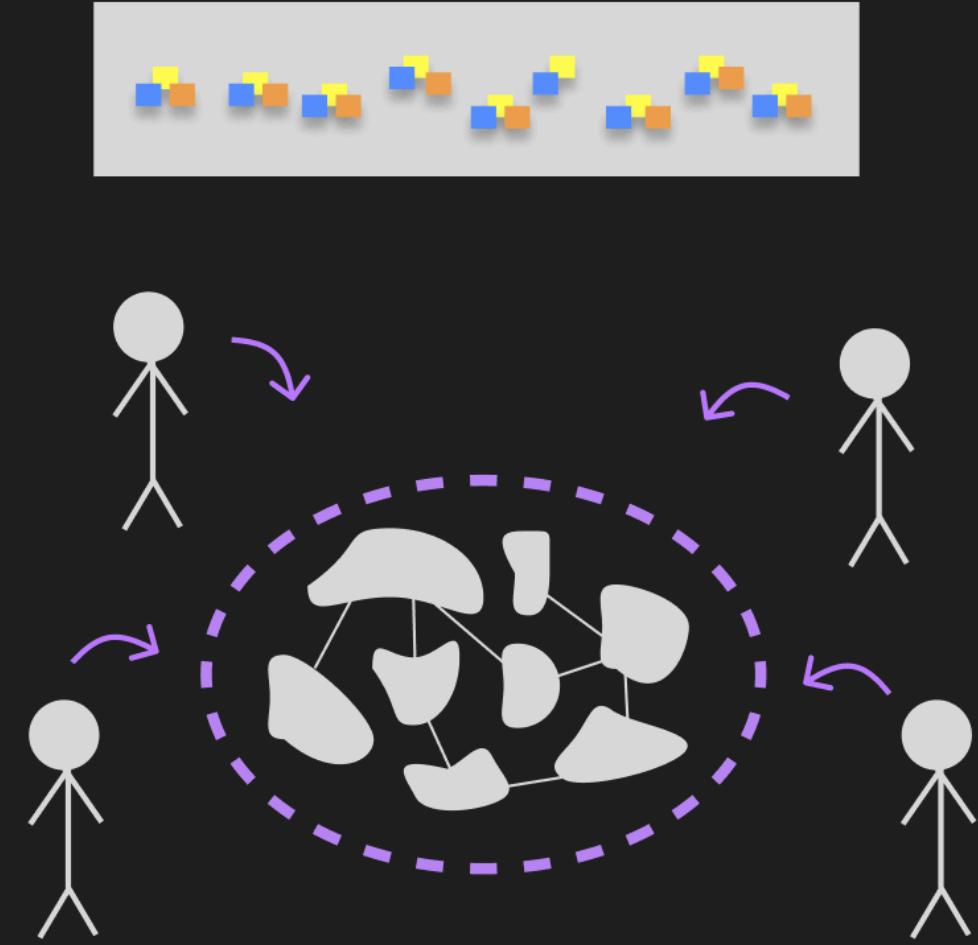




# Theoretical Essential: Domain Discovery



**Theoretical Essential**

# Domain Discovery

**Theoretical Essential**

# Domain Discovery

*The language you use matters.*

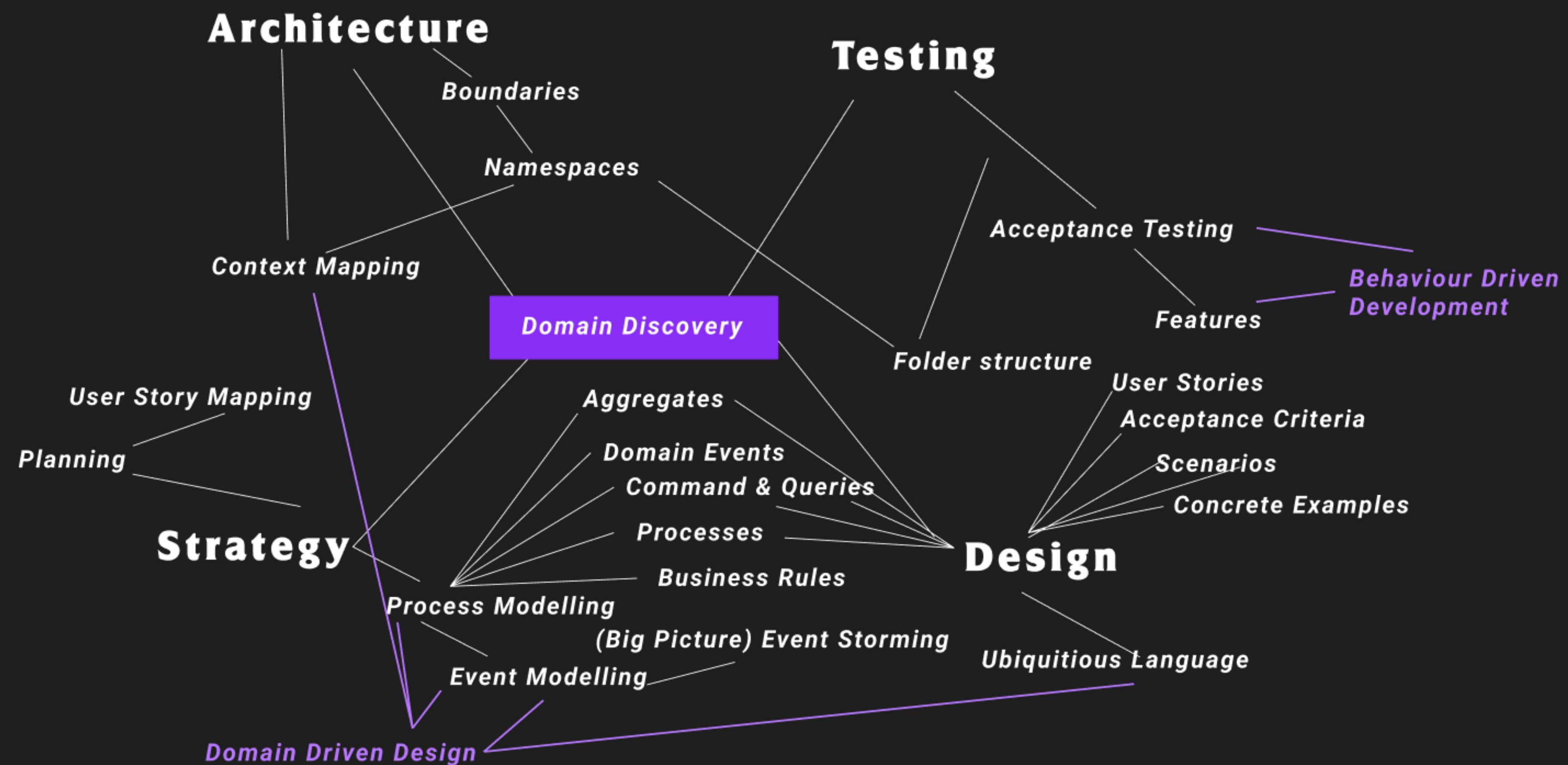
*As abstraction designers, it's your responsibility to ensure all developers are working against the same mental models  
- the same conceptual understanding of the domain.*

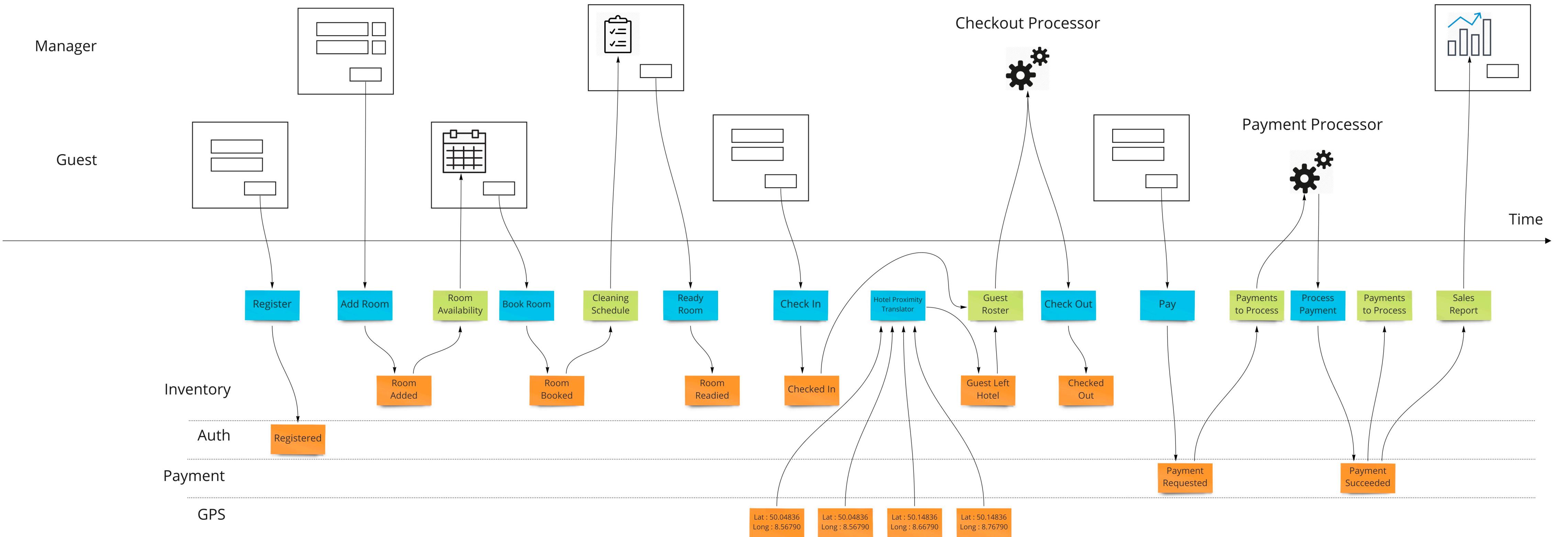
*Domain Discovery involves using a number of DDD & BDD techniques to build a shared understanding of the domain, the boundaries of the problem, how we could theoretically solve it.*

*The benefit is a rich, stable domain model through which everything else (your controllers, services and so on) rely upon.*

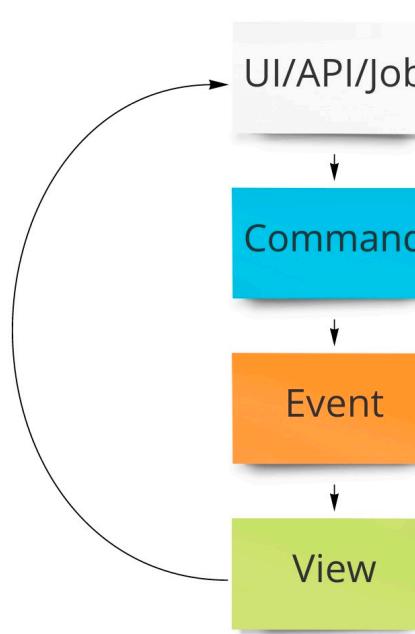
*This model is embedded in the core of your codebase.*

*A stable domain model produces way more maintainable code in the long term.*





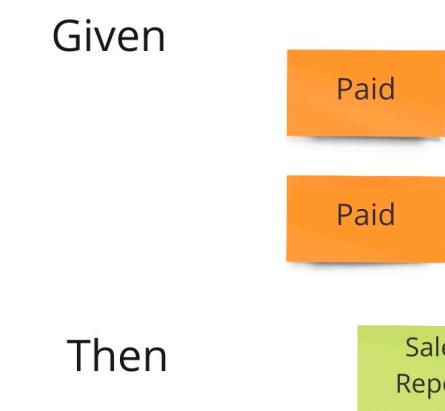
#### Legend



#### State Change



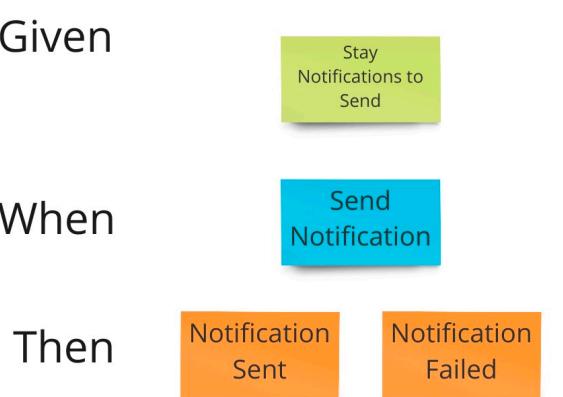
#### State View



#### External State Input

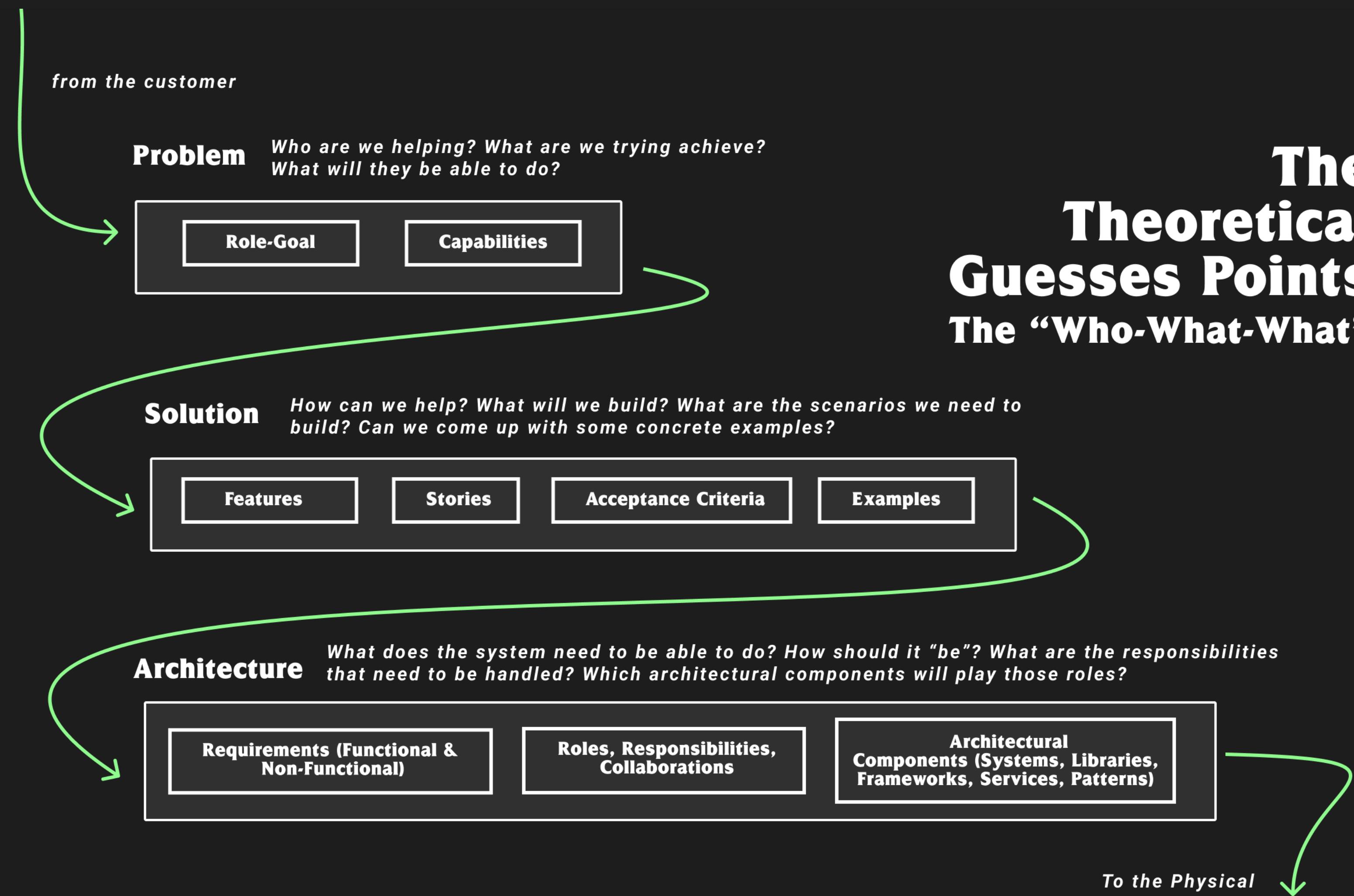


#### External State Output



# The Theoretical Guesses Points

## The “Who-What-What”



# Where we'll learn more