

20TH ANNIVERSARY EDITION

WSO2CONASIA

— PLATFORMLESS MODERNIZATION

Monetizing APIs Effectively



Derric Gilling
Co-Founder & CEO
Moesif





Derric Gilling— Nice to meet you

- CEO of Moesif, the leading **API analytics and monetization** platform, now part of WSO2
- Helping enterprises with their API productization and monetization strategy

Why Monetize APIs?

- 1 Unlock value in proprietary data and content
- 2 Customers require flexibility / pointed solutions
- 3 New revenue streams / market share
- 4 Competitive advantage

AI and automation is changing how software is consumed



- ⌘ Users log in
- 🖱 Value through **UI**
- 💰 Traditional **Subscription**

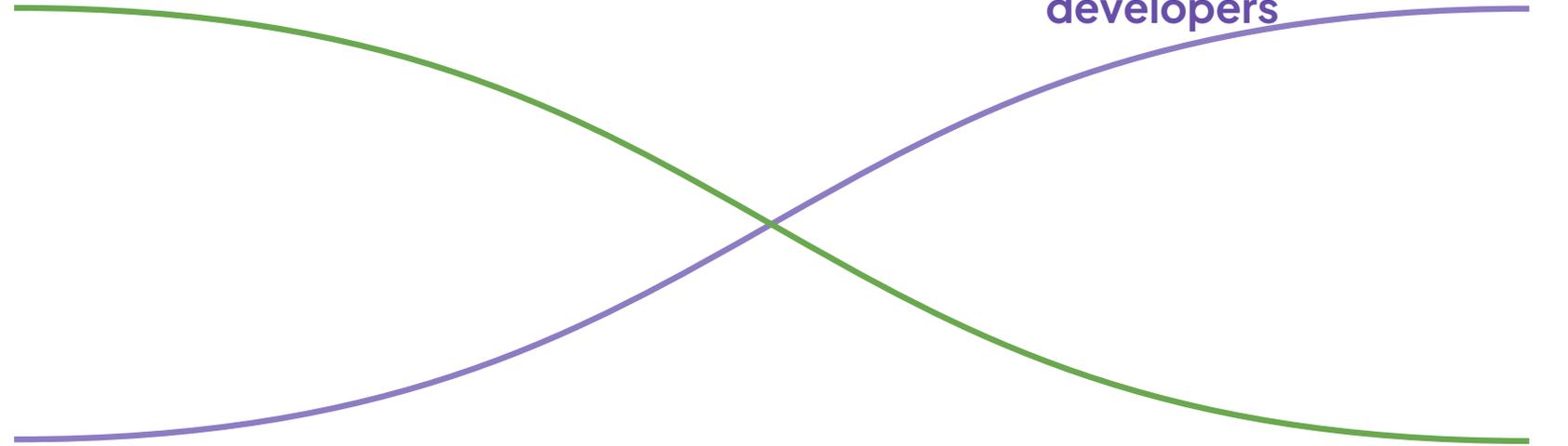


- ⌘ Automation / **AI**
- 🔄 Value through **APIs**
- ⌘ Value-based pricing

Buying power shifted to individual devs

Cost of software

Authority of individual developers



2005

Perpetual/
On-Prem

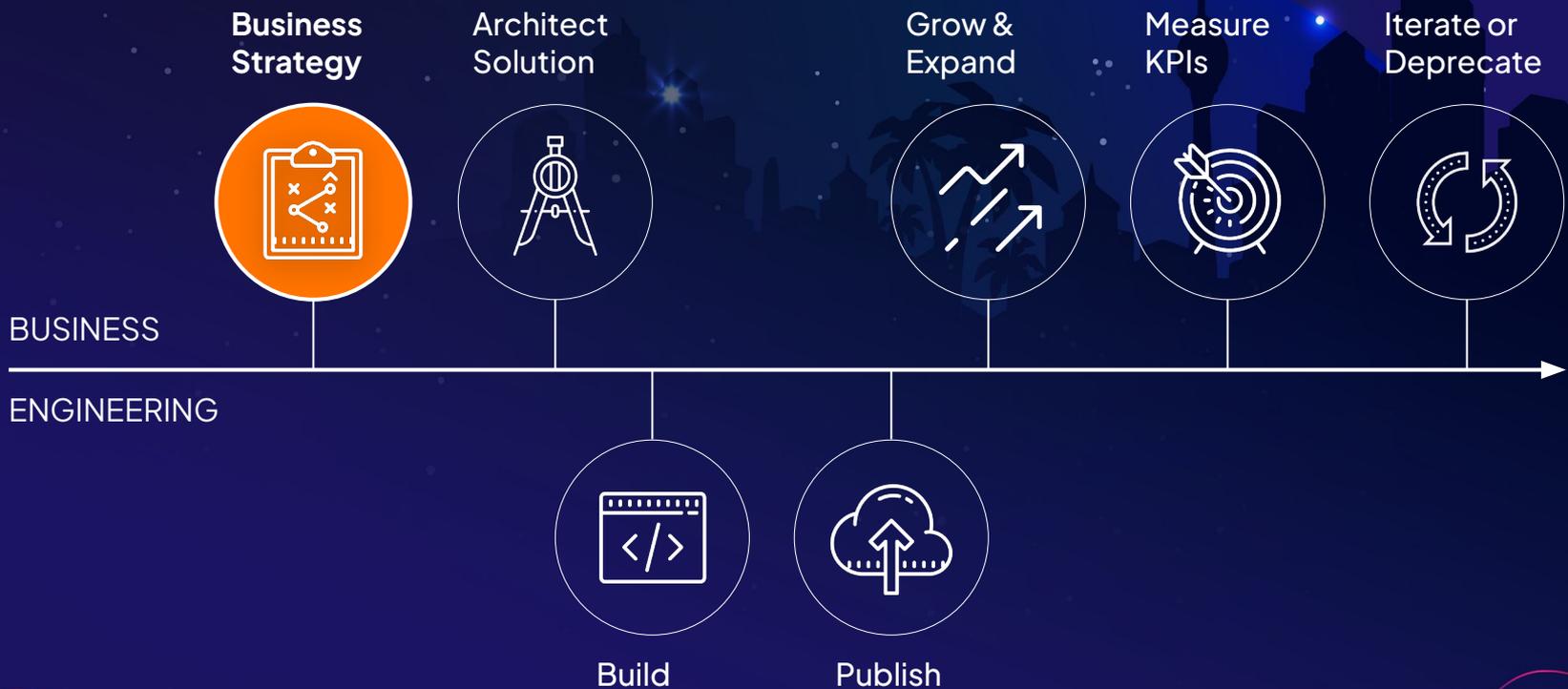
2015

SaaS

2025

API/AI

Let's first define the **API business strategy**



Each API business is different

- 1 What to charge for
- 2 Prepaid vs. postpaid
- 3 Packaging strategy
- 4 When to invoice

Pricing aligned to customer value

Transaction volume



Revenue/cost share



Data volume



User-centric



Resource



Billing strategy that works for customers

Pros:

Cons:

Prepaid billing:

Customer commits to credits / quota ahead of time before consumption

- ✓ Better cash flow for higher COGS like AI
- ✓ More familiar with traditional SaaS

- ✗ Friction in onboarding
- ✗ Harder customers with varying usage or ROI

Postpaid billing:

Customer only pays for their usage “in arrears” after service is already consumed

- ✓ Less onboarding friction as no commit needed
- ✓ Easier for unpredictable usage

- ✗ Hard for budgeting
- ✗ Can be abused / credit risk
- ✗ Unfamiliar for some industries

Packaging strategy that works for growth

Pros:

Cons:

Tiered:

Traditional SaaS tiers with predefined set of features and quota

- ✓ Enforces a min spend
- ✓ Predictable for customer
- ✓ Easy to implement

- ✗ Friction in expansion
- ✗ Rigid packages, not aligned to customer value

Pay-As-You-Go:

Usage-based or consumption-based pricing based on a unit price

- ✓ More efficient for customer
- ✓ Less friction in expansion
- ✓ Can “appear” cheaper

- ✗ Can upset customers with billing surprises
- ✗ Complex to implement

Invoicing strategy that works for finance

Pros:

Cons:

Recurring:

Customer invoiced on a schedule like each month, quarter, or year

- ✓ Easier for revenue recognition/GAAP
- ✓ Predictable revenue and cash flow

- ✗ Bad unit economics for low cost SaaS or non-cyclic usage
- ✗ Complex if prepaid

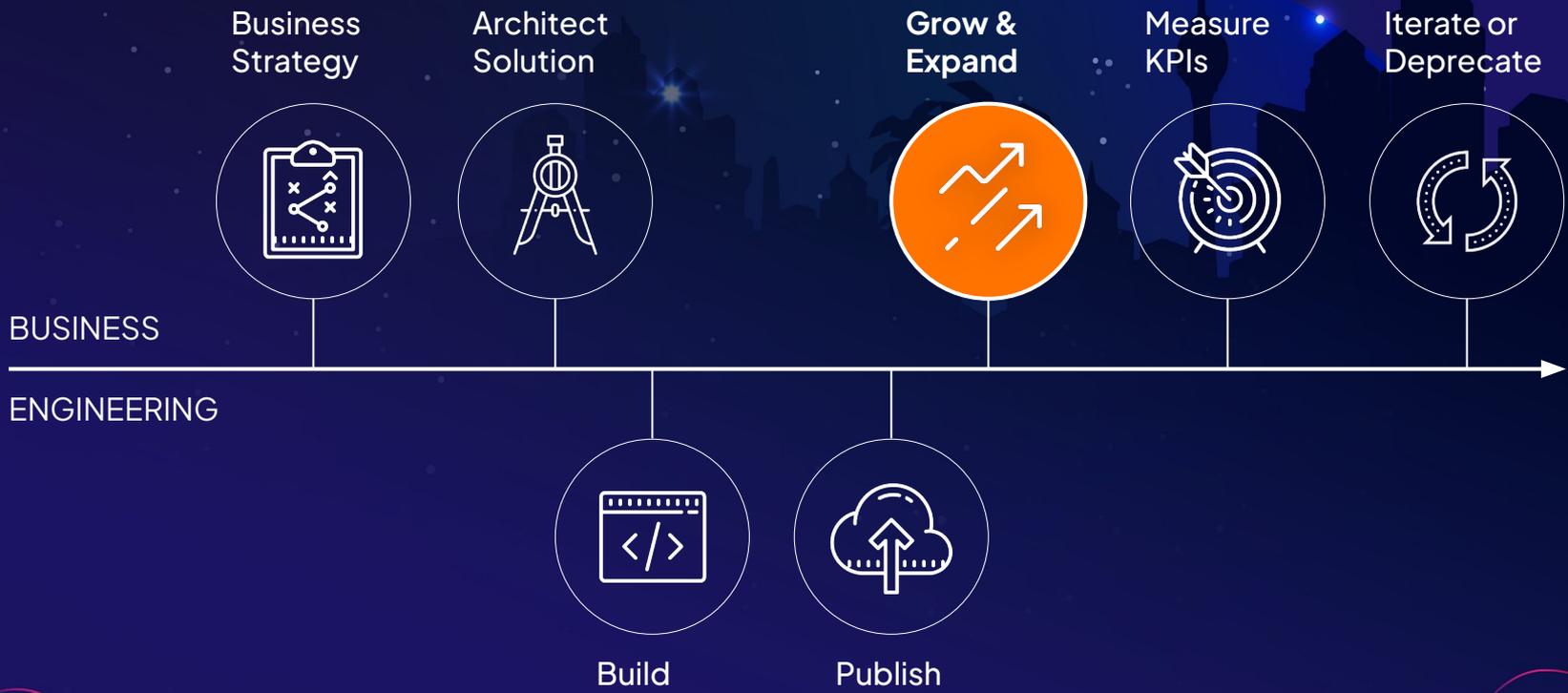
Threshold:

Customer invoiced after a credit threshold is met or auto top-off (prepaid)

- ✓ Reduces transaction overhead cost
- ✓ Auto top-off makes prepaid PAYG easier
- ✓ Better for non-cyclic usage

- ✗ Harder for finance to recognize revenue
- ✗ Cash flow becomes complex to track

Growing an API product



Each API GTM is different

- 1 Developer-first/PLG
- 2 Sales-led
- 3 Partner-led

Getting developers to adopt is hard – Many stakeholders



Legal &
Security



Project
Priorities



Budgeting &
Procurement

Reduce the number of obstacles to demonstrate initial value



Legal &
Security

Click-through
ToS



Project
Priorities

Self-service
onboarding



Budgeting &
Procurement

Paywall with
credit card

Land lots of developers first

Get developers to realize value and pay a token amount

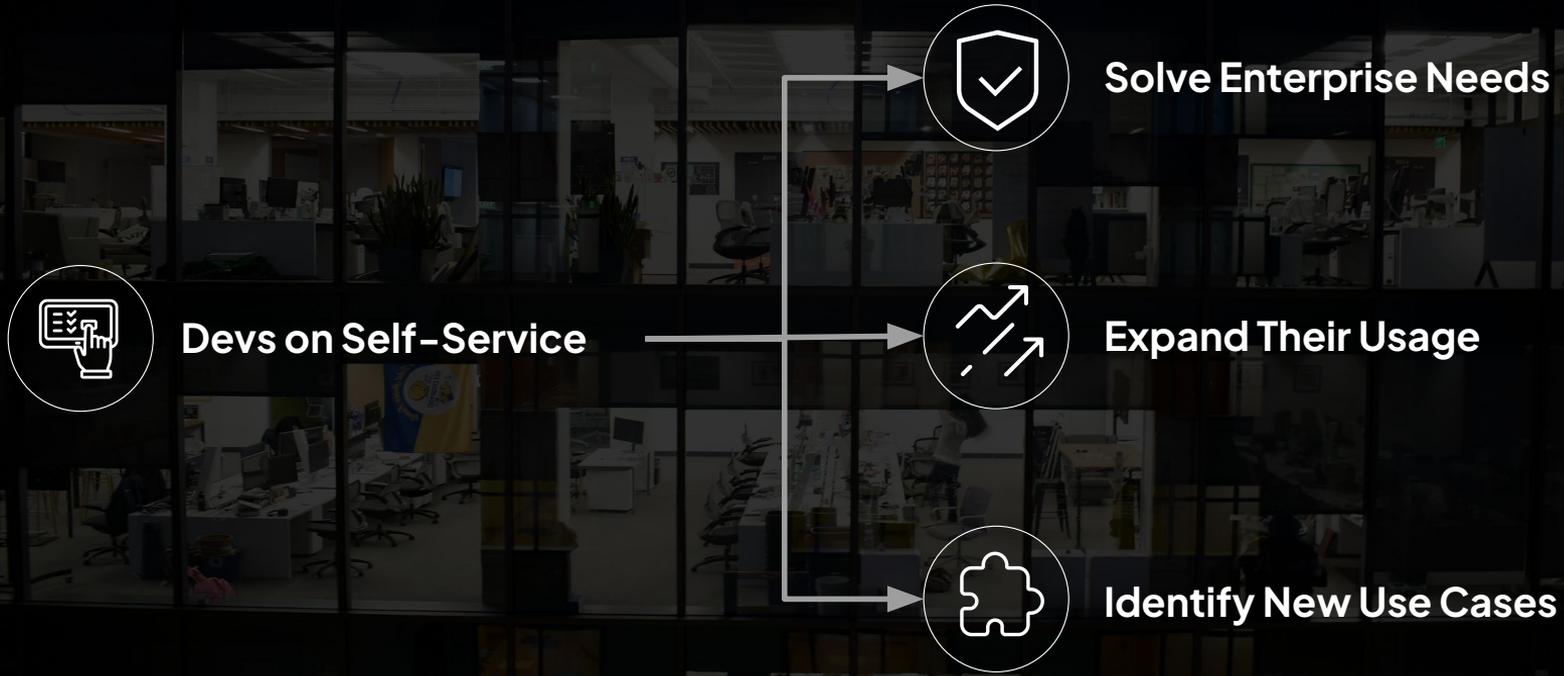


Large adoption
funnel

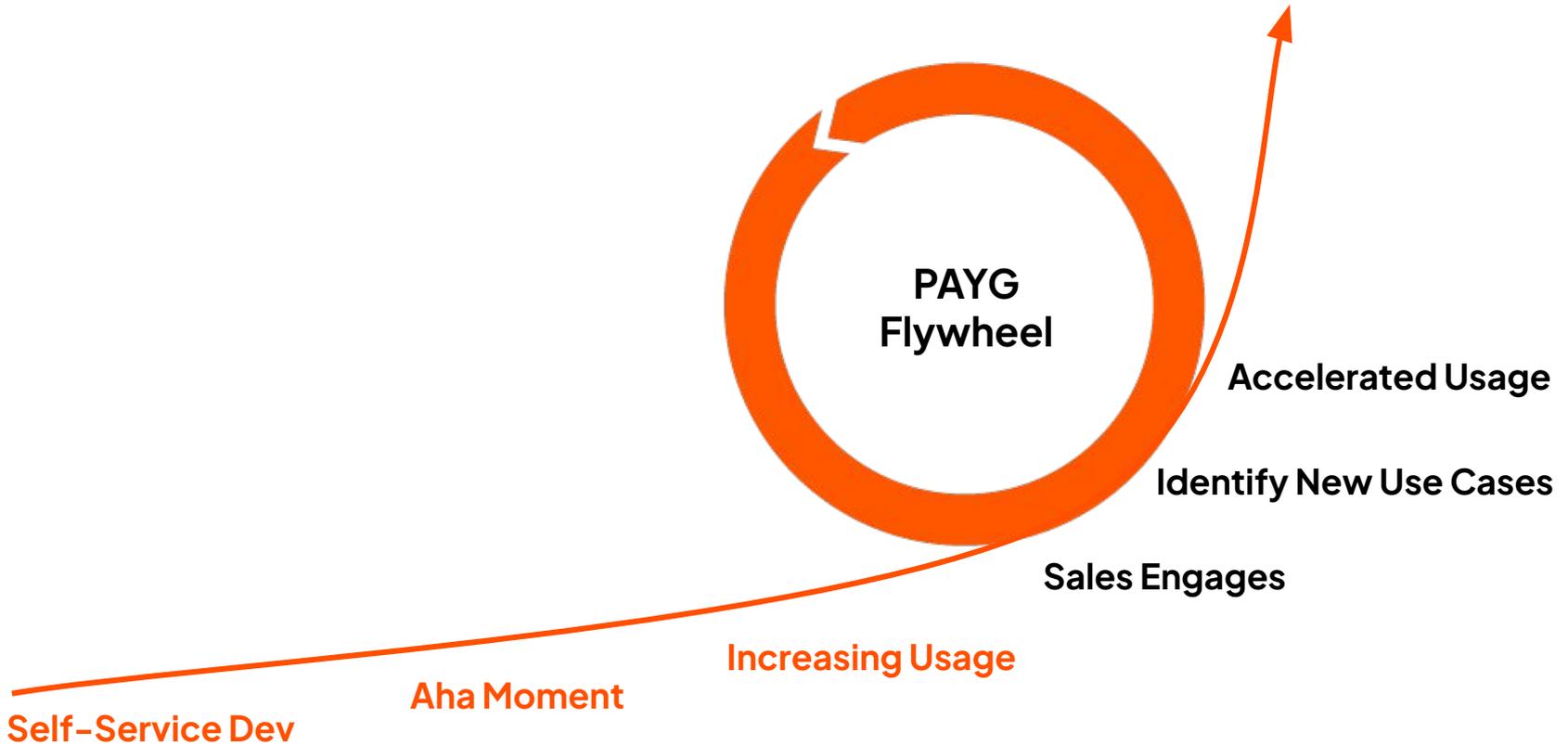


Fast time-to-
paying

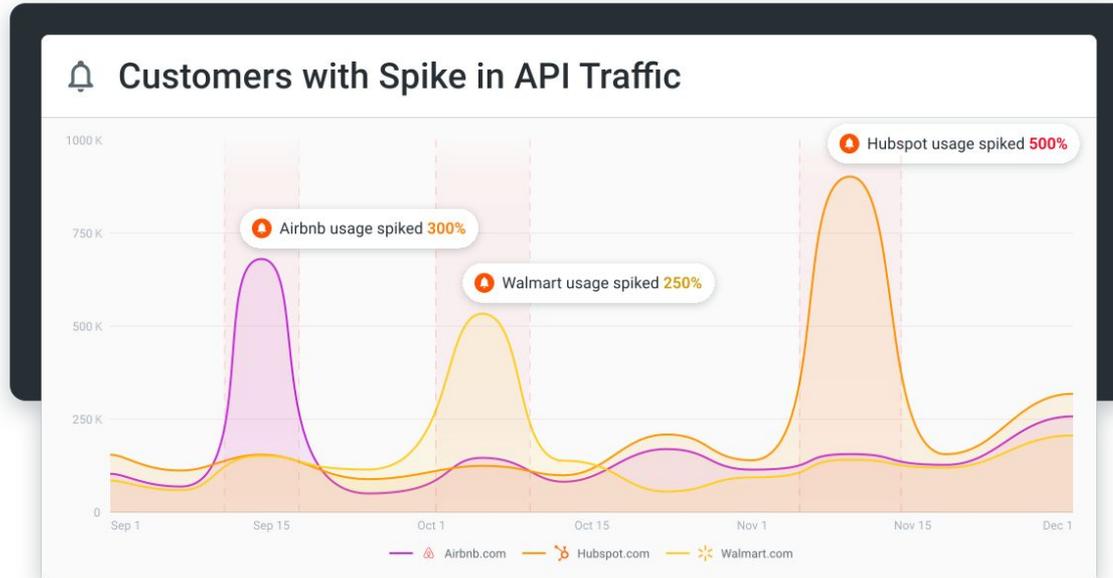
Then, sell through existing developers



Driving the developer-first/PAYG flywheel



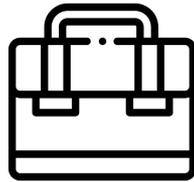
Proactively monitor usage for key accounts and reach out



Scaling developer-first is hard



Issues during
onboarding



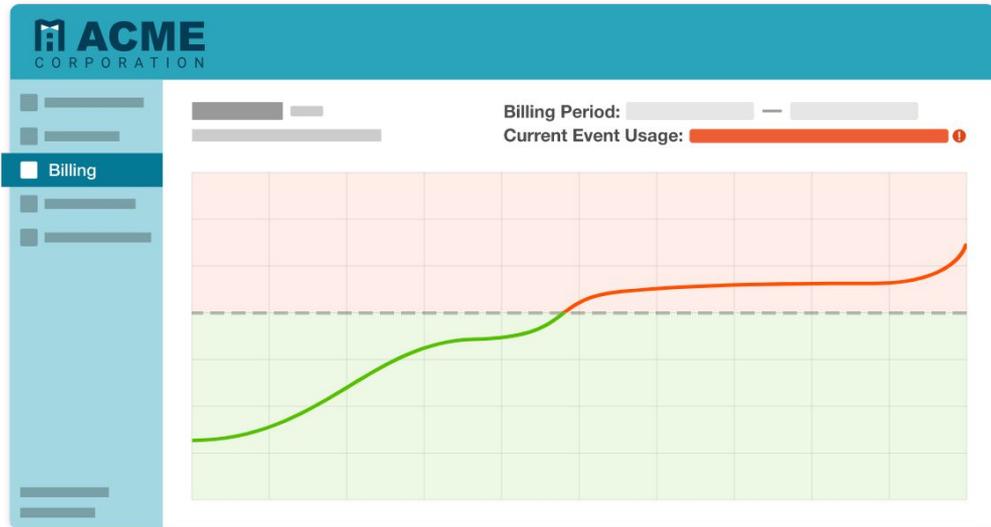
Changing
business needs



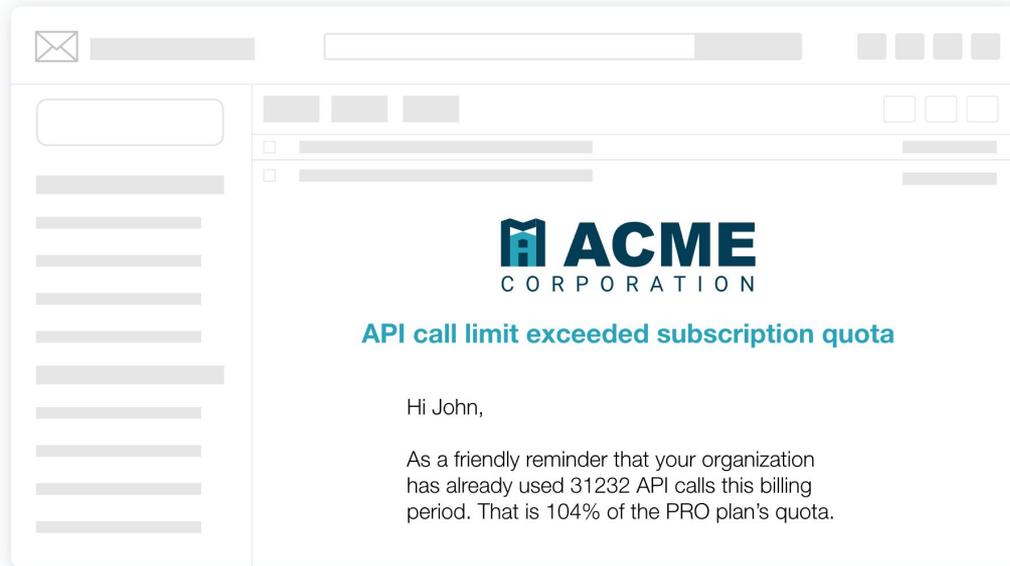
Billing surprises
and issues

Reduce support cost with a great self-service experience

- Provide visibility and reports on their usage
- Provide tools to upgrade and manage their account without involving personal



Avoid surprises by over-communicating



- Automatically notify customers when they approach quota and budget limits
- If Pay As You Go, empower customers to set their own usage thresholds

Measuring KPIs and metrics



Infrastructure metrics align to eng goals, not product goals



Uptime



Resource
Utilization



Performance



Errors



Requests
Per Minute

Product owners should focus on business goals:



Adoption

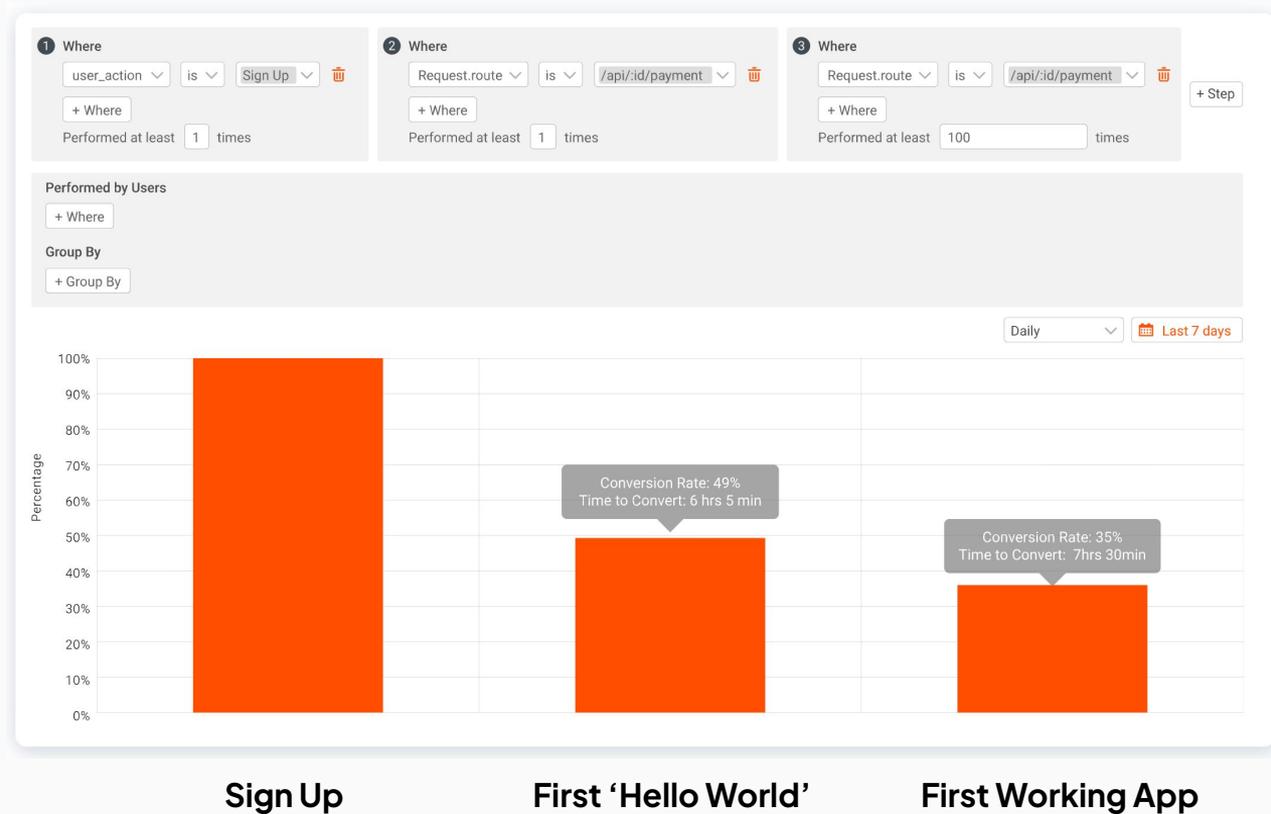


Engagement



Retention

Adoption : Reduce Time-to-Hello-World



Engagement : Drive customer value through the API



AI API

Relevant
Outcome



SaaS

Usage of
integrations



Data API

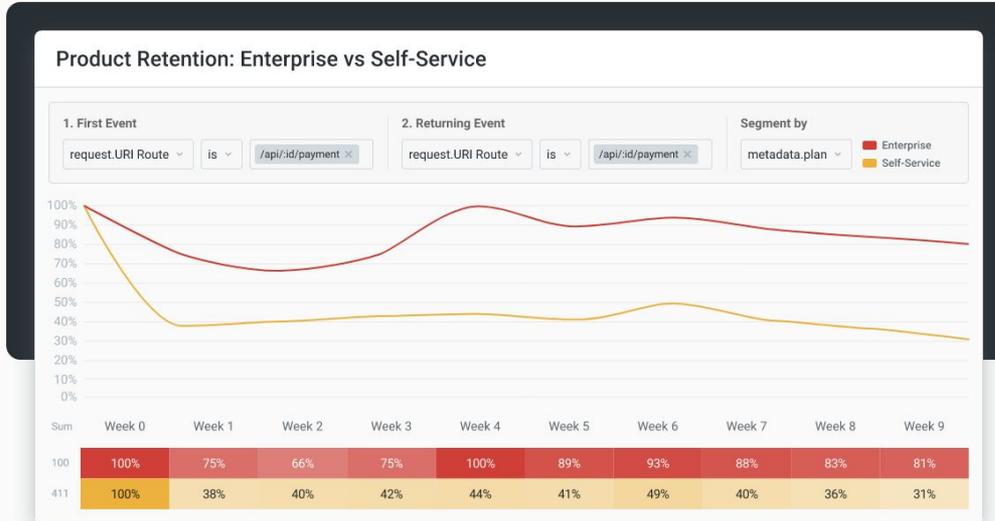
Match rate /
found in database



Logistics API

Successful orders/
Shipping SLA

Retention : Grow long-term customers

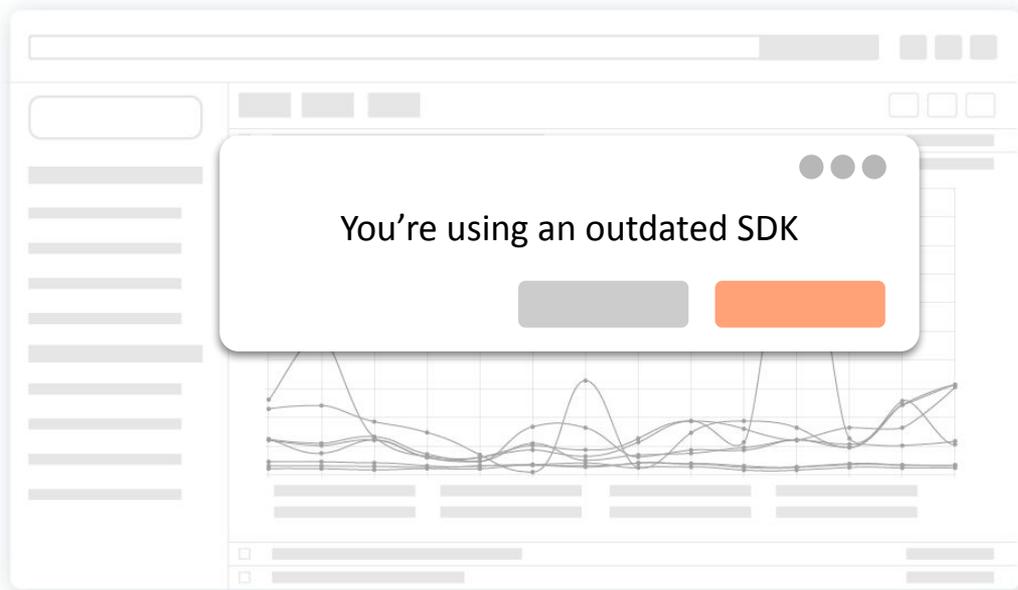


Good product retention curves flatten or look like a “smile”

End of life

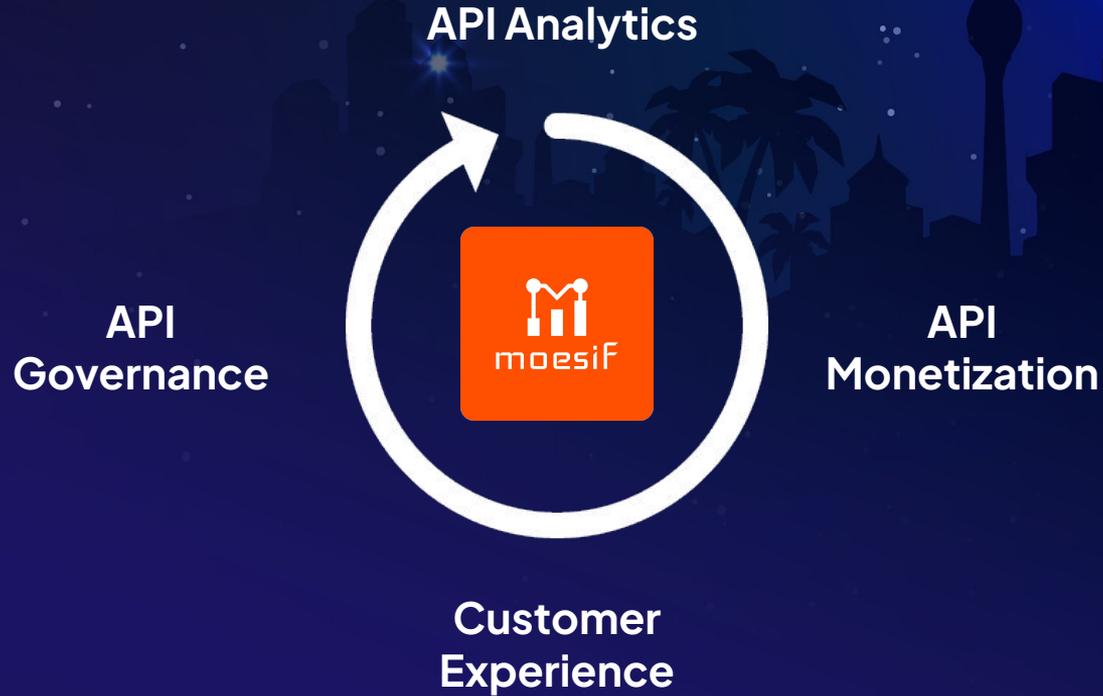


Don't break their stuff. Handle deprecation gracefully



Over-communicate changes via in-app notifications and emails

How to **monetize** APIs with Moesif



Native integrations for API Manager, Choreo/Bijira, and k8s gateway available



SIEMENS



nexhealth

 **sinch**



UpKeep

 **you.com**



 **PandaDoc**

Deloitte.



LSEG

FLOWCODE



moesif

API Analytics and Monetization

Derric Gilling

derric@moesif.com

<https://www.moesif.com/casestudies>