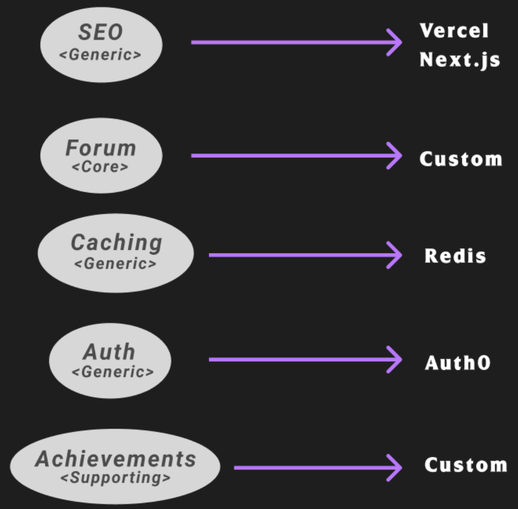
The
SOFTWARE
ESSENTIALIST

Theoretical Essential: Strategic Design

essentialist.dev



Theoretical Essential

Strategic Design

Strategic design is making smart decisions about large scale architectural components.

But there's more to architecture than just libraries, tools & frameworks

How do you know which components to use?
How do you know what to build yourself?
How do you integrate the component together?
What do you do when there isn't a component you need?

All developers need to know how to do 4 things:

1. *Stereotype components*
2. *Develop components*
3. *Integrate components*
4. *Prioritize components*

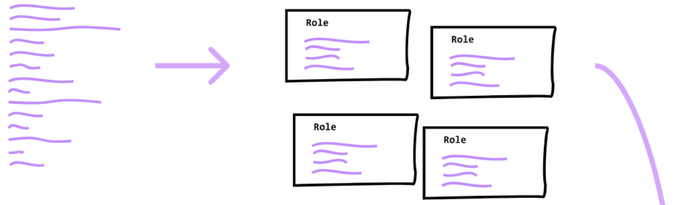
The Responsibility-Driven Design Process

Start

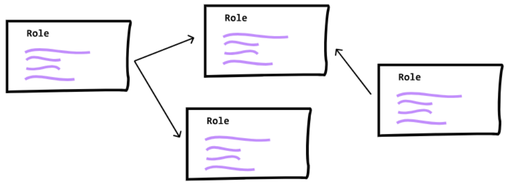
Functional Requirements | Non-Functional Requirements

Responsibilities
Extract the responsibilities from the requirements

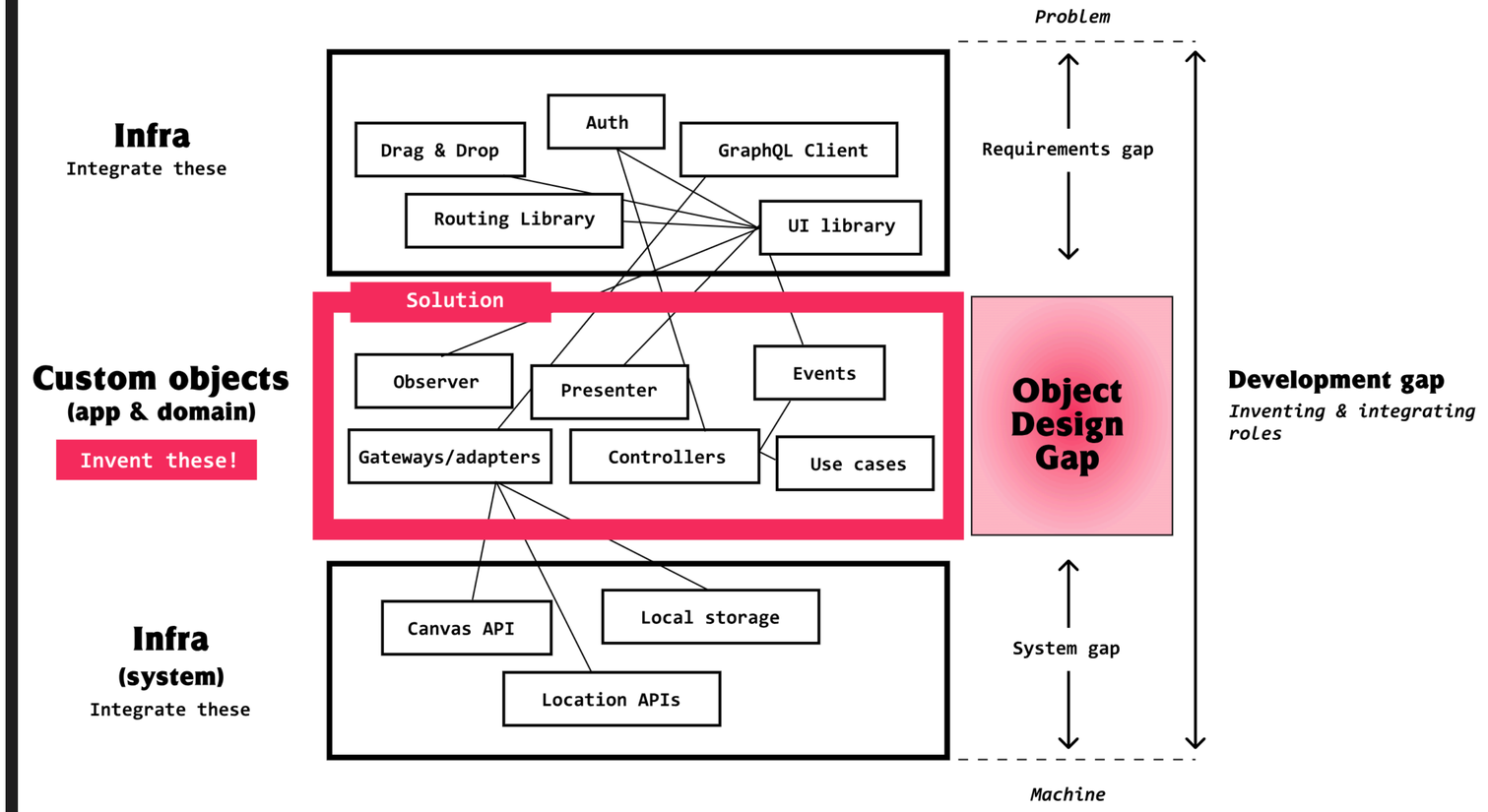
Roles
Assign responsibilities to roles



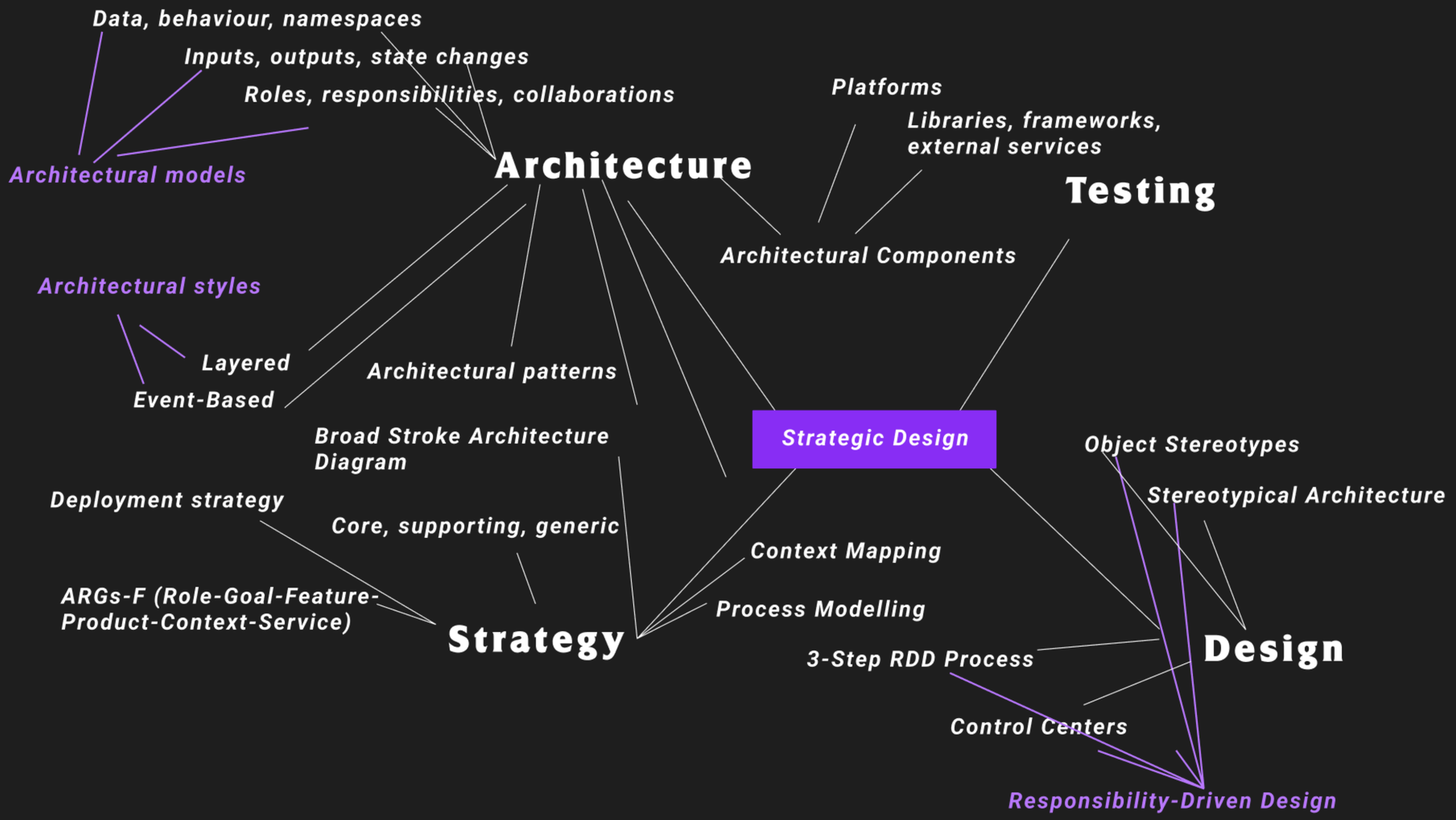
Collaborations
Design collaborations between roles



OOP (Object-oriented programming)

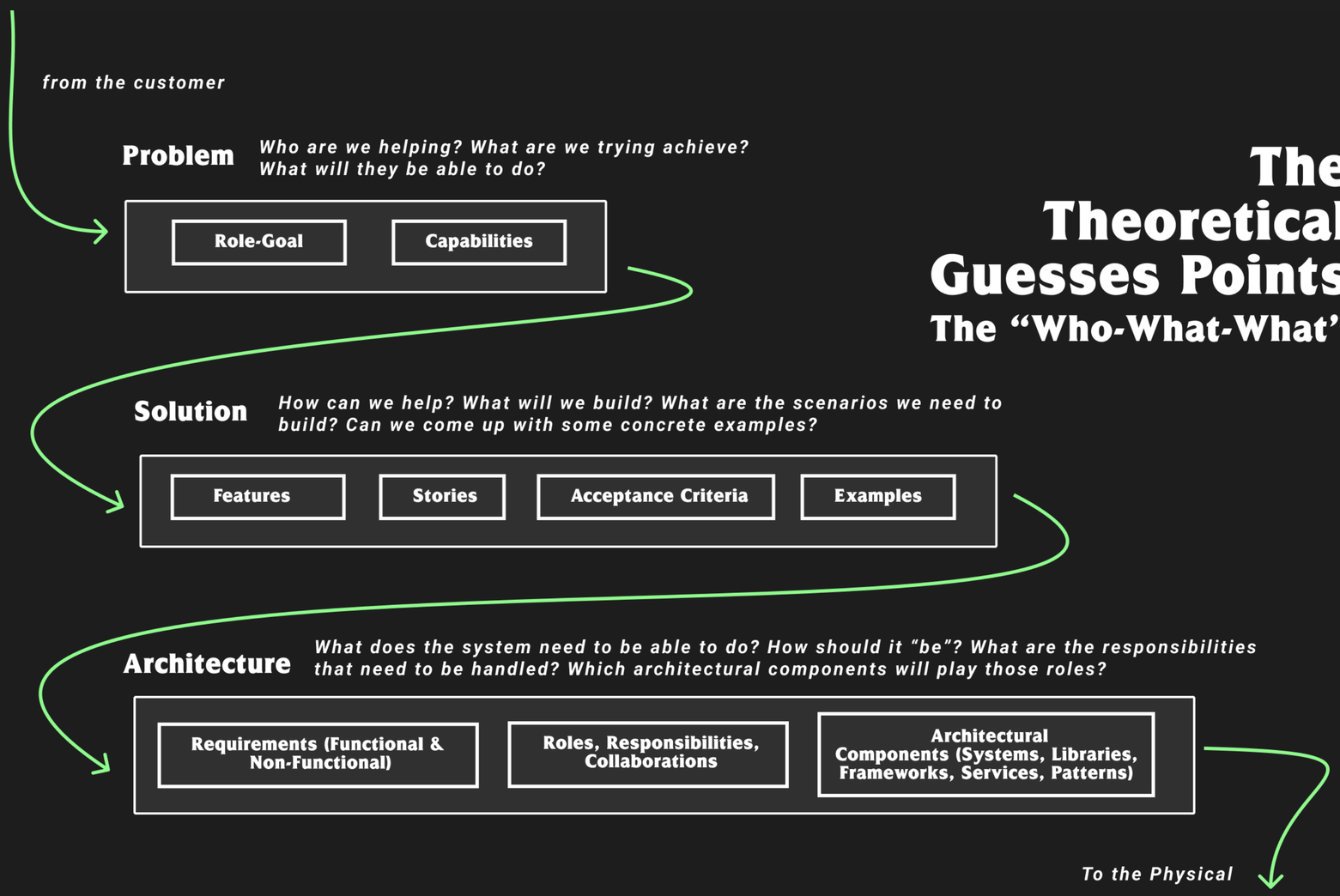


essentialist.dev



The Theoretical Guesses Points

The "Who-What-What"



Where we'll learn more