



# Boas Práticas - Codificação

**Workshop**





# DRY (Don't repeat yourself)

Se você tem um bloco de código em mais de dois lugares considere torná-lo um método separado, ou se você usar um valor codificado mais de uma vez, torne-os públicos. O benefício deste princípio de design orientado a objetos está na manutenção.

Boy scout rule. Leave the campground cleaner than you found it.

Sempre que for alterar/corrigir/adicionar código à sua aplicação, melhore um outro trecho, seja refatorando ou até mesmo reescrevendo.



Se você tem um bloco de código em mais de dois lugares considere torná-lo um método separado, ou se você usar um valor codificado mais de uma vez, torne-os públicos. O benefício deste princípio de design orientado a objetos está na manutenção.

"Keep it simple stupid. Simpler is always better. Reduce complexity as much as possible."



Always find root cause. Always look for the root cause of a problem.

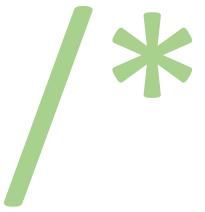
Procure sempre encontrar a raiz do problema que está resolvendo, respeite a regra do 80/20!

# Understandability tips

- Seja consistente. Se você faz uma coisa de um jeito, faça todas as coisas desse jeito!
- Dê nome significativos às suas variáveis.
- Prefira utilizar objetos no lugar de variáveis primitivas.
- Evite a dependência lógica.
- Não utilize condicionais negativos.



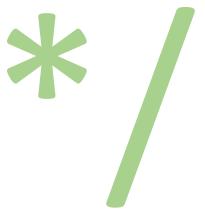
- Pequenas.
- Apenas uma funcionalidade.
- Utilize nomes descritivos.
- Tente utilizar o menor número de argumentos possíveis.
- Verifique se sua função não produz efeitos colaterais.
- Não utilize flags como argumentos.

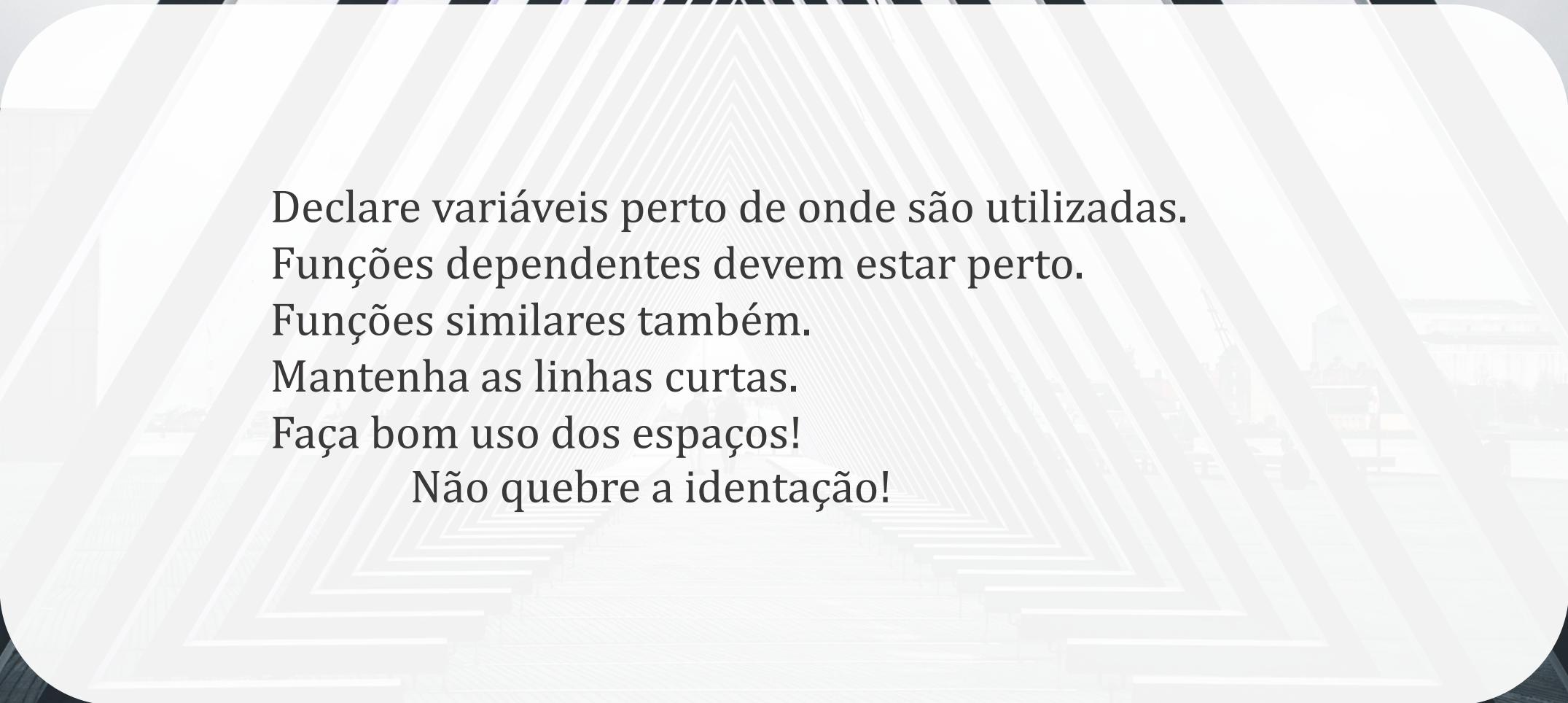


“Don’t comment bad code, rewrite it”

— Brian W. Kernighan and P. J. Plaugher

Sempre tente se explicar no código.  
Não seja redundante.  
Evite adicionar comentários sem sentido.  
Tente explicar sua intenção.  
Utilize os comentários como uma forma de  
explicitar o Código.  
Utilize como avisos!





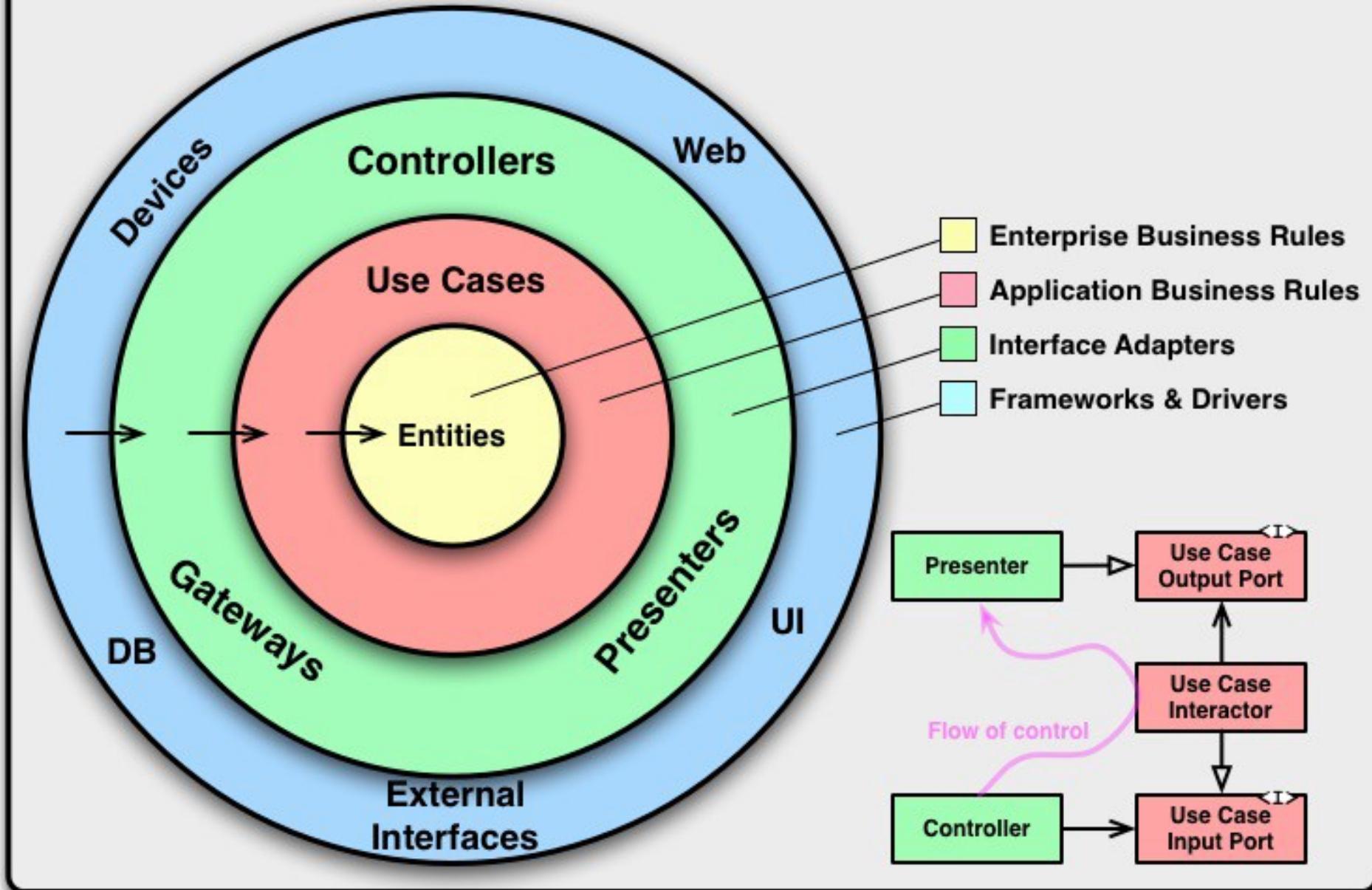
Declare variáveis perto de onde são utilizadas.  
Funções dependentes devem estar perto.  
Funções similares também.  
Mantenha as linhas curtas.  
Faça bom uso dos espaços!  
Não quebre a identação!



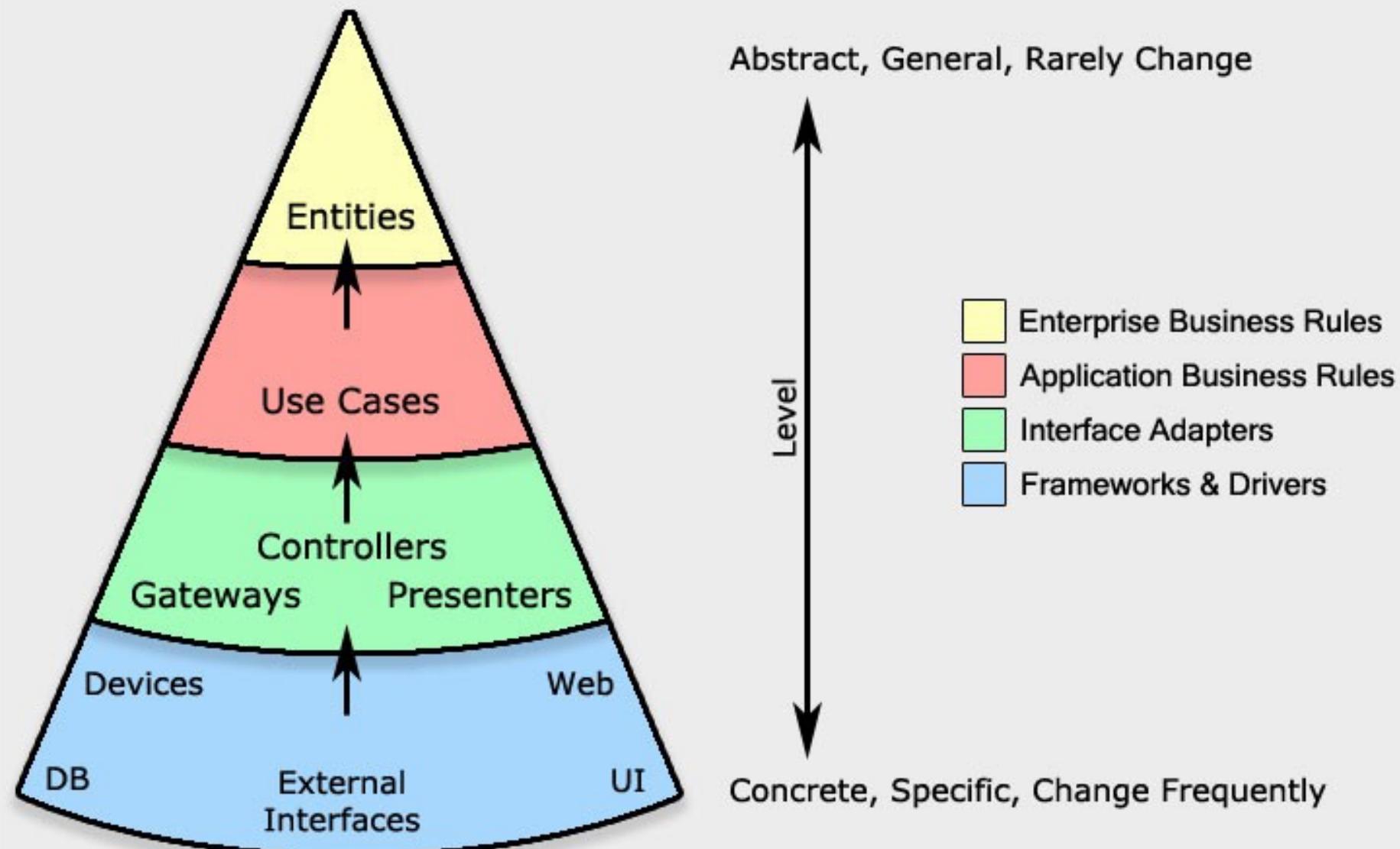
Um conceito por teste.  
Legível.  
Rápido.  
Independente.  
Reproduzível.



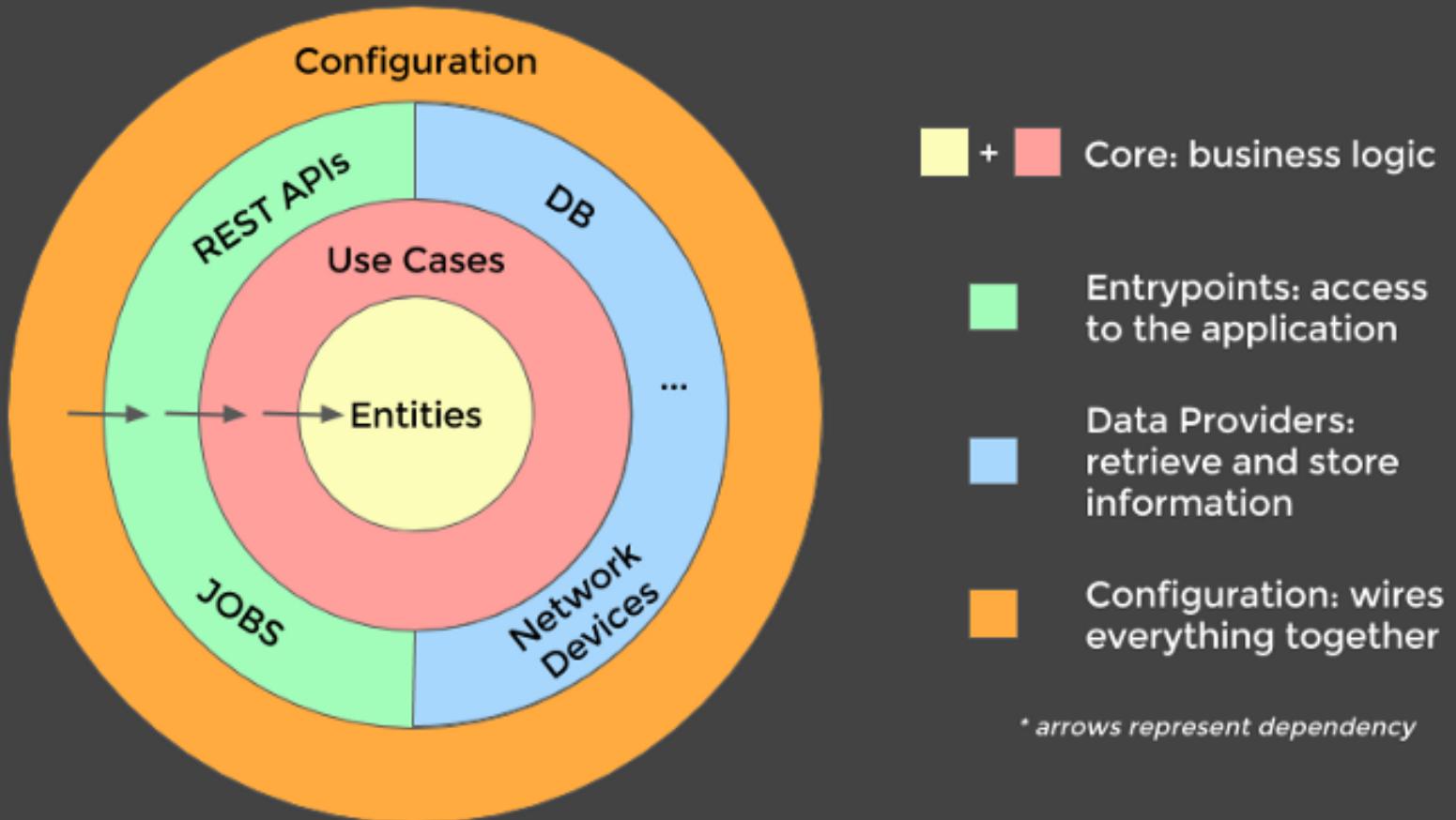
# The Clean Architecture

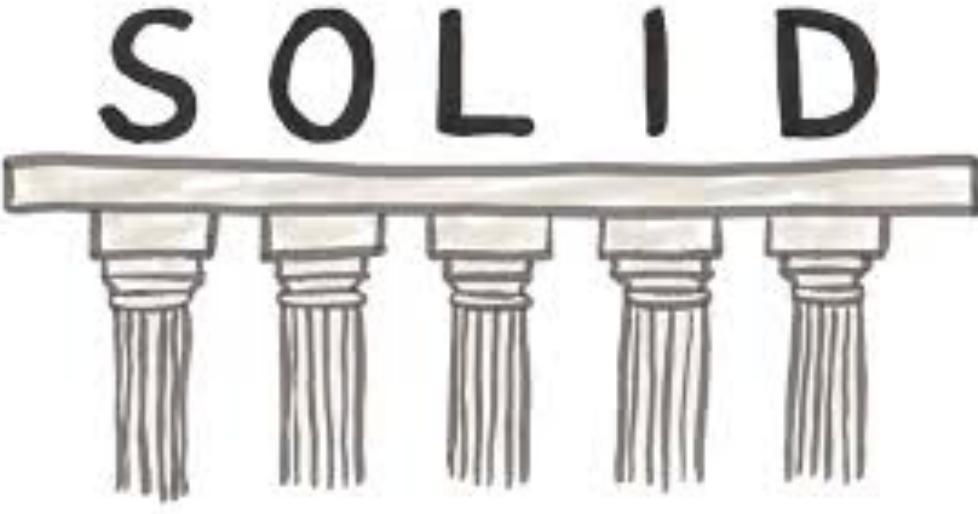


# The Clean Architecture Cone



# A.k.a (unclebob's style)





**S — Single Responsibility Principle** (Princípio da responsabilidade única)

**O — Open-Closed Principle** (Princípio Aberto-Fechado)

**L — Liskov Substitution Principle** (Princípio da substituição de Liskov)

**I — Interface Segregation Principle** (Princípio da Segregação da Interface)

**D — Dependency Inversion Principle** (Princípio da inversão da dependência)