1) Modularity Theory — core idea & vocabulary

- **What it is:** A lens for how a system's **architecture** shapes its components, their **interfaces**, and how teams must coordinate to make the whole thing work.
- **Interfaces:** The junction where two subsystems "fit" together.
- **Interdependent architecture:** Components can't be designed independently because **unpredictable interdependencies** exist across the interface. Teams must co-design/iterate together for the system to function.
- **Modular architecture:** Interfaces are **fully specified, verifiable, and predictable**. Components can be built by different teams or firms with **minimal coordination**, as long as they meet the interface spec.
- **Practical effect:** Interdependence raises coordination costs but can enable tight performance tuning; modularity lowers coordination costs and enables parallel development, sourcing flexibility, and scale.

2) Adoption implications — which type and how fast adoption happens

- **Key insight:** Choosing **modular vs. interdependent** doesn't decide *whether* a product will win, but it **predicts the speed of adoption**.
- **Interdependent early:** Useful when the problem or performance targets are novel/uncertain—tight integration helps achieve breakthrough fit or quality. **Trade-off:** Slower adoption due to heavy coordination and fewer interchangeable suppliers.
- **Modular to scale:** Once interfaces are well understood, a modular design lets multiple teams/vendors contribute, **accelerating adoption** through faster iteration, lower costs, and easier substitution.
- **Rule of thumb:** Start interdependent when specifications are not yet knowable; **modularize as interfaces stabilize** to unlock variety, competition, and rapid diffusion.