
Your technology architecture and engineering organization should coevolve as your startup grows

Abhishek Tiwari 

Citation: *A. Tiwari*, "Your technology architecture and engineering organization should coevolve as your startup grows", Abhishek Tiwari, 2020. [doi:10.59350/4hxr-v-p0v22](https://doi.org/10.59350/4hxr-v-p0v22)

Published on: February 27, 2020

The evolution of your technology architecture should depend on the size, culture, and skill set of your engineering organization. In fact, they should coevolve. There are no hard-and-fast rules to figure out interdependency between technology architecture and engineering organization but below is what I think can really work well for a typical product startup.

1-10 Engineers

Focus: Rapid product delivery to find mythical product market fit

Architecture: a well designed **12-factor** MVC app running securely on Heroku or Google App Engine, or any other PaaS platform. I suggest using a proven MVC framework like Rails, Django, Express, Sails, etc. At this stage you can use CI/CD tooling provide by source code hosting platform (Github, Gitlab) or a SaaS solution like CircleCI. Basic monitoring and logging offered by PaaS will be sufficient for now.

Skills: Induct Full-stack engineers. A few senior (2-3) engineers but mostly mid-level and junior engineers (7-8) eager to learn, improve, and master their craft.

11-20 Engineers

Focus: Still fast-paced product delivery but now have a product market fit and planning ahead for scaling team, technology, and product.

Architecture: Build APIs and decouple backend from the frontend and consume them using SPA and native apps. Serve you SPA via CDN like Netlify or Cloudfront in conjunction with S3. Use API gateway to have a singly entry point for all your APIs. Secure you APIs via standard based authentication (JWT tokens). Get test automation improved (10-20%). Add enhanced instrumentation to monitor your stack and give birds-eye view (an APM like NewRelic).

Caveats !?: If you are a mobile-first product company APIs and decoupled architecture is probably your first architecture.

Skills++: Induct specialised Frontend, Backend & QA engineers, Technical Leadership

21-30 Engineers

Focus: Building quality in products and continue to innovate while proactively planning team structure, incentives, and culture.

Architecture: Small microservices targeting new features. Start moving existing apps into a DIY cloud environment - VMs and managed DB services. Adopt DevSecOps best-practices. Introduce centralised

logging and comprehensive monitoring. Increase automated test cover (30-40%) and start implementing property tests like performance and load testing. Now is time to tighten your authentication and authorization regime using an identity and access management solution like Auth0 or Axioms.

Skills++: Induct DevOps, People Leadership

31-50 Engineers

Focus: Maintain growth trajectory of product in the market and start growing multi-layer people and technical leadership.

Architecture: Start breaking your MVC monolith into microservices. Test coverage (50-70%). Containers for local development and probably in UAT. Consider introducing a strong CI/CD toolchain like Jenkins. Explore serverless functions to create

Skills++: Induct Technical Architects, Developer Experience (DevX)

50-100 Engineers

Focus: Finding new ways to add more value quickly for your customers by exploiting data. Velocity is a feature now so reliability is super important.

Architecture: Microservices running on Kubernetes backed by polyglot persistence (NoSQL/SQL DBs). Simultaneously running multiple version of services. Now building data-driven apps and data services and where necessary incorporating machine learning web service offered by public cloud providers. Introduce site-reliability engineering best-practices (SLI/SLOs).

Skills++: Induct Site-reliability engineers, Data engineers