

# Services: Great & Small

- Service architecture is not necessarily true, as it is simply one way of regrouping functions based on their behaviors
  - May not have the boundaries separating high-level policy from low-level detail with the Dependency Rule
- Service benefits?
  - The decoupling fallacy - shared resources like data
  - The fallacy of independent development and deployment - shared data/behaviors
- Cross-cutting concerns
  - Architectural boundaries don't fall between services, instead run through the services
  - Components within the services define the architectural boundaries of the system

# The Test Boundary

## Tests as System Components

- Tests as the outermost circle in the architecture
  - independently deployable
  - the most isolated system component (support development, not operation)
- Design for testability
  - Fragile tests problem - changes to common system components can cause hundreds or even thousands of tests to break
  - *Don't depend on volatile things*
- The testing API
  - Decouple the tests from the application
  - Hide the structure of the application from the tests