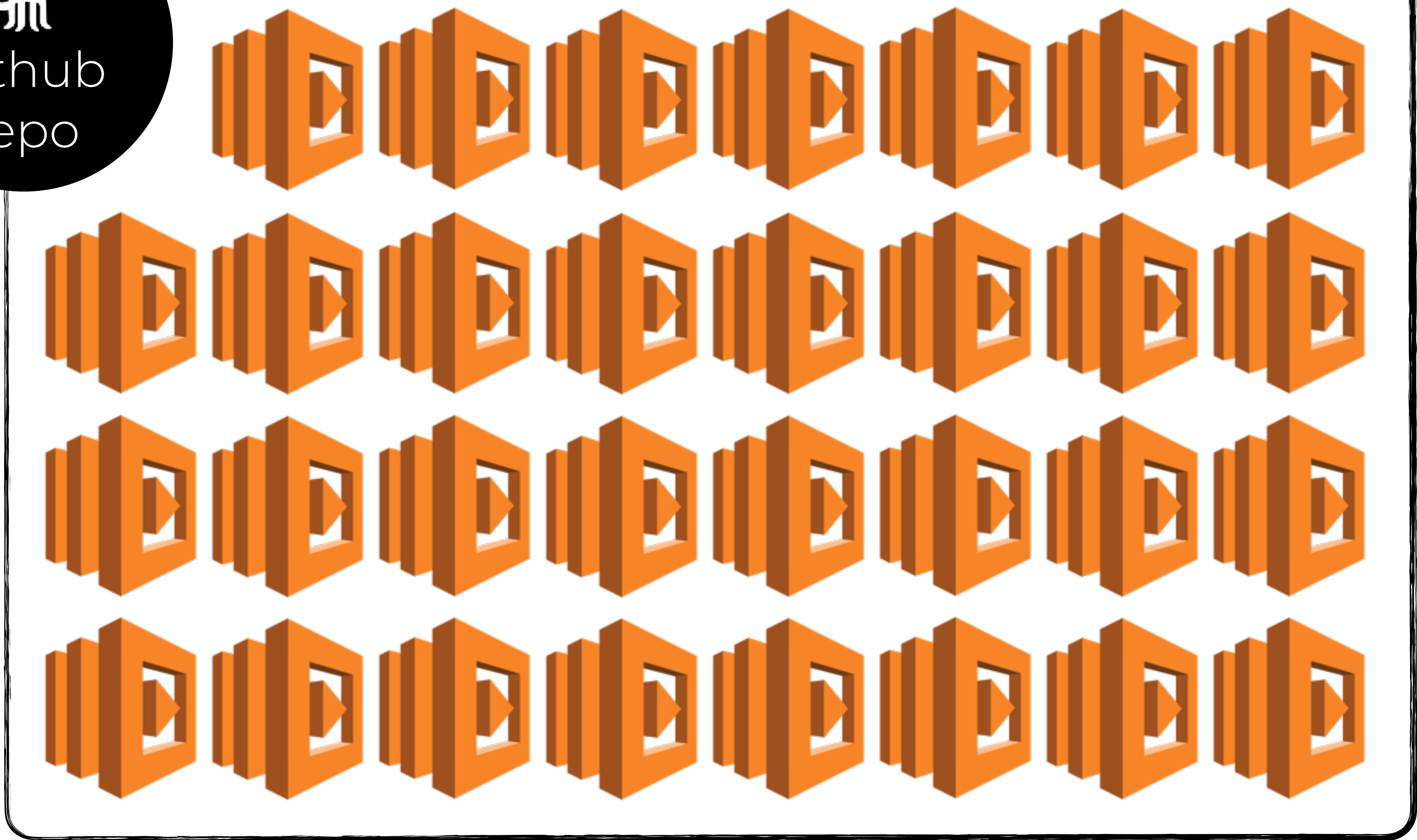






monorepo



Monorepo



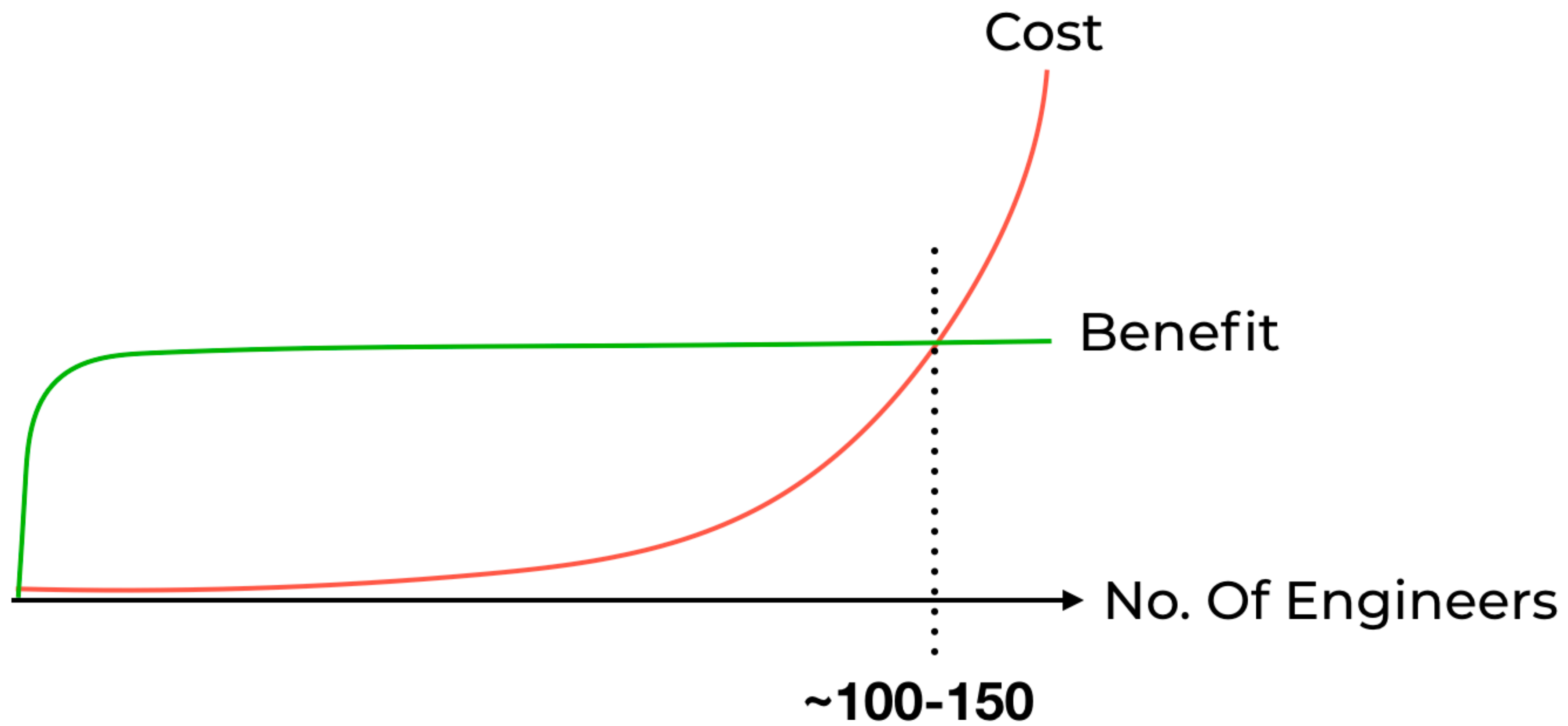




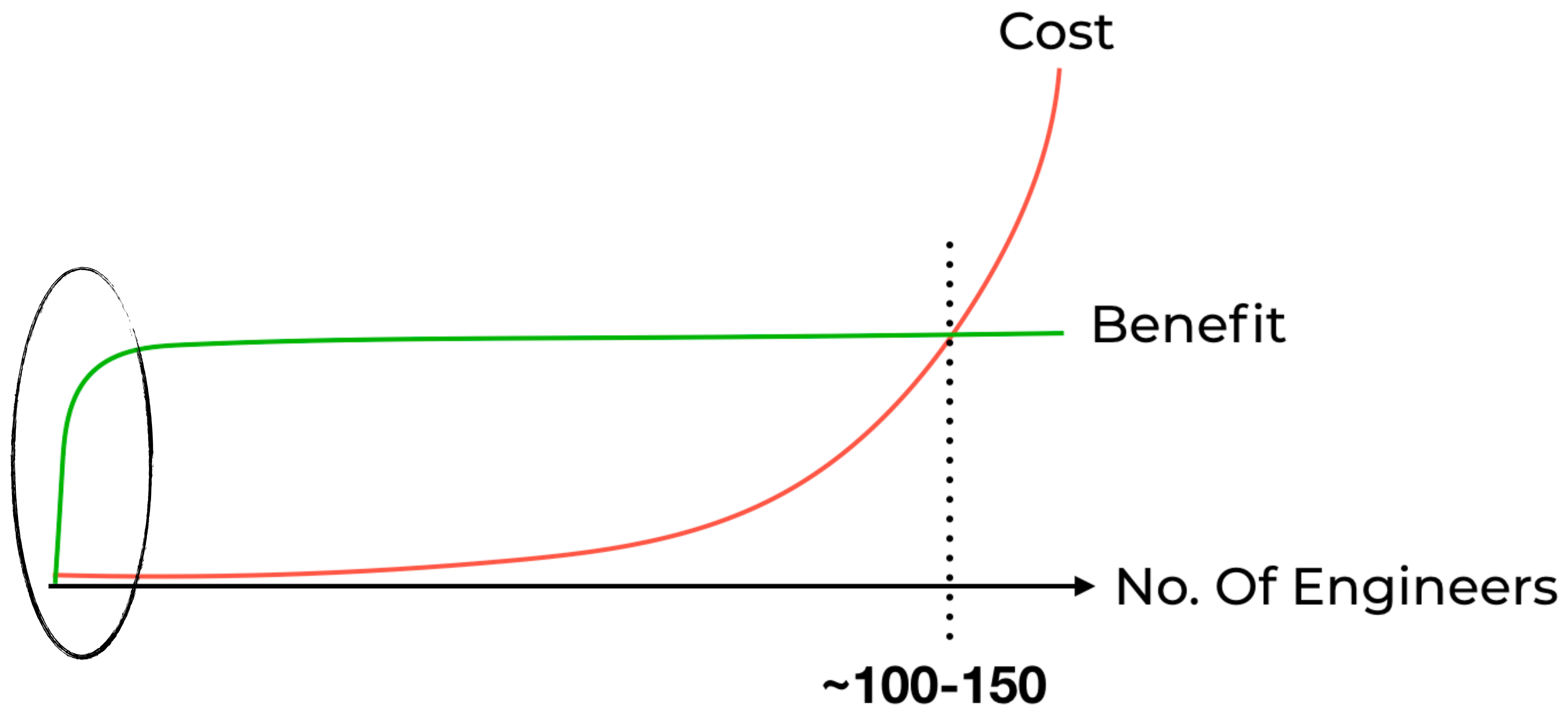
Name	^	Date Modified	Size	Kind
▶ chat-service		Today at 18:32	--	Folder
▶ explore-feed-service		Today at 18:41	--	Folder
▶ relationships-service		Today at 18:32	--	Folder
▶ search-service		Today at 18:33	--	Folder
▶ timeline-service		Today at 18:32	--	Folder
▶ user-details-service		Today at 18:32	--	Folder
▶ user-preferences-service		Today at 18:32	--	Folder

share code through symlinks + webpack

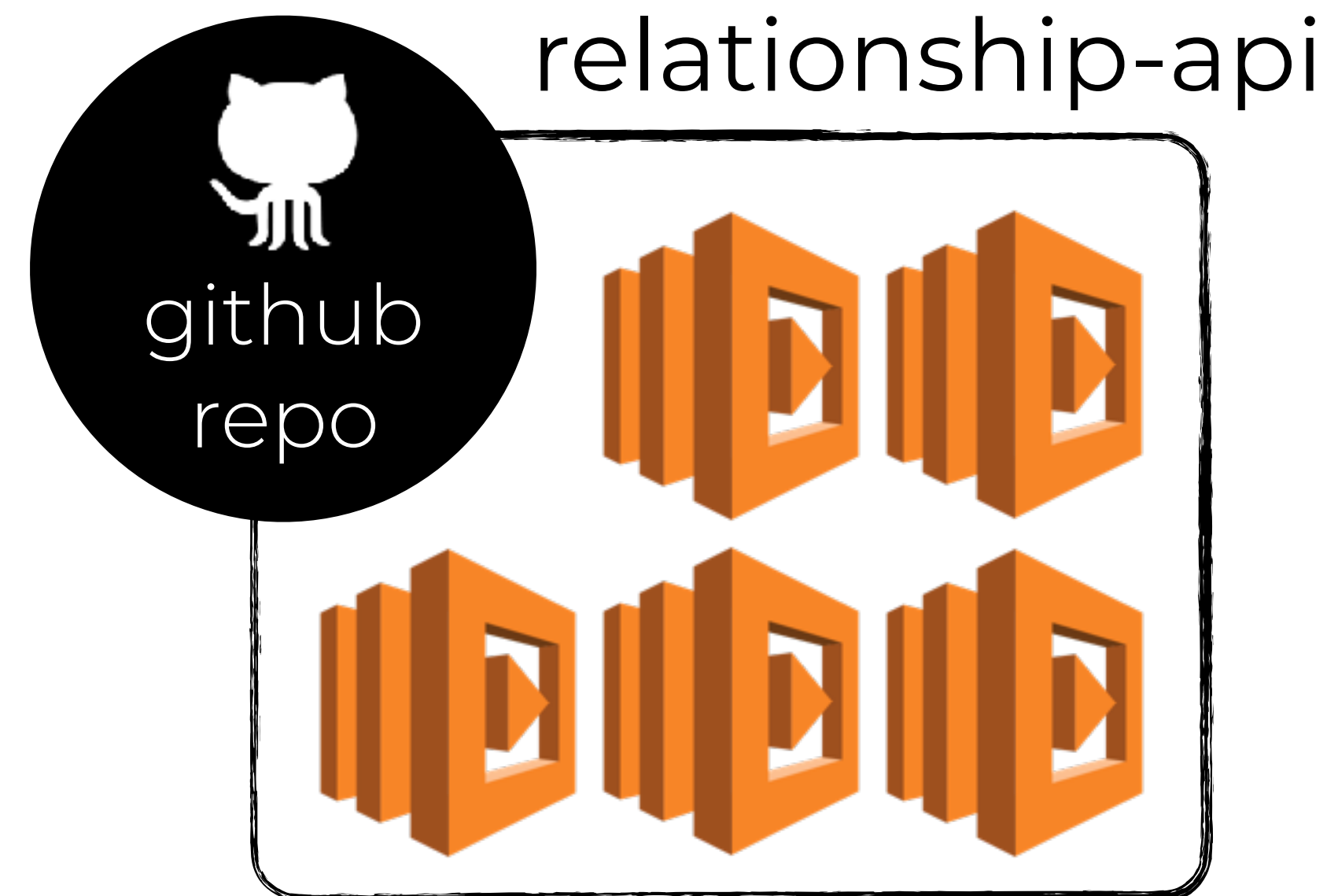
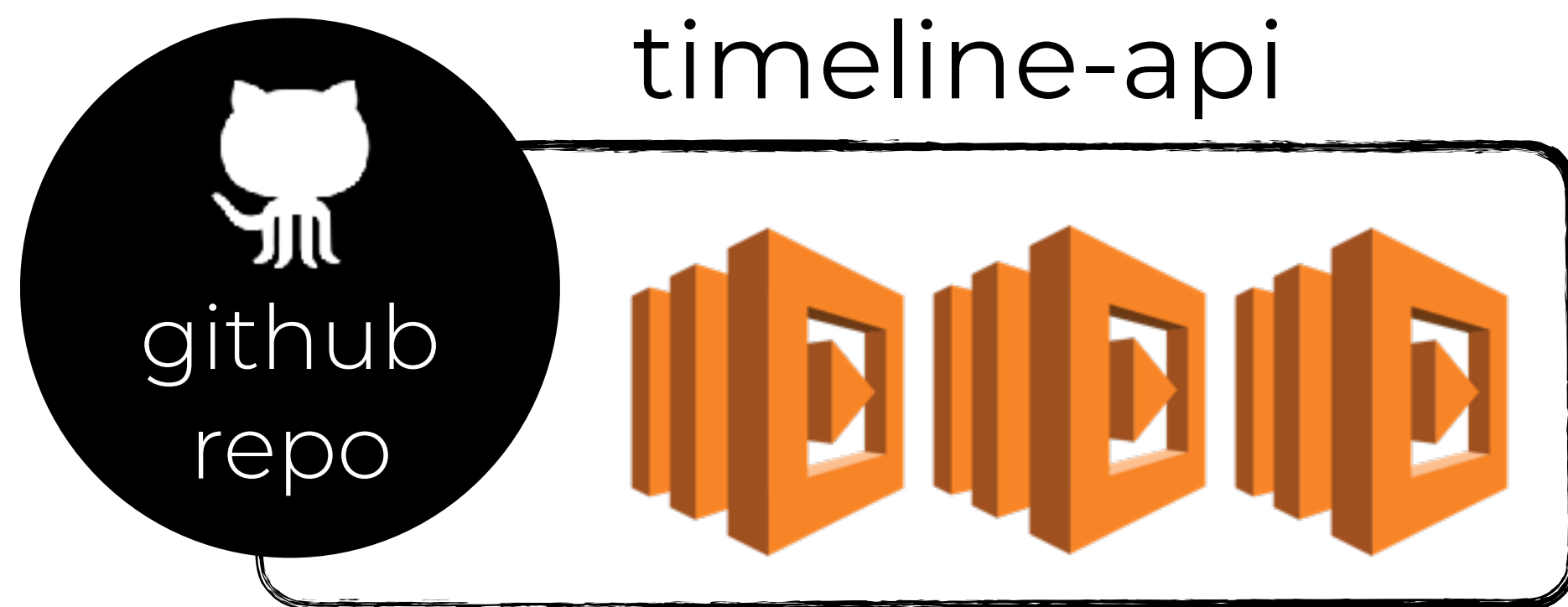
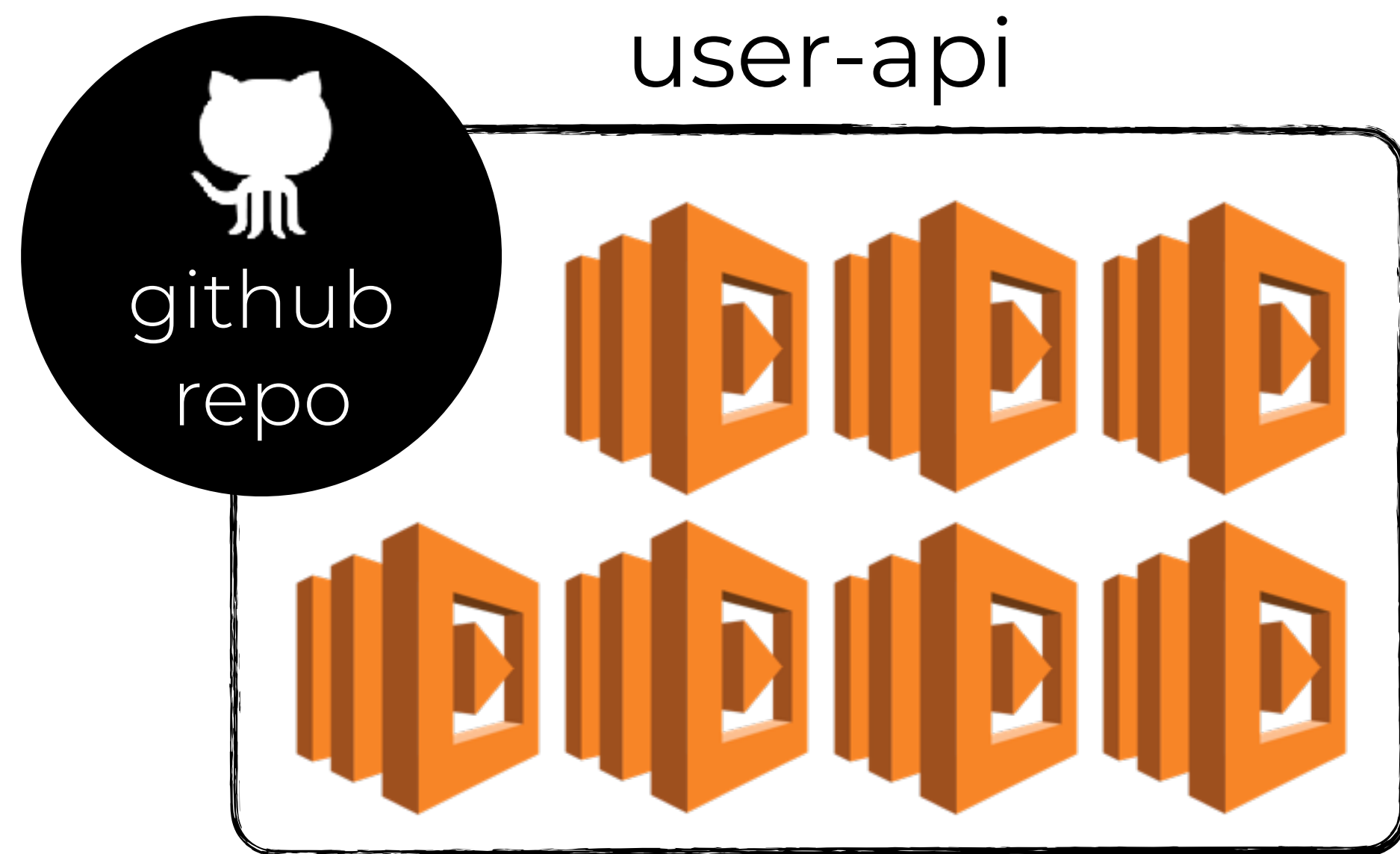
one CI/CD that deploys them all (in parallel)



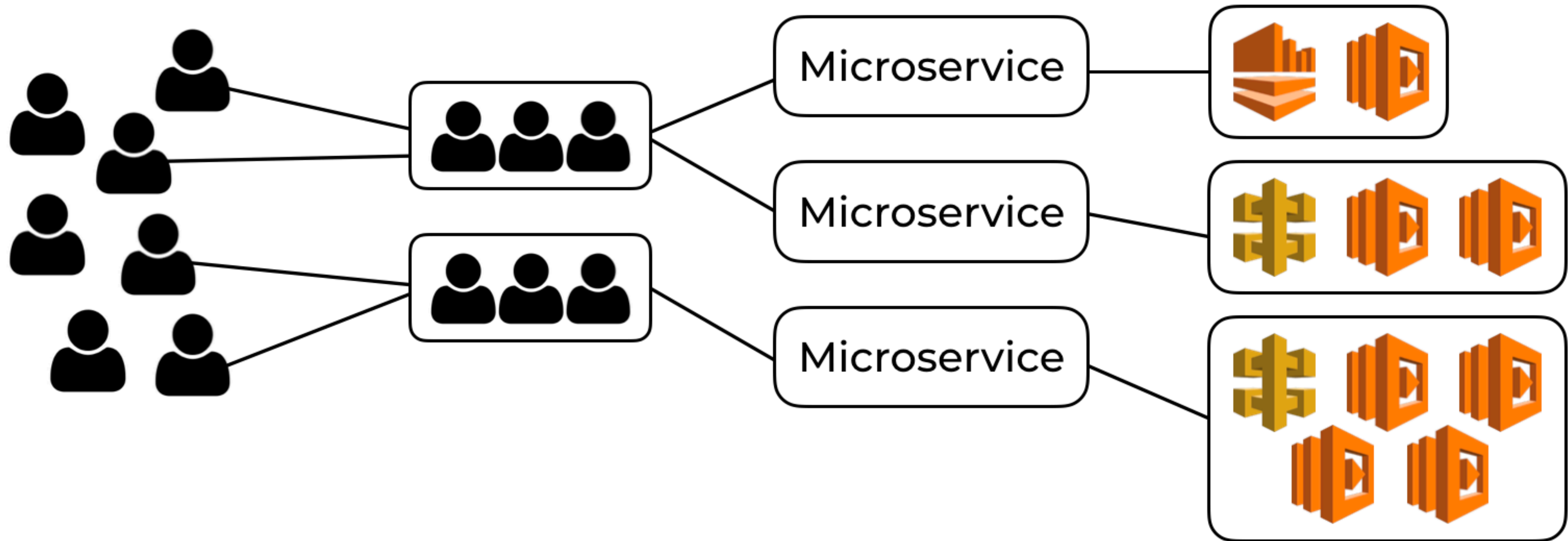
good for small teams/startups



one repo per service



Engineers $\xrightarrow{\text{work in}}$ **Teams** $\xrightarrow{\text{own}}$ **Services** $\xrightarrow{\text{consist of}}$ **Functions**



one CI/CD pipeline per service

shared infrastructure (VPCs, etc.) in separate repo

ref shared infra via CFN output, SSM param, etc.

share code through shared libs (NPM, maven, etc.)

shared code vs shared service

	shared library	service
visibility	explicit	often none
deployment	consumer	service owner
versioning	multiple active	singular/multiple active
backward compat	semantic versioning	don't break it
isolation	internals accessible	no access to internals
failure	loud & clear	it's complicated

<https://theburningmonk.com/2018/02/aws-lambda-how-best-to-manage-shared-code-and-shared-infrastructure/>