

Presenters and Humble Objects

Data pattern across architectural boundaries

- The Humble Object Pattern
 - Split the behaviors into hard-to-test behaviors (e.g., views) and testable behaviors (e.g., presenters)
 - Humble object contains all hard-to-test behaviors
- Presenters and Views

Partial Boundaries

Too expensive to implement a full-fledged architecture

- Skip the last step
 - Compile and deploy all components just as a single monolith
 - No administration of multiple components is required
- One-dimensional boundaries
- Facades
- But the costs could be very high to add a fully implemented boundary later on