

Understanding Railway-oriented Programming



Zoran Horvat

CEO AT CODING HELMET

@zoranh75

<http://csharpmentor.com>

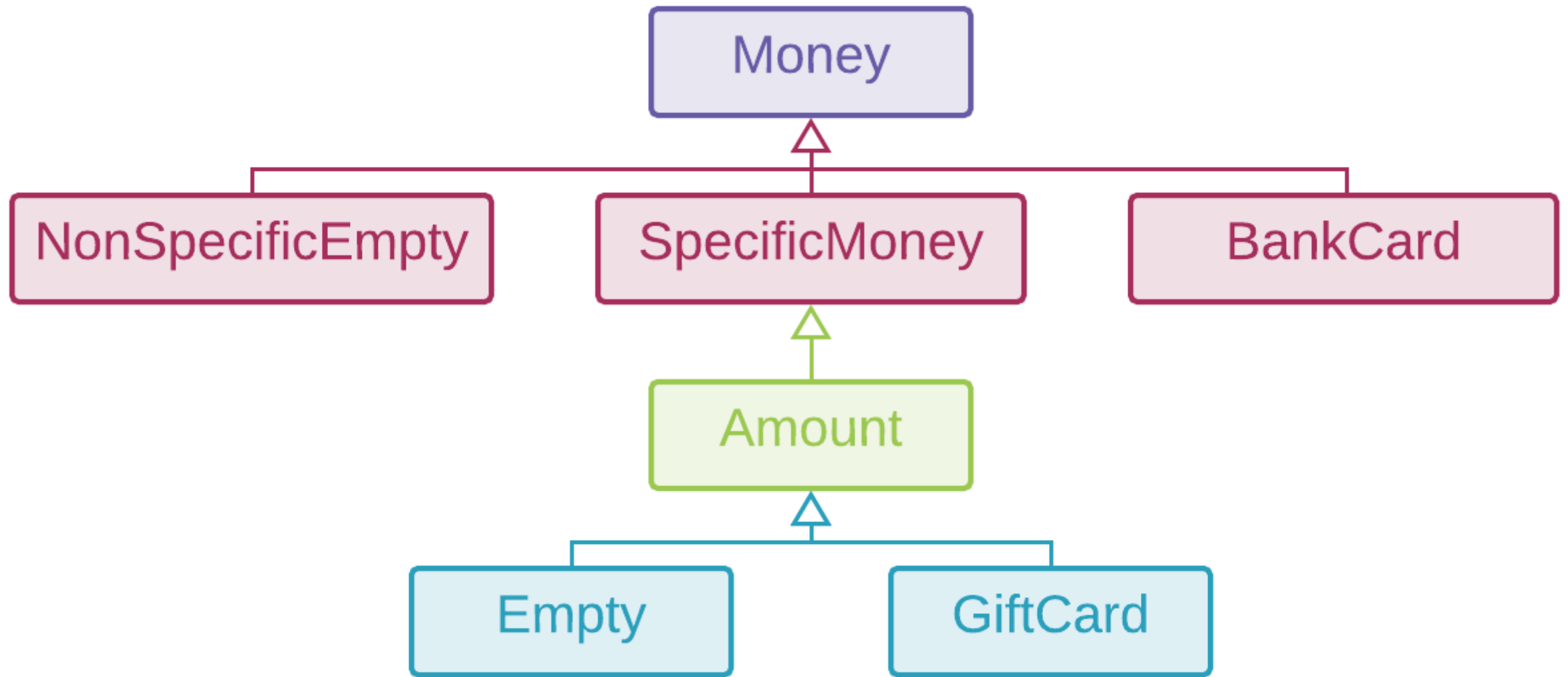


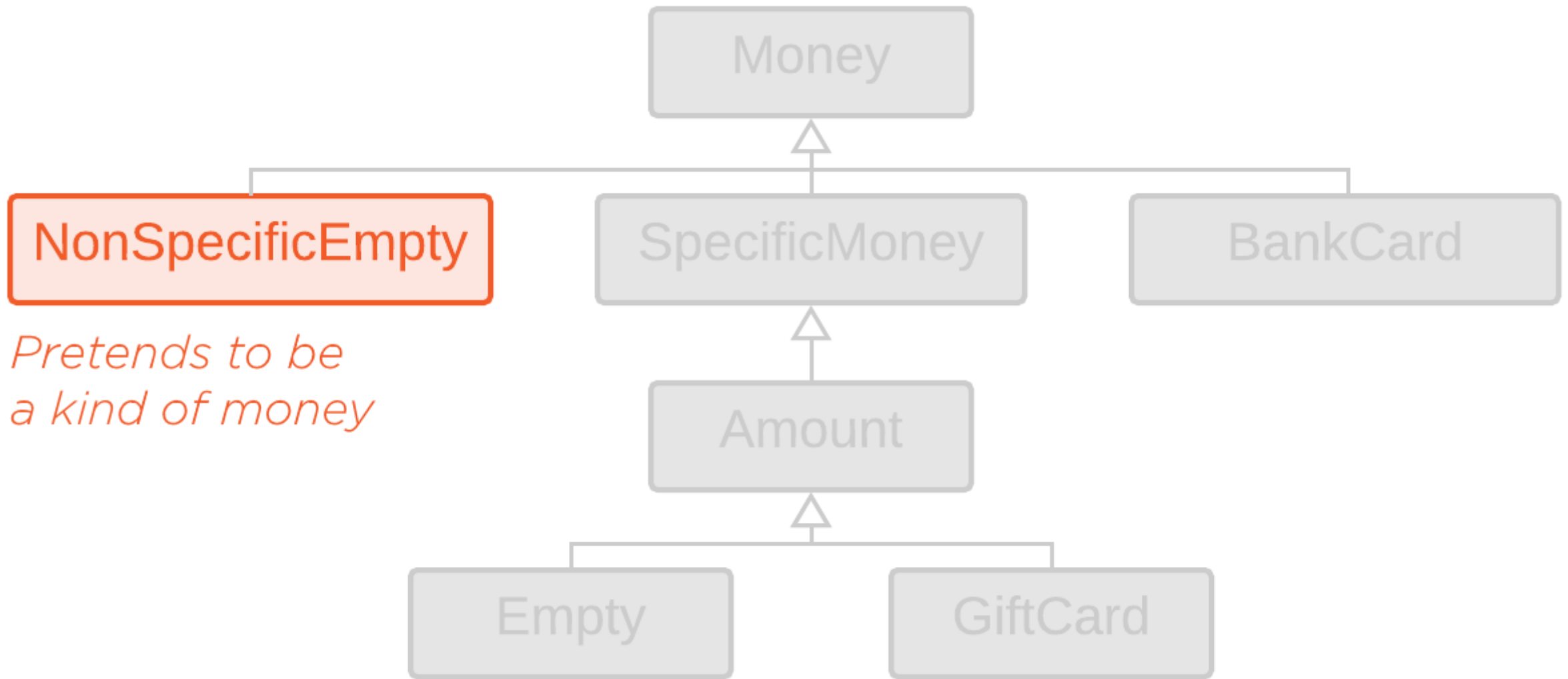
Railway-oriented Programming



Scott Wlaschin







Railway-oriented Programming

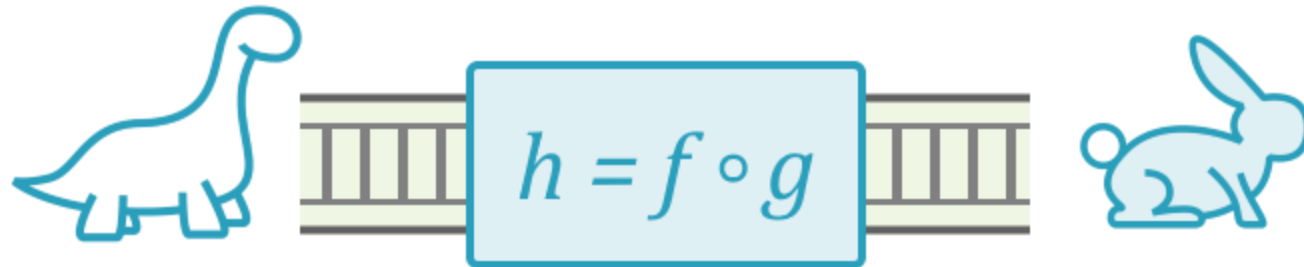


Scott Wlaschin

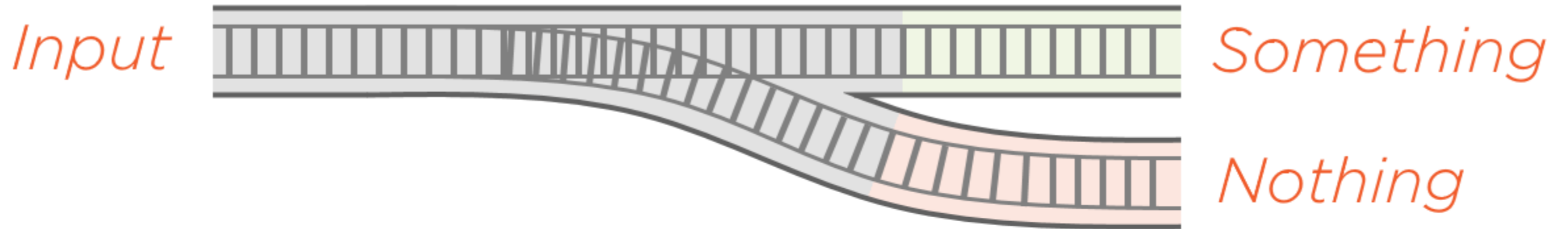
<https://fsharpforfunandprofit.com/rop/>



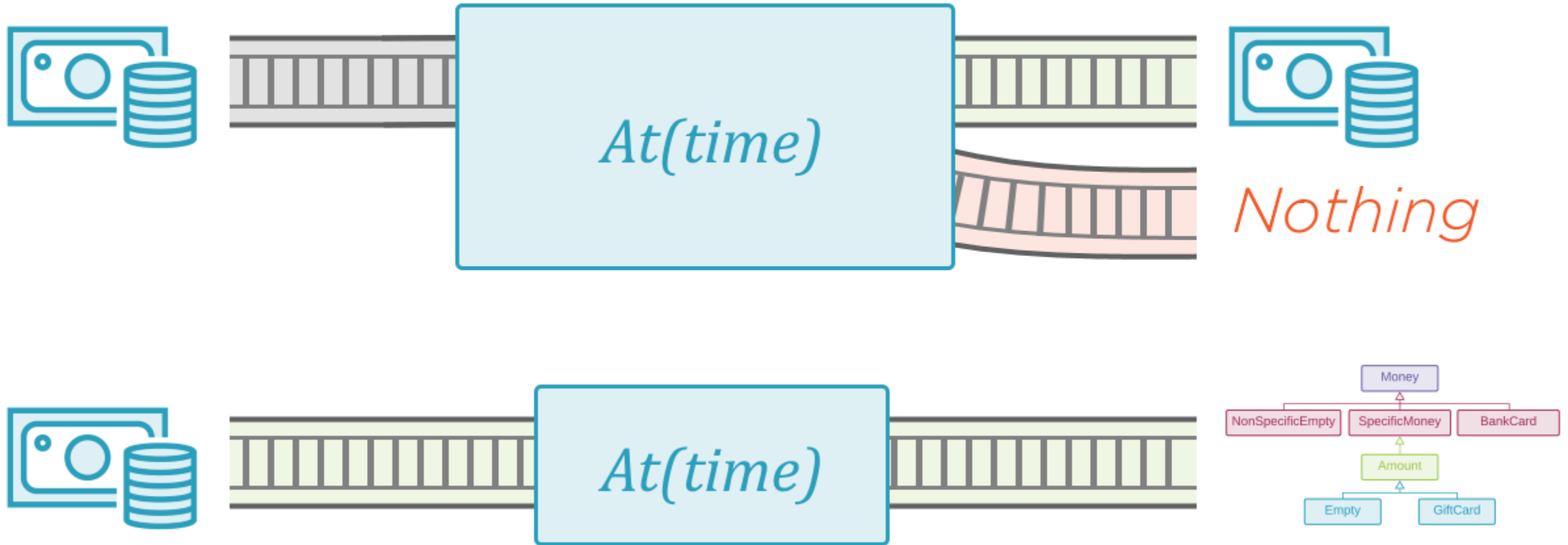
Railway-oriented Programming



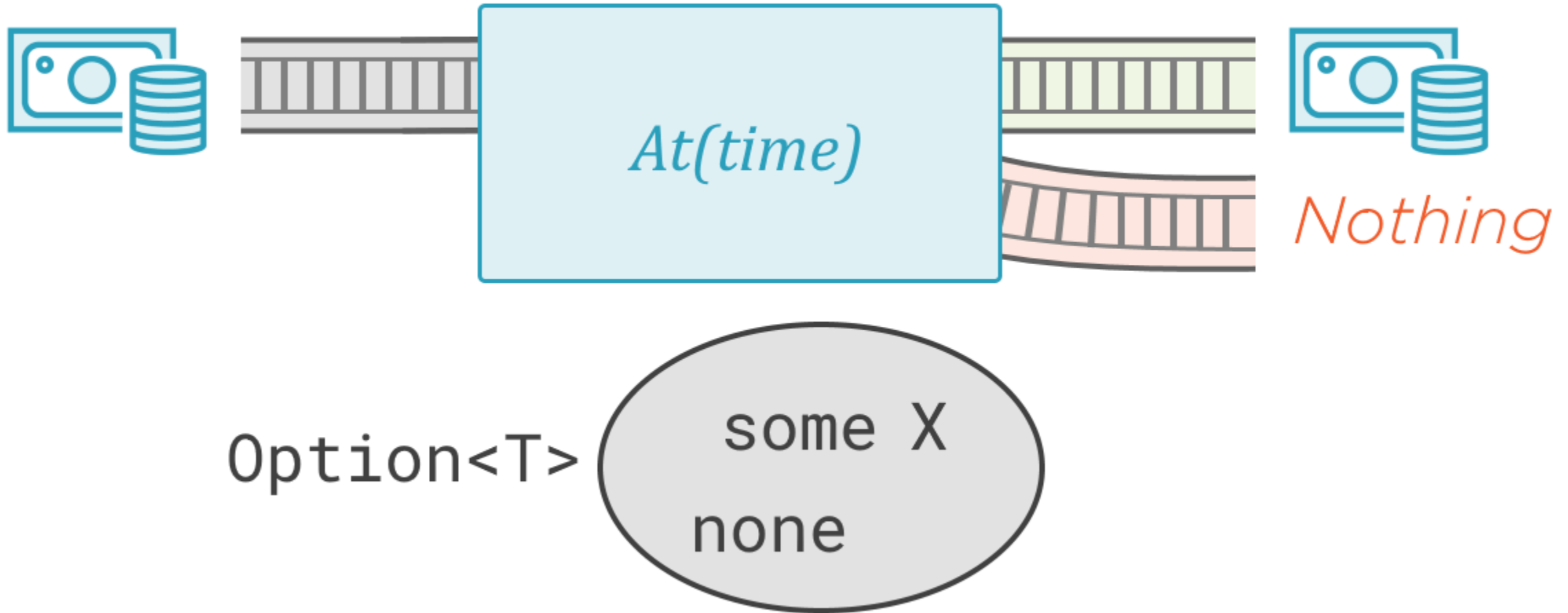
Railway-oriented Programming



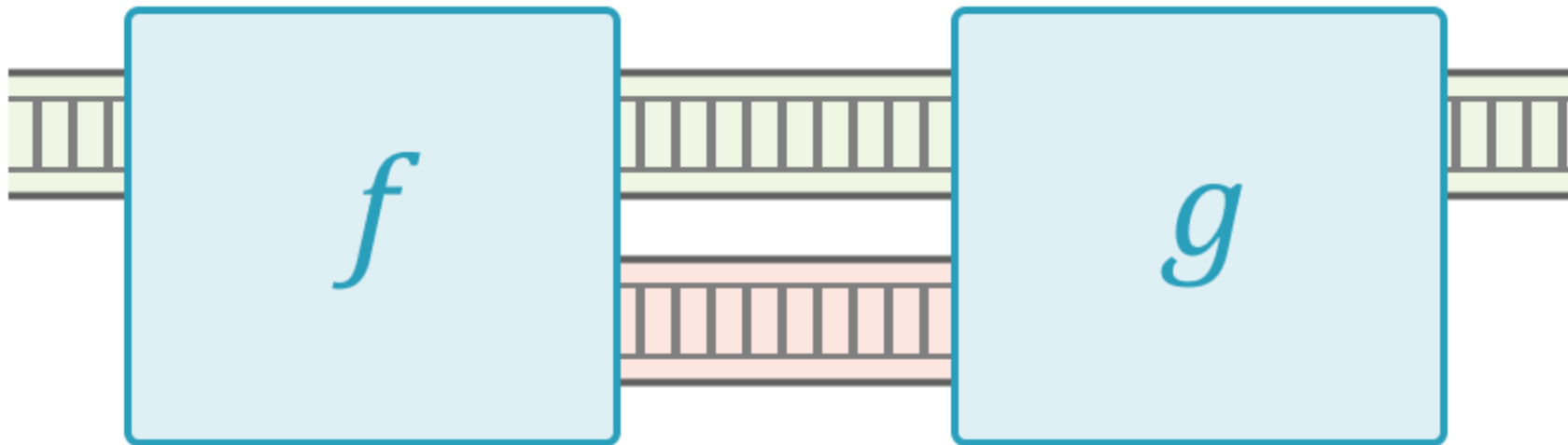
Railway-oriented Programming



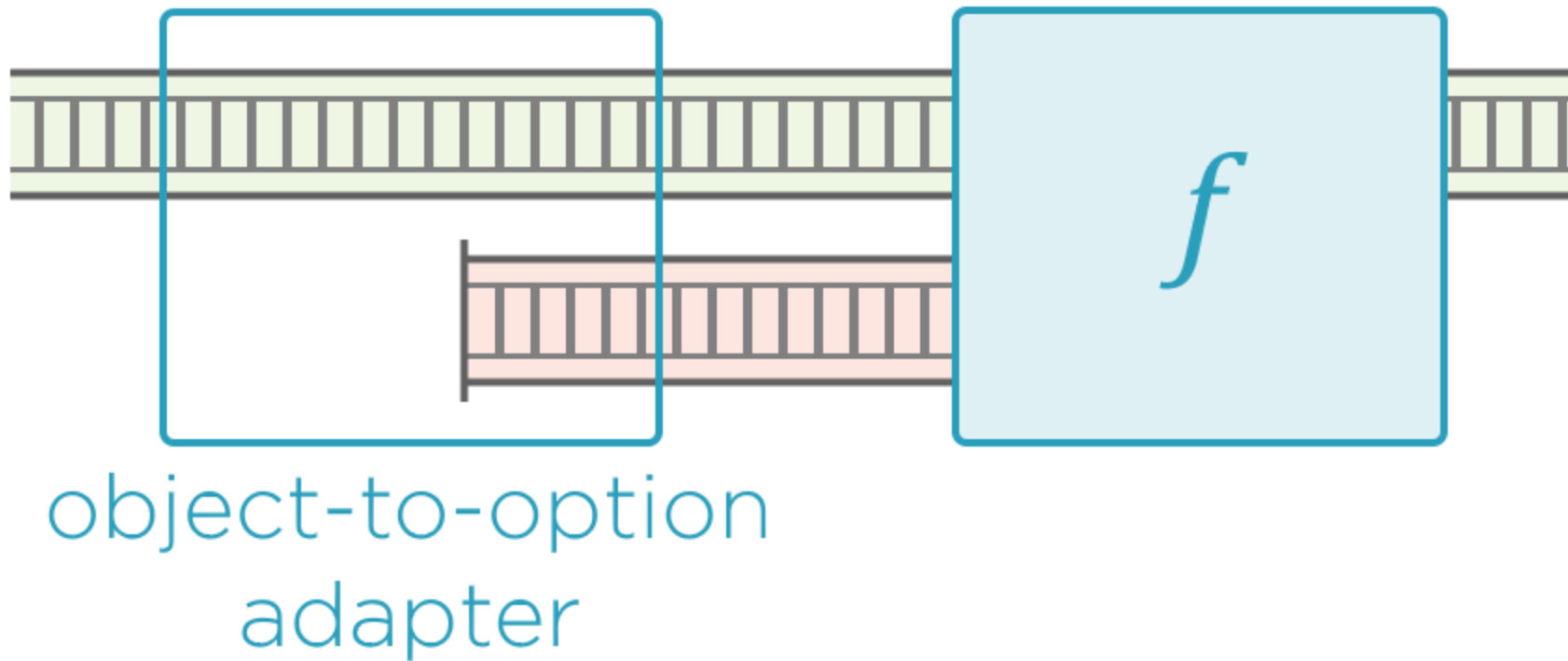
Railway-oriented Programming



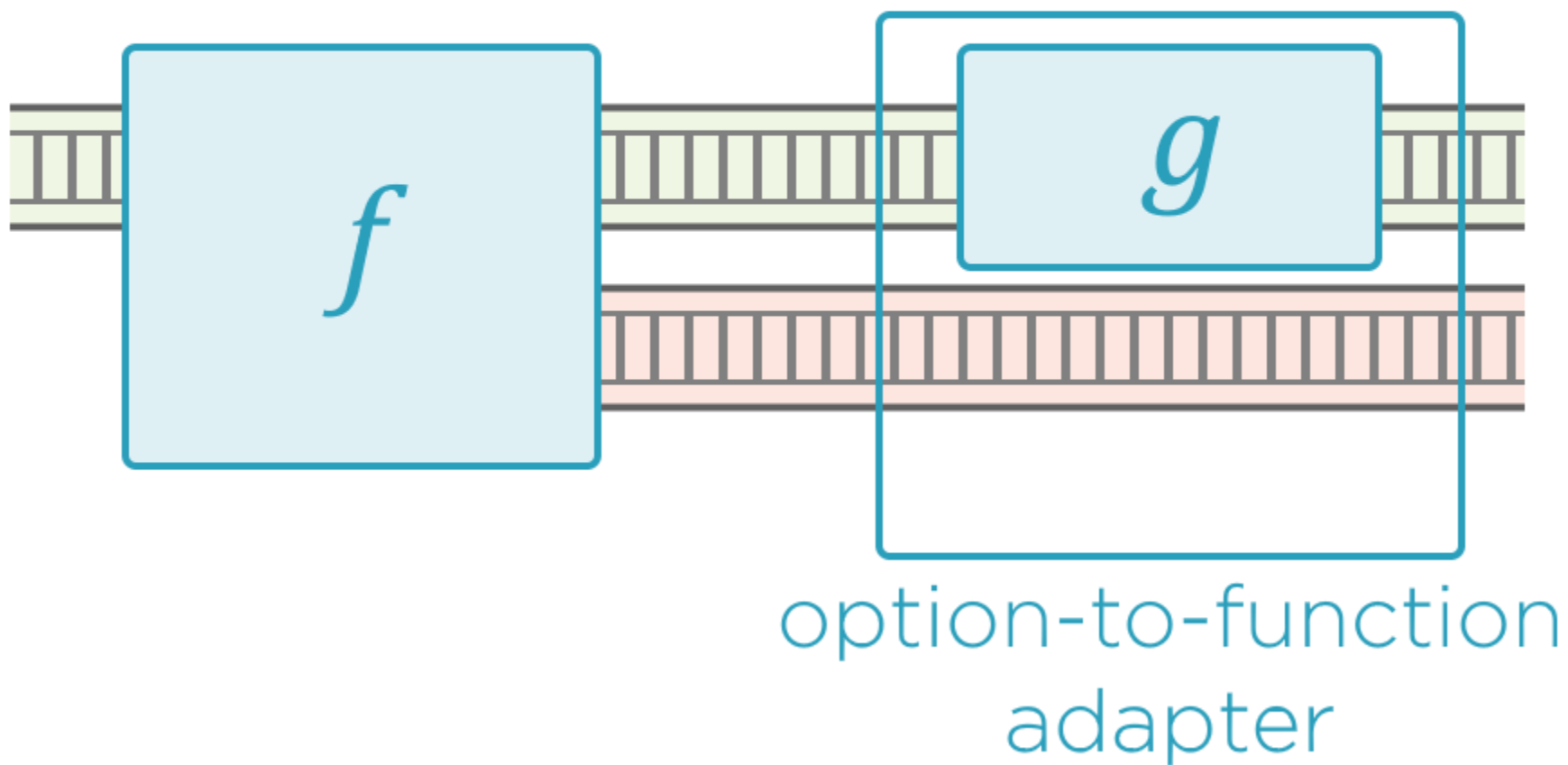
Option Adapters



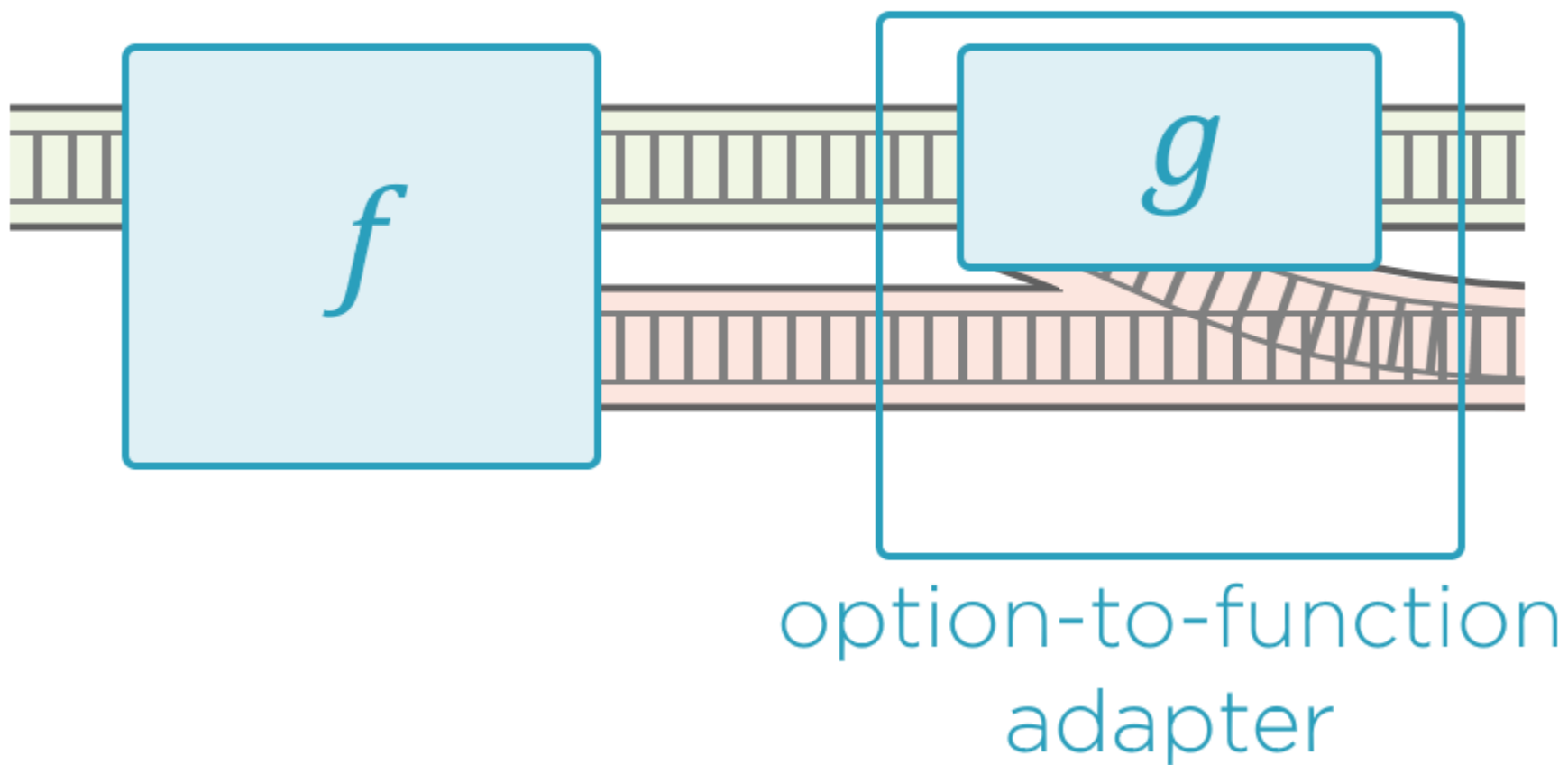
Option Adapters



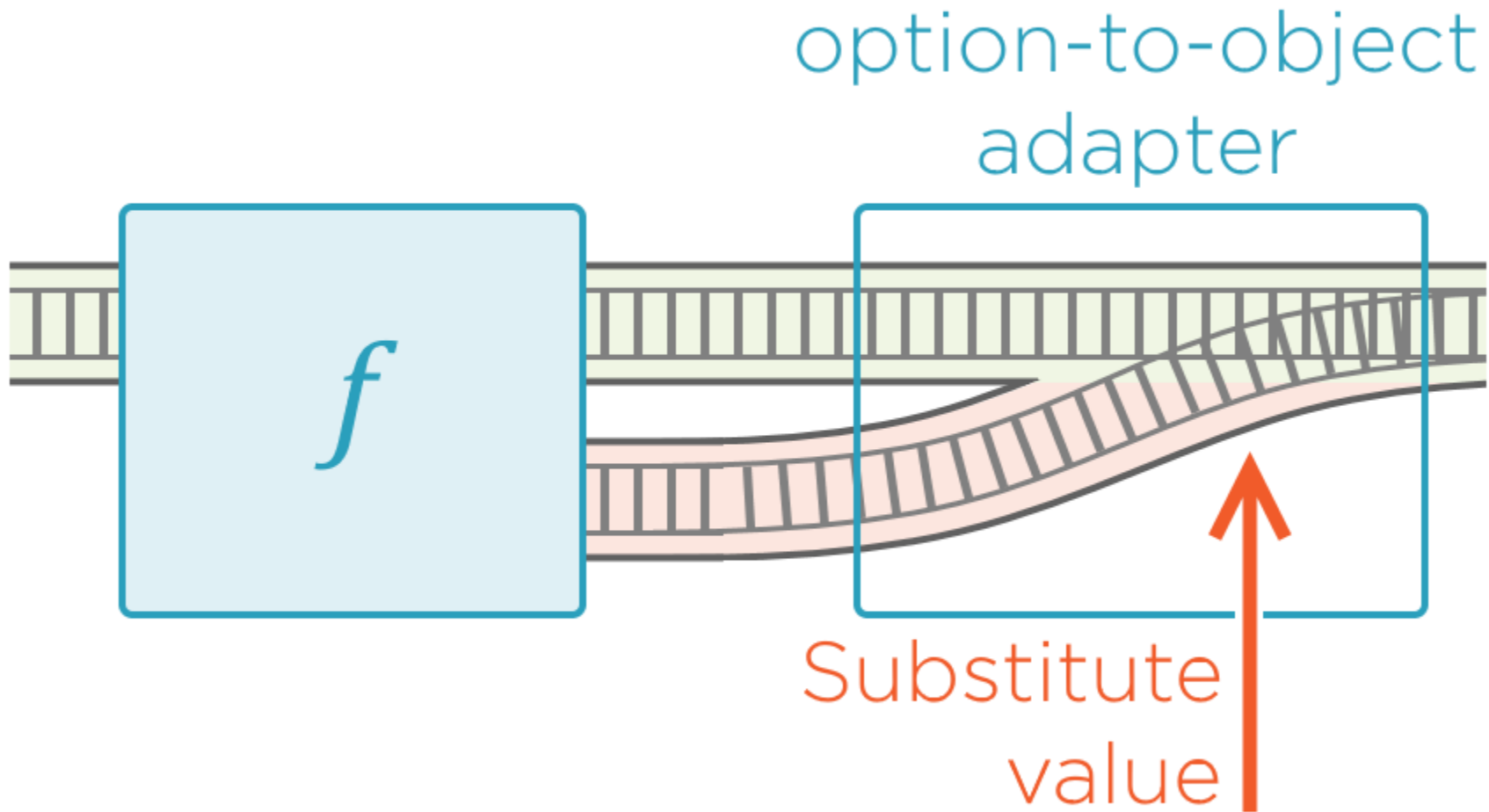
Option Adapters



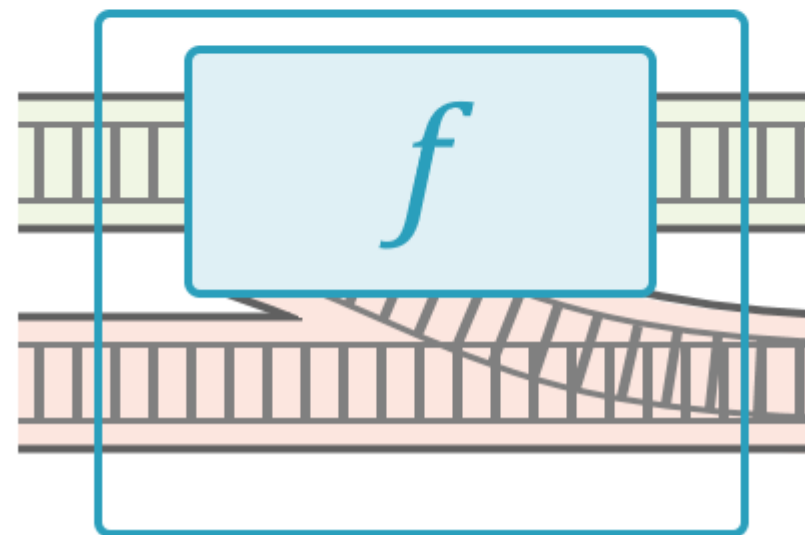
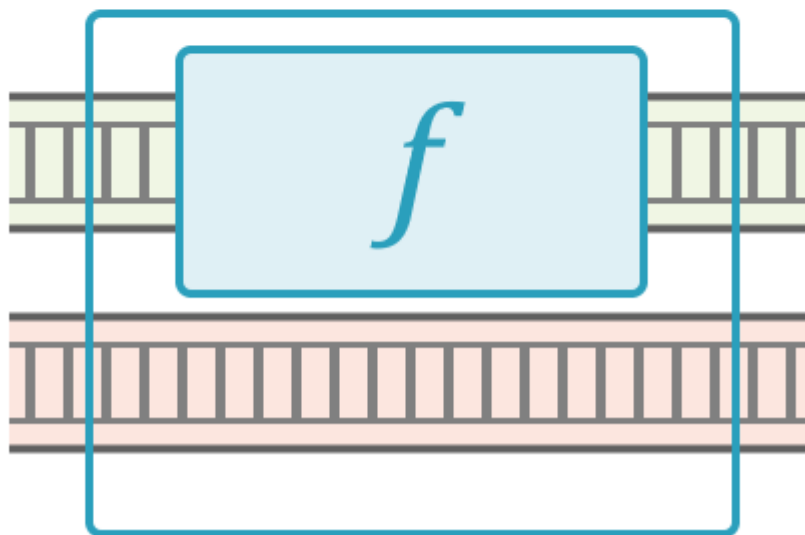
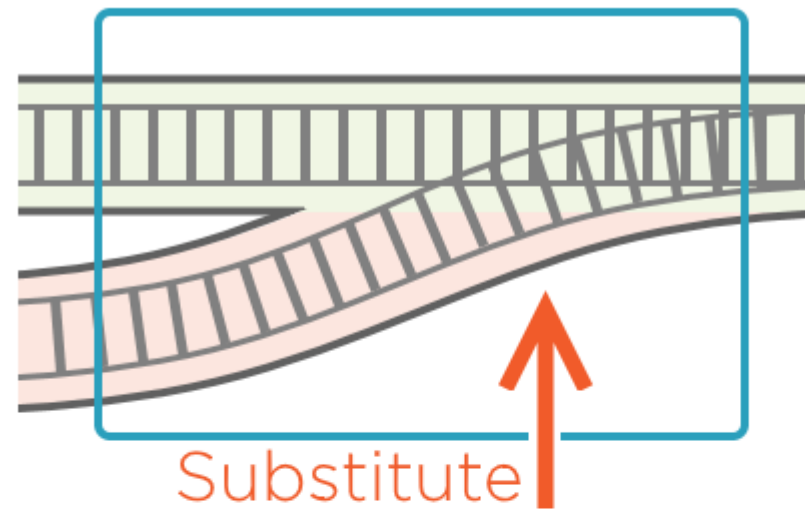
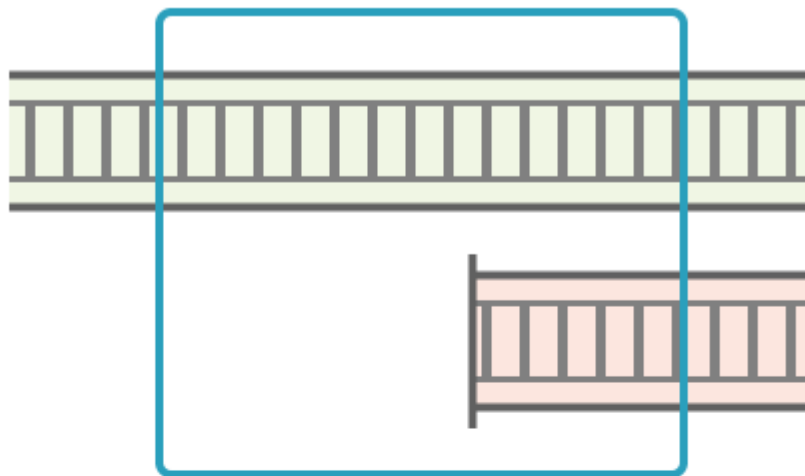
Option Adapters



Option Adapters



Option Adapters



Summary



Railway-oriented programming

- Execution progresses over one track or the other
- Regular track may contain an object
- Switch to alternate track when the object ceases to exist

Summary



Defensive coding and railways

- Both execution tracks must be defined
- None of them is a *special case*
- Omit one, and code will not compile

Function adapters

- Wrap functions into adapters
- Compose wrapped functions
- Existing functions don't have to know about optional objects



Next module:

Handling Errors in Functional Style

