

# ACME Corp:

## API Platform Maturity Roadmap workshop

- **ACME Corp:** Team introduction
- **Introduction** to the Maturity Model assessment workshop
- **Discuss:** Why did you decide to build/adopt your API platform?
- **Discuss:** What does success look like? What are your KPIs?
- **Discuss:** What's your strategic API Platform maturity level?
- **Vote:** What is your biggest strategic area of focus for your API platform?
- **Vote:** What is your biggest API development challenge?
- **Priority dimension 1:** current & desired maturity level
- **Priority dimension 2:** current & desired maturity level
- **Discuss:** How do we get there in 6 months?

**Discuss: Why did you decide to build/adopt your API platform?**


**Discuss: What does success look like? What are your KPIs?**


## Discuss:

What's your strategic

## API Platform

maturity level?

Level 0 (None)	Level 1 (Provisional)	Level 2 (Operational)	Level 3 (Scalable)	Level 4 (Optimising)
	APIs are recognised for their value, with some processes in place for development and sharing, but practices still need to be standardized.	Standardised API development and usage processes are established, but automation and integration into broader IT and business processes are in development.	APIs are systematically developed, managed, and monitored, with efforts toward automation and enhancing development velocity.	Implementing advanced automation, governance, and enhancements to maximise API development velocity and effectiveness.

## Vote:

Where's your biggest

## strategic API Platform

opportunity?

Investment	People & team	Tooling & interfaces	Process & operations	Measurement & reporting
How are resources/ funds being allocated in the development of the API platform and platform team	Who maintains/ is responsible for the API platform, and who has access?	Where is your platform deployed? How are APIs within the platform surfaced to consumers?	How are API platforms and their capabilities planned, prioritized, developed and maintained?	How do you measure and report successes and learnings?

## Vote:

What's your biggest

## API development

challenge?

API Governance	API Security and access control	API Consumer Experience	API Developer Experience	API Observability
Consistent, high quality API design and development standards across your organisation. Drive API platform adoption and ensure API discoverability.	By definition APIs put a layer between consumers and underlying applications. It's critical to ensure appropriate access controls and usage restrictions are in place to prevent bad actors or mal-use.	APIs are treated as 'first-class citizens' within an organisation and surfaced as a key digital products. Drive API adoption and repeated usage over time.	Create useful, performant APIs and deploy to your API Gateway. Design valuable APIs and expose with ease.	Viewing detailed API analytics for enhanced understanding of what's going on with your APIs under the hood. Troubleshoot with ease and realise the true value of your APIs.

# Tyk's API Platform maturity model

For Platform managers and decision makers

Maturity pillar	Maturity questions	Level 1 (Provisional)	Level 2 (Operational)	Level 3 (Scalable)	Level 4 (Optimising)
<b>People &amp; team</b>	Who maintains/ is responsible for the API platform, and who has access?	Voluntary/ individuals dotted across the organisation.	Small central tech, architecture or ops team.	Dedicated Platform team overseeing distributed development teams.	Dedicated Platform product managers and API specialists continuously iterating on priority API platform improvements.
<b>Process &amp; operations</b>	How are API platforms and their capabilities planned, prioritized, developed and maintained?	On individual requests / ad-hoc sharing basis.	Manual submission to the central resource.	Central team enablement & automated service deployment and operations.	Further bespoke workflows and distributed developer experience tooling to drive further efficiencies.
<b>Investment</b>	How are resources/ funds being allocated in the development of the API platform and platform team?	Temporary project basis / ad-hoc.	Dedicated resources/ project based.	'API Platform as a product' allows self-service for the majority of your organisation's cross application integration needs.	Platform extended to enable innovation of latest API standards or widened to incorporate bespoke edge cases.
<b>Tooling &amp; interfaces</b>	Where is your platform deployed? How are APIs within the platform surfaced to consumers?	Shareable API documentation (if at all).	Thinnest viable platform e.g. API wiki.	Mature API Catalogs as part of an IDP / External Portal.	APIs deeply integrated into products and accessible on granular user permissions.
<b>Measurement &amp; reporting</b>	How do you measure and report successes and learnings?	Basic metrics with ad-hoc reporting.	Dedicated KPIs collected and reported frequently.	Focused KPIs reported along with insights and analysis, available on a central portal.	Self-service KPI reporting dashboards targeted at technical and non-technical audiences.



# Tyk's API Platform maturity model

For Platform engineers and developers

Maturity pillar	What is it and why is it important?	Maturity questions	Level 1 (Provisional)	Level 2 (Operational)	Level 3 (Scalable)	Level 4 (Optimising)
<b>API Governance</b>	Consistent, high quality API design and development standards across your organisation. Drive API platform adoption and ensure no more shadow APIs.	How do you manage and apply API governance policies and standard practices across your organisation?	Developers run API development tasks on an ad-hoc basis.	Central API platform team with API ops still owned by distributed API dev teams, basic API ownership enforced.	Templated API development and operations golden paths standardised across the organisation, personalised 'API team' views.	Continued investment to support latest API standards and drive further innovation and efficiencies. End to end visibility of APIs. Development of Platform APIs to add further value.
<b>API Security and access control</b>	By definition APIs put a layer between consumers and underlying applications. It's critical to ensure appropriate access controls and usage restrictions are in place to prevent bad actors or mal-use.	What organisation wide API security standards do you have in place?	Basic per API rate limiting, token based authorization, allow / block lists applied by individuals.	Standardised API rate limiting and token based API authentication and authorisation, allow / block lists advised by central platform team.	Global API security enforced by central team inc. rate limits, central trust using claims to further token based API authorisation and authentication, .	Zero trust API security, pen testing, monitoring and audits. Robust compliance to sector specific or global regulations regularly achieved.
<b>API Consumer Experience (API discoverability)</b>	APIs are treated as 'first-class citizens' within an organisation and surfaced as a key digital products. Drive API adoption and repeated usage over time.	How are APIs documented, discovered, accessed, and supported?	Internal word docs shared, API access granted via manual requests e.g. email. Individual adoption.	Internal API catalogs emerging (e.g. API documentation wikis), access granted through formal API requests to central owners. API adoption by multiple teams.	Self-service API catalogs with documentation standards adhered to and access granted on an a permissions basis. API SLAs in place.	Context specific API interfaces, with granular permissions and automated role or subscription based access control baked in. SLAs continually met. API adoption by non technical teams.
<b>API Developer Experience (API Operations)</b>	Create useful, performant APIs and deploy to your API Gateway. Design valuable APIs and expose with ease.	What does your end to end API development and deployment workflow / SDLC look like? What best best practices are followed?	Click Ops or manual process to move APIs between environments. Adhoc API design and development workflows.	GitHub actions, config scripts and basic GitOps or CI/CD deployment pipeline best practice for APIs achieved.	Native technologies in use (CI/CD ops tools), automated API creation, testing and deployment achieved.	Consistent API development tooling in use at all stages of the API SDLC e.g. API design, linting, SDKs or IDEs. Seamless collaboration within teams.
<b>API Observability</b>	Viewing detailed API analytics across for enhanced understanding of what's going on with your APIs under the hood. Troubleshoot with ease and realise the true value of your APIs.	What level of insights do you have across your API platform stack?	Basic API telemetry used to analyse API stability, error rates and traffic levels.	Comprehensive API analytics summary dashboards, accessible for API platform management and API development.	Granular API tracing, API telemetry exported and viewed within central observability platform.	Mature API observability dashboards for variety of BAU (alerting and monitoring) and strategic platform KPI / value reporting.



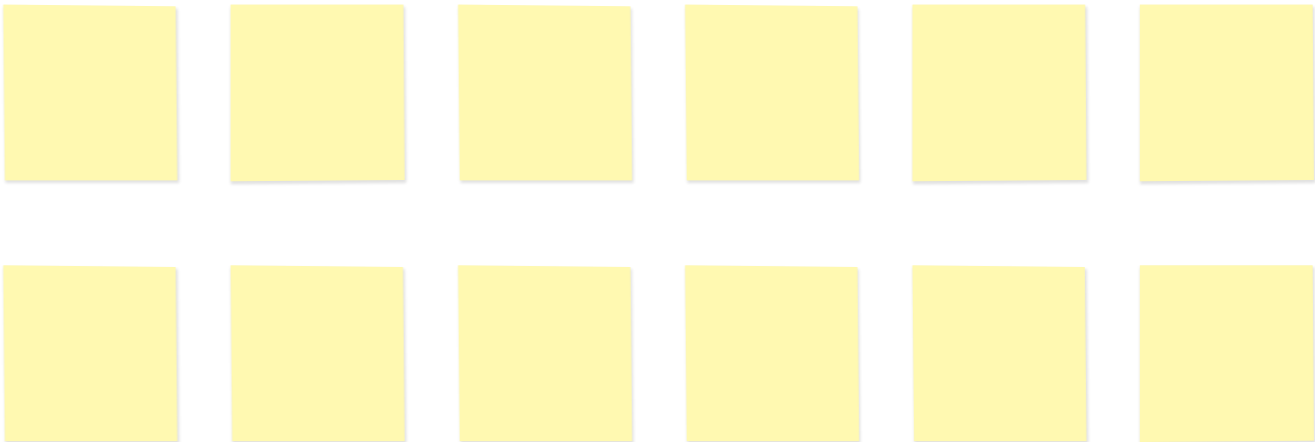
Priority dimensions - steps to accelerate maturity

Pillar 1 -  
\_\_\_\_\_

Current level => \_\_\_\_

Desired level => \_\_\_\_

How do we get there?

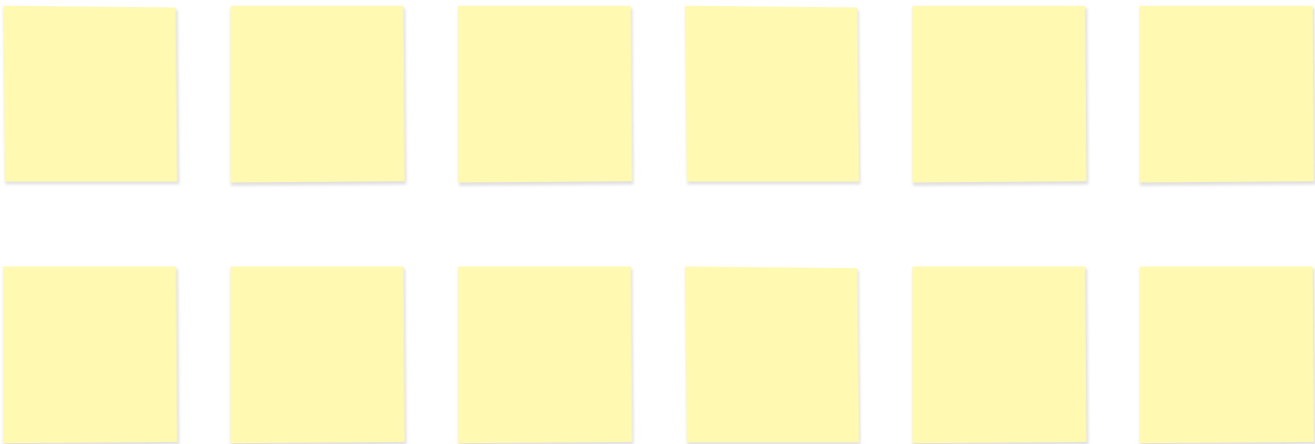


Pillar 2 -  
\_\_\_\_\_

Current level => \_\_\_\_

Desired level => \_\_\_\_

How do we get there?



Pillar 3 -  
\_\_\_\_\_

Current level => \_\_\_\_

Desired level => \_\_\_\_

How do we get there?

