k3s1 version • MGMT_CLUSTER : k3s-gloo-mgmt-server-192.168.0.21-cluster • REMOTE_CONTEXT1: k3s-gloo-gateway1-192.168.0.22-cluster • REMOTE_CONTEXT2: k3s-gloo-gateway2-192.168.0.23-cluster
	NETWORK	• LoadBalancer 구성 - metallb : v0.13.9
	Application	• hellloworld 서비스 (demo 편의상 v1, v2로 분리 배포함) - helloworld v1 (cluster1) - helloworld v2 (cluster2)

Gloo Mesh Multi-cluster 라우팅 데모 구성도 (정상 시나리오)



Gloo Mesh Multi-cluster 라우팅 데모 구성도 (장애발생 시나리오)



Summary

Gloo Gateway

- Gloo Gateway 설치로 API Gateway + Kubernetes Ingress 모든 기능을 사용 가능
- 컨트롤 plane 과 데이터 plane 분리하여 Multi namespace 환경의 확장이 대단히 용이
- 복잡성을 추상화한 간소화된 API로 간단한 설치 및 조직 단위의 운영관리 용이
- API Gateway및 Ingress 기능 외에도 보안, WAF, DLP, GraphQL 등 다양한 기능 제공

Gloo Mesh

- Gloo Mesh는 멀티 클러스터 환경에 최적화된 제품
- 간소화된 서비스 검색, 보안 통신 ,Zero Trust 및 글로벌 Observability 기능을 제공하여 효율적인 운영관리 용이
- 쉽고 관리가 용이한 Multi-cluster 라우팅 아키텍처로 서비스 장애 대응에 용이
- Kubernetes 기반 멀티테넌시에 대한 새로운 개념의 Workspace 제공

아래 링크를 통해 Solo.io 데모를 신청하세요.

담당자가 필요한 정보를 가지고 곧 연락 드리겠습니다.

Contact

solo.io

OSC Korea 문의

sales@osckorea.com / 02-539-3690

www.osckorea.com/contact

감사합니다

㈜오에스씨코리아
서울 서초구 신반포로 339 논현빌딩 101호 06531 | 02.539.3690
sales@osckorea.com | www.osckorea.com