

# Kubernetes as an Abstraction Layer for a Connected Home

KubeCon + CloudNativeCon - Europe 2018



Scott Nichols  
@n3wscott



Kubernetes



OPEN SERVICE  
BROKER API



Service Catalog



GKE





Kubernetes



OPEN SERVICE  
BROKER API



Service Catalog



GKE

# In collaboration with



# I had an idea...



+



+



Service Catalog

+



=





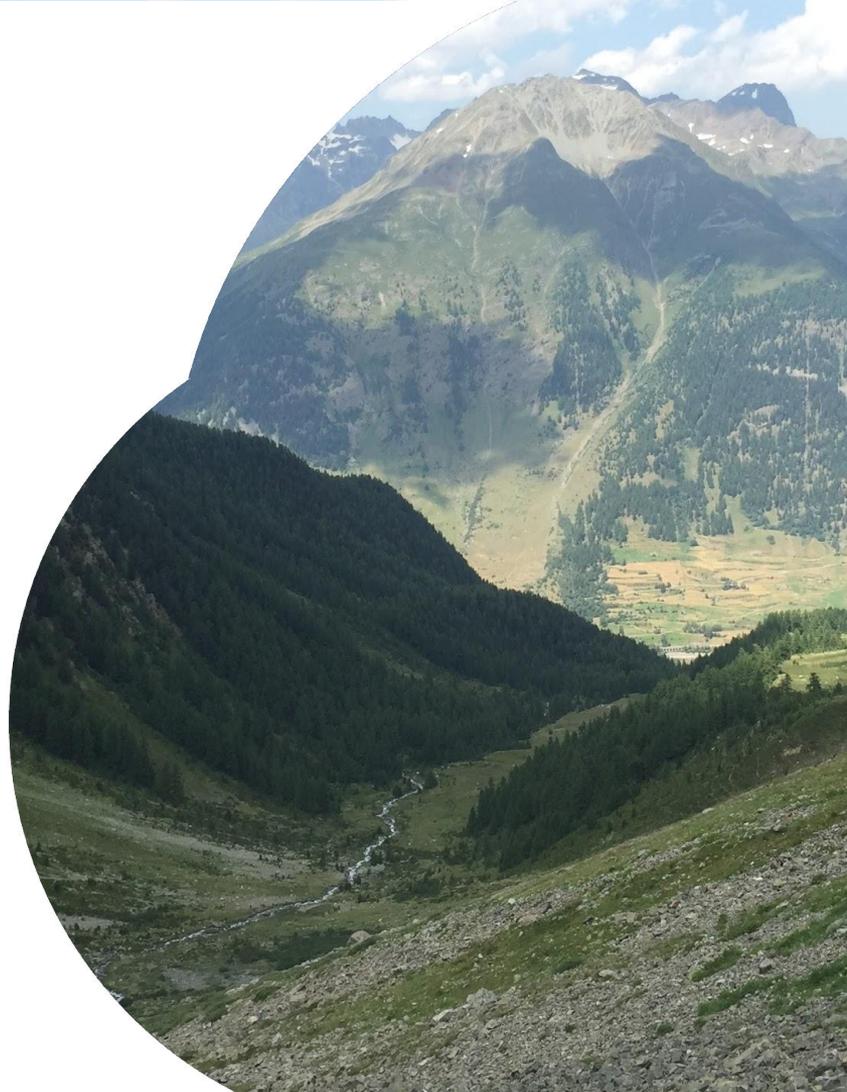
What if I pretend my house is an existing limited resource on-prem deployment solution, my smart lights are services, and I use Kubernetes to control them from the cloud.



# Because...

- having something tangible to explain
- pain point in my past experience
- migrating between deployment solutions
- expansion into the cloud
- hybrid?
- maybe a bit of fun :D

This is that journey.



# Agenda

Connected Home Idea

Overview of Open Service Broker API

Abridged Kubernetes Service Catalog

Peek at Google Cloud Platform Service Broker

LedHouse Deep Dive

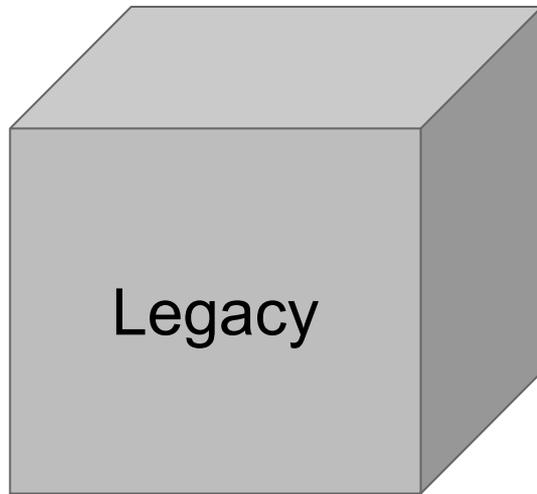


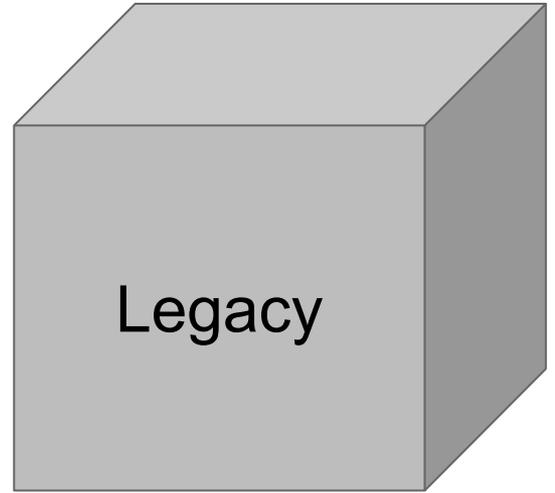
# A Connected Home

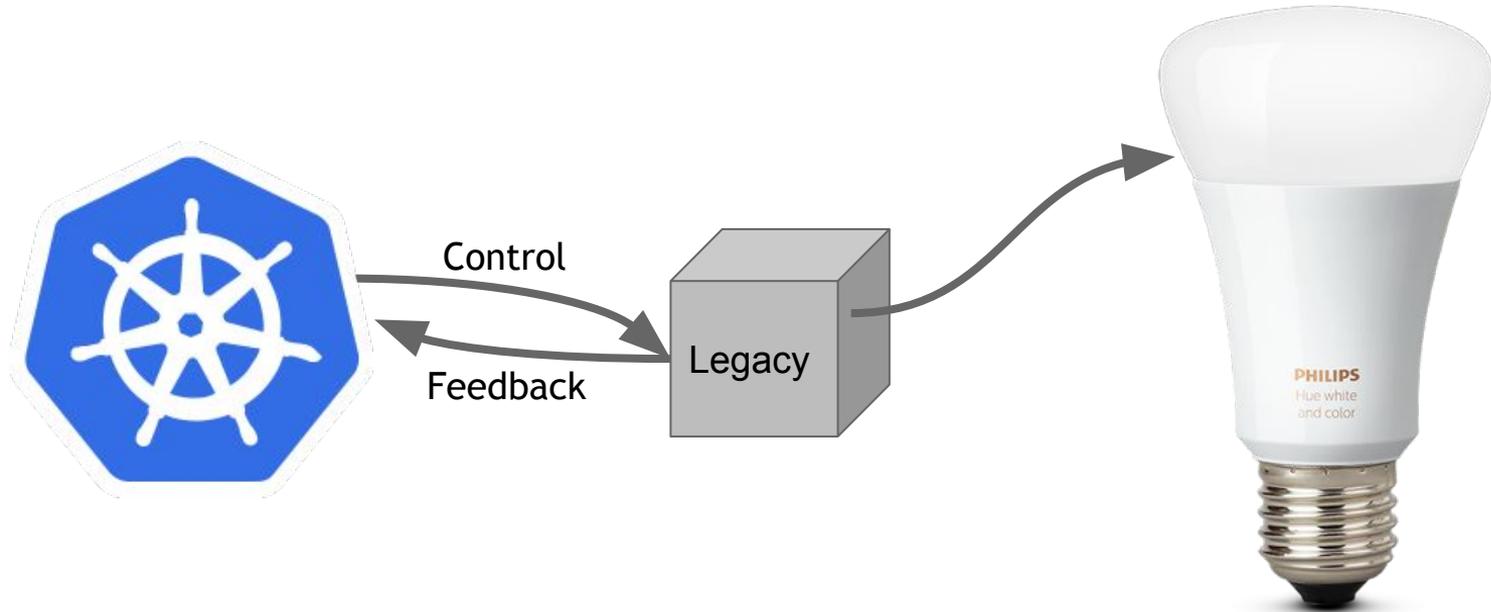
a metaphor for on-prem

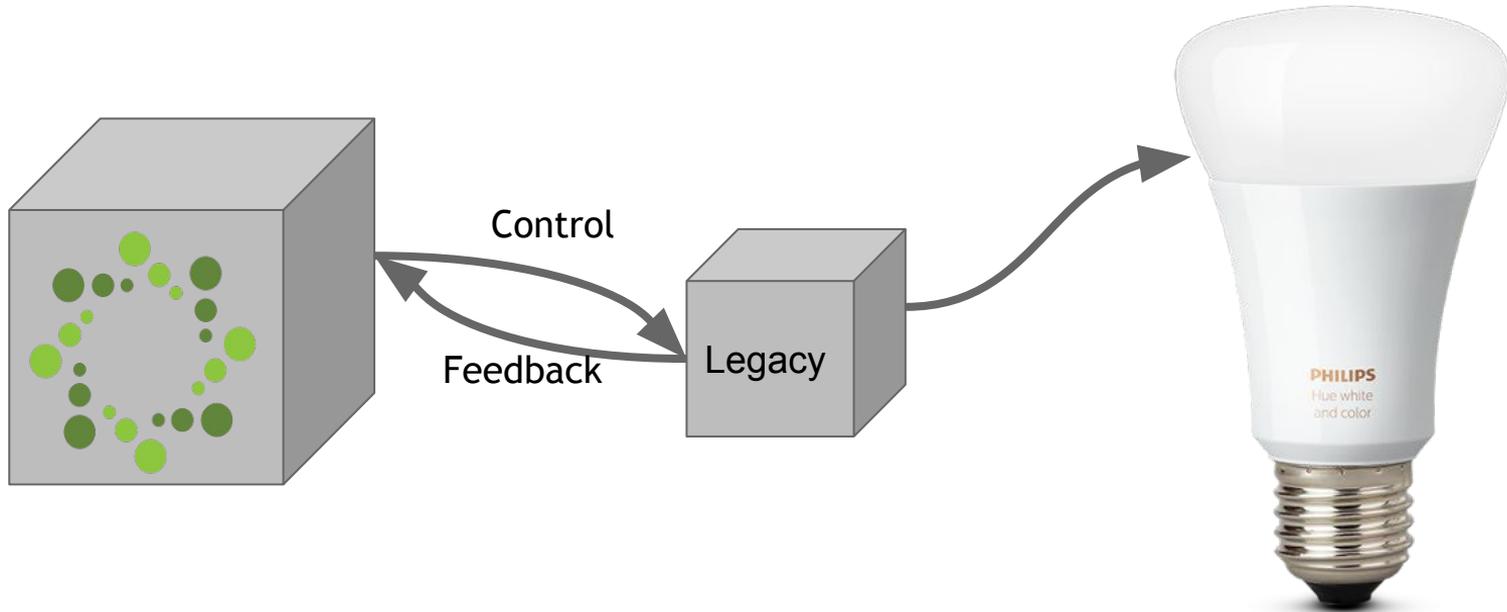








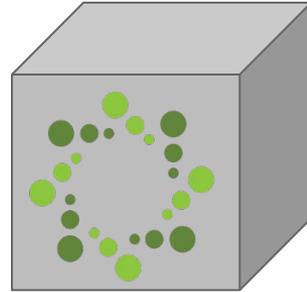








Service Catalog



# Components in play

## Kubernetes

Kubernetes is a portable, extensible open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.

## Open Service Broker API

The Open Service Broker API project allows developers, ISVs, and SaaS vendors a single, simple, and elegant way to deliver services to applications running within cloud native platforms

## Service Catalog

Service Catalog is an extension API that enables applications running in Kubernetes clusters to easily use external managed software offerings, such as a datastore service offered by a cloud provider. It provides a way to list, provision, and bind with external Managed Services. from Service Brokers.



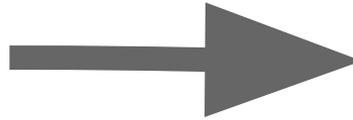
# Open Service Broker API

an overview





# OPEN SERVICE BROKER API

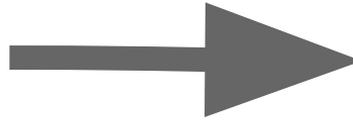


- Service Broker
- Catalog
- Service Offering
- Service Plan
- Service Instance
- Service Binding





OPEN SERVICE  
BROKER API



- GetCatalog
- Provision
- Bind
- Update
- Unbind
- Deprovision



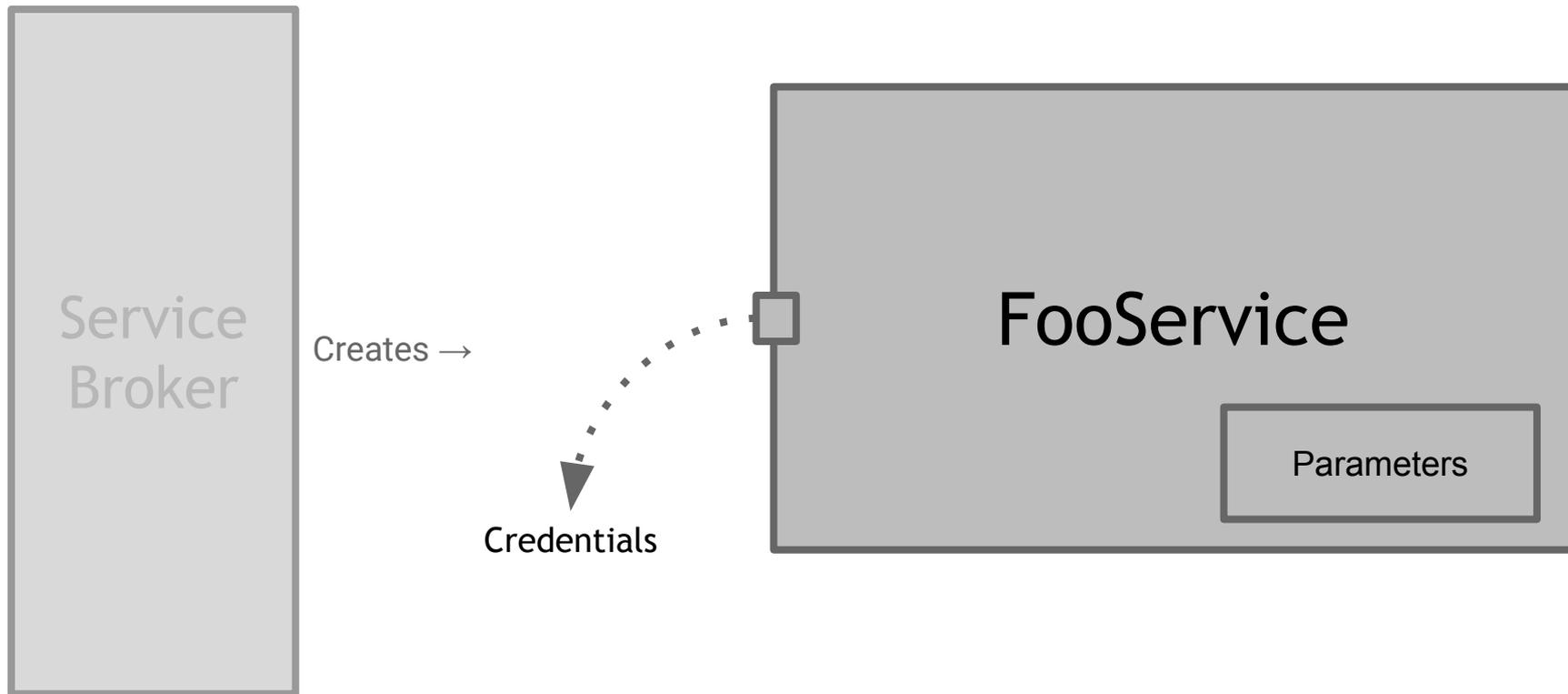
# Service Broker

Service Brokers manage the lifecycle of Services. Platforms interact with Service Brokers to provision, and manage, Service Instances and Service Bindings.



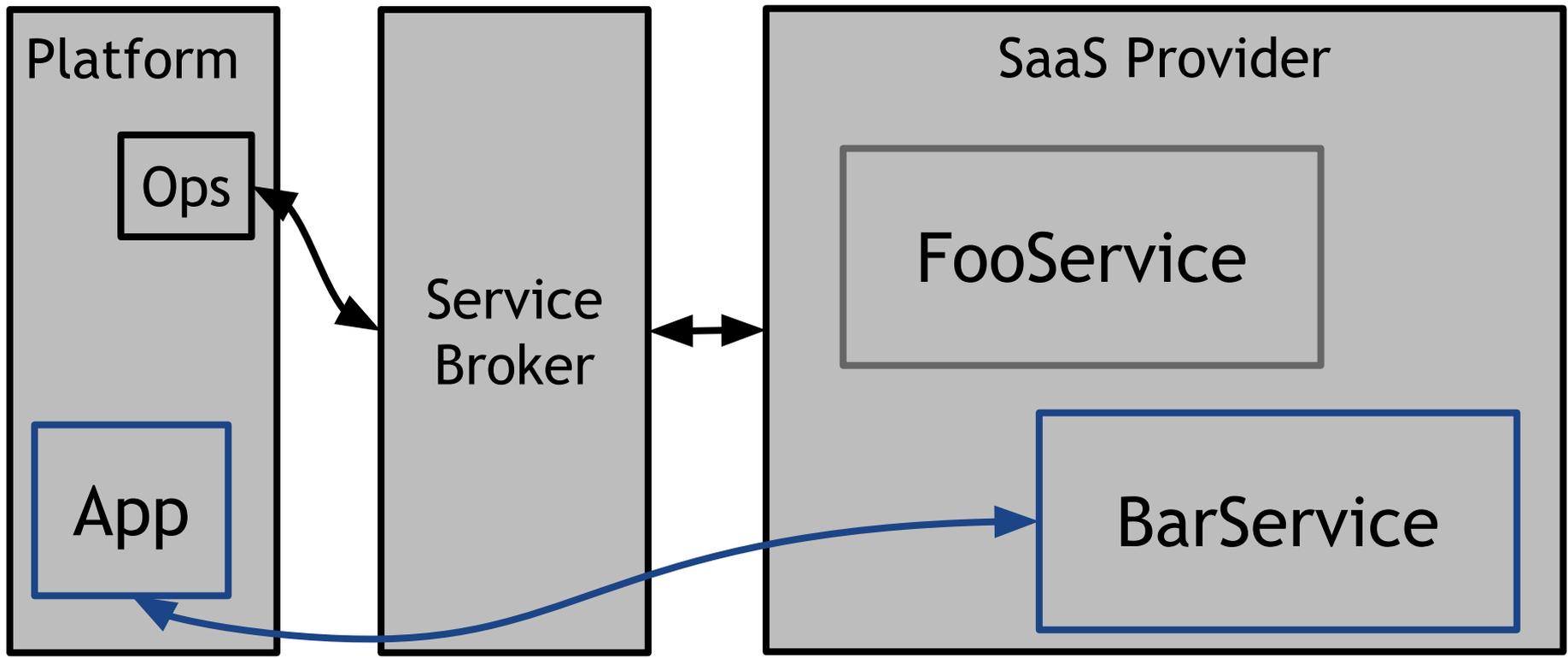


# OPEN SERVICE BROKER API



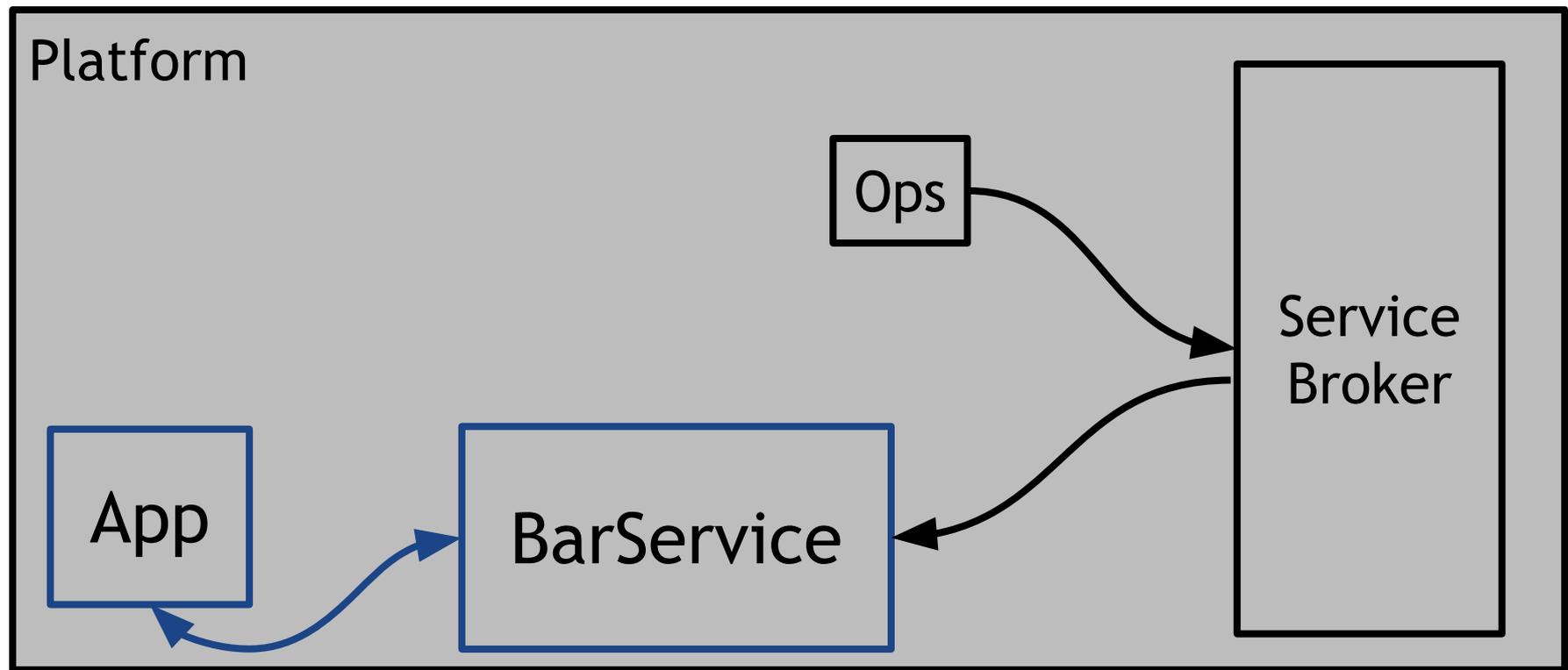


# OPEN SERVICE BROKER API





# OPEN SERVICE BROKER API



# What is in a Catalog?

## Service Offering

The advertisement of a service that a Service Broker supports.

## Service Plan

The representation of the costs and benefits for a given variant of the service, potentially as a tier.



```
$ curl "http://my-broker/v2/catalog"
```

```
{
  "services": [{
    "id": "location-1b",
    "name": "1B",
    "description": "A set of lights in 1B",
    "bindable": true,
    "plans": [{
      "id": "location-1b-kind-red",
      "name": "Red",
      "description": "Light type Red"
    }, {
      "id": "location-1b-kind-green",
      "name": "Green",
      "description": "Light type Green"
    }, {
      "id": "location-1b-kind-blue",
      "name": "Blue",
      "description": "Light type Blue"
    }
  ]
}, ...]
}
```

# Get Catalog





# OPEN SERVICE BROKER API

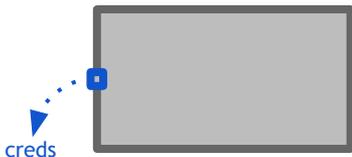
## Provision

Creates an instance of a Service Offering.



## Bind

Grants access to a Service Instance.



## Update

Modify parameters of a Service Instance.



## Unbind

Remove access to a Service Instance.



## Deprovision

Destroy a Service Instance.



# Google Cloud Platform Service Broker

## Cloud Pub/Sub - Nouns

### Service Offering

A Topic.

### Service Plan

Not used.

### Binding

- A. Access to the topic.
- B. Creation + Access of a Subscription for the Topic.



# Google Cloud Platform Service Broker

## Cloud Pub/Sub - Verbs

### Provision

Creates an instance of a **Topic**

### Bind

- A. *Publisher*, grants a service account access to **Topic**.
- B. *Subscriber*, creates and grants access to a **subscription**.

### Update

Modifies *some* parameters, such as queue timeout and push/pull subscription support.

### Unbind

- A. Revokes access for publisher service account.
- B. Deletes Subscription and revokes access.

### Deprovision

Deletes the Topic.



# Google Cloud Platform Service Broker

## Cloud MySQL - Nouns

Service Offering

A database + schema.

Service Plan

Not used.

Binding

Auth to that database and schema.



# Google Cloud Platform Service Broker

## Cloud MySQL - Verbs

### Provision

Creates a database, provide schema via parameters, database settings.

### Bind

Grants a service account access to the database and provides the url for the database.

### Update

Updates schemas and database settings.

### Unbind

Revokes access for service account.

### Deprovision

Removes the instance of the database.



# Light Broker for LedHouse

## Nouns

### Service Offering

A Light in a **Room**.

### Service Plan

A type of Light.

### Binding

A token for the Light Registry, via HTTP or Pub/Sub.



# Light Broker for LedHouse

## Verbs

Provision

Reserve a light from the Light Registry.

Bind

Get a token to be used in the Light Registry API, the Light Registry URL, and the Light Registry Topic it is monitoring.

Update

Not supported.

Unbind

Delete the access token.

Deprovision

Release the light.



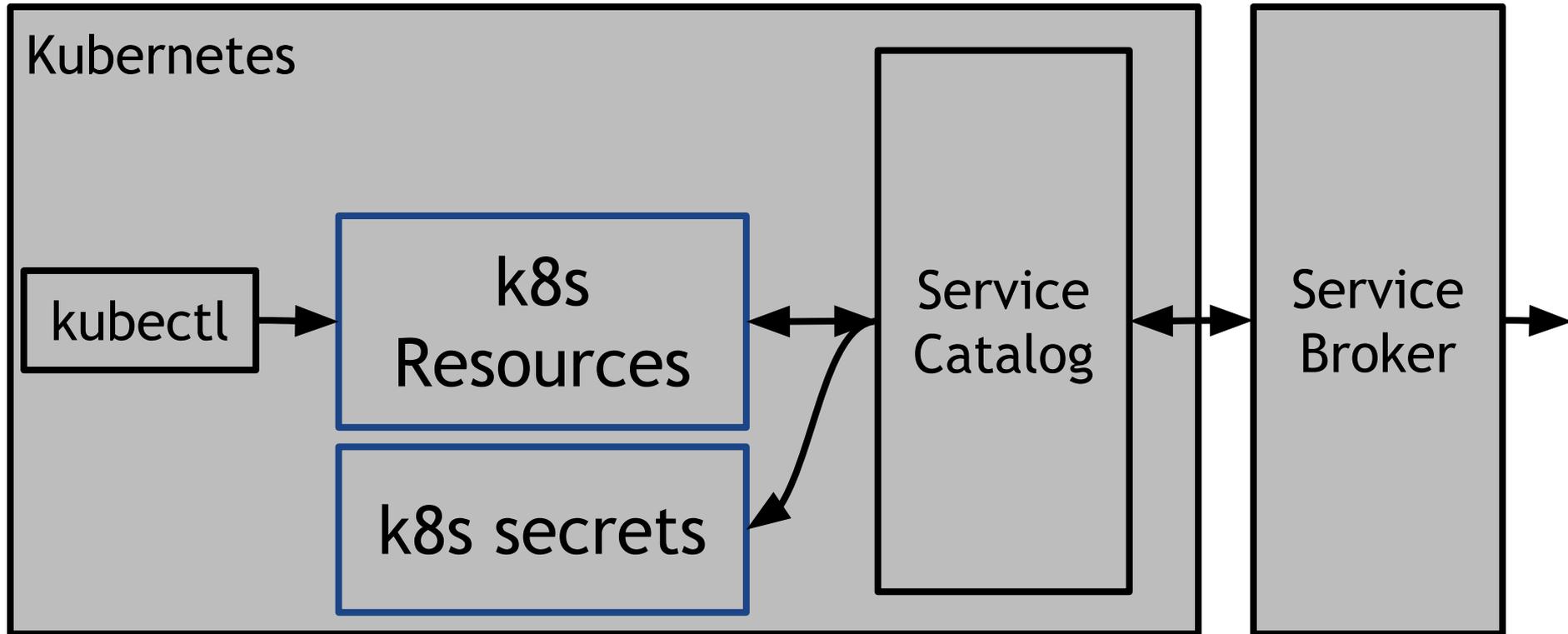
# Kubernetes Service Catalog

an overview





# Service Catalog



# Service Catalog integration details

- Currently an Aggregate API Server
- Provides new resource types for k8s
- A ServiceInstance is the combo of Service Class + Plan + Parameters
- Async is supported

Service Catalog translates *declarative* k8s actions into *imperative* OSBAPI Broker calls





# Service Catalog

Provision

Create a  
ServiceInstance

Bind

Create a  
ServiceBinding, get a  
Secret

Update

Update a  
ServiceInstance

Unbind

Delete a  
ServiceBinding

Deprovision

Delete a  
ServiceInstance



# Google Cloud Platform Service Broker

an introduction



# Google Cloud Platform Service Broker

To make it easier to connect to Google Cloud Platform (GCP) services from either a GCP-hosted Kubernetes cluster or an on-premises Kubernetes cluster, we are releasing a new services framework: Kubernetes Service Catalog, a collection of services available to Kubernetes running on GCP, and the Google Cloud Platform Service Broker, a hosted service that connects to a variety of GCP services. These offerings are based on the Kubernetes Catalog SIG and the Open Service Broker API.

Based on open-source APIs, Kubernetes Service Catalog and the Service Broker give you access to a rich ecosystem of services to incorporate into your applications. Brokers for Cloud Foundry and other environments are already available.

This beta release allows you to focus on the services you need to get your job done without the hassle of knowing how the services are built or worrying about the infrastructure you need to run them. Support for the Kubernetes Service Catalog will be rolling out in the Google Cloud Console UI over the next few days. We'll add more GCP services to the Service Broker as we move forward, opening up a whole new range of services for your applications.



# Google Cloud Platform Service Broker

⊙ `svcat get class`

NAME

DESCRIPTION

NAME	DESCRIPTION
<code>cloud-spanner</code>	The first horizontally scalable, strongly consistent, relational database service
<code>cloud-iam-service-account</code>	Specialized service which provisions Google Cloud Platform IAM service accounts
<code>cloud-pubsub</code>	Ingest event streams from anywhere, at any scale, for simple, reliable, real-time stream analytics
<code>cloud-sql-mysql</code>	A fully-managed MySQL database service
<code>bigquery</code>	A fast, highly scalable, cost-effective and fully-managed enterprise data warehouse for analytics at any scale
<code>cloud-bigtable</code>	A high performance NoSQL database service for large analytical and operational workloads
<code>cloud-storage</code>	Google Cloud Storage is unified object storage for developers and enterprises

# Google Cloud Platform Service Broker, Simple app deployed in GKE:

```
apiVersion: servicecatalog.k8s.io/v1beta1
kind: ServiceInstance
metadata:
  name: foo
  namespace: pubsub
spec:
  clusterServiceClassExternalName:
cloud-pubsub
  clusterServicePlanExternalName: beta
  parameters:
    topicId: local-to-proxy
```

```
---
apiVersion: servicecatalog.k8s.io/v1beta1
kind: ServiceBinding
metadata:
  name: foo-binding
  namespace: pubsub
spec:
  instanceRef:
    name: foo
  parameters:
    roles:
      - roles/pubsub.publisher
      - roles/pubsub.viewer
```

---

```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: publisher
  namespace: pubsub
spec:
  replicas: 1
  selector:
    matchLabels:
      app: publisher
  template:
    metadata:
      name: publisher
    labels:
      app: publisher
  spec:
    volumes:
      - name: service-account
        secret:
          secretName: foo-binding
  containers:
    - name: publisher
      image: gcr.io/foo:latest
      volumeMounts:
        - name: service-account
          mountPath:/service-account
```

```
env:
  - name: GOOGLE_APPLICATION_CREDENTIALS
    value:/privateKeyData
  - name: GOOGLE_CLOUD_PROJECT
    valueFrom:
      secretKeyRef:
        name: publisher
        key: projectId
  - name: PUBSUB_TOPIC
    valueFrom:
      secretKeyRef:
        name: publisher
        key: topicId
```

# Google Cloud Platform Service Broker, Simple app deployed in k8s on-prem:

```
apiVersion: servicecatalog.k8s.io/v1beta1
kind: ServiceInstance
metadata:
  name: foo
  namespace: pubsub
spec:
  clusterServiceClassExternalName:
cloud-pubsub
  clusterServicePlanExternalName: beta
  parameters:
    topicId: local-to-proxy
```

```
---
apiVersion: servicecatalog.k8s.io/v1beta1
kind: ServiceBinding
metadata:
  name: foo-binding
  namespace: pubsub
spec:
  instanceRef:
    name: foo
  parameters:
    roles:
      - roles/pubsub.publisher
      - roles/pubsub.viewer
```

```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
  name: publisher
  namespace: pubsub
spec:
  replicas: 1
  selector:
    matchLabels:
      app: publisher
  template:
    metadata:
      name: publisher
    labels:
      app: publisher
  spec:
    volumes:
      - name: service-account
        secret:
          secretName: foo-binding
  containers:
    - name: publisher
      image: gcr.io/foo:latest
      volumeMounts:
        - name: service-account
          mountPath:/service-account
```

```
env:
  - name: GOOGLE_APPLICATION_CREDENTIALS
    value:/privateKeyData
  - name: GOOGLE_CLOUD_PROJECT
    valueFrom:
      secretKeyRef:
        name: publisher
        key: projectId
  - name: PUBSUB_TOPIC
    valueFrom:
      secretKeyRef:
        name: publisher
        key: topicId
```

# Google Cloud Platform Service Broker

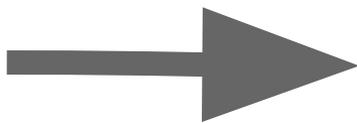
<https://cloud.google.com/kubernetes-engine/docs/concepts/add-on/service-broker>

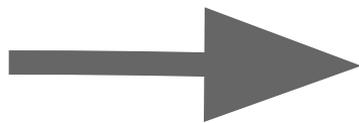


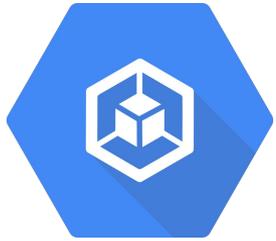
# LedHouse

a deep dive

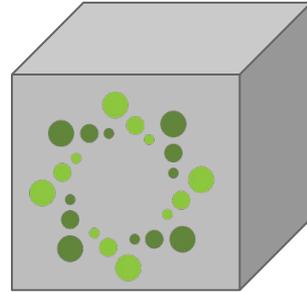




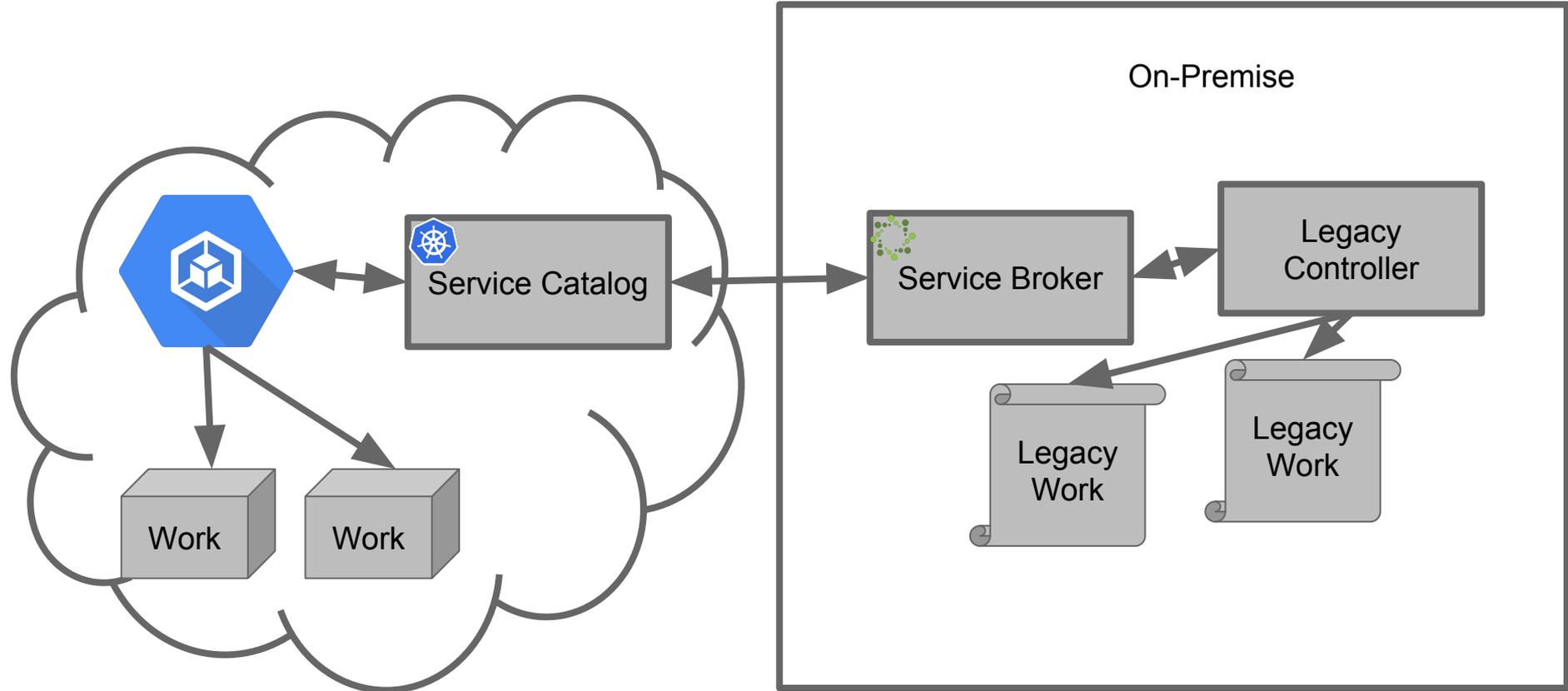




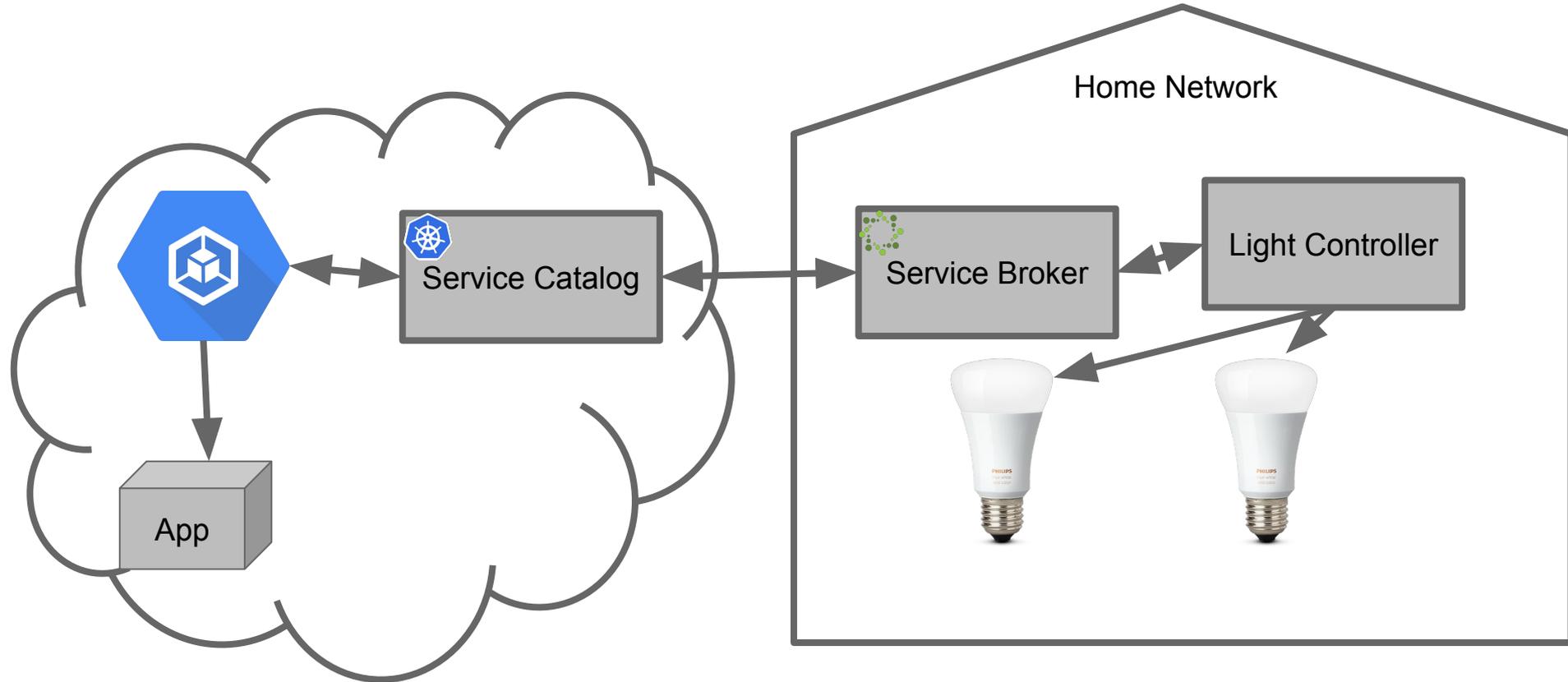
Service Catalog



# Hybrid.



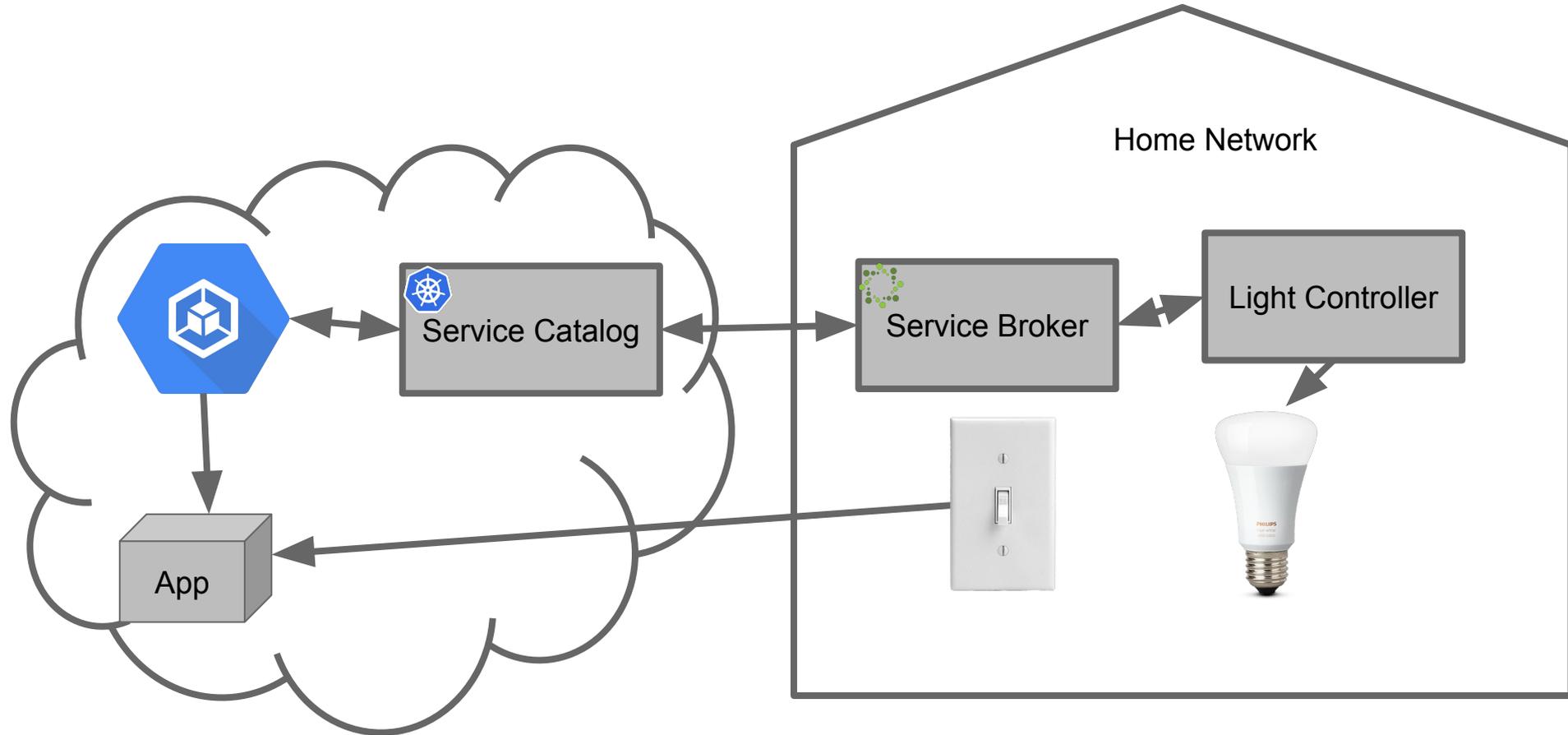
# Hybrid looks like home automation?



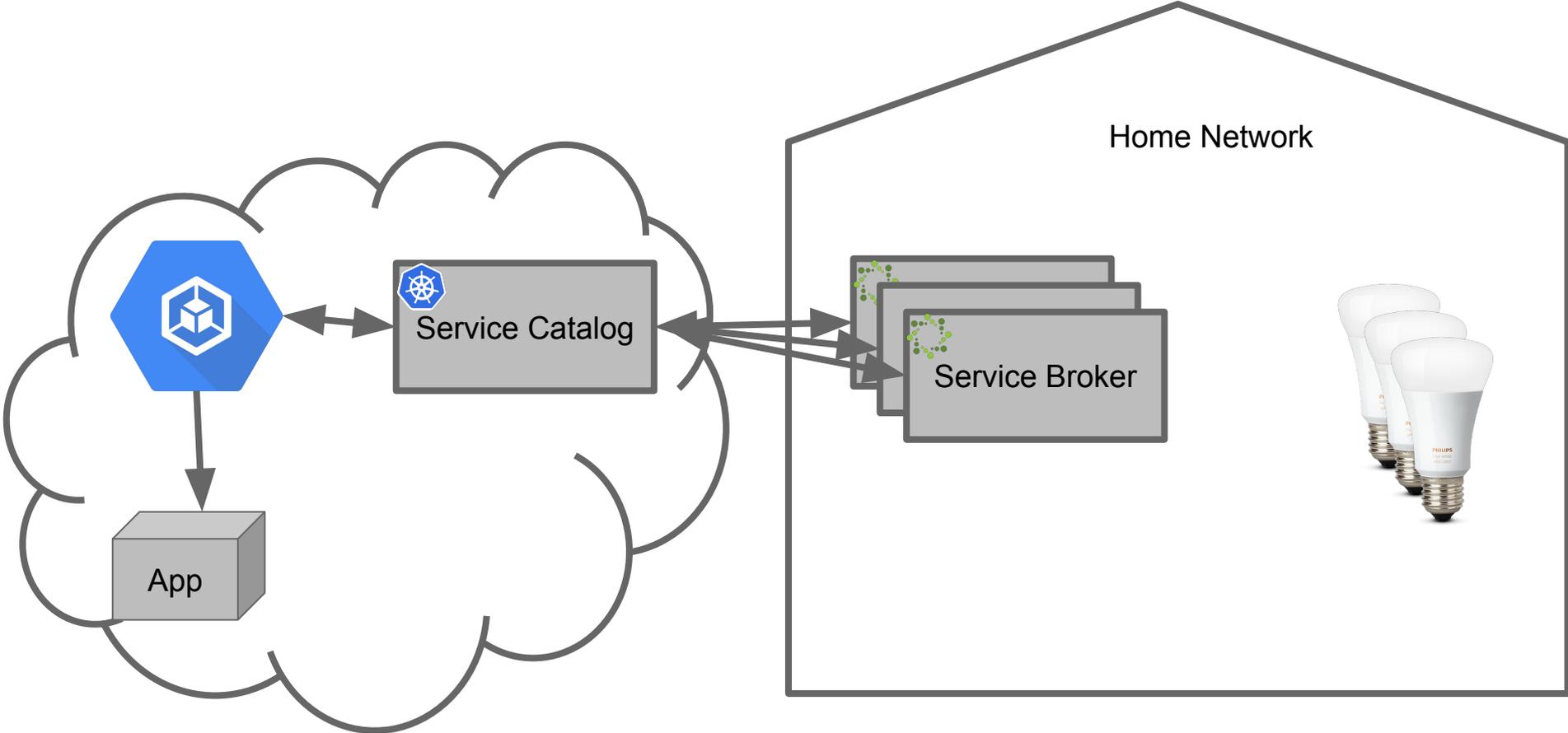
# Demo



# Extending functionality



# Multiple brokers...

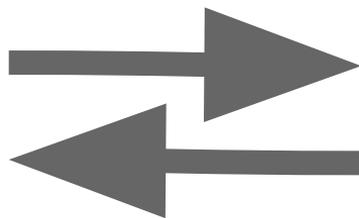


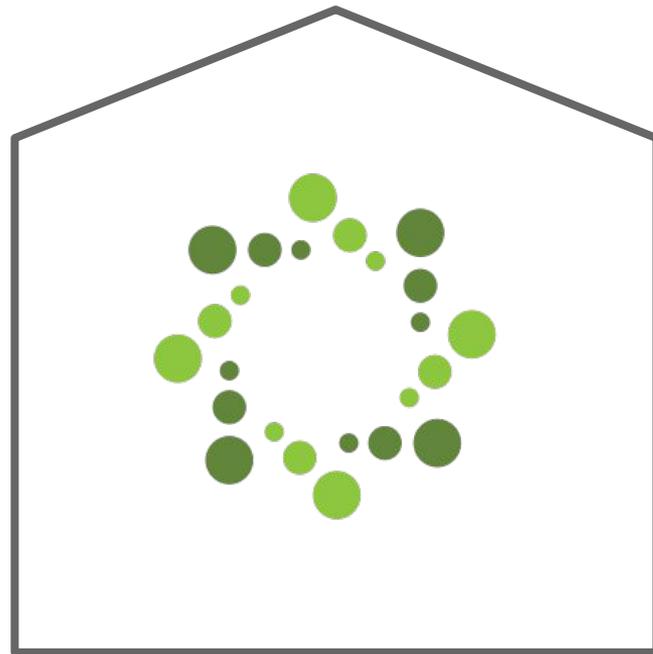
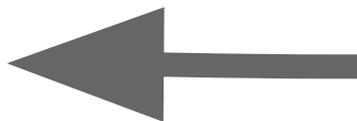


Lolz what? The cloud talks  
to your house? Sounds  
risky...

*Everyone in the audience.*

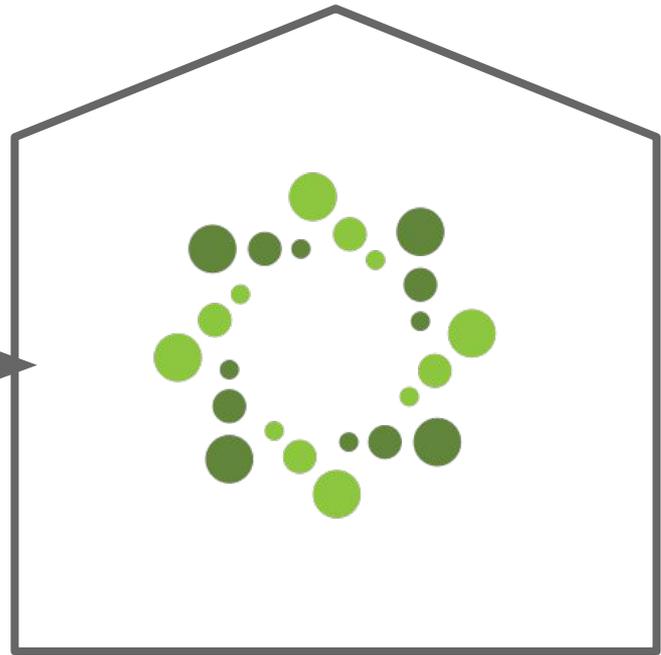


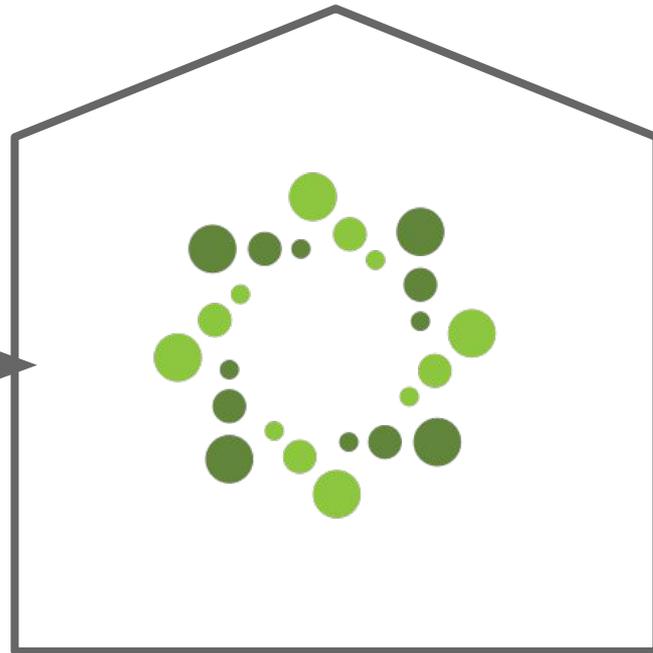


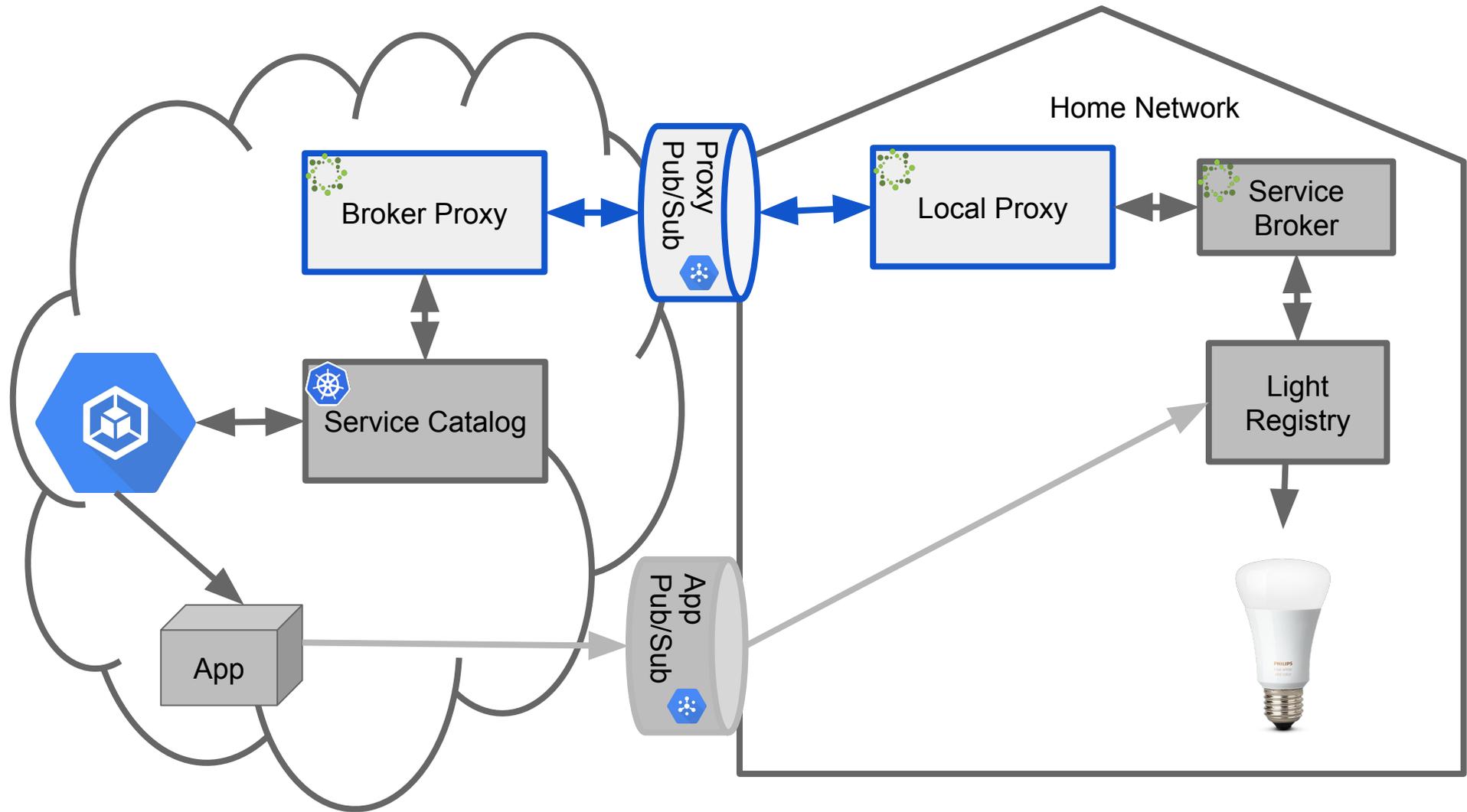


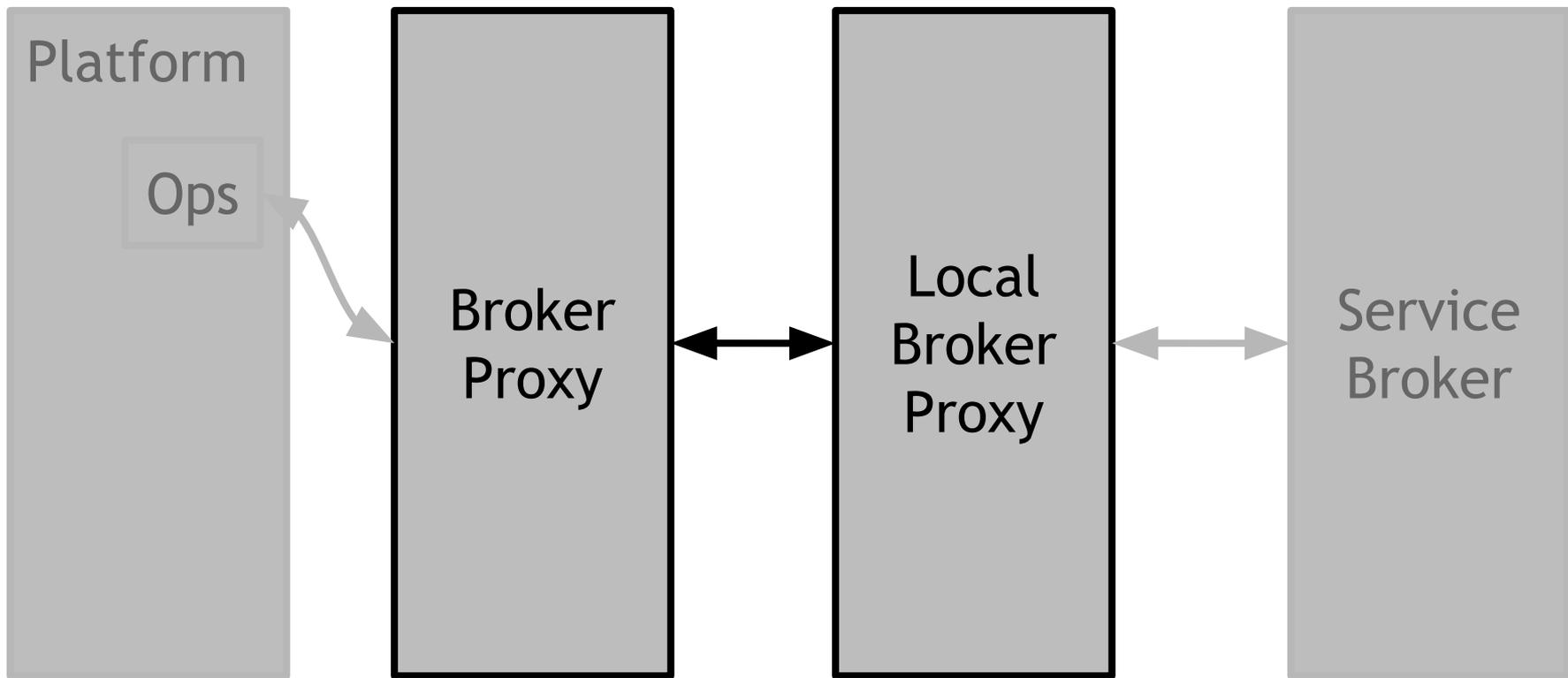


クラウド

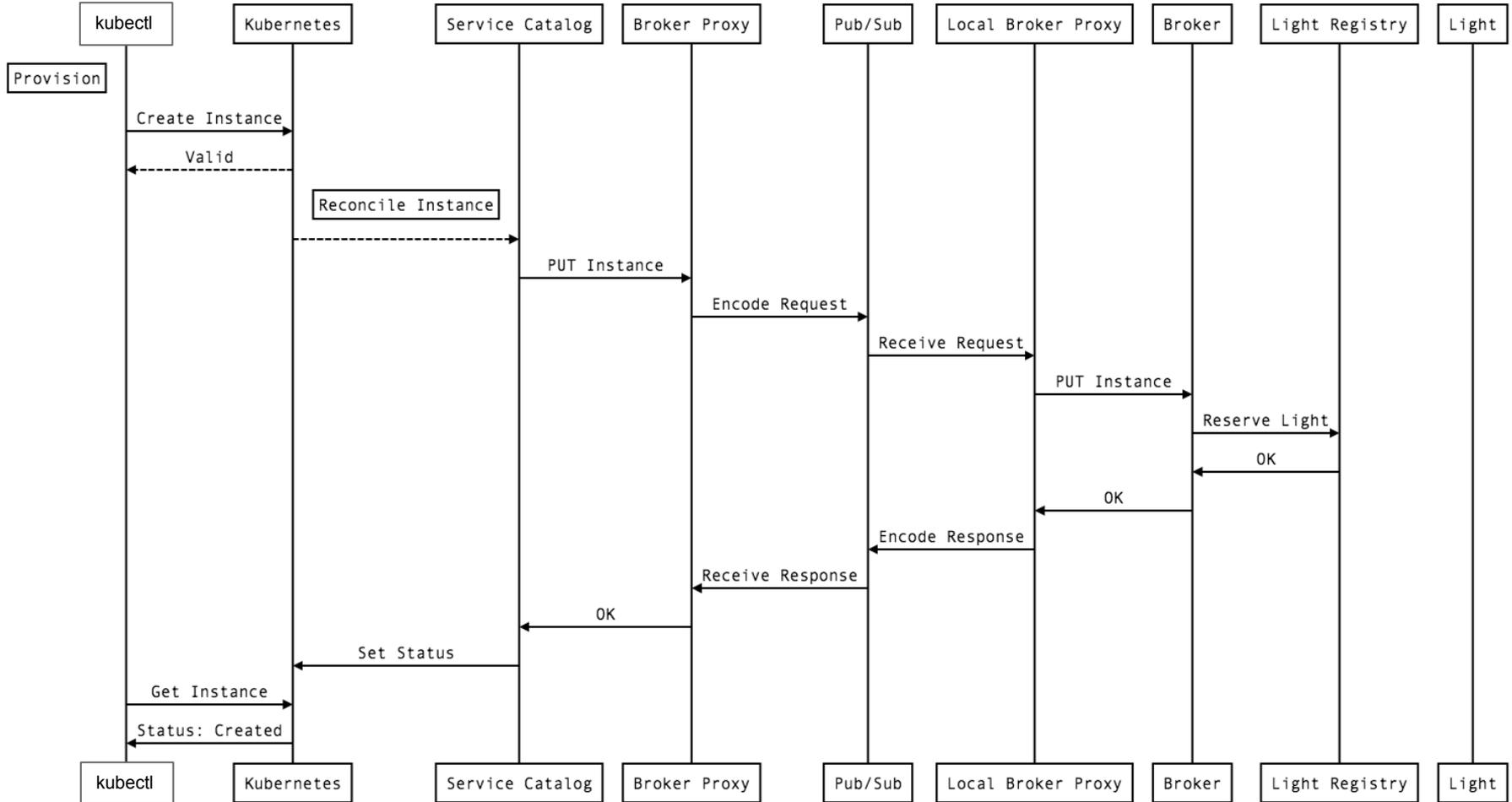


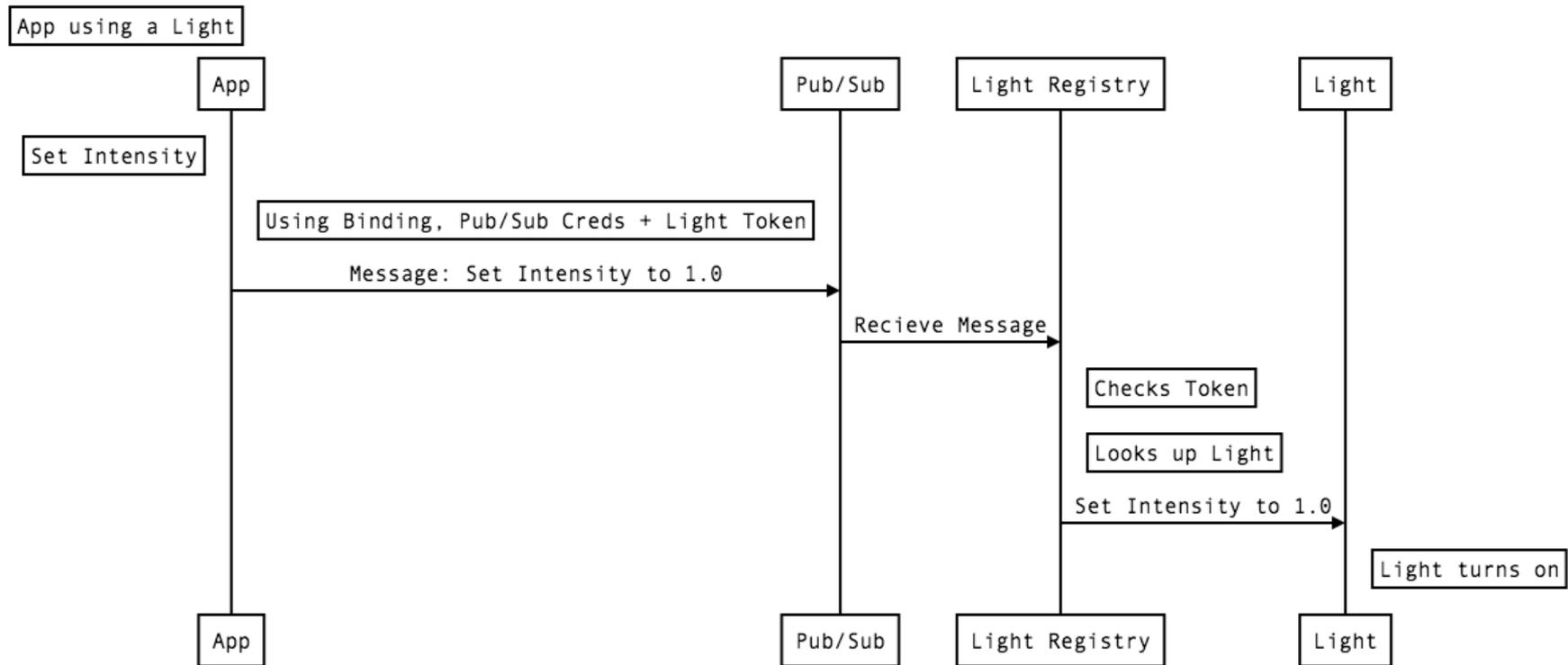






# Light Registry via Service Catalog





# Demo

```
#ledhouse <room> <color> <intensity>  
#ledhouse <[1-4]><a|b|c> <red|green|blue> <[0-10]>
```

*Examples:*

#ledhouse 1b green 5 ← First floor middle green light to 50%.

#ledhouse 3c red 10 ← Third floor right side red to 100%

#ledhouse 4a blue 1 ← Fourth floor middle side blue to 10%



# Git Repos to get you started

<https://github.com/openservicebrokerapi/servicebroker>

<https://github.com/kubernetes-incubator/service-catalog>

<https://github.com/pmorie/osb-starter-pack>

<https://github.com/pmorie/osb-broker-lib>

For this demo:

<https://github.com/n3wscott/ledhouse>

<https://github.com/n3wscott/ledhouse-broker>

<https://github.com/n3wscott/k8s-broker-proxy>

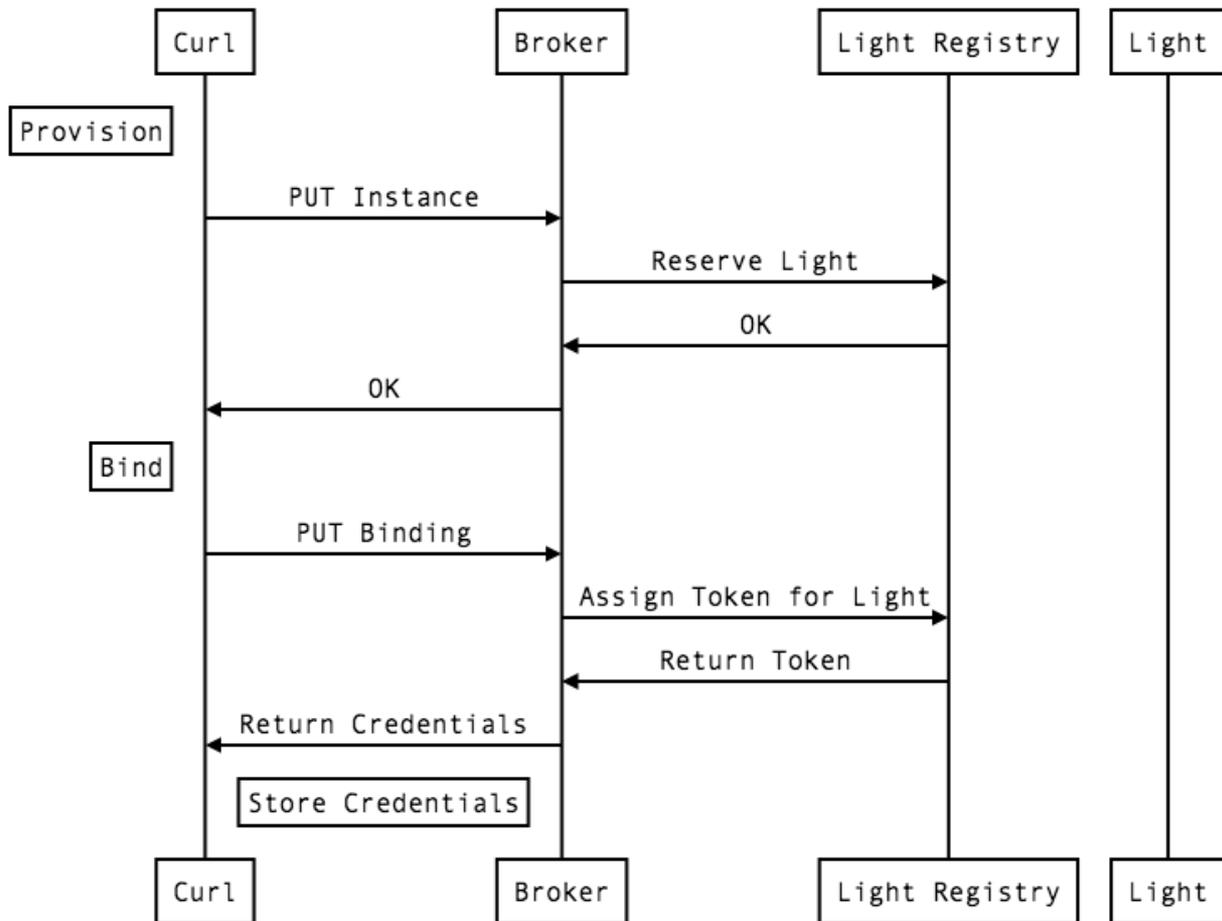


Thank you

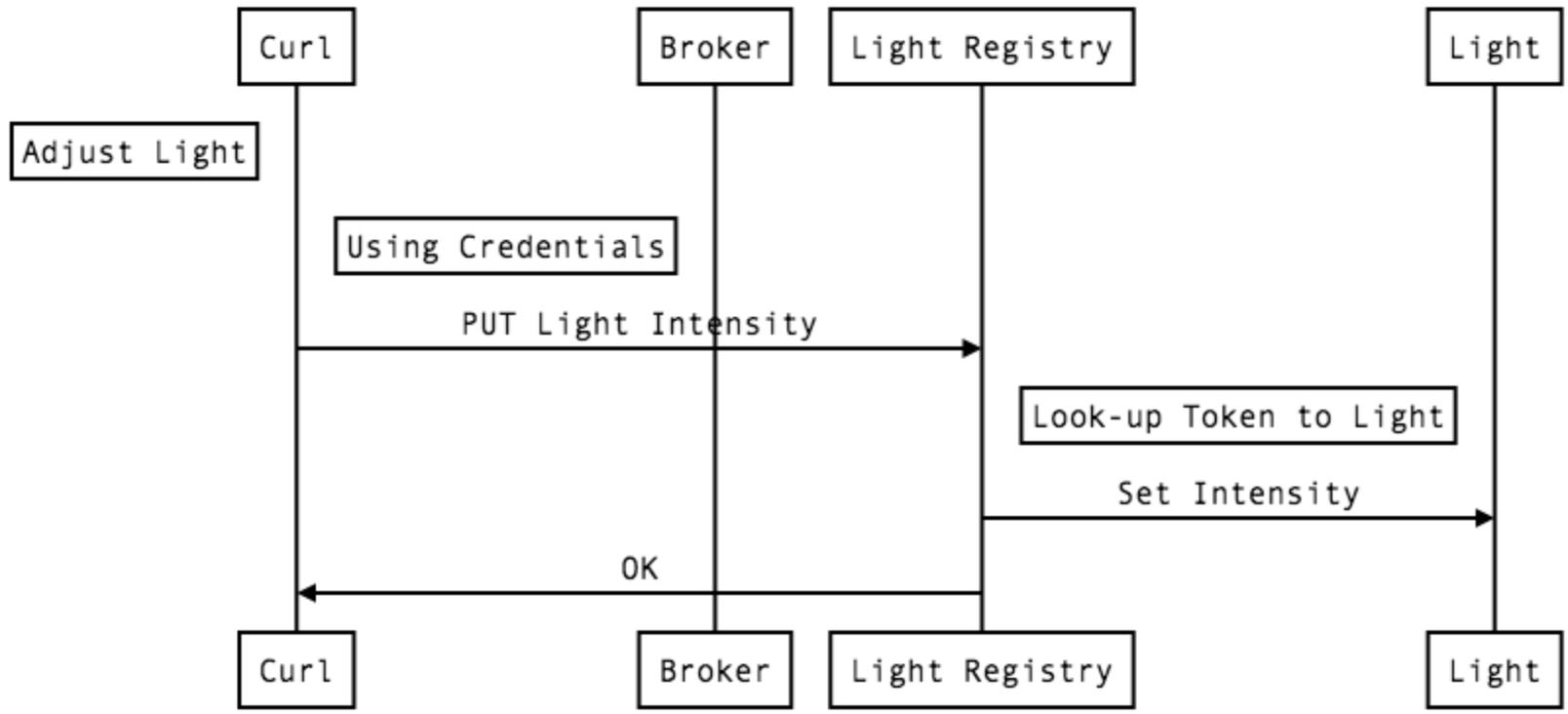
Scott Nichols  
@n3wscott



# Light Registry via Broker and Curl



# Light Registry via Broker and Curl



# Light Registry via Broker and Curl

