API Governance

Navigating the challenges of implementing it

Marcelo Araujo

/in/mlaraujo @apiglue

About me

Led API Strategy at 3 Fortune 500

Tell me about you

What is API Governance

Framework ensuring APIs are

- Discoverable
- Consistent
- Compliant
- Secure
- Collaborative

Signs you might need API governance

Manager: “I learn there’s no way to know who’s using my API. How would I know if there’s suspicious activity?”

Front-end folks: “Do you know how much time I spent writing a customer parser for each #$!@ API I consume?”

Backend devs: “I don't want to have to think about every single 4xx I need to implement“

Signs you might need API governance

Backend devs: “Don't tell me how to design APIs. I know exactly what to do”

SLT: “Why you(api gov team) haven’t budget for this(anything, really) last year?”

Security: “We need an API Gateway for the API gateway“ - This is a joke. I ❤️ security

We’ll cover today

1 - Balancing standardization with flexibility

2 - Aligning cross-functional teams with varying priorities and levels of API maturity

3 - Governance elements and their boundaries

4 - Overcoming resistance to change and fostering a culture of accountability

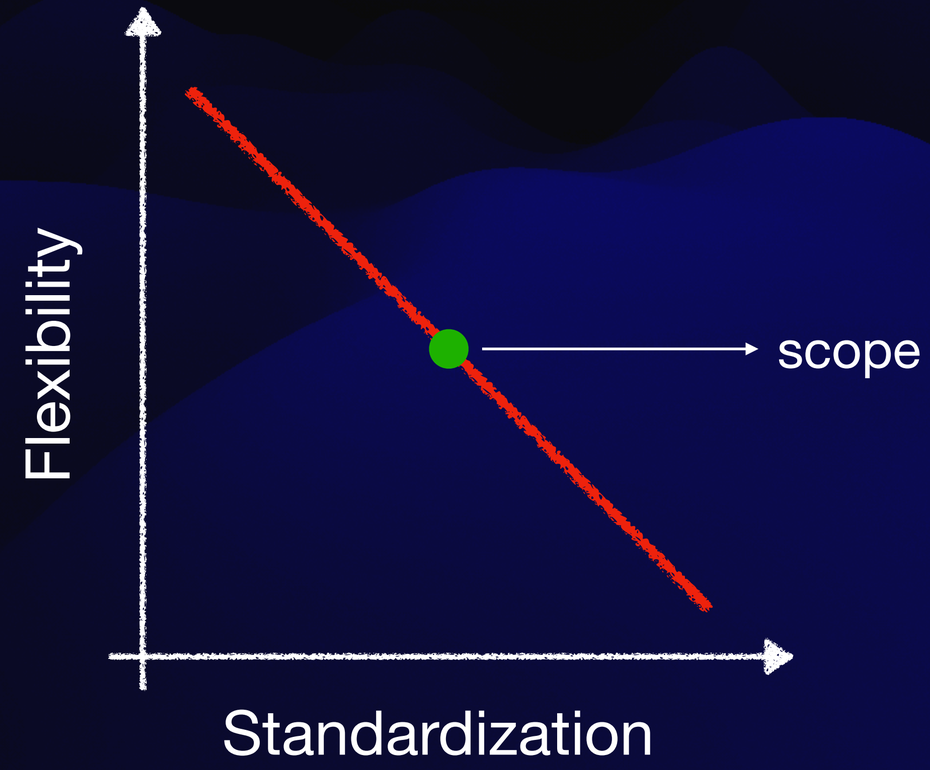
5 - Integrating governance processes without stifling innovation or agility

Chapter 1

Balancing standardization with flexibility

Standardization with Flexibility

The dilemma

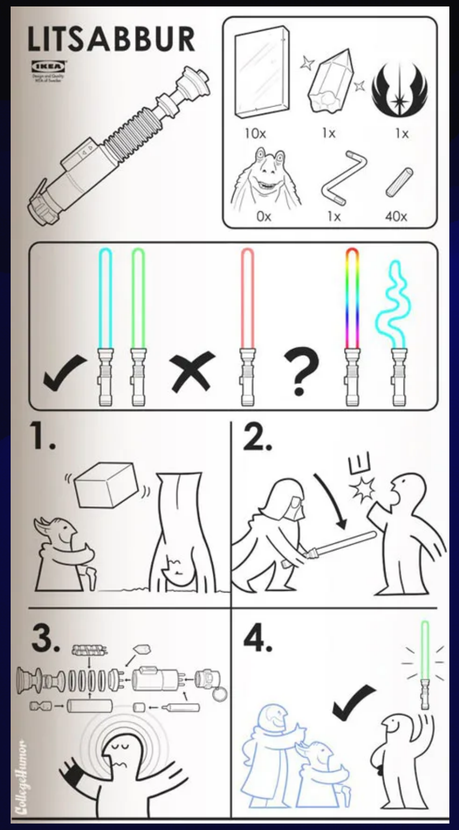


Standardization with Flexibility

Why both are essential

Standardization with Flexibility

Why both are essential



Created by Caldwell Tanner, Susanna Wolff and Conor McKeon of College Humor

Standardization with Flexibility

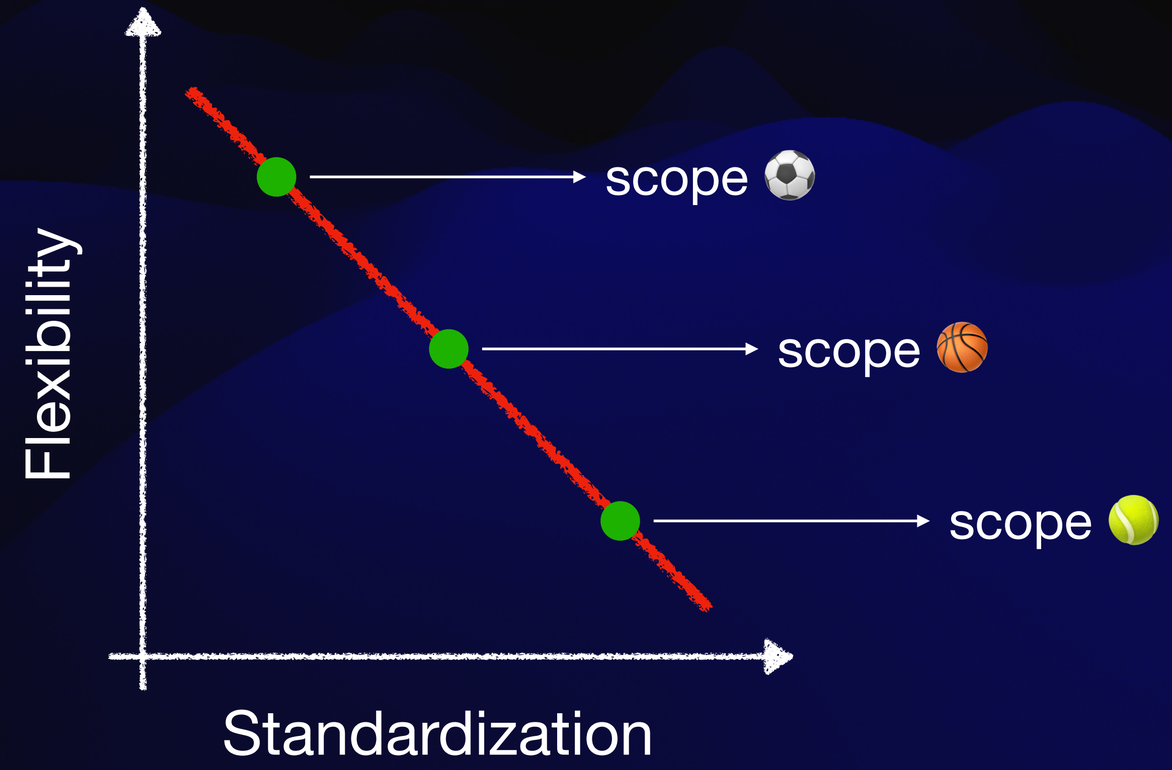
Core problem

Teams fear one-size-fits-all rules and leaders fear fragmentation and risk.

The way out is to decouple standards from innovation using risk-based levels and versioned guidance.

Standardization with Flexibility

The dilemma



Standardization with Flexibility

Framework to solve

- Define levels by risk: Public/external, partner, internal, experimental.
- *Codify* must/should/may into versioned standards.
- Offer paved roads: scaffolds, templates, starter repos, SDKs, examples.
- Provide an exception path: temporary waivers, and planned convergence.

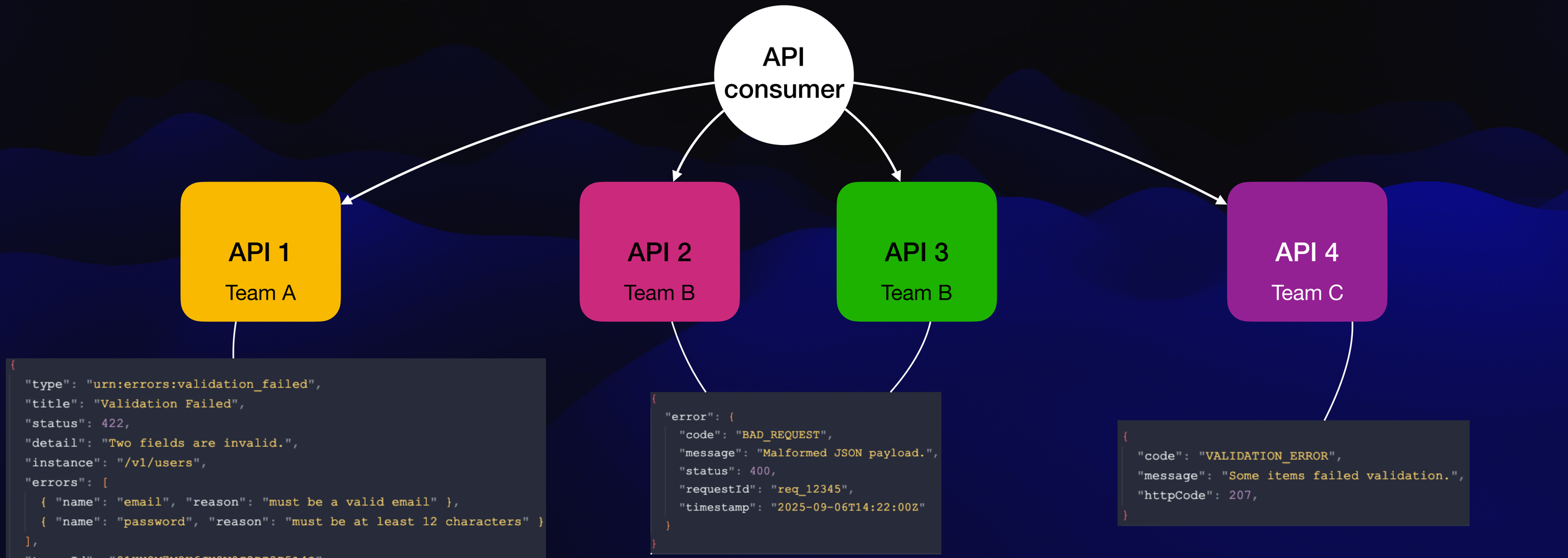
Standardization with Flexibility

Comparison

Dimension	Standardization	Flexibility
Goal	Consistency, safety, reuse	Speed, fit-for-purpose, learning
Mechanism	Codified standards, paved roads	Exceptions, experiments
Risk control	Mandatory checks, policy-as-code	Time-bound waivers, risk tiers

Standardization with Flexibility

Practical examples



Middle ground

Governance Do’s and dont’s

DO	DON’T
Risk-based standardization documents	Worry about ALL APIs
Standardized security	Careful with reusable models
Have non-domain common models	Assume it happens w/o incentives
SLx for reusable APIs	Think governance is about tooling
Inventory and catalog	Disrupt devs workflow (too much)
Measure progress (i.e TTFC)	

Chapter 2

Aligning cross-functional teams with varying priorities and levels of API maturity.

Objective

The organization need consistent, secure and stable access to its core capabilities through APIs

Assumption

Core capabilities are delivered by more than one team

Challenges

Alignment across departmental boundaries are not prioritized in the same way as within department boundaries.

As a result

Access to core capabilities through APIs is neither possible nor consistent

Now what

Steering towards objective

What to focus on

API standards

Organizational objectives

Capability maps and roadmaps

API Standards

What to focus on

Interface design

Architecture and Security

Visibility/reuse

Monitoring and observability

More on chapter 3

Organizational objectives

What to focus on

Define terms of API ownership, including capacity allocation

Define Time-to-first-call and consumption ratio as metric

Organizational objectives

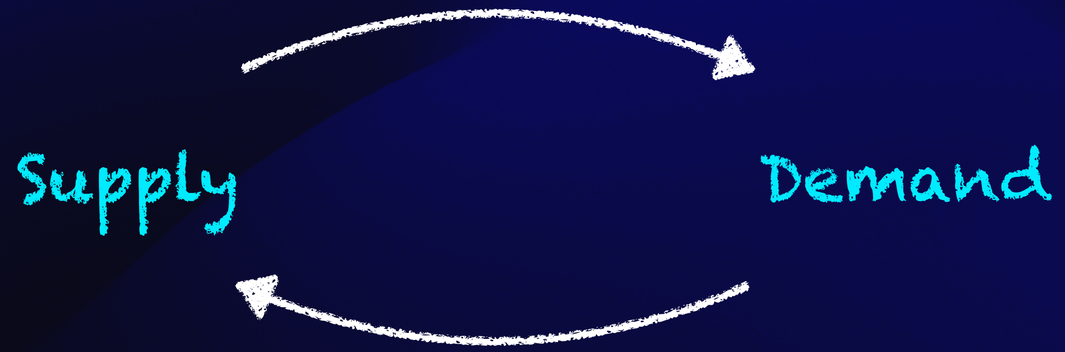
What to focus on

Define reuse rate as early metric:

$$\frac{\text{Number of unique consumers}}{\text{Number of reusable APIs}} > 1$$

Capability maps and roadmaps

A “production” pipeline and “needs” are publicly available



Capability maps and roadmaps

For project-driven organizations

With strategic funding:

Scope your APIs around capabilities

Capability maps and roadmaps

For project-driven organizations

Without strategic funding:

Define boundaries around capabilities, not project.

Capability maps and roadmaps

For product-driven organizations

Without strategic funding:

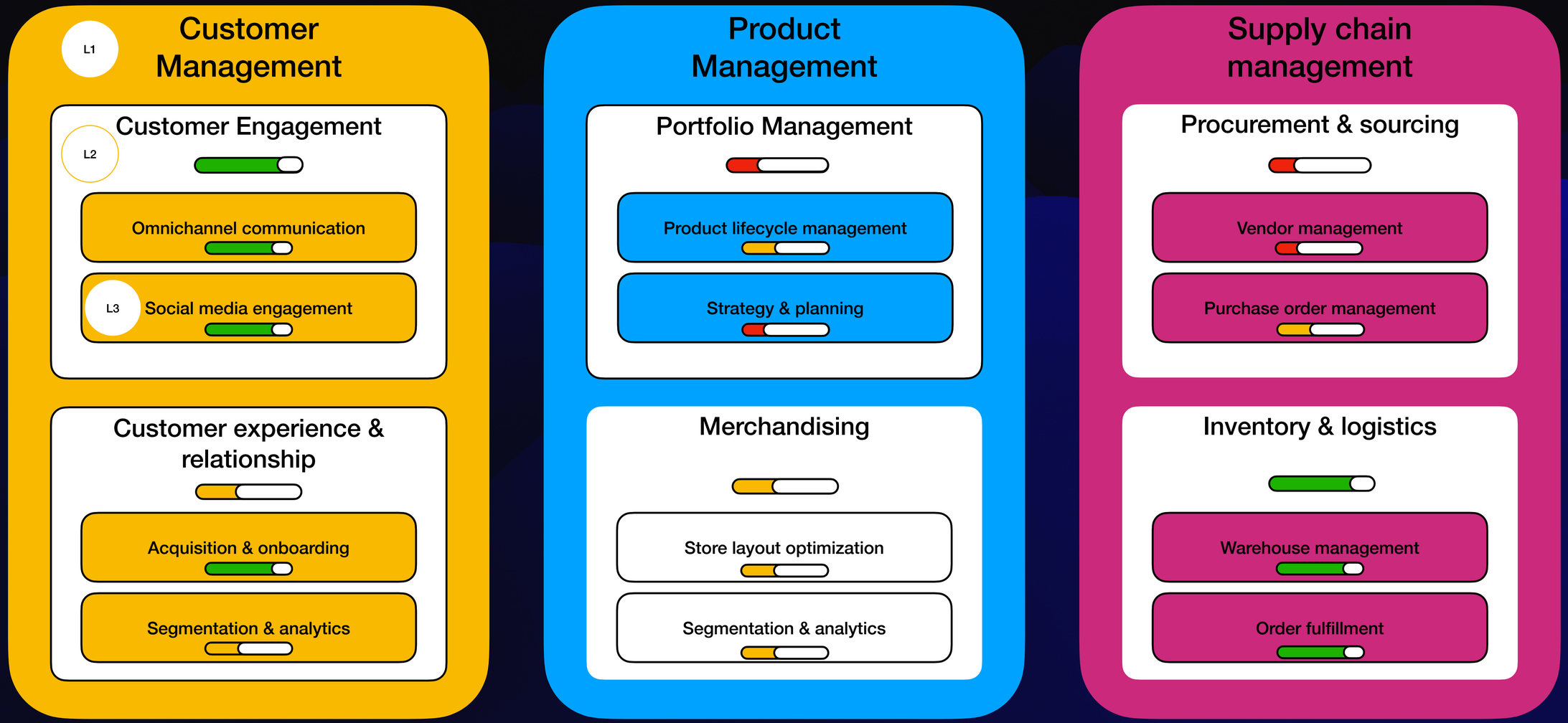
Same approach. Define boundaries around capabilities.

With strategic funding:

Build ahead. Look for capabilities gap.

Capability maps and roadmaps

What to focus on



Chapter 3

Governance elements and their boundaries

Elements of API Governance

- Interface design
- Architecture & Security
- Automation
- Visibility

Elements of API Governance

Interface design

The inverse Conway’s maneuver

Elements of API Governance

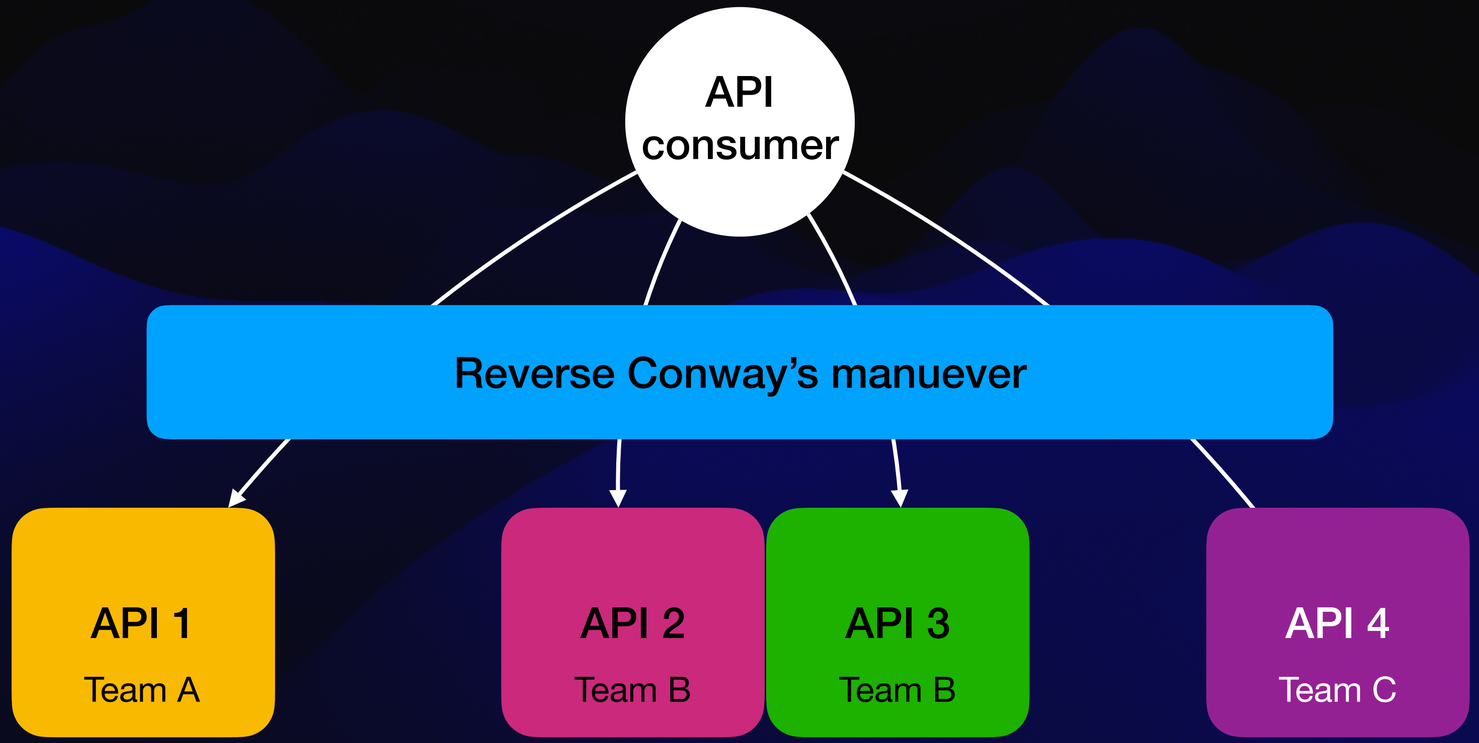
Interface design

> “Any organization that designs a system will inevitably produce a design whose structure mirrors its own communication structure.”

Melvin Conway, 1967

Elements of API Governance

Interface design



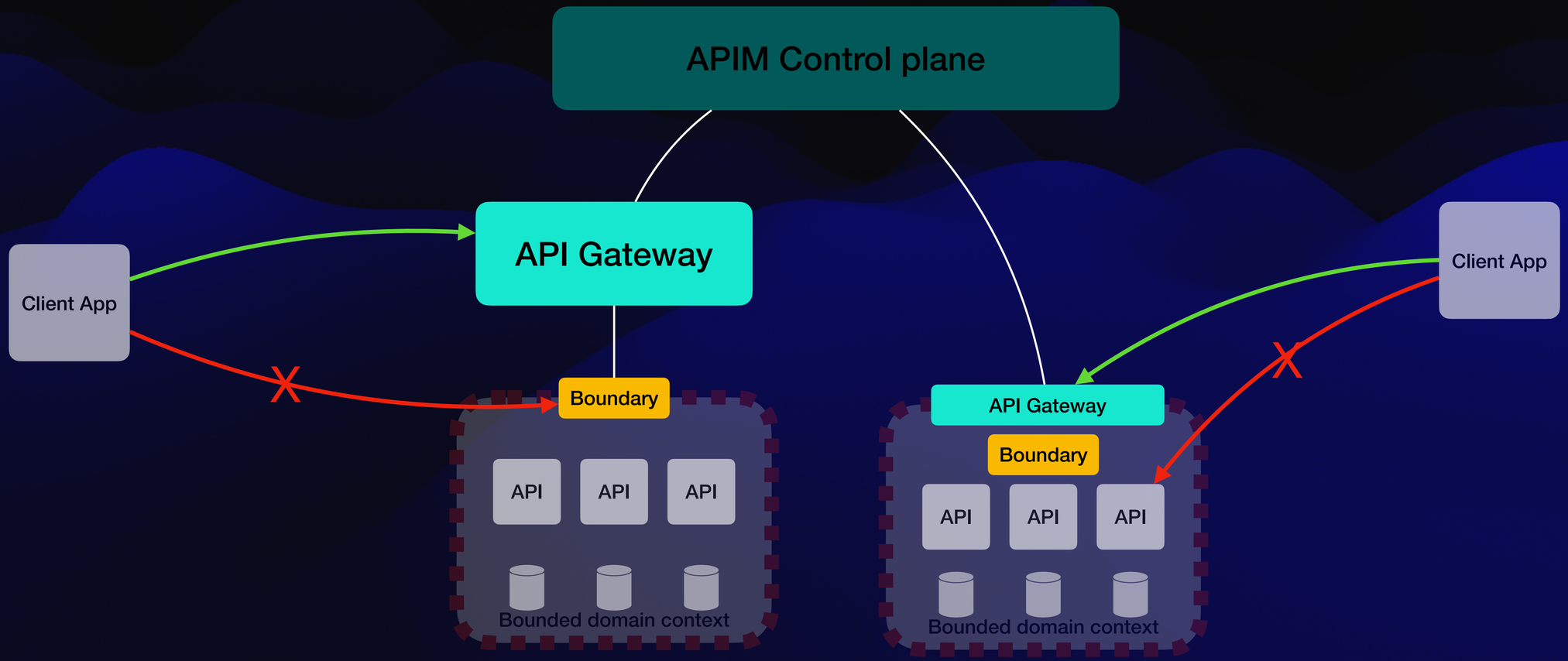
Elements of API Governance

Architecture & Security

Method	Use-case	Strenghts	Limitations
oAuth 2.0	Third-party integrations	Fine-grained access control	Complex setup
OIDC	Identity Management	AuthZ/N	Steep learning
JWT	Microservices	Stateless, performance	No revocation, token size
API keys	Internal services	Easy implementation	Limited security, no expiration
Basic Auth	Legacy systems	Simple setup	High security risk

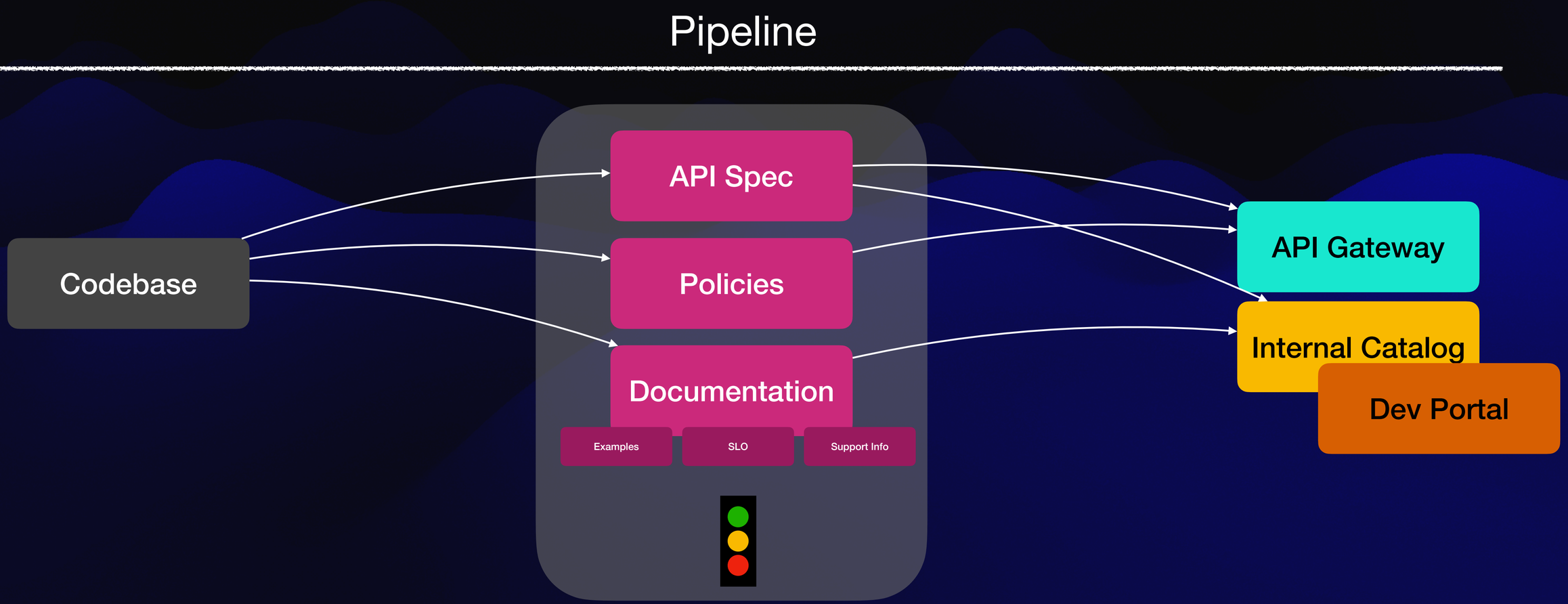
Elements of API Governance

Architecture & Security



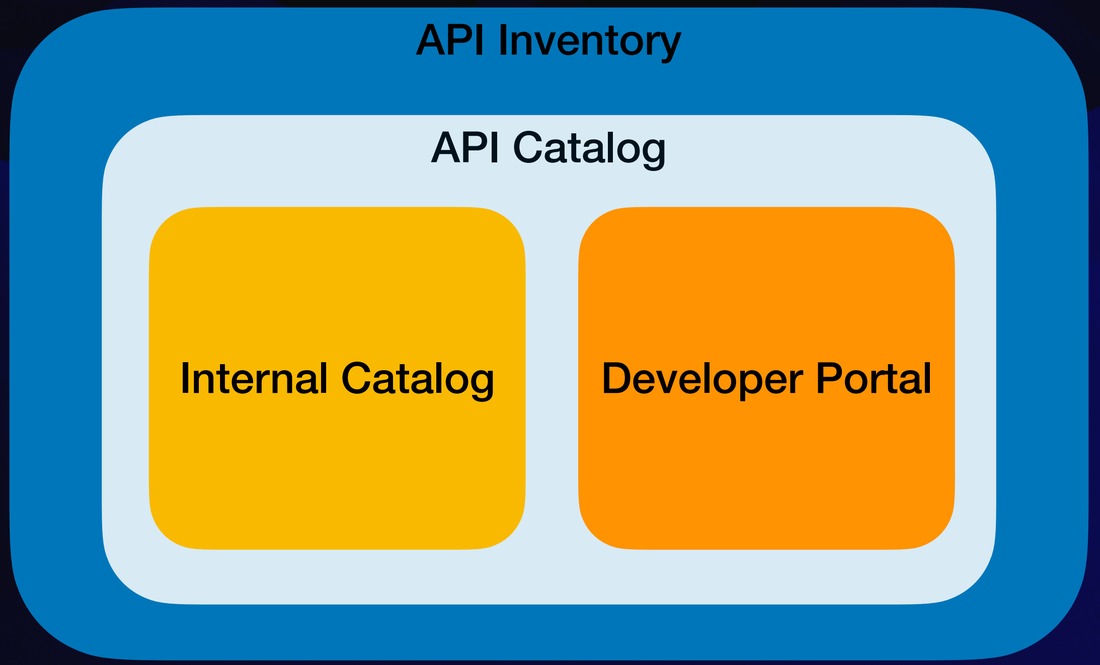
Elements of API Governance

Architecture & Security



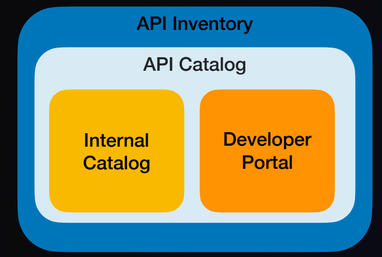
Elements of API Governance

Visibility - Inventory and catalogs



Elements of API Governance

Visibility - Inventory and catalogs



API Inventory

Contains all APIs
Not available to end users
Mainly used for governance purposes

API Catalog

Internal Catalog

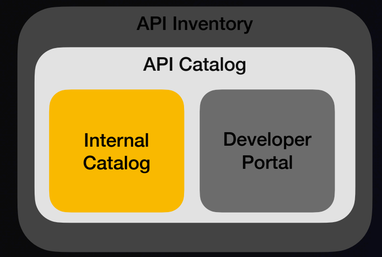
Reusable internal APIs
Self-service
Documentation > DX

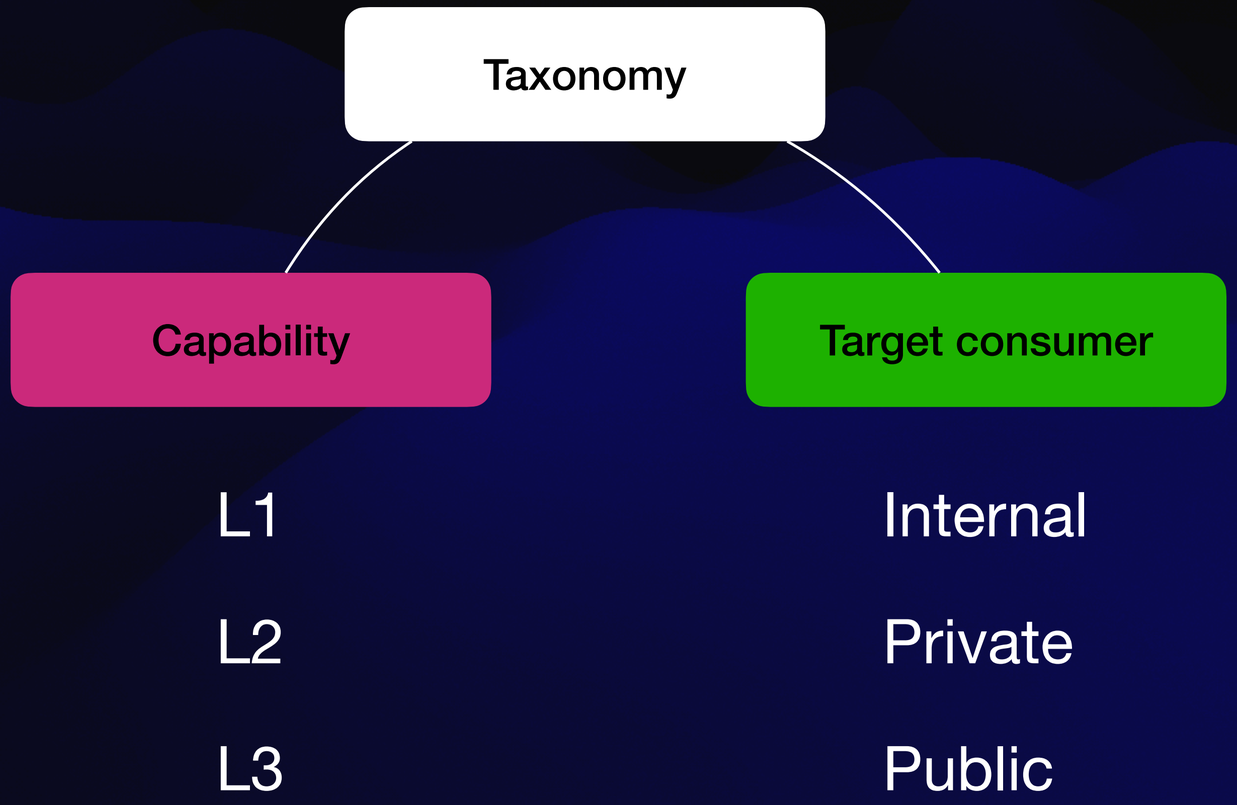
Developer Portal

Partner/Public APIs
Brand/product centric
Documentation + DX

Elements of API Governance

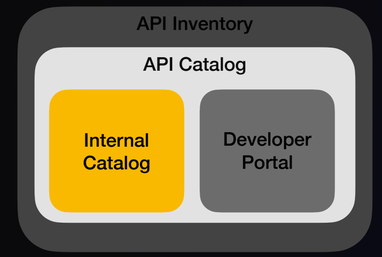
Visibility - Taxonomy

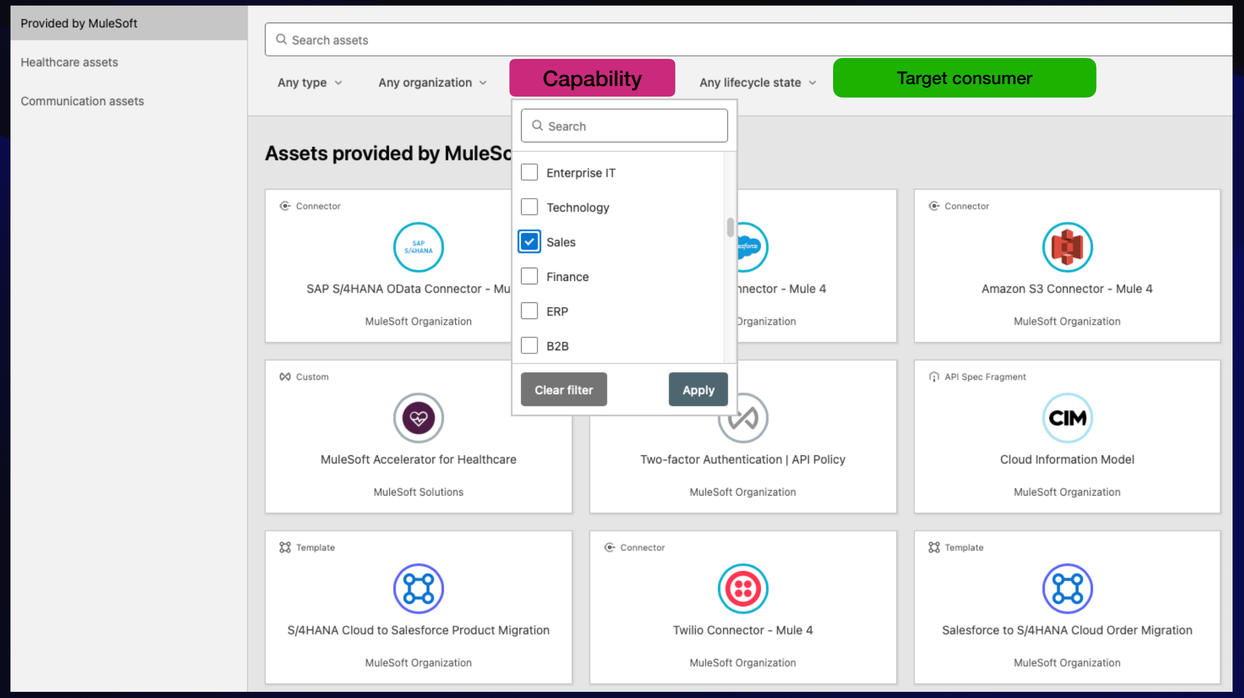




Elements of API Governance

Visibility - Taxonomy





Elements of API Governance

Playbook

Element	Primary goals	Boundaries	Key artifacts	Automation	Accountable
Design	Consistency, consumer UX	Style guide vs domain semantics	OpenAPI/AsyncAPI, error model, versioning policy	Linting (style/semver), breaking-change detection	Architecture + API Guild
Security	Protection by default	Controls vs threat modeling depth	AuthN/Z patterns, scopes, data classification	Policy-as-code, SAST/DAST, secrets scan	Security Engineering
Architecture	Fit-to-platform, reliability	Reference arch vs solution design	ADRs, reference diagrams, NFRs, SLOs	Template arch, IaC modules validation	Architecture Board
Observability	Operability, SLOs, MTTR	Required telemetry vs analysis	Tracing, logging, metrics, error taxonomy	Trace ID checks, log schema tests, SLO monitors	SRE/Platform
Visibility	Discoverability, reuse	Metadata minimums vs curation	API catalog, ownership, taxonomy, lifecycle state	Catalog CI publish, contract validation	Platform PM
Lifecycle	Change safety, consumer trust	Gateways vs team autonomy	Versioning, deprecation, EOL plans	Auto-notify consumers, usage analytics, EOL timers	API Program

Chapter 4

Overcoming resistance to change and fostering a culture of accountability.

Overcoming resistance

Core problem

“This slows us down” and “not invented here” create cultural drag.

Bilbo - The uninterested

Meet the characters

“We are plain quiet folk, and I have no use for adventures. Nasty, disturbing, and uncomfortable things”

Jordan - The relentless

Meet the characters

“ Act as if you’re a wealthy man, rich already, and then you’ll surely become rich. Act as if you have unmatched confidence, and people will be drawn to you ”

Erin - The fair

Meet the characters

“ Not personal? That is my work! My sweat! My time away from my kids! If that’s not personal, I don’t know what is. ”

Winning them over

Or not!

My Strategy: Ignore.

As a result: His (public metrics didn’t look too good).

My Strategy: Find middle ground. Hold him accountable for compromising

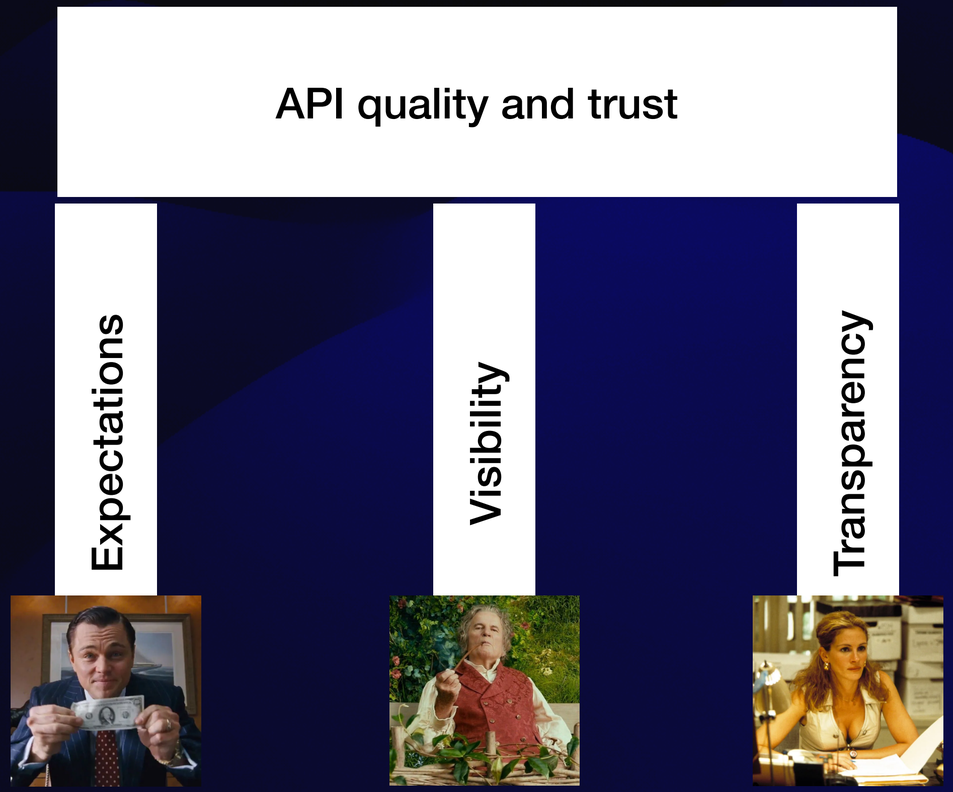
As a result: he got praised and returned the favor

My Strategy: Active listening.

As a result: She became a key advocate

Culture of accountability

Meet the characters



Overcoming resistance

Playbook

Narrative	Governance as enablement, not enforcement; show “before vs after” outcomes.
Champions network	One person per team advocates for patterns and collects feedback
Change levers	Dojo, Office hours, short videos, brown-bags, internal docs, and FAQ.
Transparency	Team scorecards and wall-of-fame for successful APIs
Psychological safety	Treat exceptions as learning, not violations
Incentives	Organizational goals, personal objectives, etc…

Chapter 5

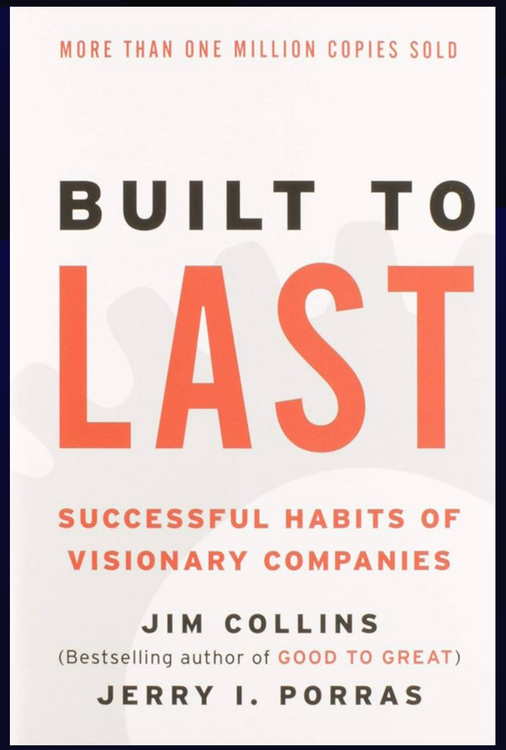
Integrating governance processes without stifling innovation or agility.

Governance vs innovation

The false dichotomy

Genius of the AND

Against the tyranny of the OR

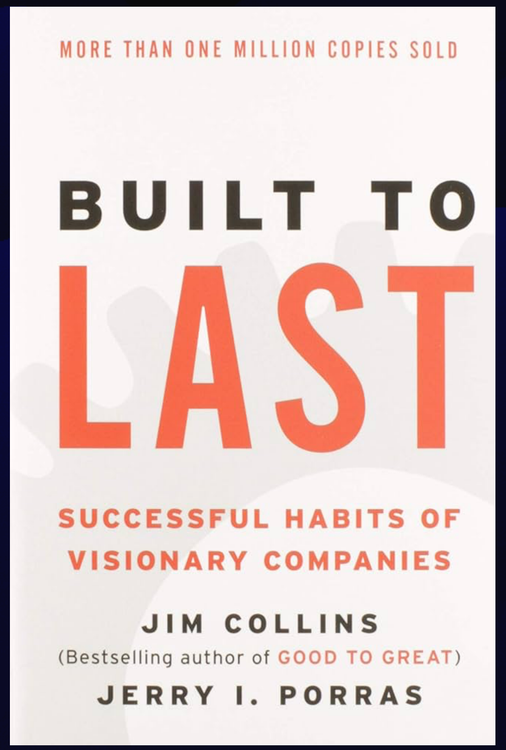


“[…] Builders of greatness reject the "Tyranny of the OR" and embrace the "Genius of the AND. […]”

- Jim Collins

Genius of the AND

Against the tyranny of the OR

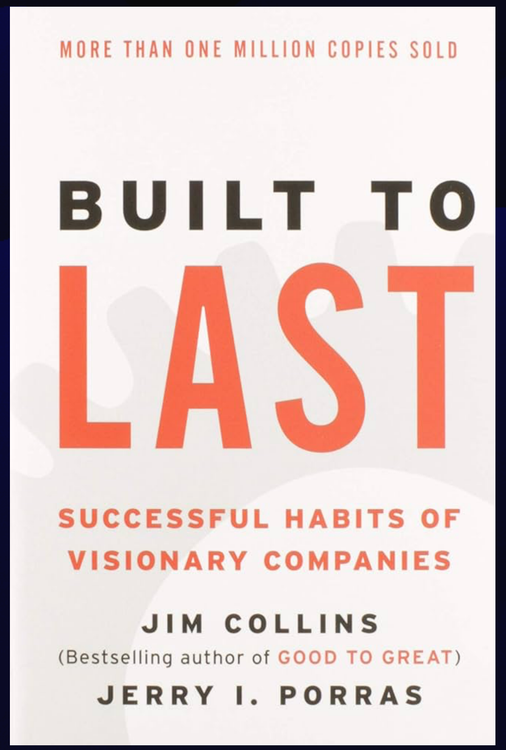


"They embrace both extremes across a number of dimensions at the same time”

- Jim Collins

Genius of the AND

Against the tyranny of the OR



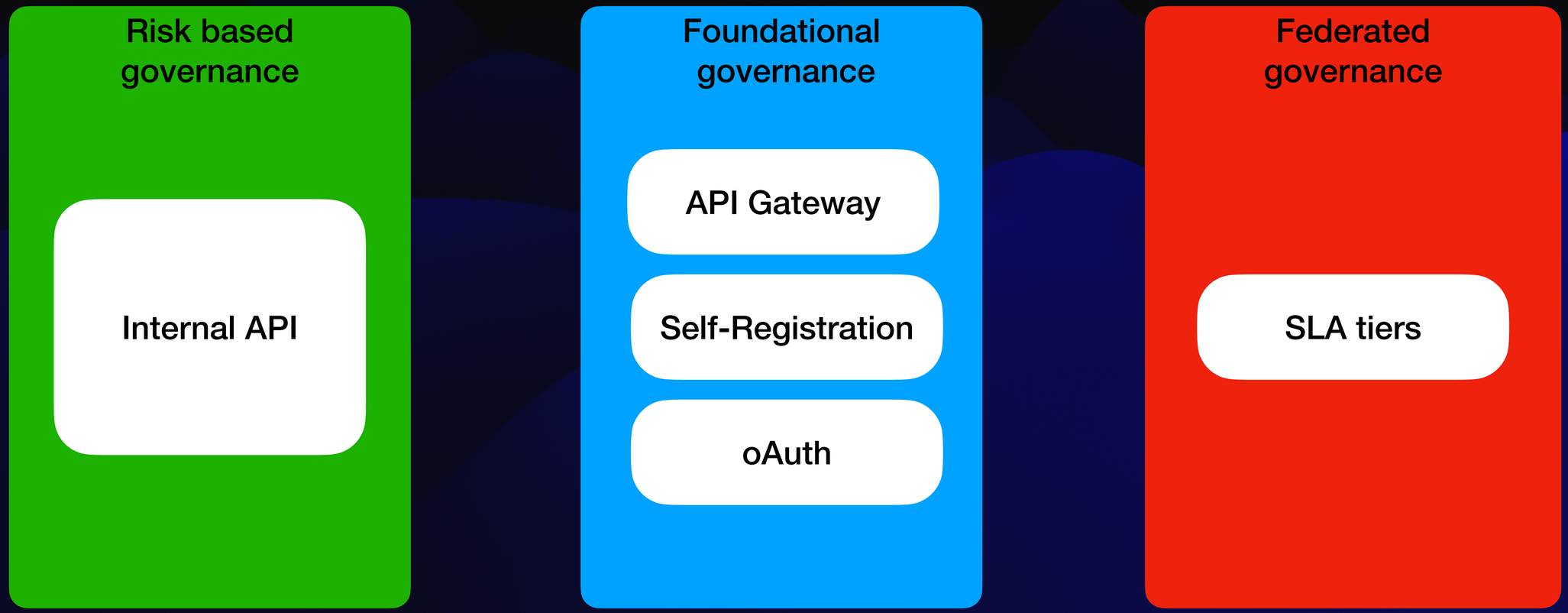
“ […] purpose AND profit, continuity AND change, freedom AND responsibility, discipline AND creativity, humility AND will, empirical analysis AND decisive action, etc.”

- Jim Collins

Composite governance

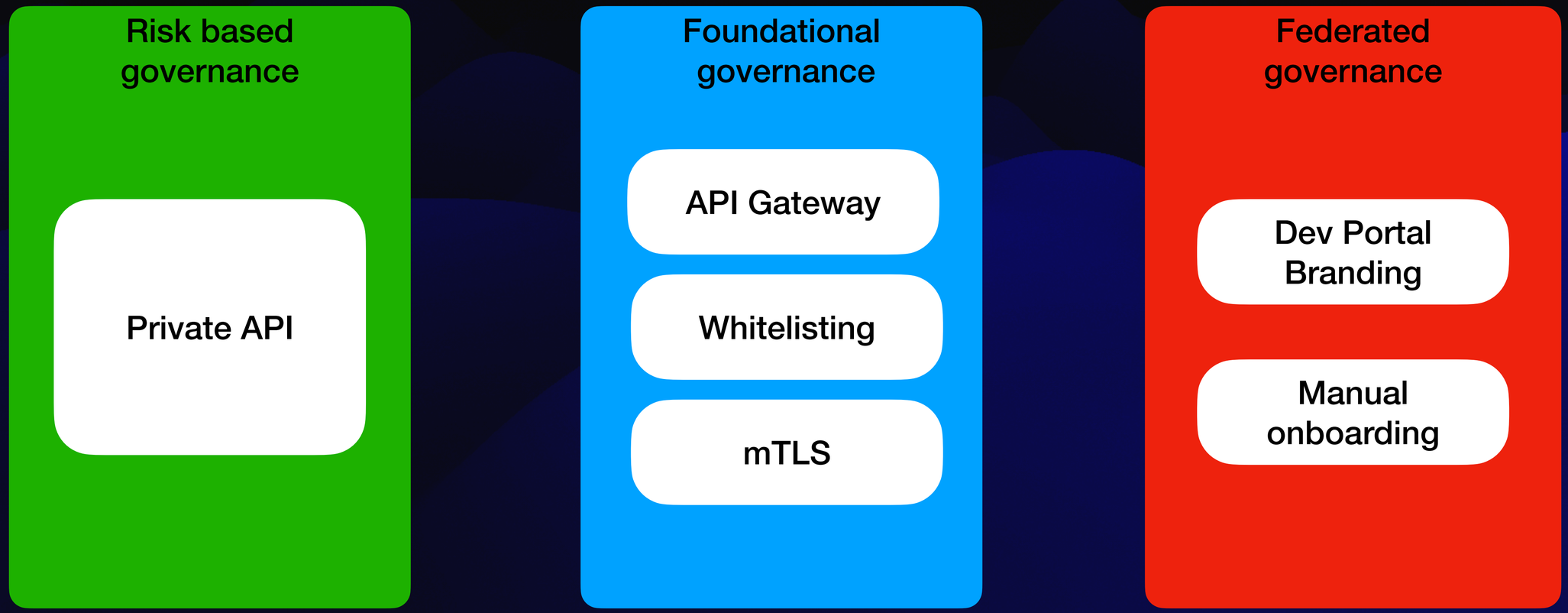
Composite governance

3-tiered governance



Composite governance

3-tiered governance



Composite governance

Playbook

Tier	Goal
Risk-based governance	Define the risk taxonomy
Central governance	Set of risk-aware, minimum requirements for healthy oversight
Federated governance	Freedom to extend Risk and Central governance based on specific needs.

Summary

Key takeaways

Takeaways from Chapter 1

Balancing standardization with flexibility

Define taxonomy for risk-based API governance

Think twice before wanting to “govern them all”. Focus on subsets of APIs.

Takeaways from Chapter 2

Aligning cross-functional teams

Establish API standards

Establish organizations objectives to increase API governance adoption

Make capability maps and roadmaps visible and engaging

Measure progress through well defined metrics

Takeaways from Chapter 3

Governance elements and their boundaries

Determine the elements of your governance framework. As an example:

- Interface design
- Architecture & Security
- Automation
- Visibility

Takeaways from Chapter 4

Overcoming resistance to change

Identify all stakeholders: The uninterested, the relentless, the fair

Build trust by actively listening to the relentless and the fair

Ask those with high emotions: What would you do differently?

Takeaways from Chapter 5

Governance and innovation

Leverage composite/layered governance

Define governance elements of each cross-cutting risk category

Thank you for your time!

Marcelo Araujo

 **/in/mlaraujo @apiglue**