KubeCon | CloudNativeCon

Europe 2023

KubeCon | CloudNativeCon

Europe 2023

Story of Our Transition to a Custom Kubernetes Operator for an API Gateway

Vincent Behar - Ubisoft

ABOUT ME

Vincent Behar

Senior Engineer

Ubisoft

French video game company

Twitter

https://twitter.com/vbehar

Mastodon

https://hachyderm.io/@vbehar

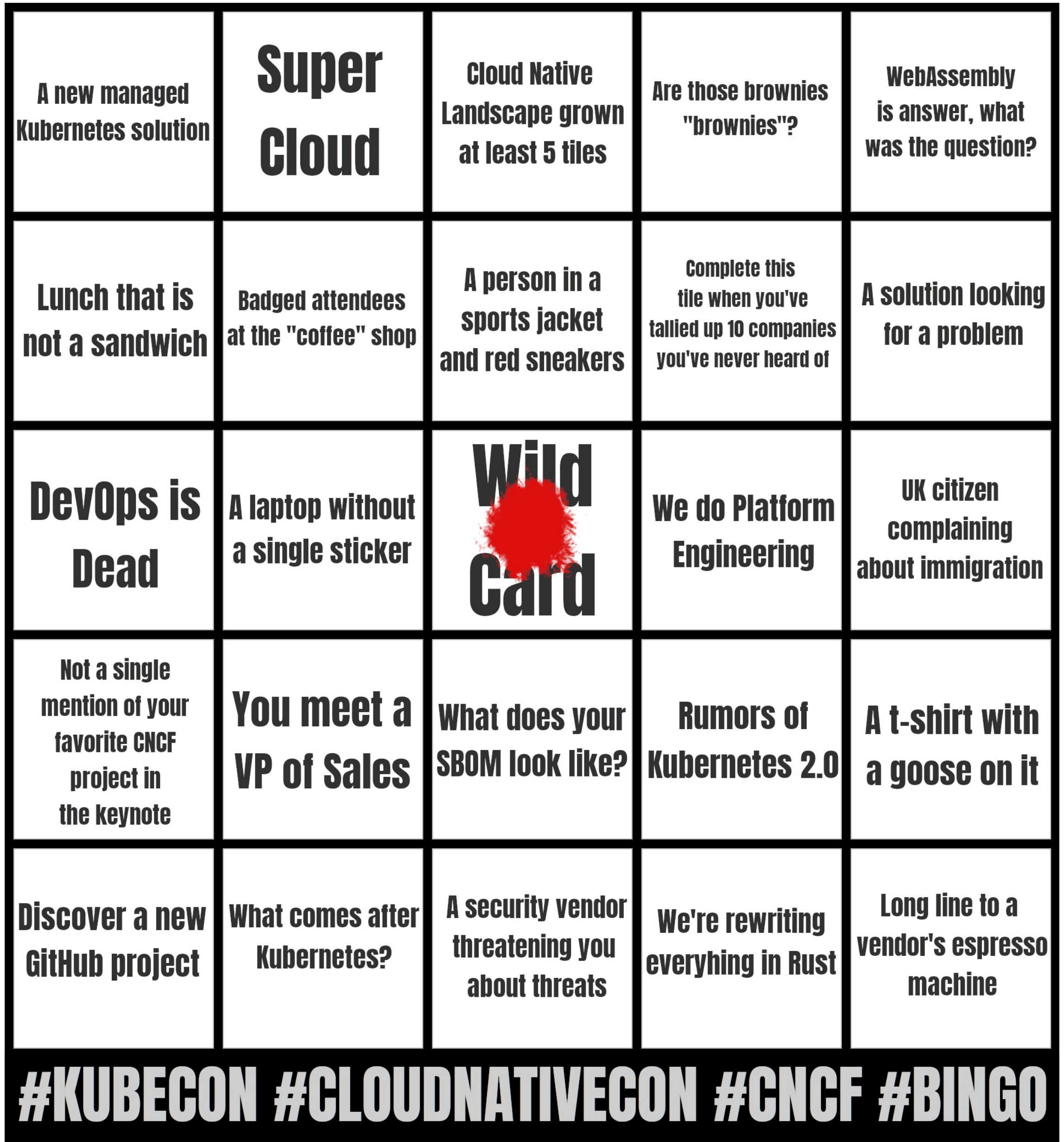
Slack CNCF/Kubernetes

@vbehar

GitHub

https://github.com/vbehar

BINGO GAME



A new managed Kubernetes solution	Super Cloud	Cloud Native Landscape grown at least 5 tiles	Are those brownies "brownies"?	WebAssembly is answer, what was the question?
Lunch that is not a sandwich	Badged attendees at the "coffee" shop	A person in a sports jacket and red sneakers	Complete this tile when you've tallied up 10 companies you've never heard of	A solution looking for a problem
DevOps is Dead	A laptop without a single sticker	Wild Card	We do Platform Engineering	UK citizen complaining about immigration
Not a single mention of your favorite CNCF project in the keynote	You meet a VP of Sales	What does your SBOM look like?	Rumors of Kubernetes 2.0	A t-shirt with a goose on it
Discover a new GitHub project	What comes after Kubernetes?	A security vendor threatening you about threats	We're rewriting everyhing in Rust	Long line to a vendor's espresso machine

#KUBECON #CLOUDNATIVECON #CNCF #BINGO

Story of Our Transition to a Custom Kubernetes Operator

For an API Gateway

- Context
- Why an operator – and which one
- User experience

INTERNAL DEVELOPER PLATFORM

Providing managed services – across cloud providers

Managed Services

- Kubernetes clusters / namespaces
- Databases
- Vault namespaces
- …
- **Opened to contributions**

Unified Experience

- Central management of projects, users, …
- Billing
- JSON REST API
- **OpenAPI**

PLATFORM'S API GATEWAY

Single entry point for control plane – based on **Kong**

Rules

- **OpenAPI**
  - Monitoring
  - Custom behaviour
- Authentication
  - OIDC / Azure AD
- Authorization
  - Open Policy Agent

Management

- **Gateway admins**
- Config as code
- YAML
- Python
- Kong DecK
- CI-Ops
- GitLab Pipelines

CHALLENGES

Enforcing Security

- **Authentication** everywhere
- **Authorization** for project-scoped routes
- Mapping between OpenAPI operations and Kong routes
- Dependency on human knowledge

Scaling

- Single team managing all configs
- Time to onboard new services
- Ownership of config
- Copy/paste from OpenAPI doc
- Propagation between environments
- Syntax requires Kong knowledge

GOALS

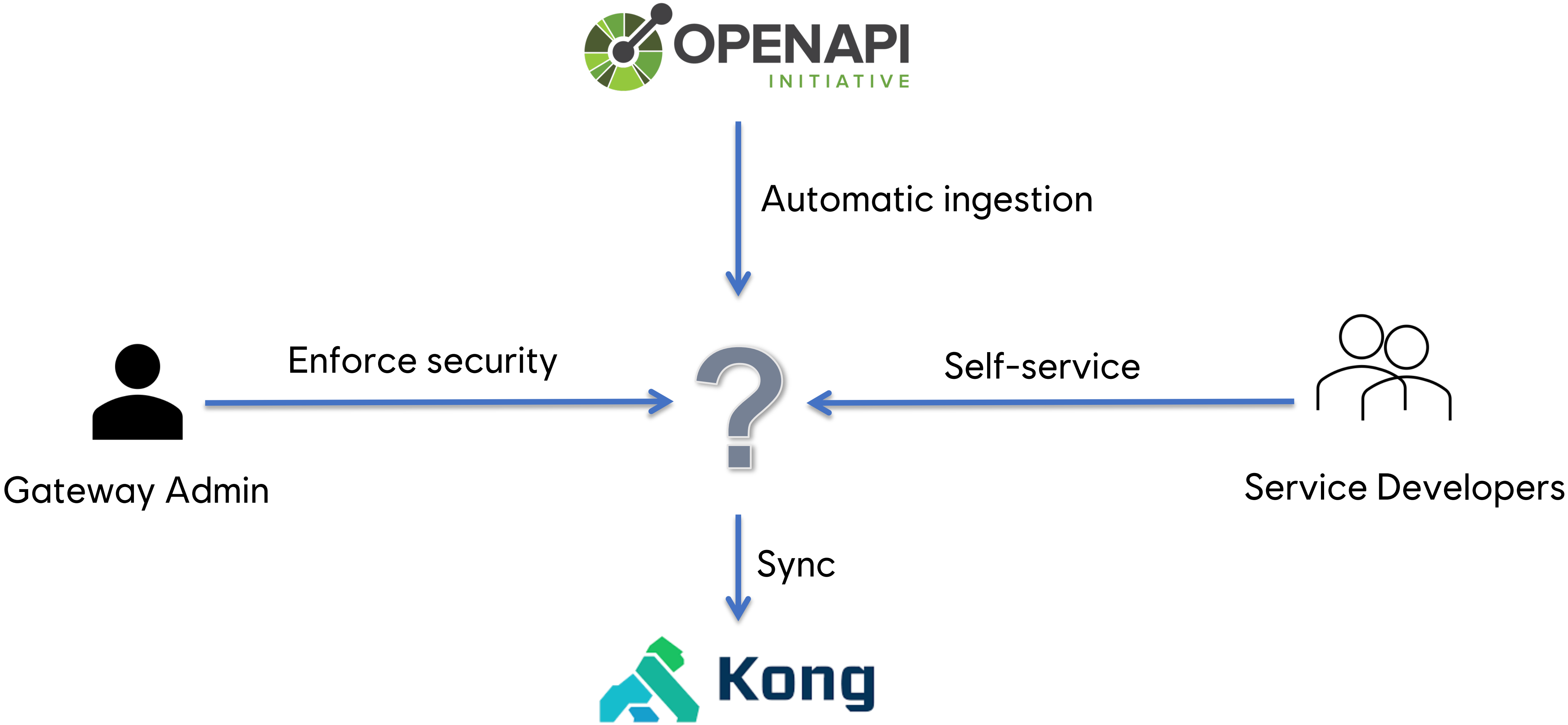
Safe

- **Auto-generation** from OpenAPI
  - Enforced path prefix
- Enforced security
  - **Defined by gateway admins**
  - Authentication everywhere
  - Automatic authorization
- Config **validation**
- Self healing

Shift left

- **Self-service** for service devs
- Autonomy
- Don't require Kong knowledge
- Don't hide Kong
  - Plugins ecosystem…
- **Clear ownership**

GOALS



OPENAPI INITIATIVE

Automatic ingestion

Gateway Admin

Enforce security

?

Self-service

Service Developers

Sync

Kong

KUBERNETES OPERATOR

Why?

✅ Self Service
✅ Self Healing
✅ Validations
✅ Kubernetes API

❓ Enforcing our own rules
❓ Based on OpenAPI

Which one?

Kong Ingress Controller 2.9

kubebuilder

Kubernetes Gateway API

ALTERNATIVES

Kong Ingress Controller

- Ingress + Gateway API
- **Custom** Resources

✅ Kong-integration

〜 OpenAPI-based routing

❌ Enforced rules by admins
❌ Kong plugin precedence

Ingress API

- **Ingress** resource

✅ Stable & well-known API

❌ OpenAPI-based routing
❌ Enforced rules by admins

Gateway API

- **Gateway** resource
- **HTTPRoute** resource

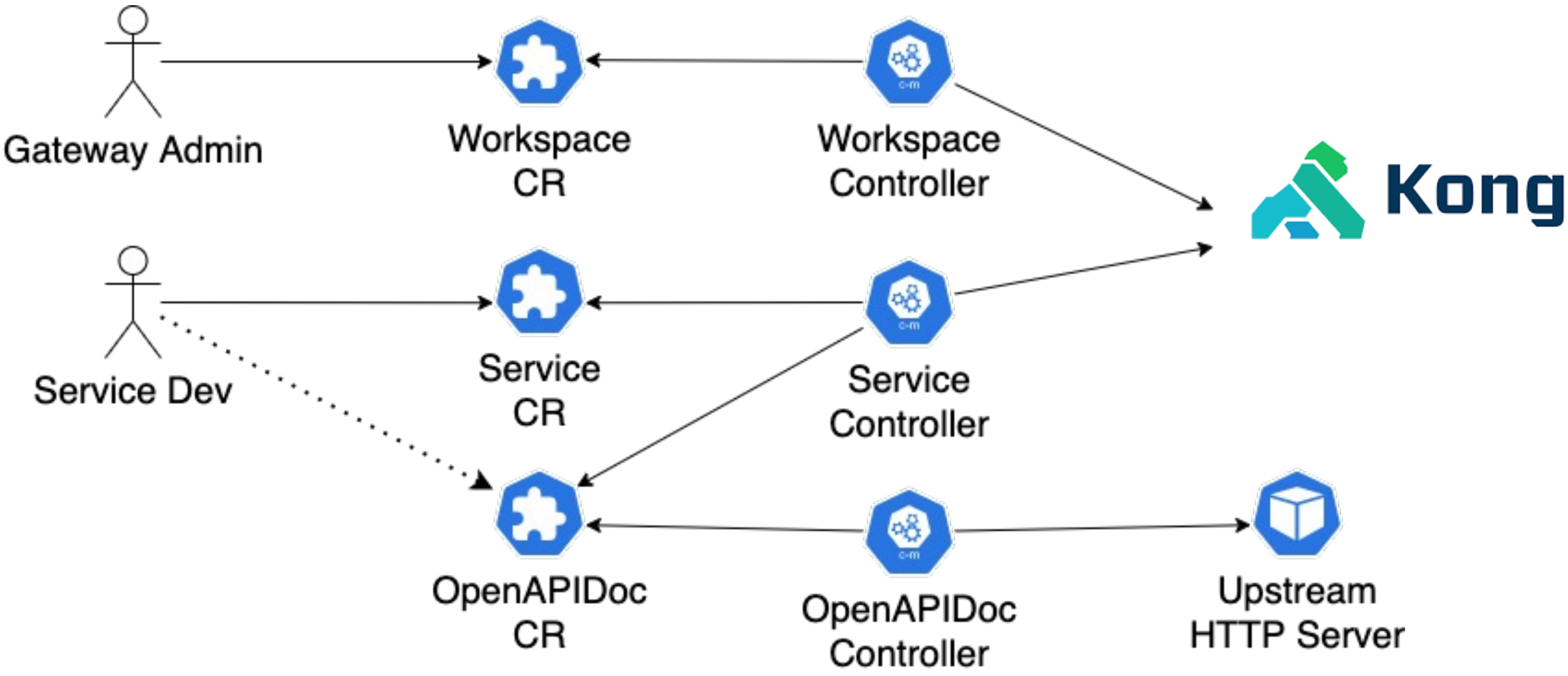
✅ admin & dev roles

〜 Enforced rules by admins

❌ Beta
❌ OpenAPI-based routing
❌ Complex to implement

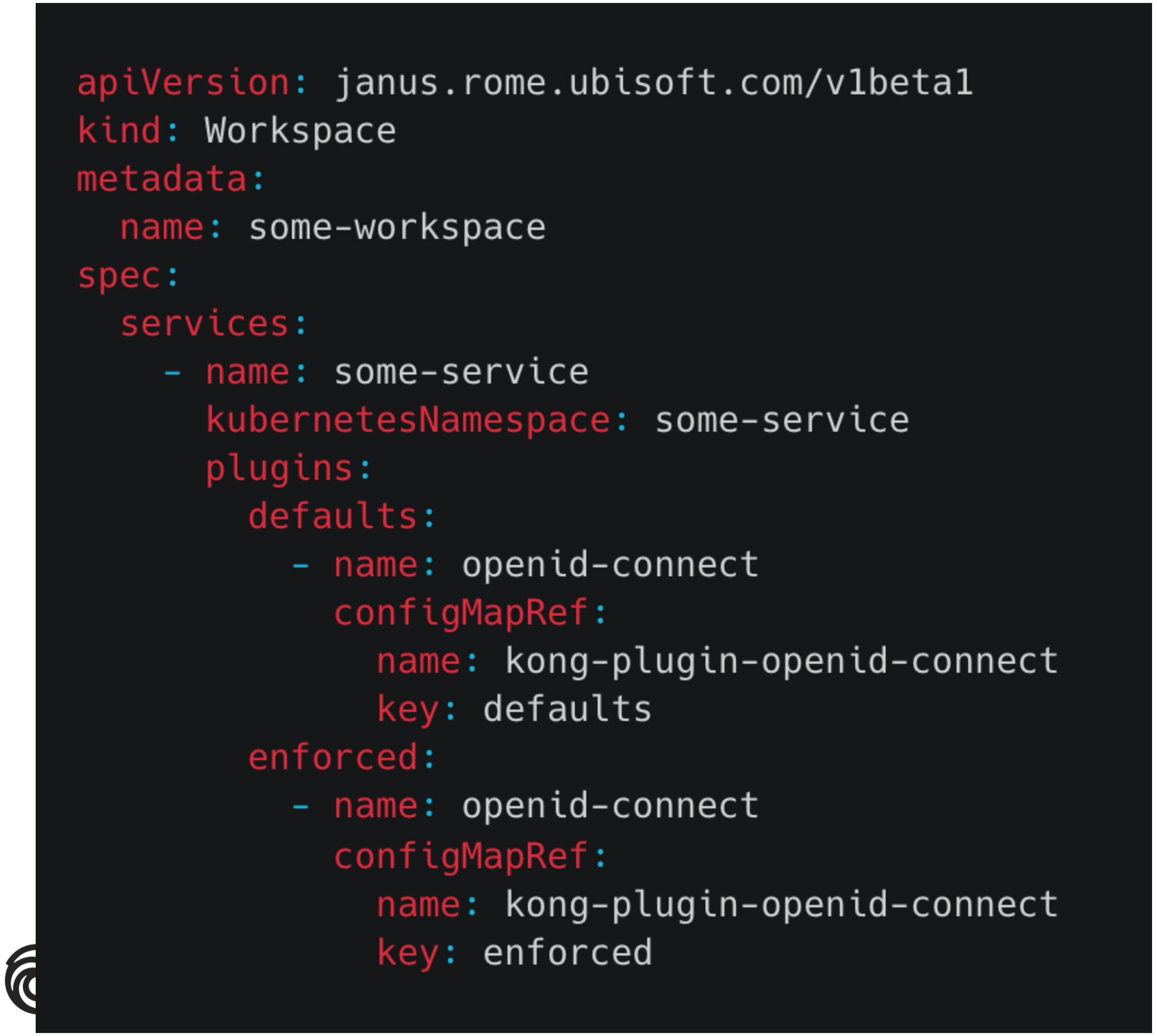
CUSTOM OPERATOR

CRDs & Controllers

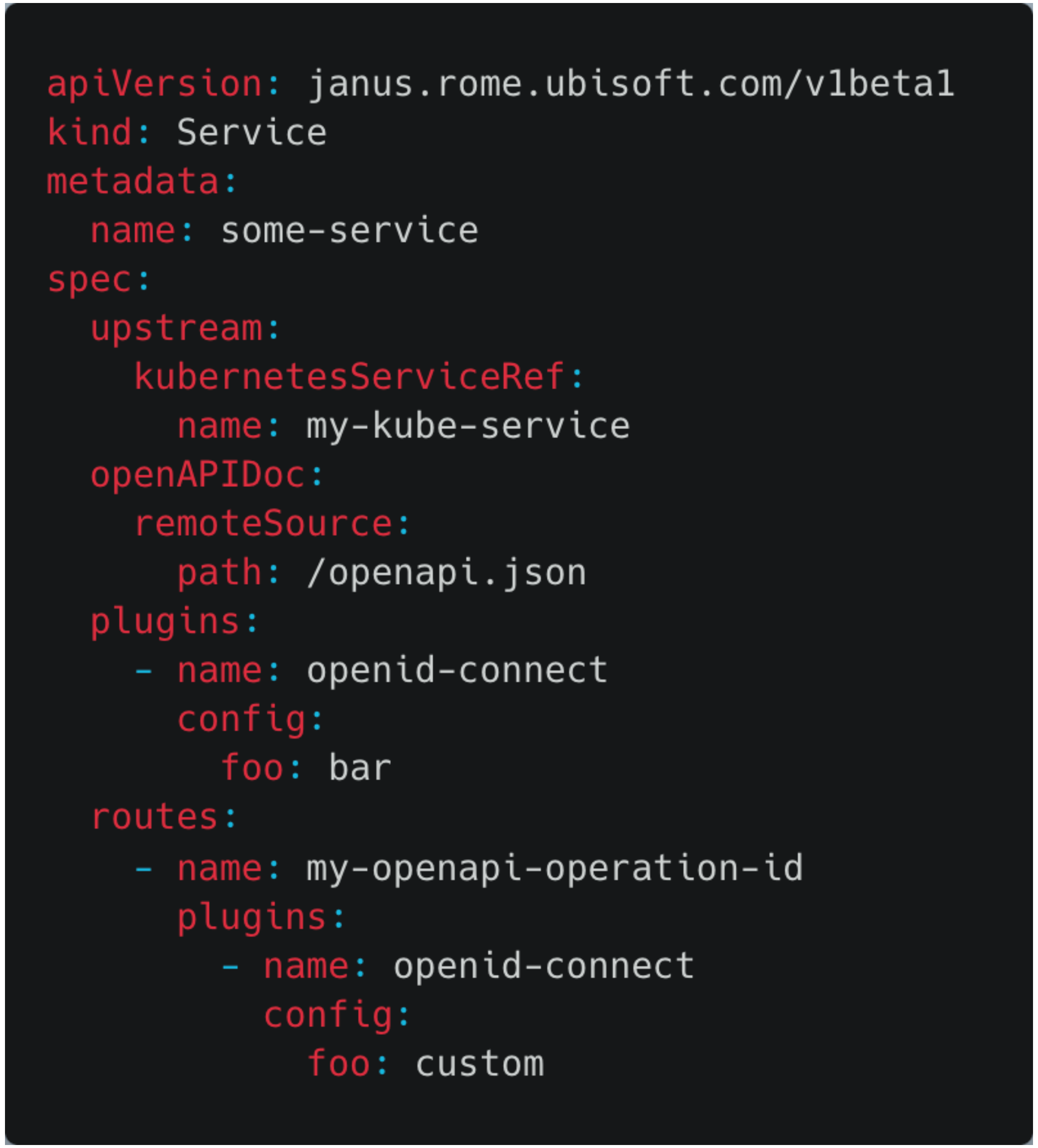


CRD SAMPLES

Workspace & Service

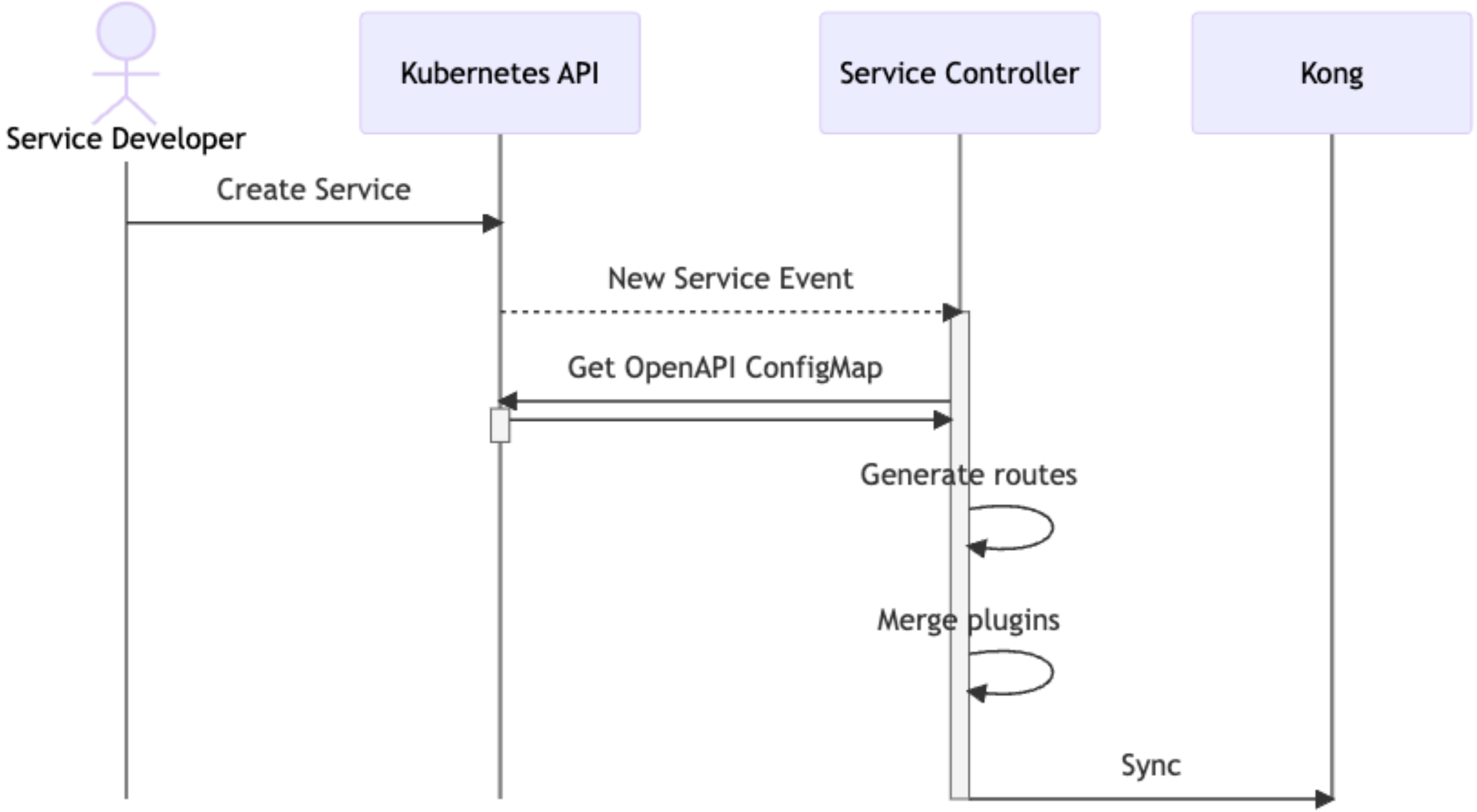


```yaml
apiVersion: janus.rome.ubisoft.com/v1beta1
kind: Workspace
metadata:
  name: some-workspace
spec:
  services:
    - name: some-service
      kubernetesNamespace: some-service
      plugins:
        defaults:
          - name: openid-connect
            configMapRef:
              name: kong-plugin-openid-connect
              key: defaults
        enforced:
          - name: openid-connect
            configMapRef:
              name: kong-plugin-openid-connect
              key: enforced
```



```yaml
apiVersion: janus.rome.ubisoft.com/v1beta1
kind: Service
metadata:
  name: some-service
spec:
  upstream:
    kubernetesServiceRef:
      name: my-kube-service
  openAPIDoc:
    remoteSource:
      path: /openapi.json
  plugins:
    - name: openid-connect
      config:
        foo: bar
  routes:
    - name: my-openapi-operation-id
      plugins:
        - name: openid-connect
          config:
            foo: custom
```

SERVICE CONTROLLER



KONG SYNC

From scratch, or extending?

Extending Kong Ingress Controller

- Generate Kong custom resources

✅ Kubernetes API only

〜 Simple reconciliation

❌ 2 operators to manage
❌ Extra latency
❌ Limited status

Building from scratch

- Interact with Kong Admin API

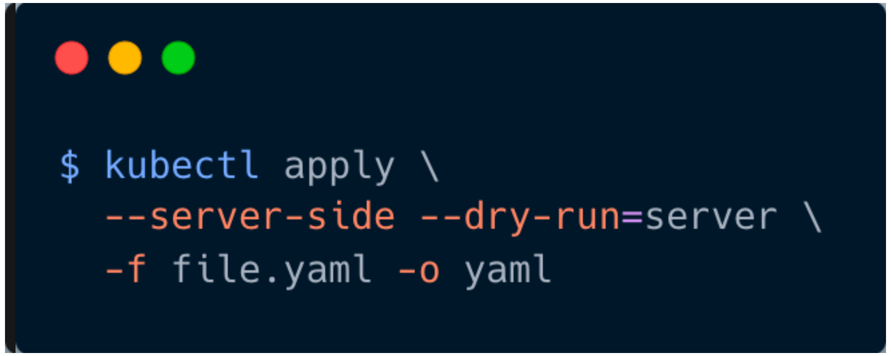
✅ Reconcile against Kong API (DecK)
✅ Less components
✅ Full access to Kong status
✅ **User experience**: dry-run

USER EXPERIENCE

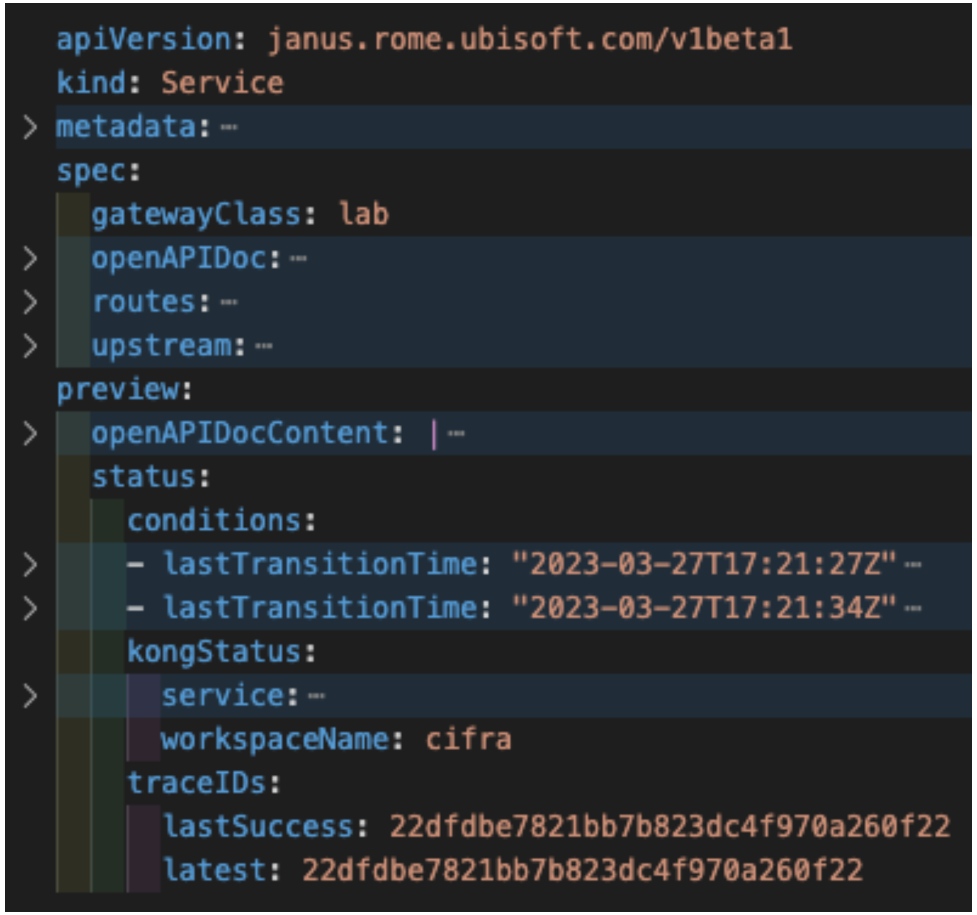
Self service for the service developers…

Dry Run

- Server-side apply
- Server-side dry-run
- Mutating admission webhook



```
$ kubectl apply \
  --server-side --dry-run=server \
  -f file.yaml -o yaml
```



```yaml
apiVersion: janus.rome.ubisoft.com/v1beta1
kind: Service
metadata: ⋯
spec:
  gatewayClass: lab
  openAPIDoc: ⋯
  routes: ⋯
  upstream: ⋯
preview:
  openAPIDocContent: | ⋯
  status:
    conditions:
    - lastTransitionTime: "2023-03-27T17:21:27Z" ⋯
    - lastTransitionTime: "2023-03-27T17:21:34Z" ⋯
    kongStatus:
      service: ⋯
      workspaceName: cifra
    traceIDs:
      lastSuccess: 22dfdbe7821bb7b823dc4f970a260f22
      latest: 22dfdbe7821bb7b823dc4f970a260f22
```

USER EXPERIENCE

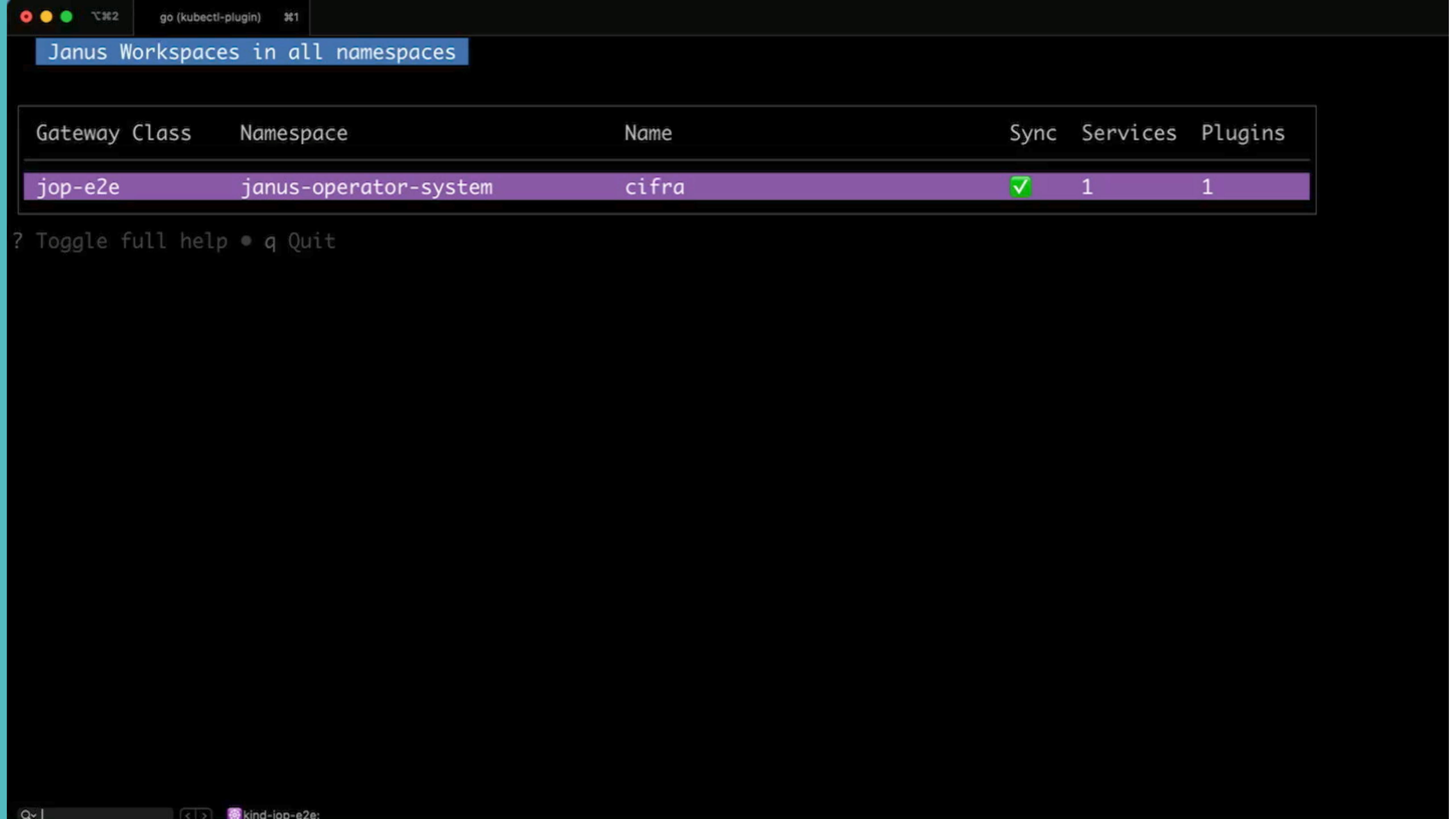
What makes a good operator?

User Feedback

- Status field
- Conditions
- Events

Kubectl plugin

- **Interactive** visualization
- Aggregated view
- Integration with other systems
    - Prometheus metrics
    - OpenTelemetry traces



```
 Janus Workspaces in all namespaces

 Gateway Class    Namespace                      Name                       Sync  Services  Plugins
 ──────────────────────────────────────────────────────────────────────────────────────────────────
 jop-e2e          janus-operator-system          cifra                      ✅    1         1

? Toggle full help • q Quit
```

CONCLUSION

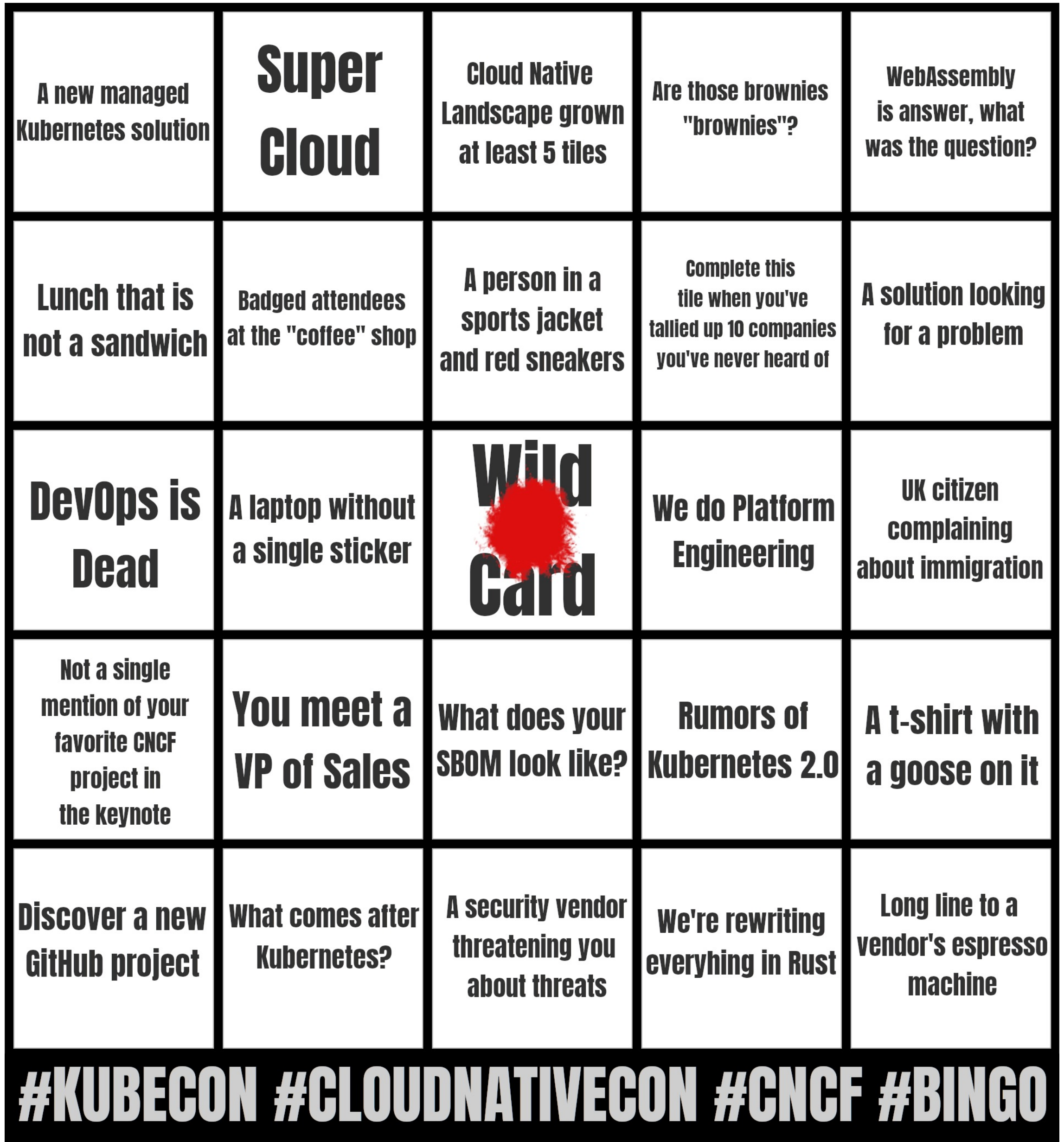
When to build your own operator?

- Self service
- Kubernetes API
- Well defined use-case(s)
  - Avoid versions upgrade
- Go / Kubernetes experience
- No existing solution

Benefits

- Complete control
  - Flexibility
  - Custom integrations
- **User experience**

BINGO GAME



A new managed Kubernetes solution	Super Cloud	Cloud Native Landscape grown at least 5 tiles	Are those brownies "brownies"?	WebAssembly is answer, what was the question?
Lunch that is not a sandwich	Badged attendees at the "coffee" shop	A person in a sports jacket and red sneakers	Complete this tile when you've tallied up 10 companies you've never heard of	A solution looking for a problem
DevOps is Dead	A laptop without a single sticker	Wild Card	We do Platform Engineering	UK citizen complaining about immigration
Not a single mention of your favorite CNCF project in the keynote	You meet a VP of Sales	What does your SBOM look like?	Rumors of Kubernetes 2.0	A t-shirt with a goose on it
Discover a new GitHub project	What comes after Kubernetes?	A security vendor threatening you about threats	We're rewriting everyhing in Rust	Long line to a vendor's espresso machine

#KUBECON #CLOUDNATIVECON #CNCF #BINGO

KubeCon | CloudNativeCon

Europe 2023

Thank you.
Questions?

Please scan the QR Code above
to leave feedback on this session