



From **Dark APIs** to **Federated** Platforms



Thomas Lawrence

Head of Professional Services, NEMEA

JUN 2026

APIs in production

No specs No catalogue **Just traffic...**

No Visibility

No Contracts

No Portal

No Metrics

Specs exist - Portal is live

Governance is breaking at **Scale...**

**Standards
Drift**

**Schema
Divergence**

**Governance
Bottleneck**

**Decentralised
Chaos**



WHICH?

**They're the same company.
Just at different points in the journey.**

Our Play Today

Act I

Technology

Dark APIs into the
Light

Small Demo >

Act II

People & Process

Governance scaling
to Strategic Impacts

Strong Opinions >

ACT 1

The **Dark API** Problem

APIs in production — no specs, no contracts, no visibility

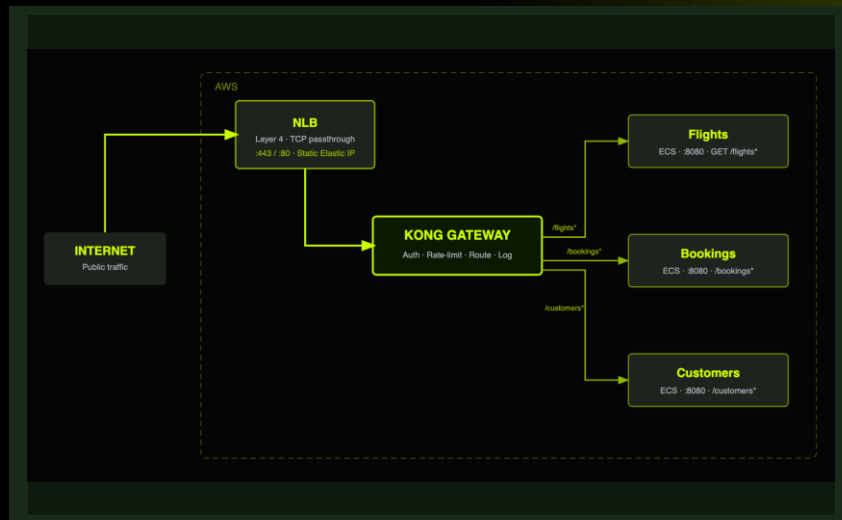
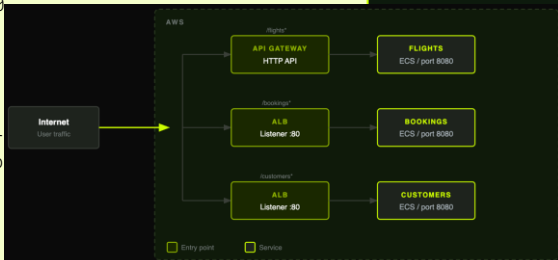
INFRASTRUCTURE AS CODE

The entry point to our story

Engineering Principle:

Every change to infrastructure tracked, versioned, and reviewable

```
resource "aws_lb_target_group" "flights" {
  name     = "kongair-flights"
  port     = 8080
  protocol = "HTTP"
  vpc_id   = var.vpc_id
}
resource "aws_lb_listener_rule" "flights" {
  listener_arn = aws_lb_listener.http.arn
  priority     = 10
  condition {
    path_pattern {
      path_pattern {
      }
    }
  }
  action {
    type = "forward"
    target_group_arn = aws_lb_target_group.arn
  }
}
```



"This isn't how I'd recommend setting this up in production — but it illustrates the principle perfectly."

OBSERVABILITY

O11y Brings **Light** to Endpoints and Interfaces

The Terraform config declared /flights. It said nothing about /flights/{id} or /flights/{id}/details — both of which are being called in production right now

Note: just a bandaid to bring initial management for APIs

● Config

- * /flights
- * /routes
- * /customers

Requests

- GET /flights
- GET /flights/*/details
- POST /bookings
- PUT /bookings
- GET /routes
- GET /routes/{id}

Add New Endpoints

Provide more management for endpoints
More customisation for the flows

Spec Generation

Using logs and AI can generate the initial versions for publication

Dark APIs → **Managed Platform** in 3* steps

1

**Single Entry Point /
Control Plane**

Infrastructure-as-code for traceability

2

**O11y to Specs
Using AI**

Single place to review all traffic

3

**Catalogue &
DevPortal**

Consumer visibility and scorecards

You've reached Company B.

Now what?

APIs are onboarded. Specs exist. There's a portal and a scorecard.
Success — but success creates a new set of problems.

ACT 2 ↓

ACT 2

From **Managed** to **Federated**

Scaling governance without creating bottlenecks

SPEC-FIRST

Design Before You Build

CODE-FIRST

- Code → Spec/CRDs
- **Design = Afterthought**
- Could work with strong discipline

DESIGN FIRST

- Design for *Consumer*
- Design for *Function*
- Align / Approve with Consumer/Stakeholders

Spec-first doesn't slow you down. It shifts the thinking earlier — where it's cheapest to catch mistakes.

CENTRAL PIPELINE

One to *Rule* them All



**Team
Repo**

Code
Spec
Config

**Pull
Request**

Review internally
Team Pipelines

**Central
CI**

Shared to Teams
Re-usable cross-org

**Lint &
Validate**

Spectral Engine
Company Rules
Deployment Rules

**Deploy &
Publish**

Push to Gateway
Push to DevPortal

Ownership

Team Control

Central Control

Checks

Live

AUTOMATED LINTING

Governance at Point of Authoring

Layer0

Technical Rules

- Private / Public Target
- Approved Plugins
- Rejected Plugins
- Security Inclusion

Layer1

Generic Rules

- Missing descriptions on paths or operations
- No request/response examples
- Undeclared response codes
- Reuse Spectral base

Layer2

Company Rules

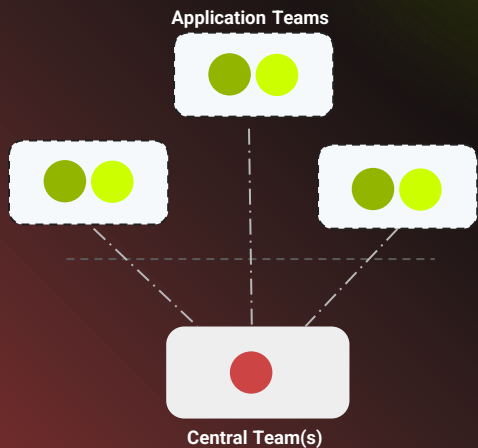
- Required x-owner header on every API
- Mandatory error envelope format
- API versioning convention
- Required security scheme declarations

These are governance board decisions made durable in code.

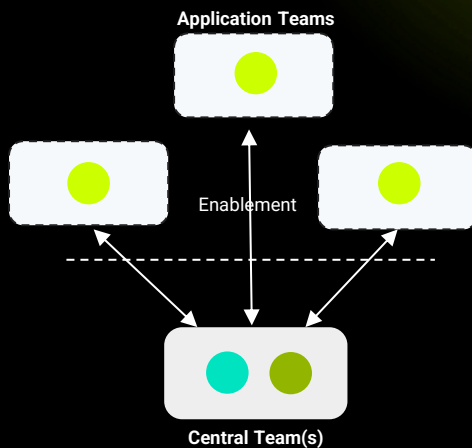
DRIVING AUTONOMOUS CONTROL

The balancing act...

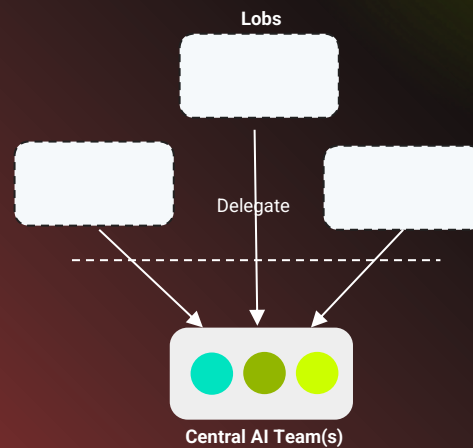
Fully Decentral



Federated



Fully Central



THE GOVERNANCE BOARD

Structure & Centralisation Model

Who Sits

- Domain Owner
- Platform Rep
- No permanent chair

Does What

- Ratifies rules
- Approves objects
- Exception requests

Does NOT

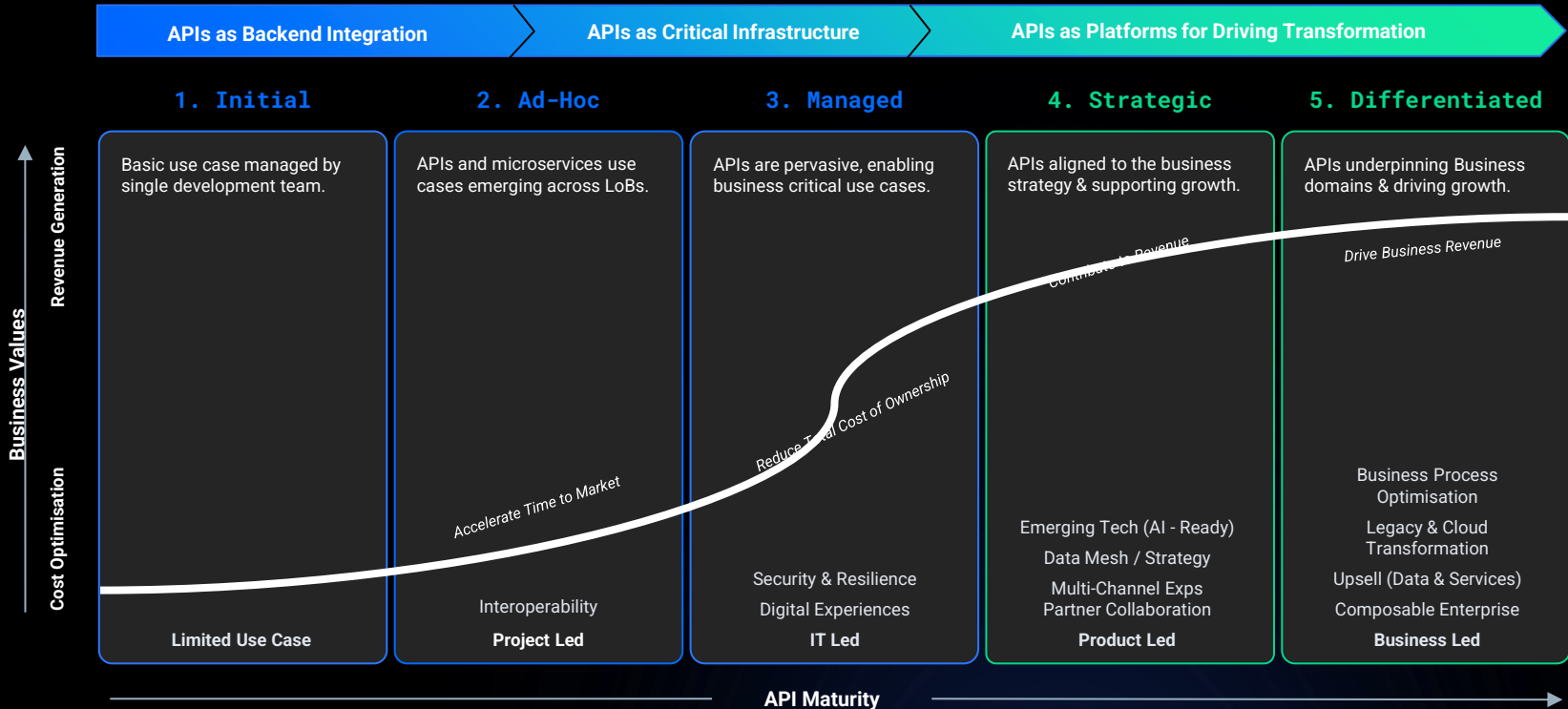
- Review specs
 - Pipeline does
- Be a bottleneck

Closing Out Towards Strategic API-M

With Long Term Goal of Becoming **Differentiated**

Stages of API & Platform Evolution

Business value is shaped by the maturity of your technology platform and API maturity



Questions & Answers (Opinions / Ideas)

Ask me **anything** about this...

Thank you!



Thomas Lawrence

Head of PS, NEMEA

thomas.lawrence@konghq.com

Kong Inc.

contact@konghq.com

44 Montgomery Street
San Francisco, CA 9410, USA

[Konghq.com](https://konghq.com)

Kong

