A **sidecar container** is a helper container that runs alongside the main application container **in the same pod**, sharing the same network and storage. It’s used to add supporting features like logging, proxying, or monitoring **without changing the main app’s code**. A common example is using **Envoy** as a sidecar proxy in a service mesh (like Istio), where Envoy handles **all incoming and outgoing traffic** for the main app container, adding features like load balancing, retries, and security. This pattern helps keep the application code simple while offloading complex operations to the sidecar.

A **monolithic architecture** is a single, large application where all components (like UI, business logic, and database access) are tightly integrated and run as one unit, making it simpler to develop initially but harder to scale and update. In contrast, a **microservices architecture** breaks the application into smaller, independent services that communicate over a network, allowing teams to develop, deploy, and scale each service separately. While monoliths are easier to start with, microservices offer greater **flexibility, scalability, and fault isolation** in large, complex systems.