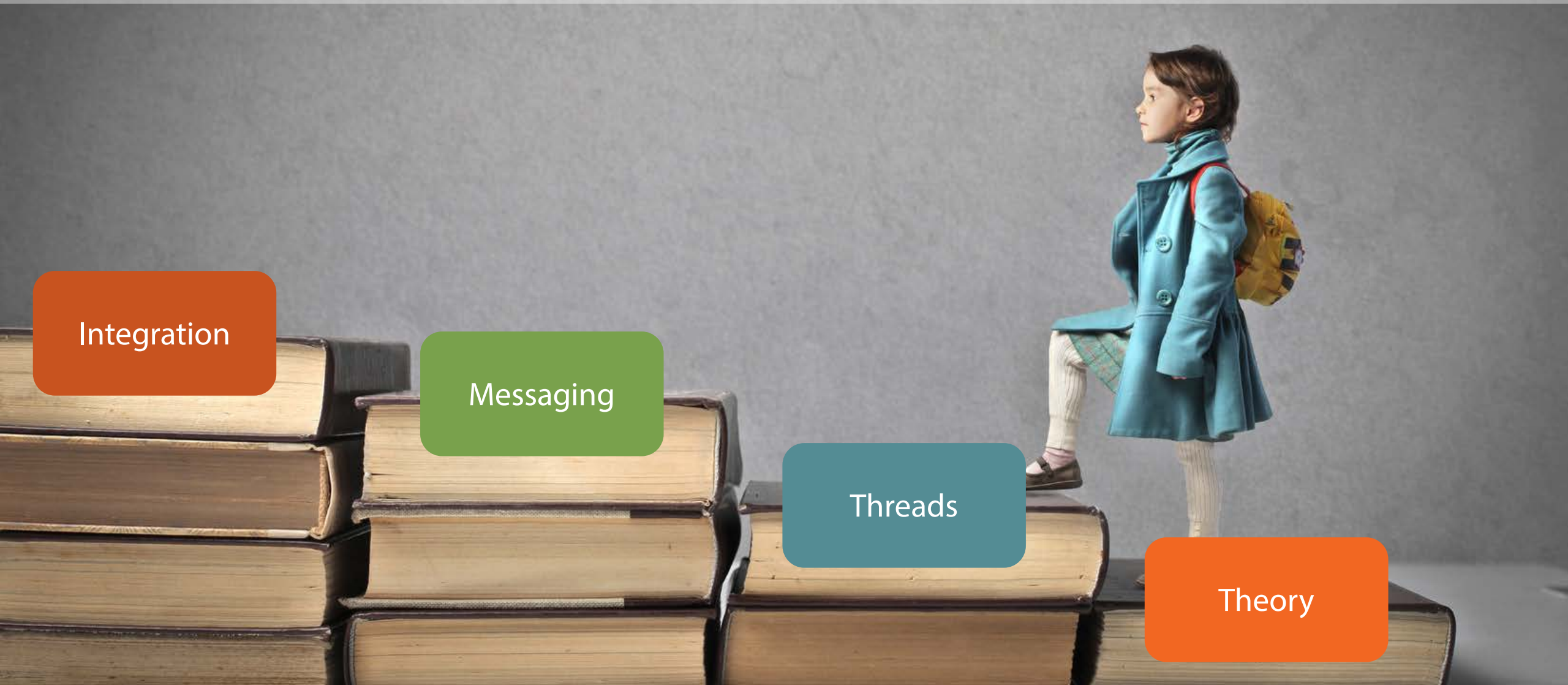
Concurrent Programming with Go

Concurrency Patterns

Michael Van Sickle

http://allibec.com | @vansimke

Course Overview



Who is this Course For?

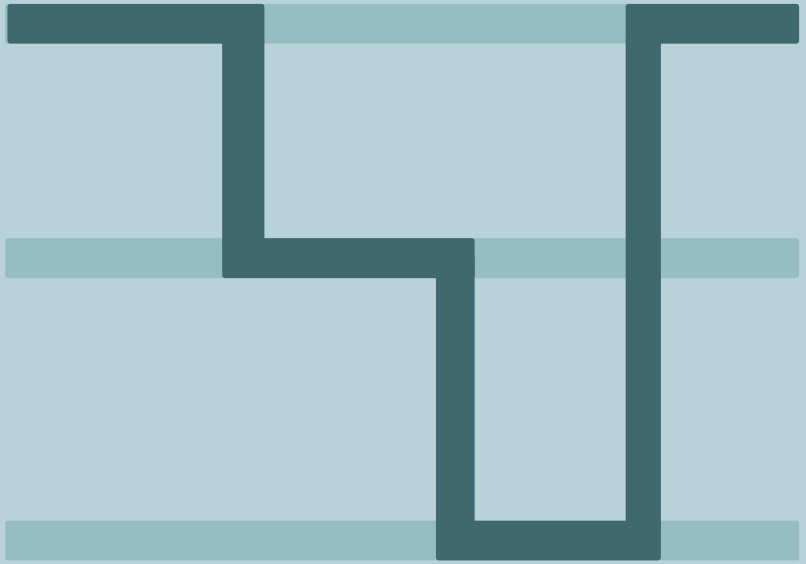
Learning Go

Systems Developer

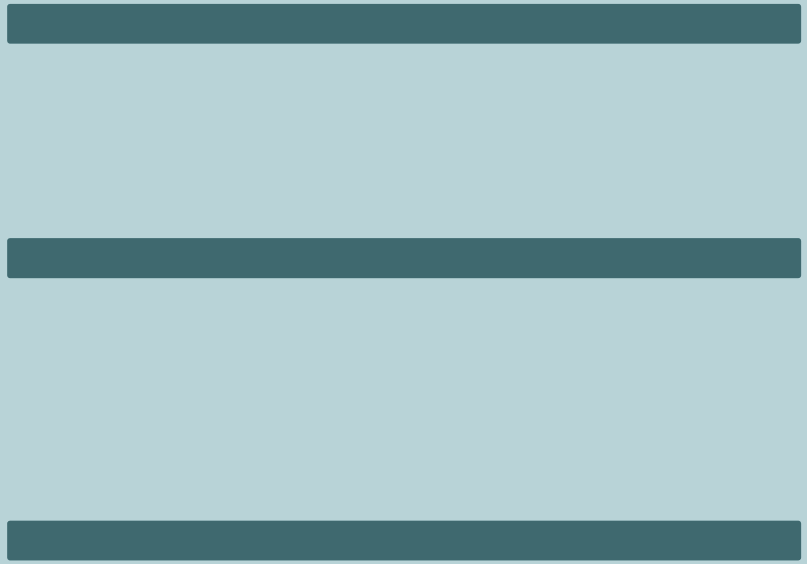
Explorer


The Go Programming Language
John Sonmez
Creating Web Applications with Go
Mike Van Sickle
pluralsight

Concurrent



Parallel



Concurrency Models

- Processor Threads
- Events
- Callbacks and Promises
- Communicating Sequential Processes

Concurrency Models

- Processor Threads
- Events
- Callbacks and Promises
- Communicating Sequential Processes


Process
Mutex, …
Thread

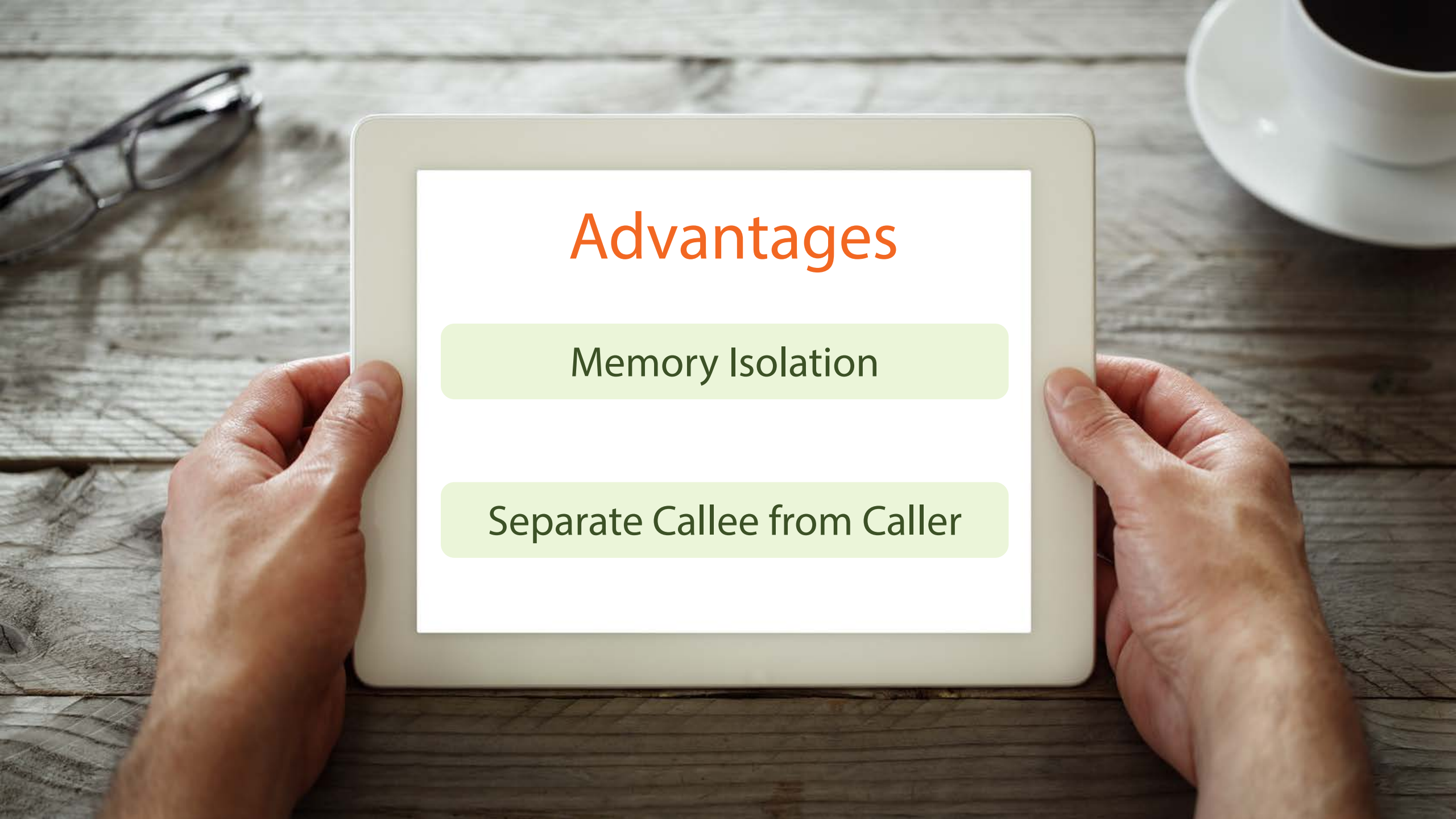

Advantages
Control
Performance (maybe)
Responsive User Interface


Disadvantages
Poor Performance
Memory Consumption
Shared Memory
Race Conditions

Concurrency Models

- Processor Threads
- **Events**
- Callbacks and Promises
- Communicating Sequential Processes


Event
Object
Emitter
Receiver


Advantages
Memory Isolation
Separate Callee from Caller


Disadvantages
Traceability
Synchronizing Receivers

Concurrency Models

- Processor Threads
- Events
- **Callbacks and Promises**
- Communicating Sequential Processes


Promise
Callback


Advantages
Memory Isolation
Simplify Async Operations


Disadvantages
Pyramid of Doom
Handling Multiple Receivers

Concurrency Models

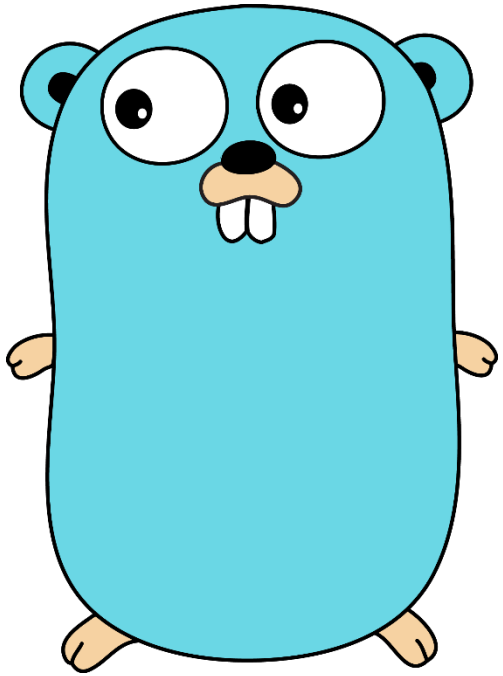
- Processor Threads
- Events
- Callbacks and Promises
- Communicating Sequential Processes


Actor
Message


Advantages
Fully Decoupled
Multiple Handlers
Memory Isolation


Disadvantages
Complicated Mental Model
Traceability

Concurrency in Go



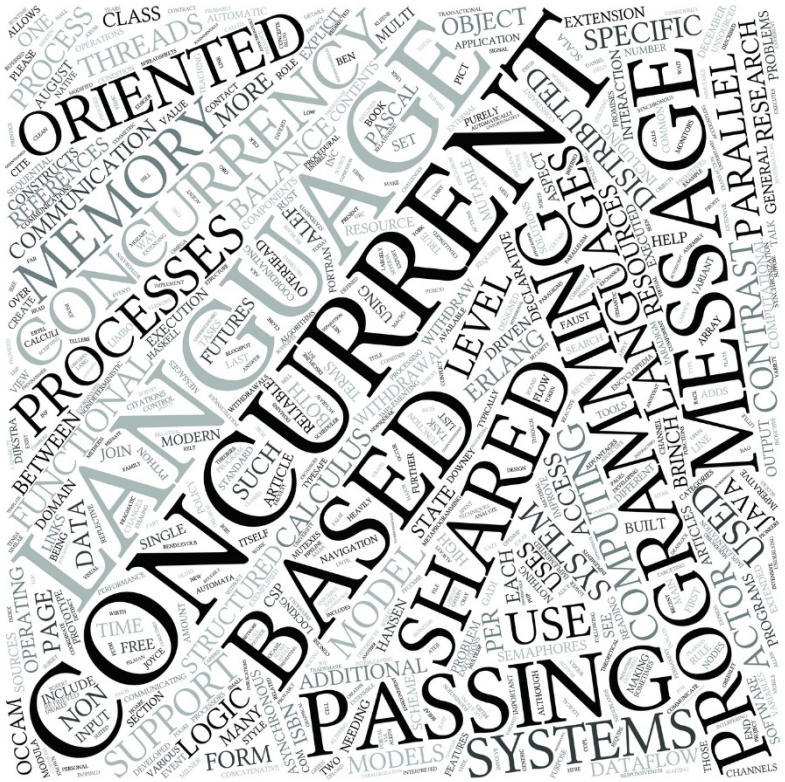
No Thread Primitives

Goroutines

Channels

Review of Dev Env

Summary



Concurrency Strategies

Go's Concurrency Model