Shift Right?!

What's Behind the Self-Service Easy Button?

Practical Strategies for Operations Leaders in Platform Engineering

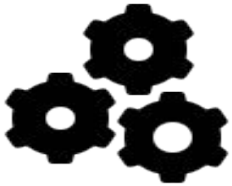
Rob Hirschfeld, CEO of RackN
Social Handle @zehicle

WHY PLATFORM ENGINEERING?

NOTHING WORKS TOGETHER!

WHY PLATFORM ENGINEERING?
NOTHING WORKS TOGETHER!

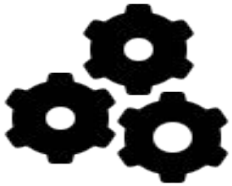
We have
great tools



But we can't use them consistently

WHY PLATFORM ENGINEERING?
NOTHING WORKS TOGETHER!

We have
great tools



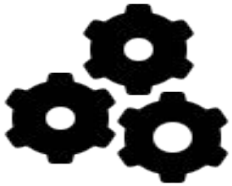
But we can't use them consistently

We have
great people

But can't control our processes

WHY PLATFORM ENGINEERING?
NOTHING WORKS TOGETHER!

We have
great tools

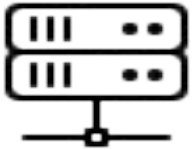


But we can't use them consistently

We have
great people

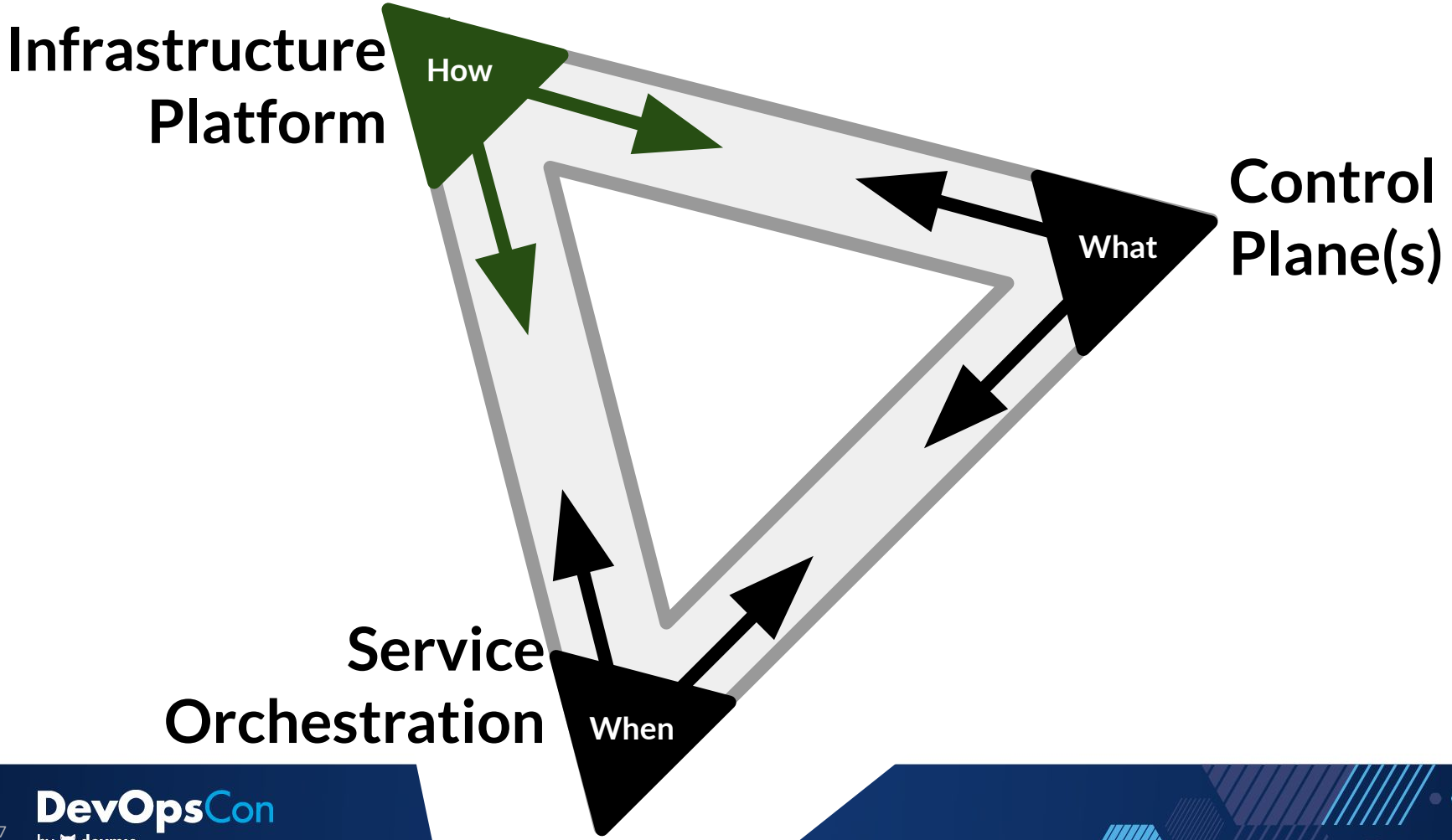
But can't control our processes

We have
great vendors



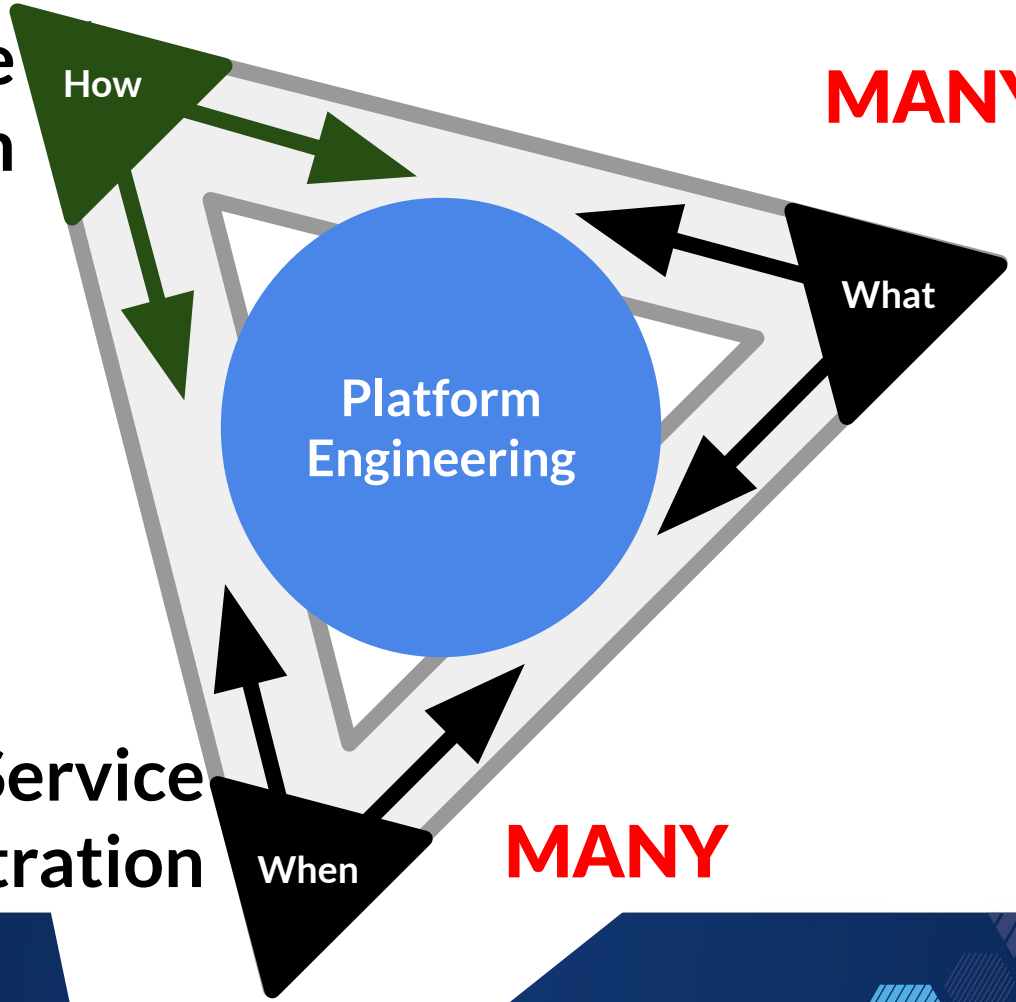
But we can't manage the complexity

Tool sprawl is expensive!


Infrastructure Platform
How
Control Plane(s)
What
Service Orchestration
When

Infrastructure Platform

ONE



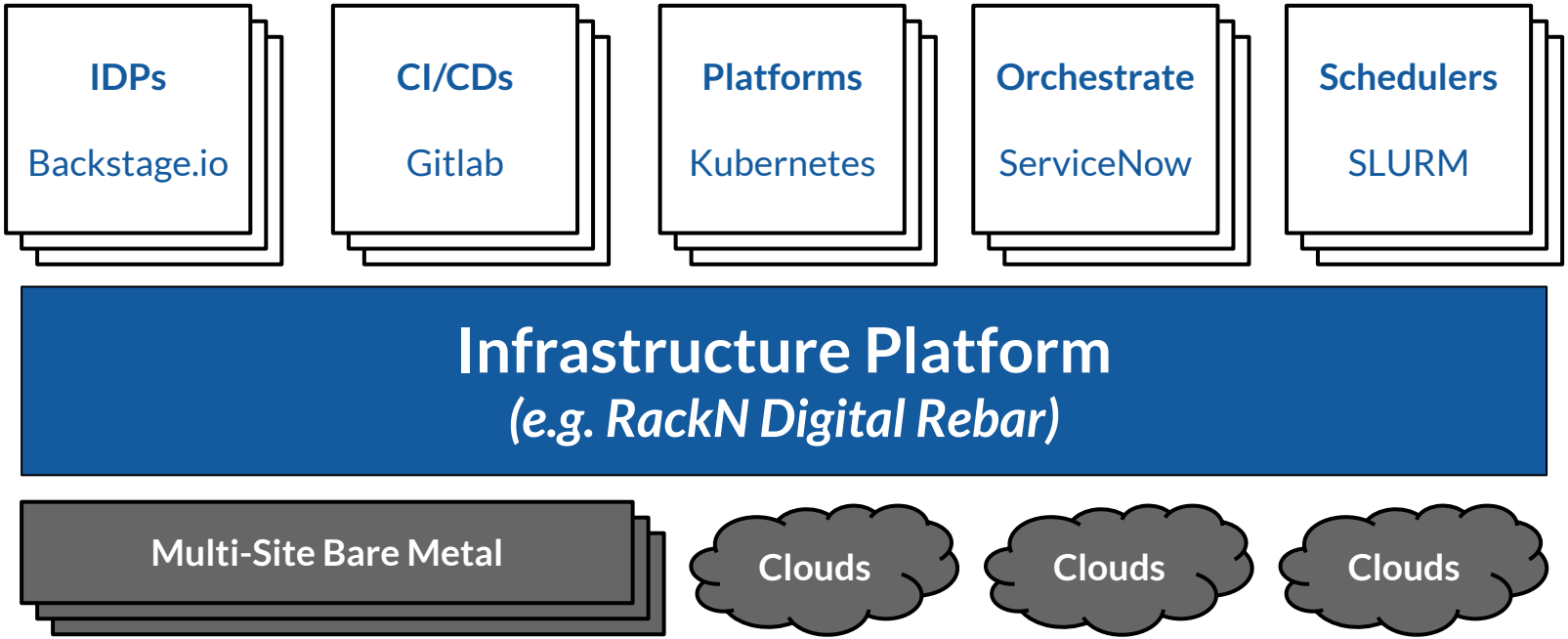
MANY

Control Plane(s)

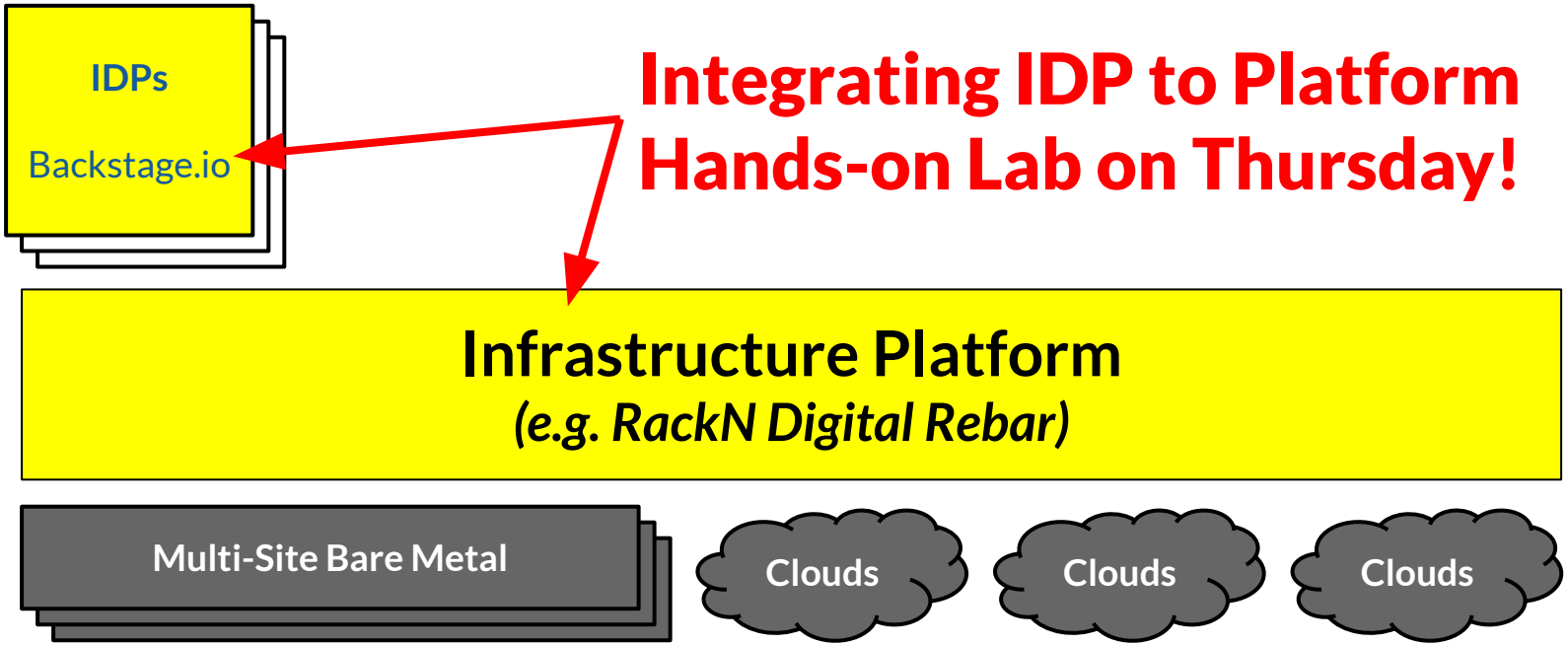
Service Orchestration

MANY

Many Devs with One Ops

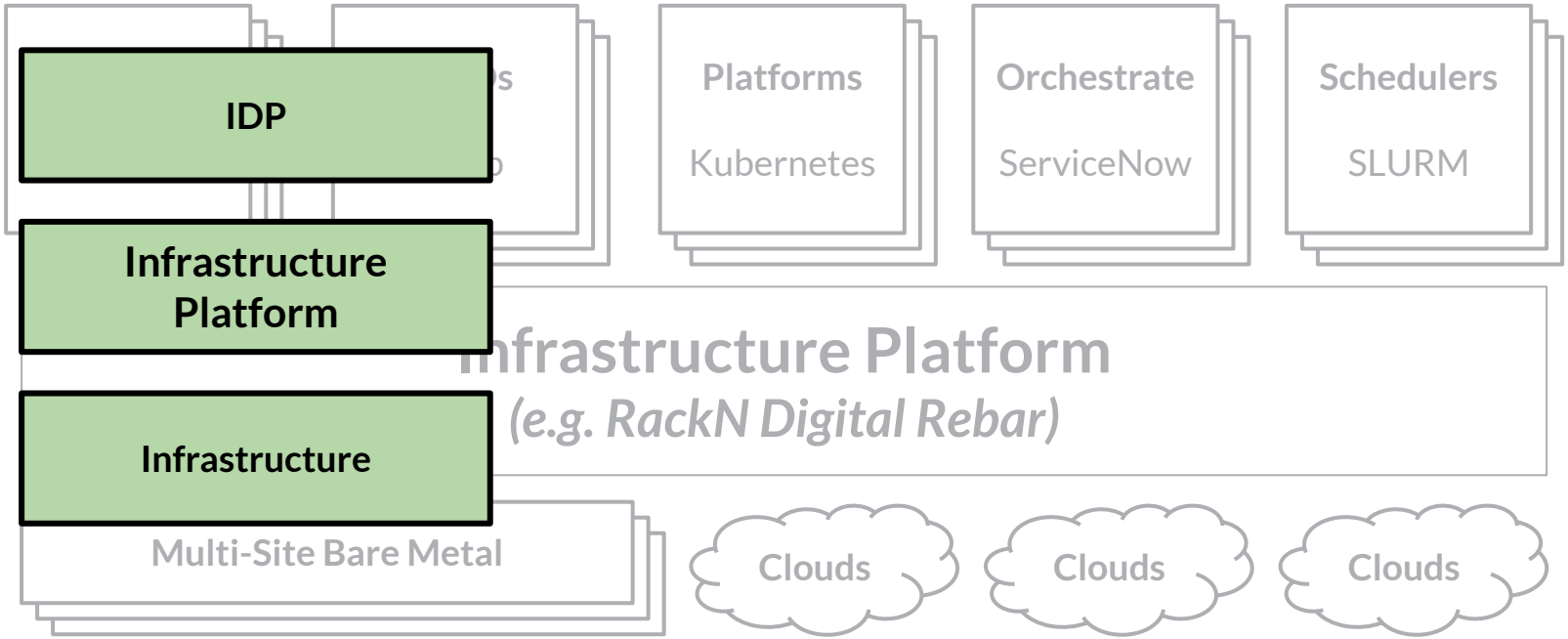


Many Devs with One Ops

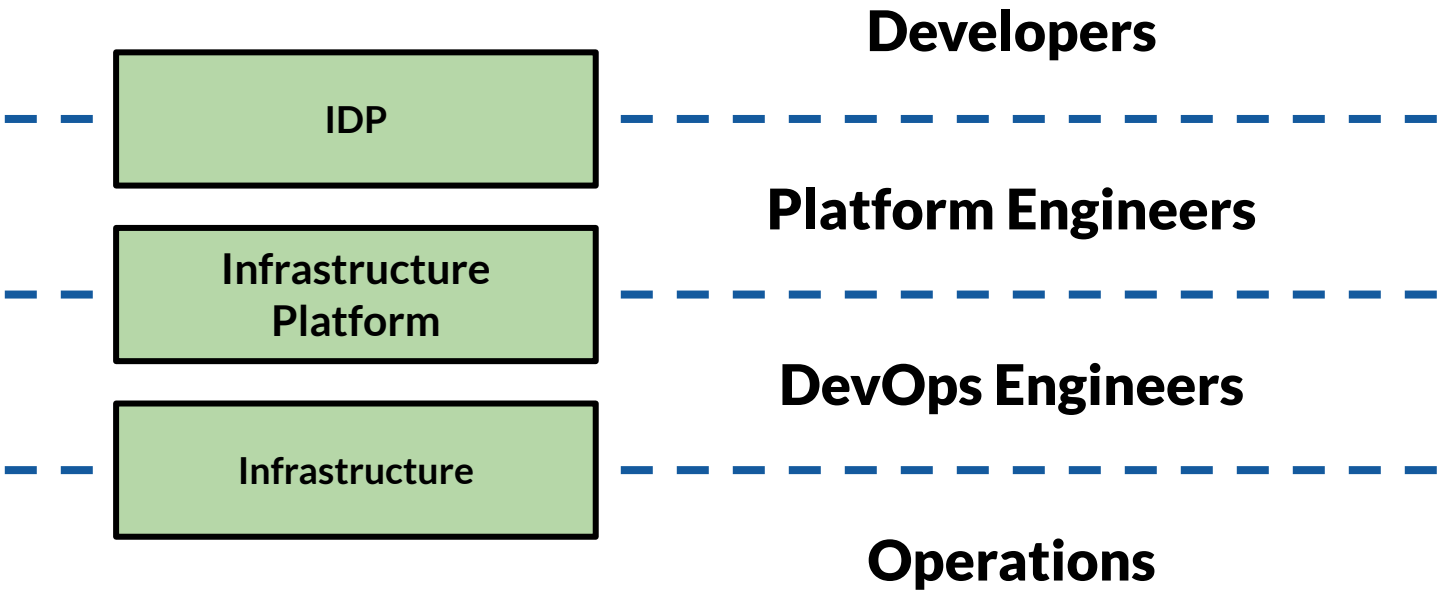


Pexels.com Photo by Jeffry Surianto:

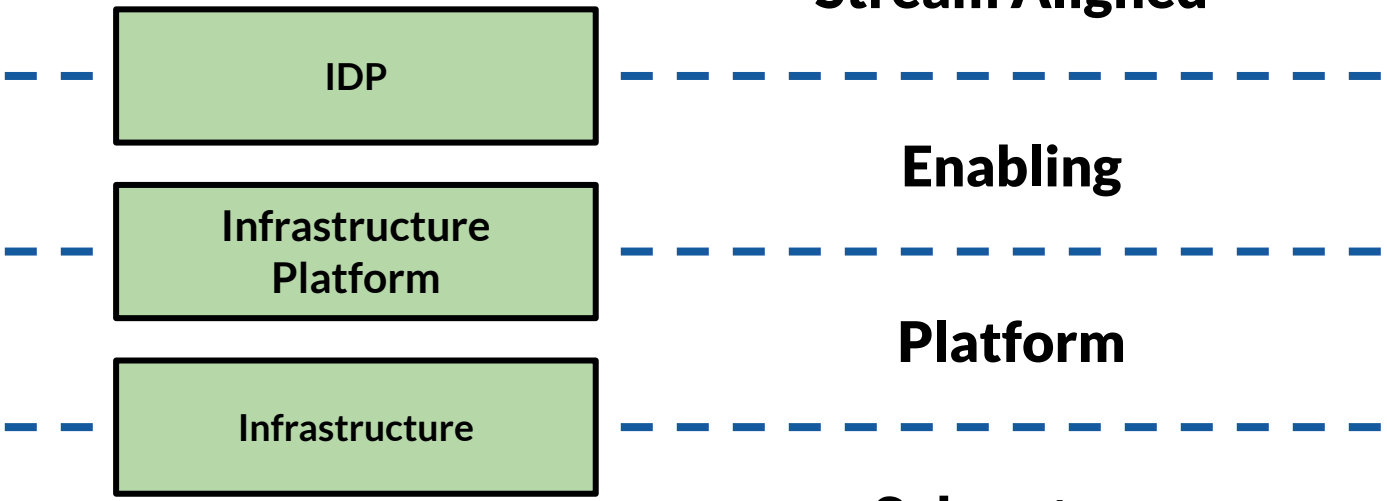
A Layered Approach

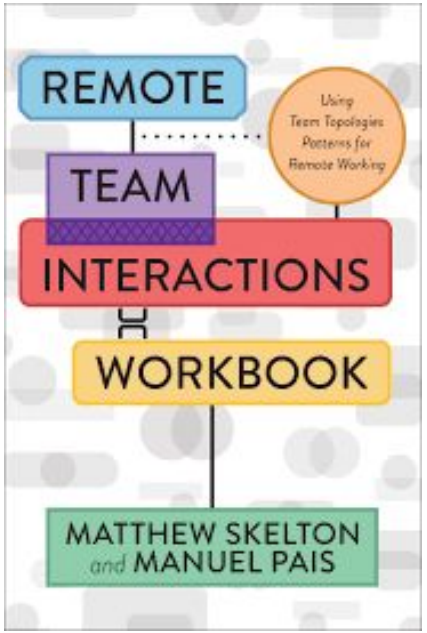


Who does what? (Personas)



Who does what? (with Team Topologies Lingo)



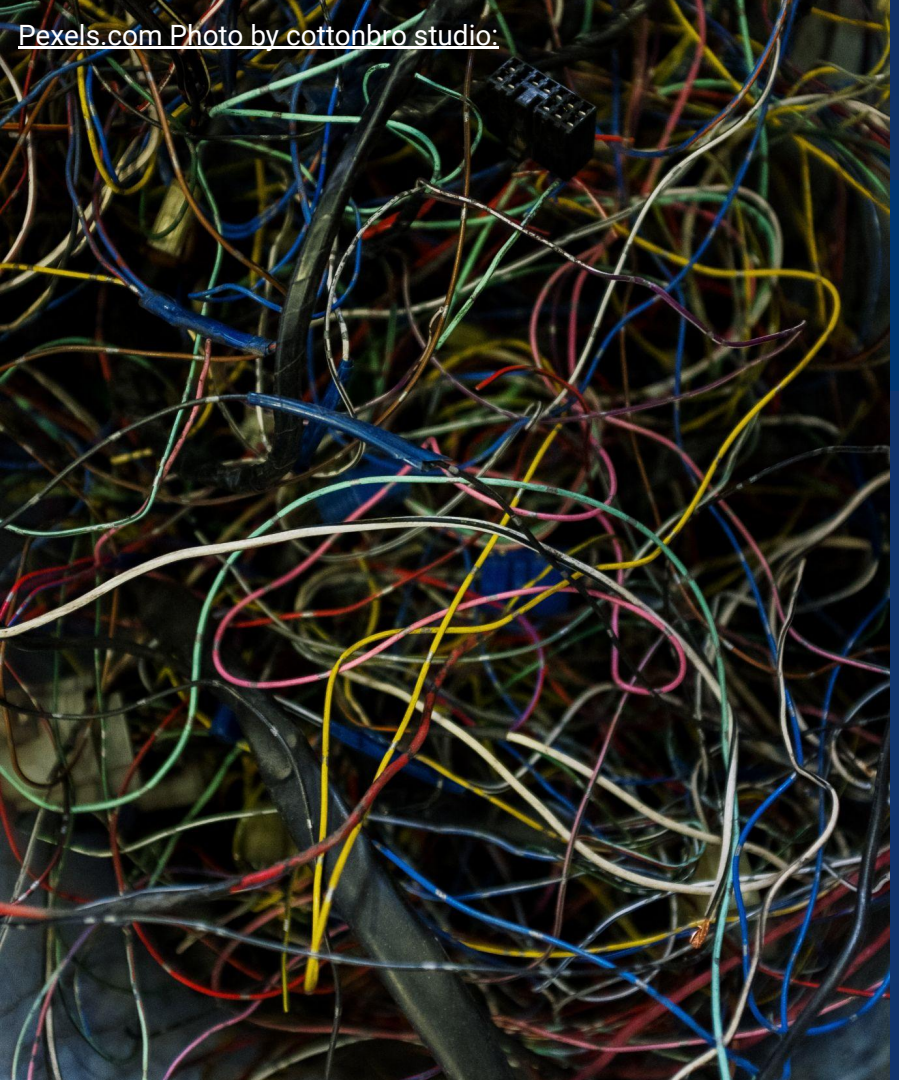


Pexels.com Photo by Kenneth Jencks:

How do we control infrastructure?

1. APIs
2. Events
3. Pipelines

RackN


Pexels.com Photo by cottonbro studio:

Using APIs for Platforms

1. REST & JSON
2. State
3. Tagging
4. Errors

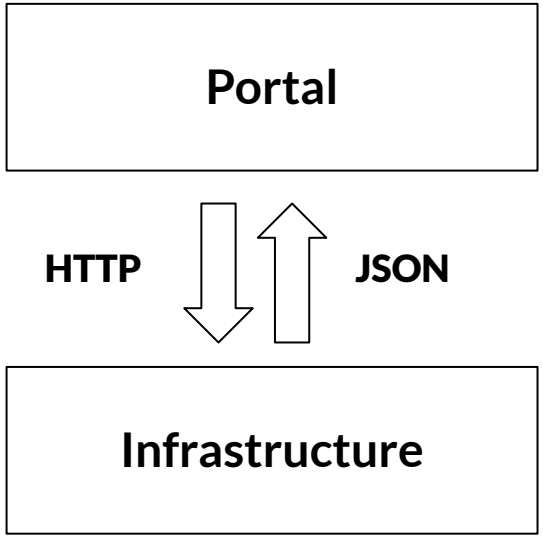
RackN

Using APIs with Platforms

Ideally: call API -> get result

But it's never that simple!

- format / syntax
- access controls
- headers
- error handling



Protip: Use PATCH if possible

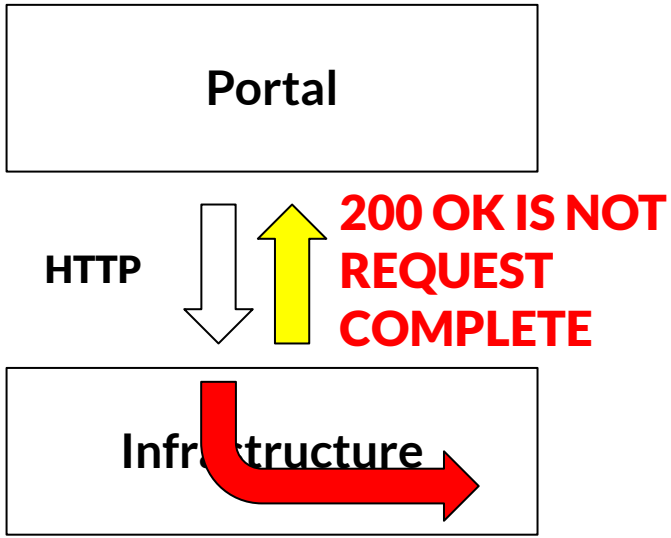
Using APIs with Platforms

Ideally: call API -> get result

But it's never that simple!

- format / syntax
- access controls
- headers
- error handling
- **YOU ARE STARTING A PROCESS**

Protip: Use PATCH if possible



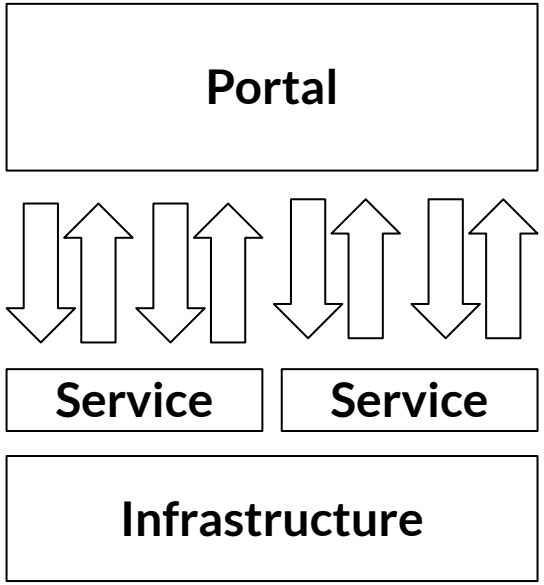
In Practice: APIs with Platforms

Sadly, it's not that simple.

Infrastructure is a collection of many systems and distinct APIs.

To streamline a user request, the platform needs to:

- **decompose the request**
- **coordinate calls to APIs**
- **sequence and check results**



Decompose, Coordinate & Sequence

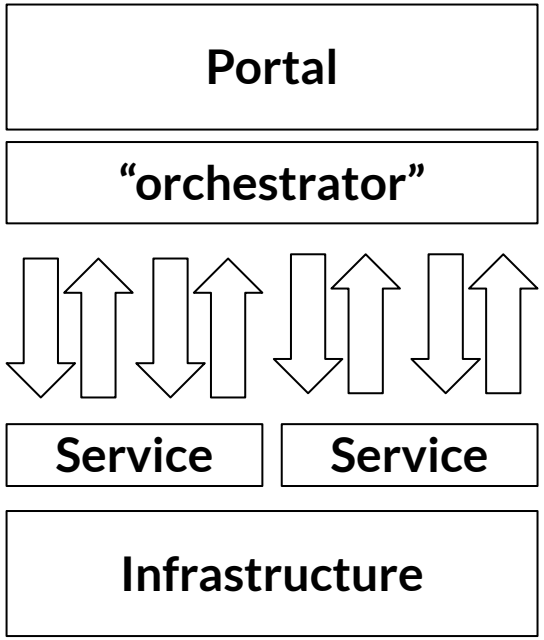
That's a lot of work!

Yes - We call that *Orchestration*

A lot of IDPs embed lightweight orchestration systems

- **Backstage Scaffolding**
- **Terraform**
- **Plumni**

Terraform is lightweight orchestration!?

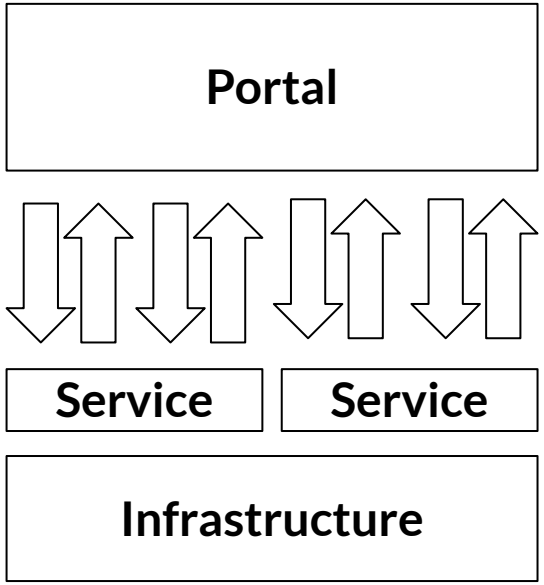


Challenges with APIs

While REST APIs are the best way to control infrastructure, they are not a magic control plane!

You still need to understand best practices!

- **Reconciling State over Timed**
- **Danger of Undo**
- **Version & Protocol Drift**
- **Troubleshooting**
- **Privacy & Trust**



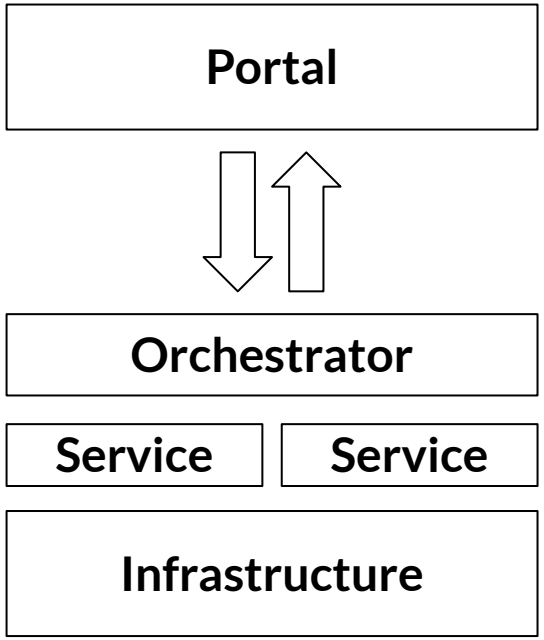
Orchestrators! Back to Simple API?

Ultimately, we want simple API calls:

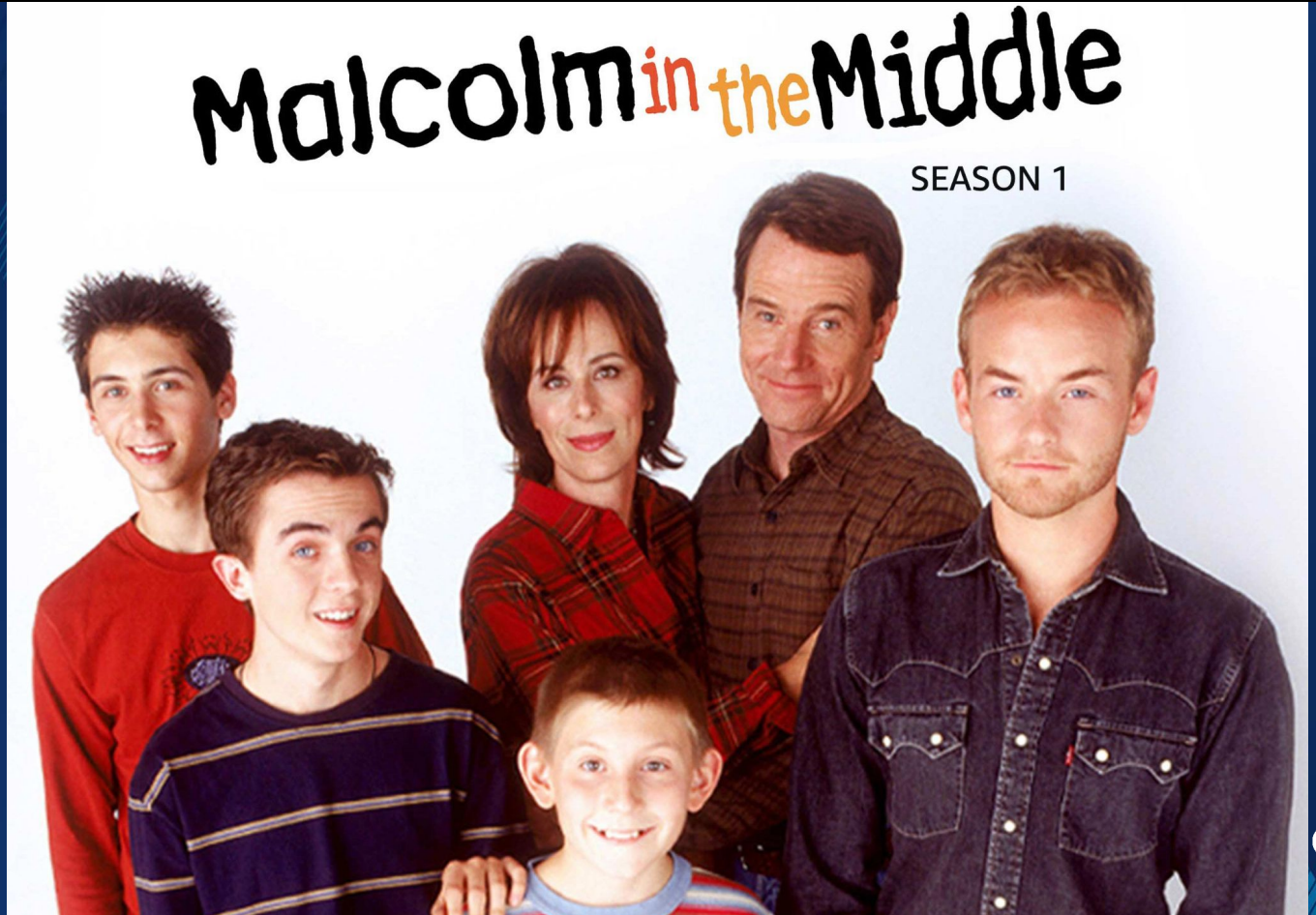
- **Single call to start process**
- **Fast "I got it" response**
- **Ability to check progress**

We need to have a *simple way* to start a process to keep our portal simple.

We'll come back to this when we talk about pipelines.



APIs leave you in the middle



RackN

Pexels.com Photo by Brett Sayles

Using Events for Platforms

1. Handlers
2. Context
3. Scaling
4. Orchestration

RackN

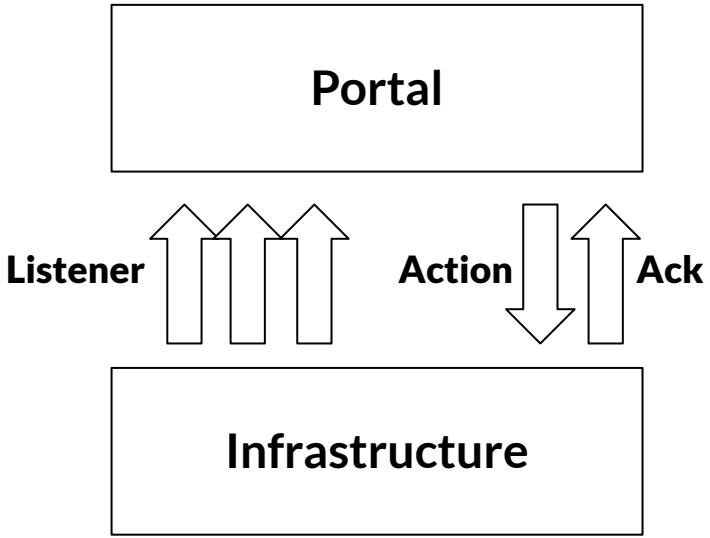
Using Events with Platforms

Ideally: wait for Event -> process it

But it's never that simple!

- handler
- format / syntax
- access controls
- headers
- error handling

Protip: Make sure you can simulate



Improving on Simple Events

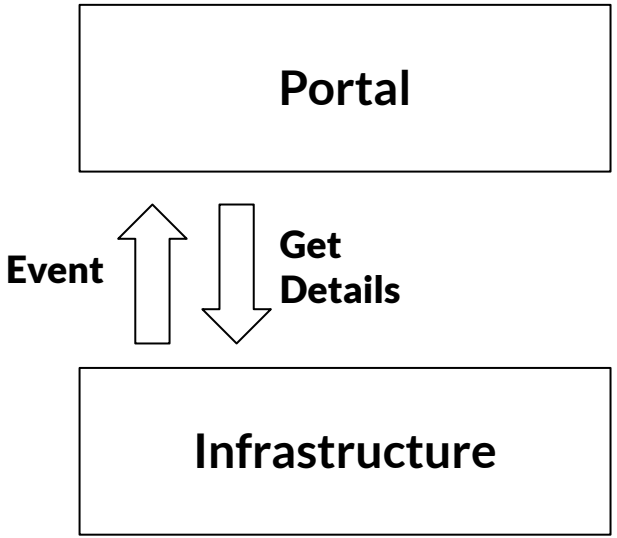
Events need to have *small payloads*

You'll need to get more context

To handle events expect

- **Poll to get additional information**
- **Protect from event floods**
- **Recover from delivery failure**

Still hard manage state



Pexels.com Photo by Josh Hild:

But why does State Tracking matter?

Infrastructure is a resource and you need to keep track of it.

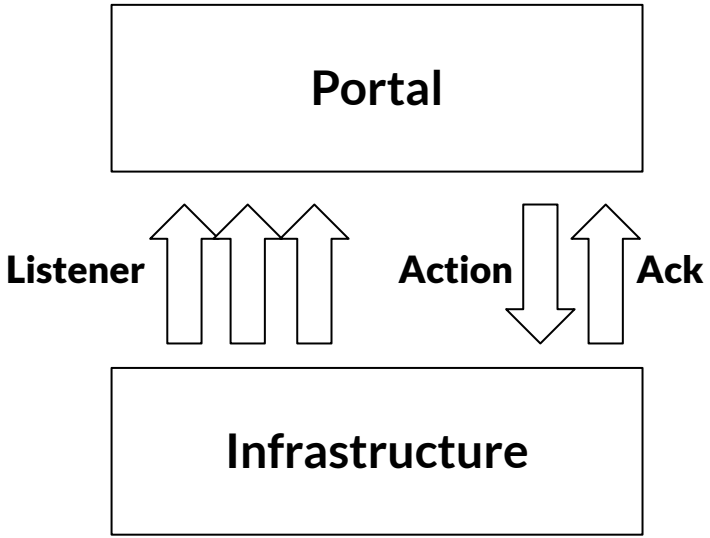
RackN

Challenges with Events

Events make our user experience more responsive, but there's a price.

Challenges:

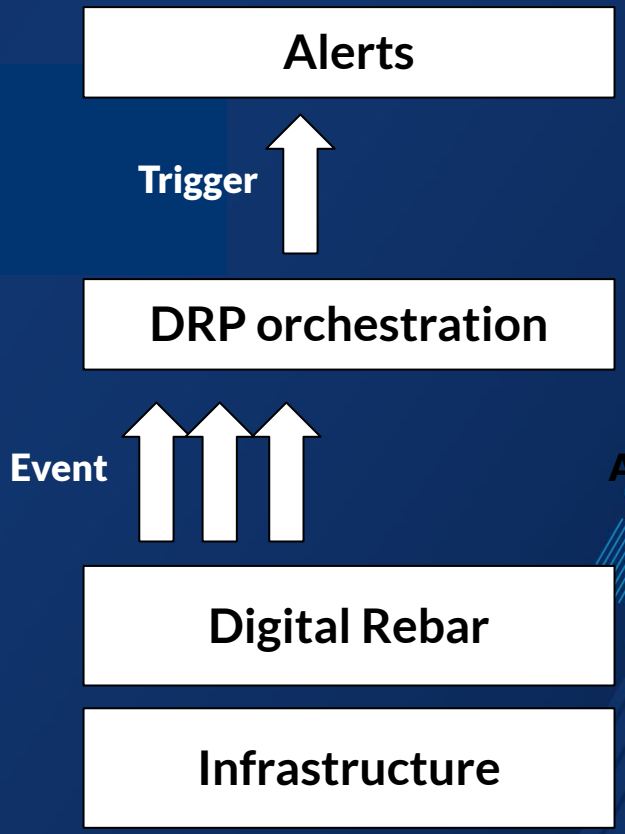
- How to scale event processing?
- How to connect user to event?
- What if user misses event?
- How do you secure event data?
- What happens when there's an error? Is that an event too?



Digital Rebar Story Time!

Digital Rebar has events. We added triggers to process them.

For USERS, we also needed to add alerts.



RackN

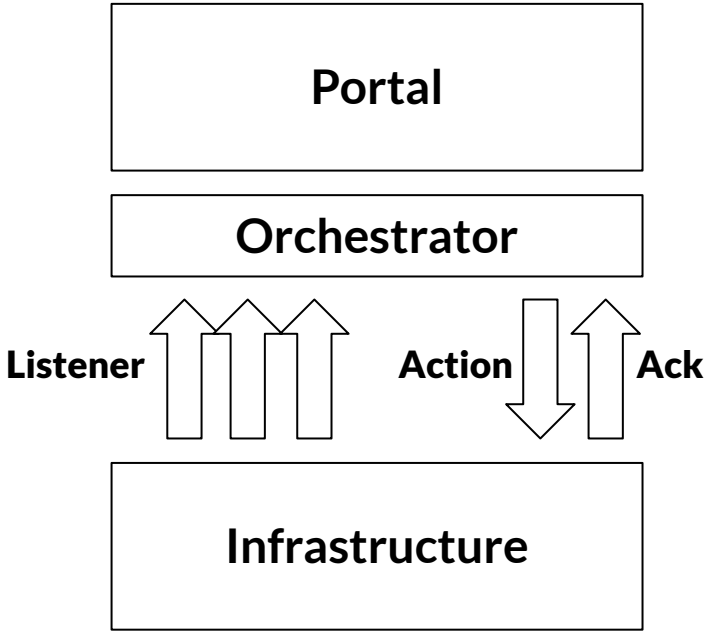
And Platforms Need a Orchestrator!

Portals are for USERS
Orchestrators are for EVENTS

Need simple “if-then-do-that” rules
Source and Sink logic

Ideally

- **standard rules for common sources**
- **ability to define / reuse rules**
- **incorporate business logic**



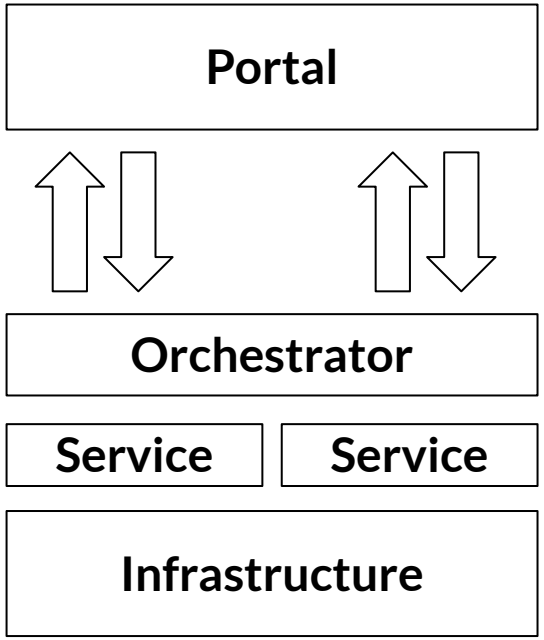
Orchestrators! Back to Simple Events?

Ultimately, we want simple Events:

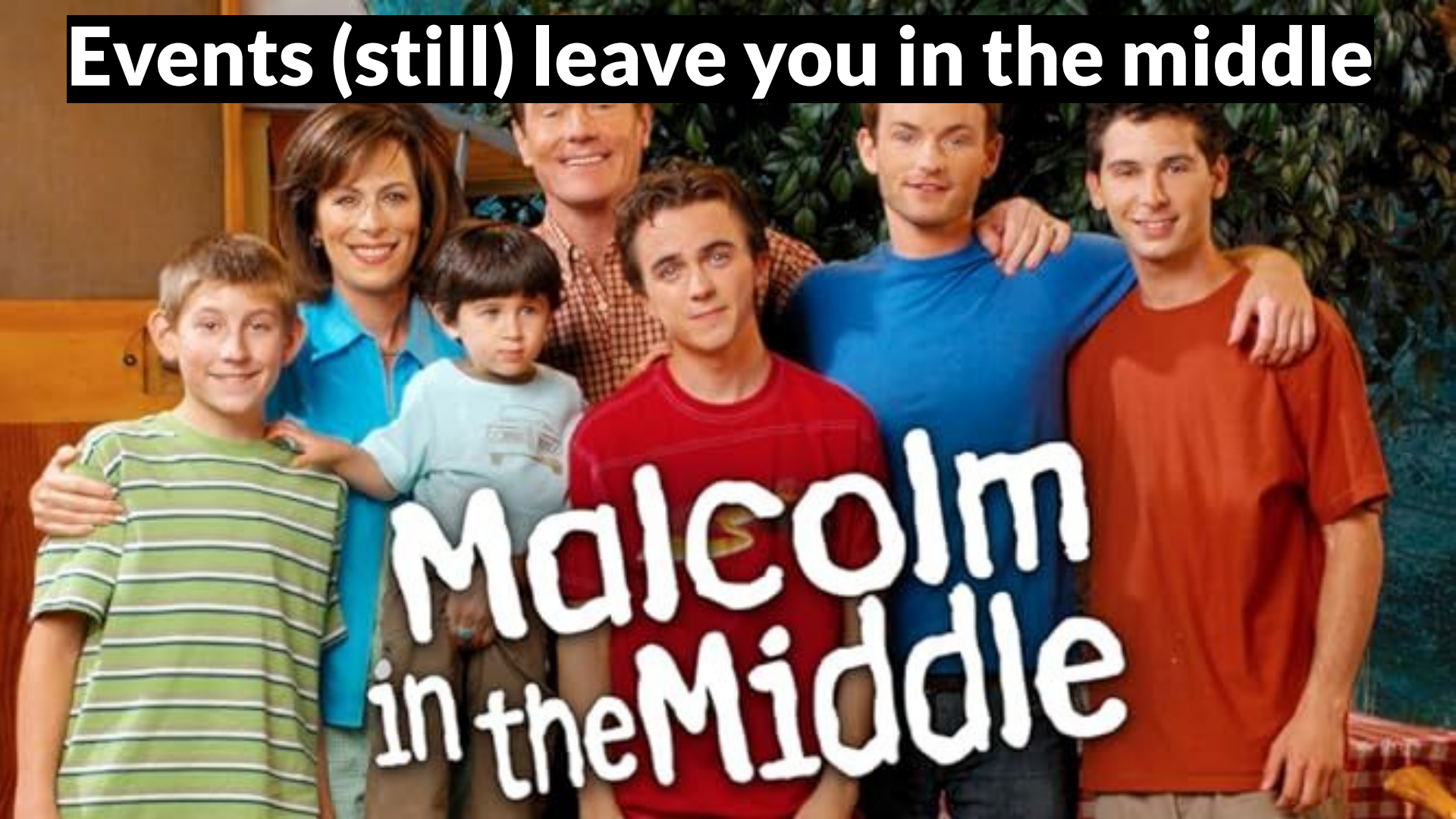
- **Few events linked to user action**
- **Minimal rules or triggers**
- **Ability to observe systems**

We need a *simple way* to handle user events keep our portal simple.

We'll come back to this when we talk about pipelines.



Events (still) leave you in the middle



Pexels.com Photo by - Manouar

Building Infrastructure Pipelines

1. Workflow Focus
2. Observable
3. Multi-Tool

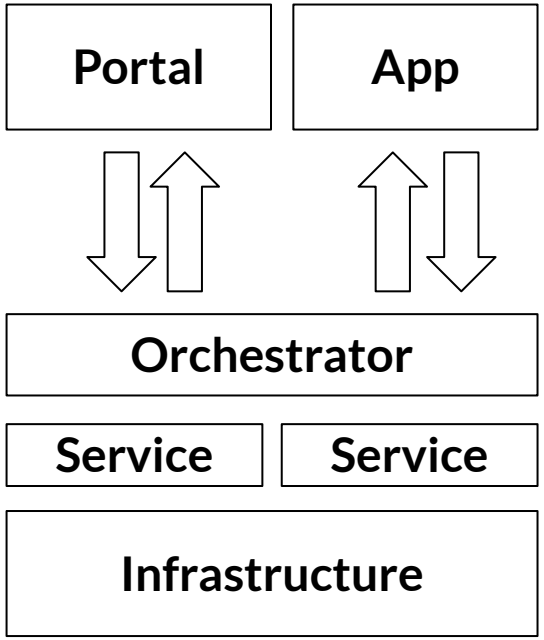
RackN

APIs, Events & Orchestrators

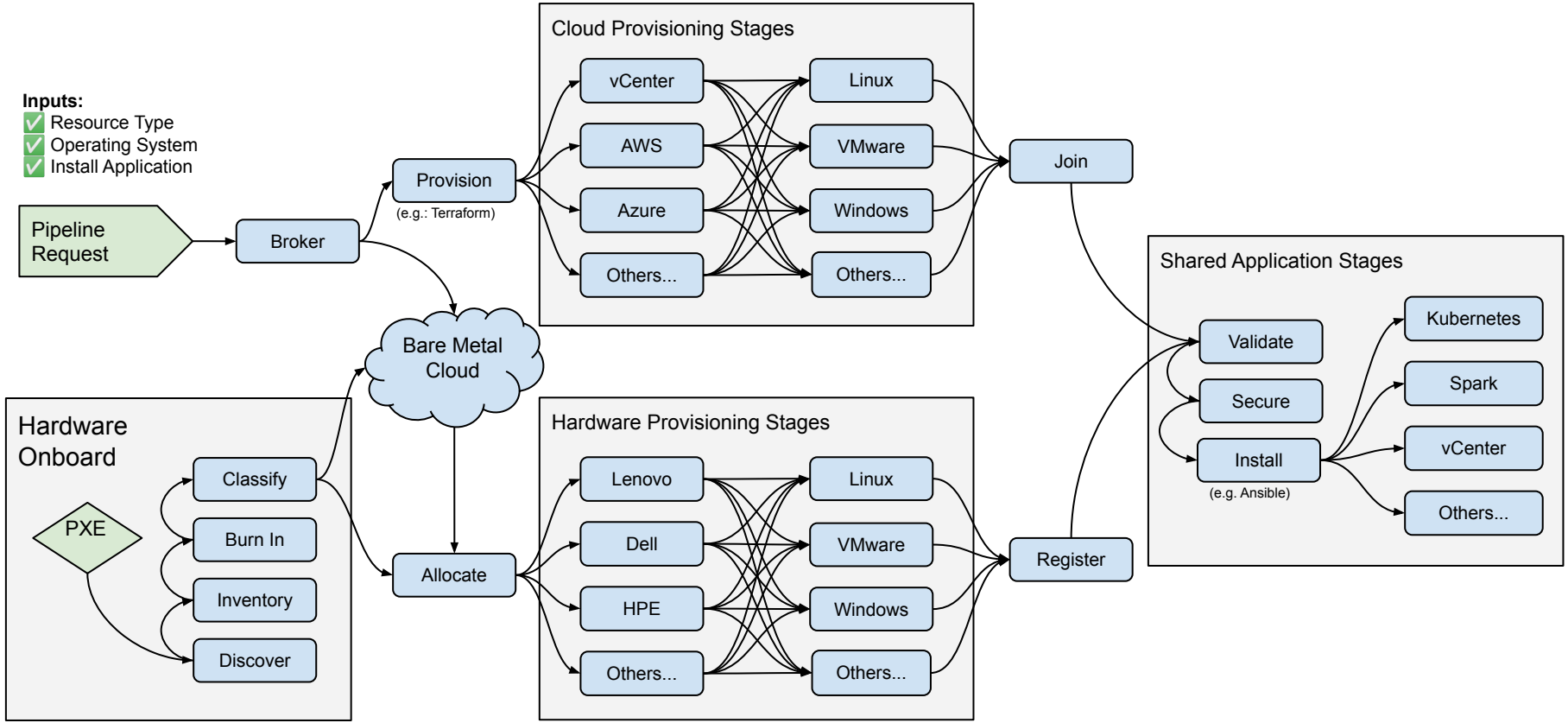
How can we use these together?

- **Portal makes simple API calls**
- **Orchestrator coordinates long process chains (pipelines)**
- **Pipelines are reusable for multiple consumers**

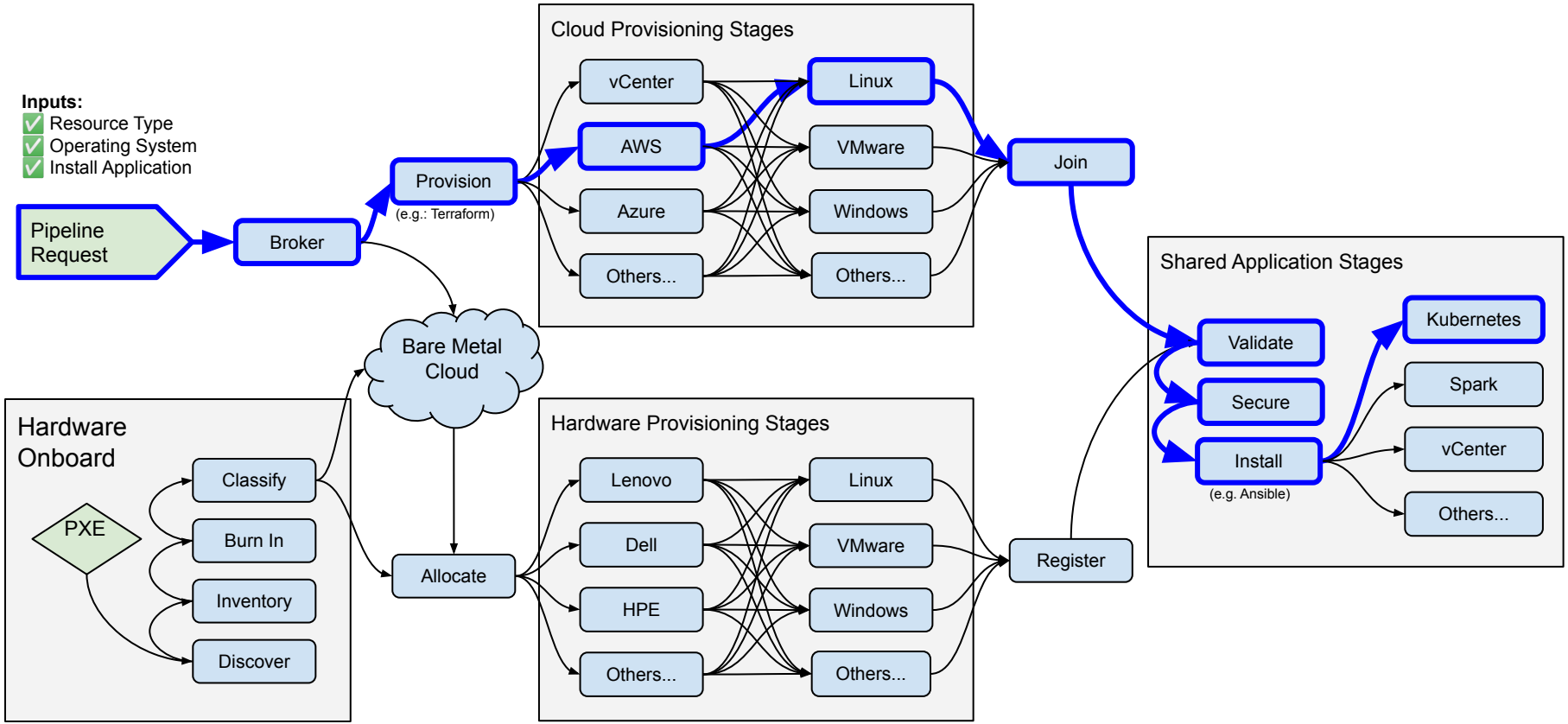
Allows operations teams to have offer streamlined and standardized service catalogs.



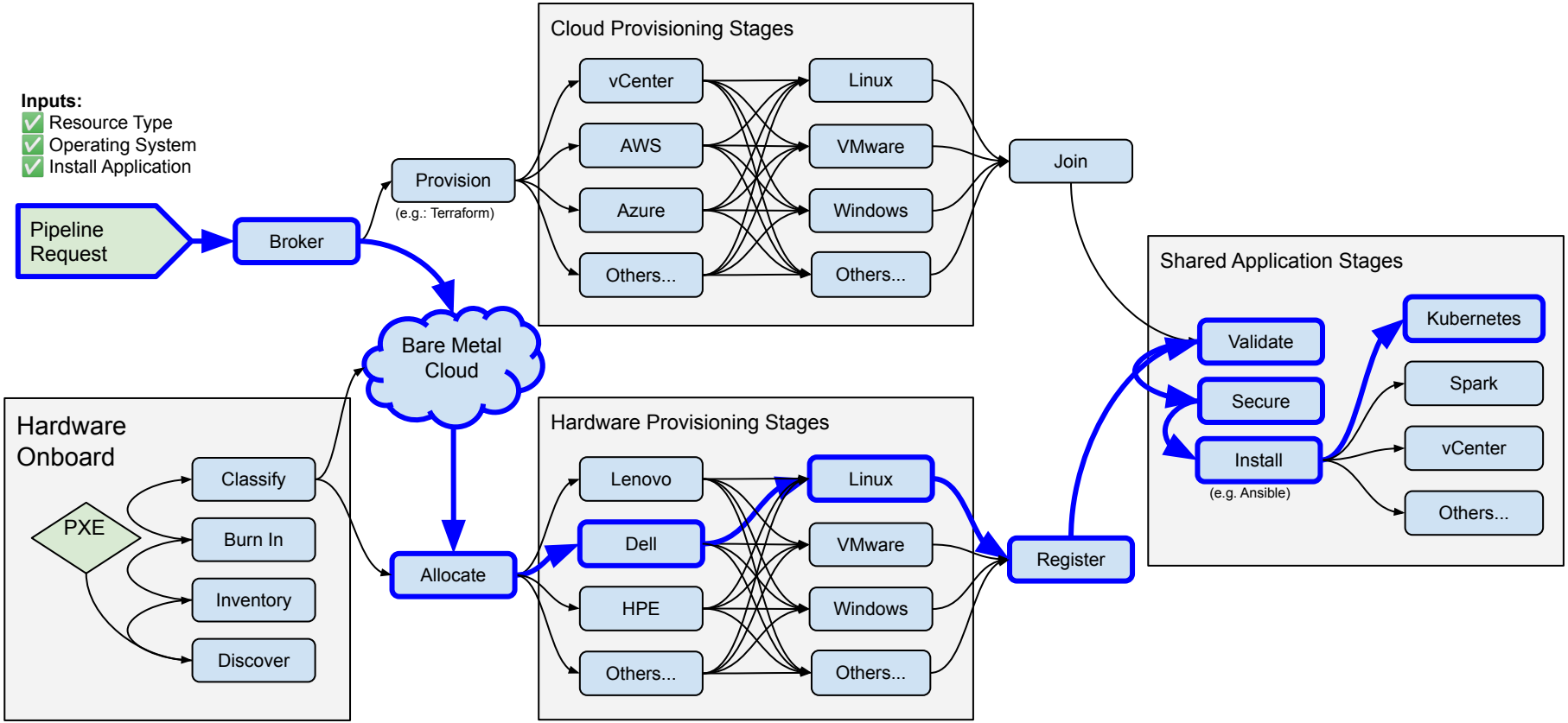
What are Infrastructure Pipelines?



Example: Cloud Infrastructure Pipelines

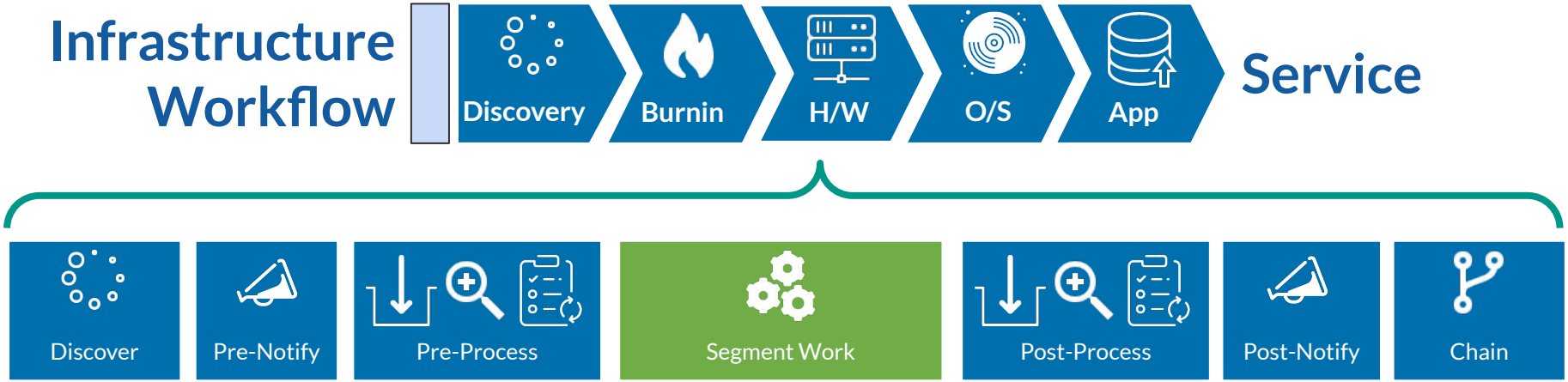


Example: Metal Infrastructure Pipelines

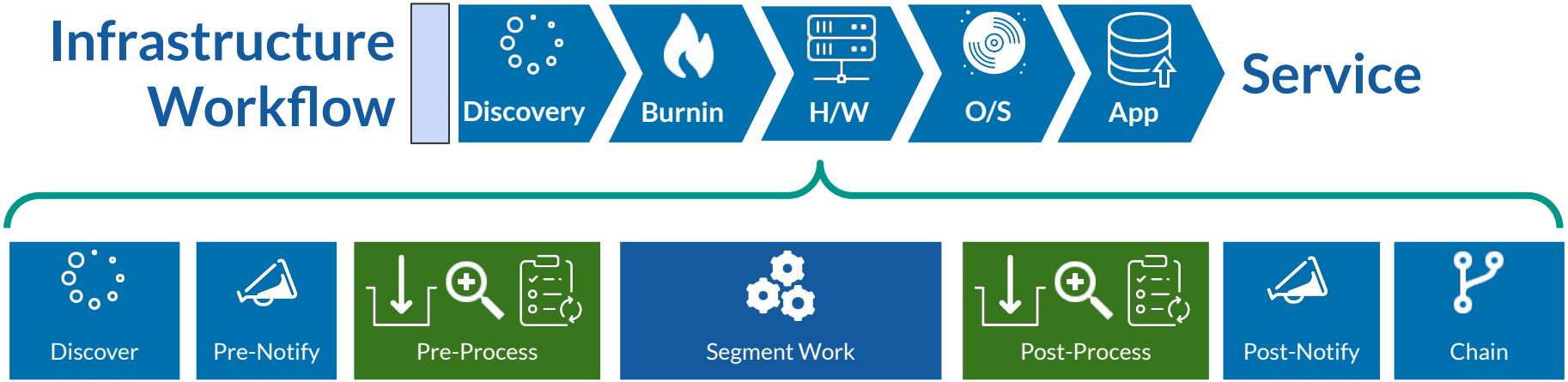


RackN Composable Workflow

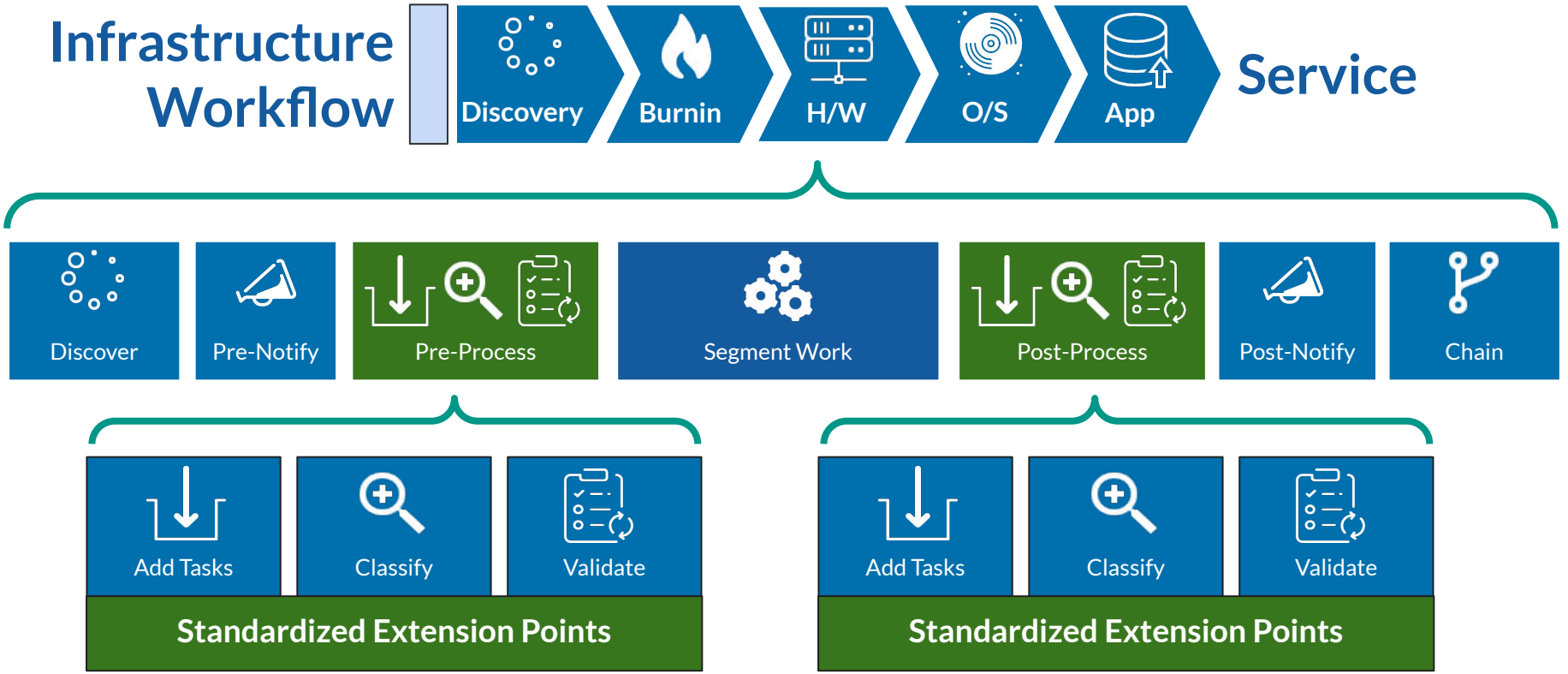
RackN Composable Workflow



RackN Composable Workflow



RackN Composable Workflow

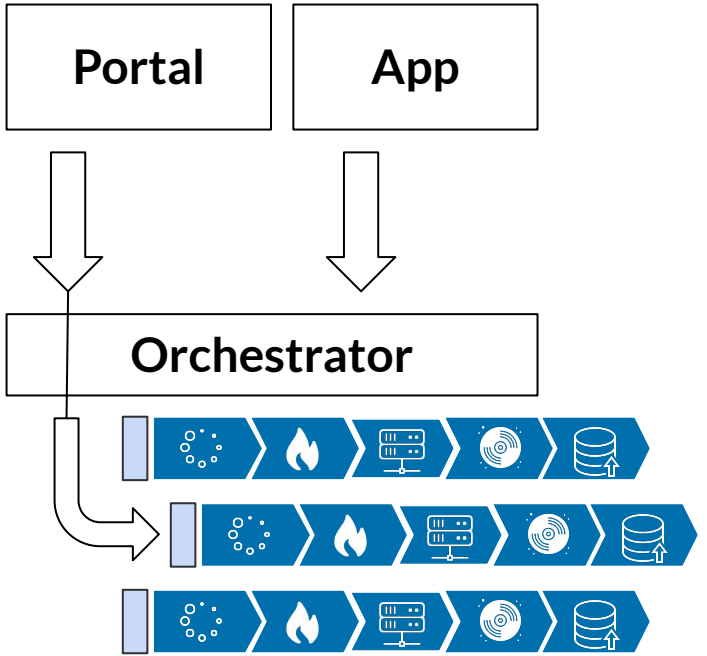


Request Driven Pipelines

Pipelines

- Portal makes simple API calls
- Orchestrator coordinates long process chains (pipelines)
- Pipelines are reusable for multiple consumers

Allows operations teams to have offer streamlined and standardized service catalogs.



Pexels.com Photo by Harrison Haines:

“We Already Have Too Much Automation”

VP of Infrastructure, Top US Bank

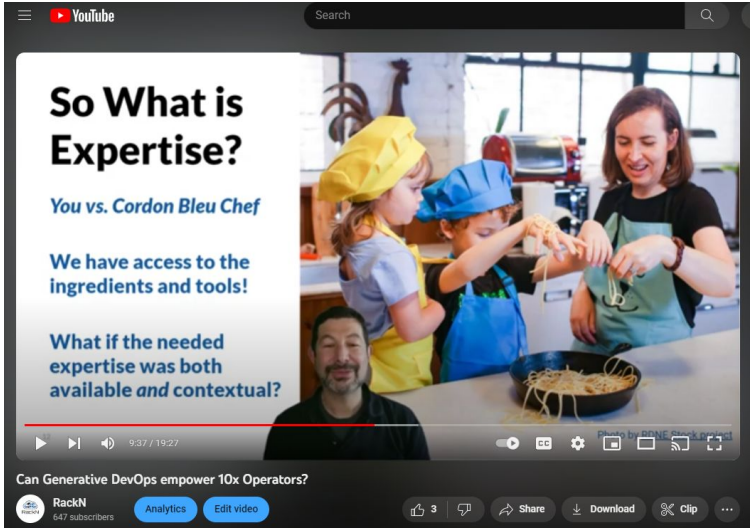
If so, Platforms will just add to the existing a dumpster fire!

How can we fix automation?

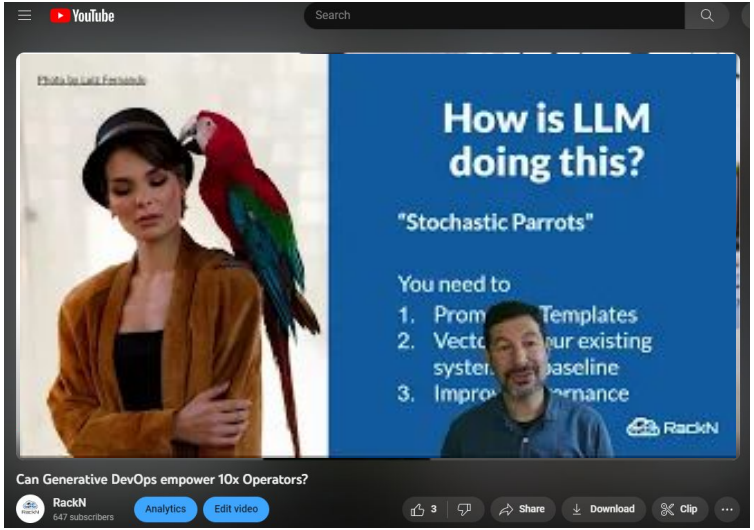
RackN

Learn Generative DevOps

[Can Generative DevOps empower 10x Operators?]

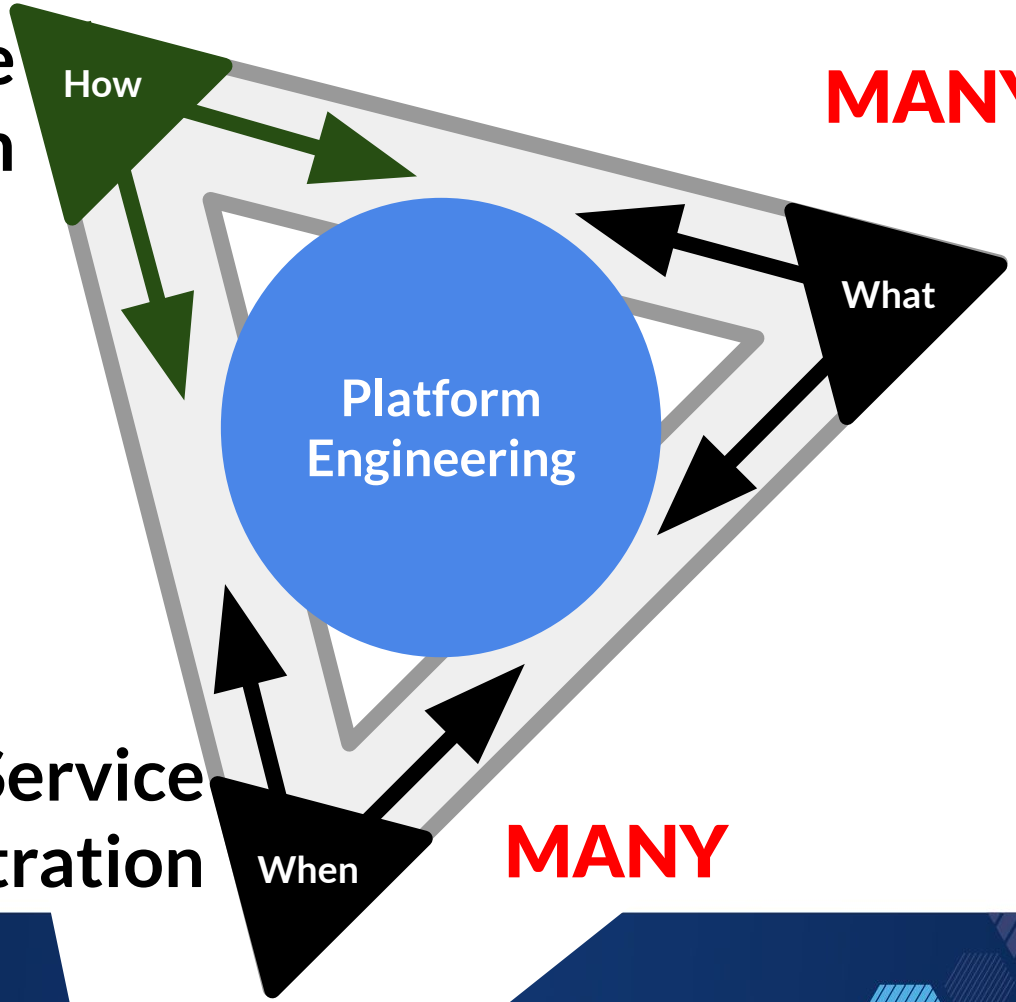


[Better DevOps with Generative AI]



Infrastructure Platform

ONE



MANY

Control Plane(s)

Service Orchestration

MANY

Questions?

Want to become 10x Operator?
Visit us at RackN.com

Rob Hirschfeld
RackN CEO & Co-Founder
@RobHirschfeld
@Zehicle