The Go Programming Language

- Open Source – Most of Core Team @ Google
- Statically Typed, Compiled, Garbage Collected
- Intended for large dev teams (think Googleplex) working on widely distributed server systems
- 5 years old ~ version 1.5.x

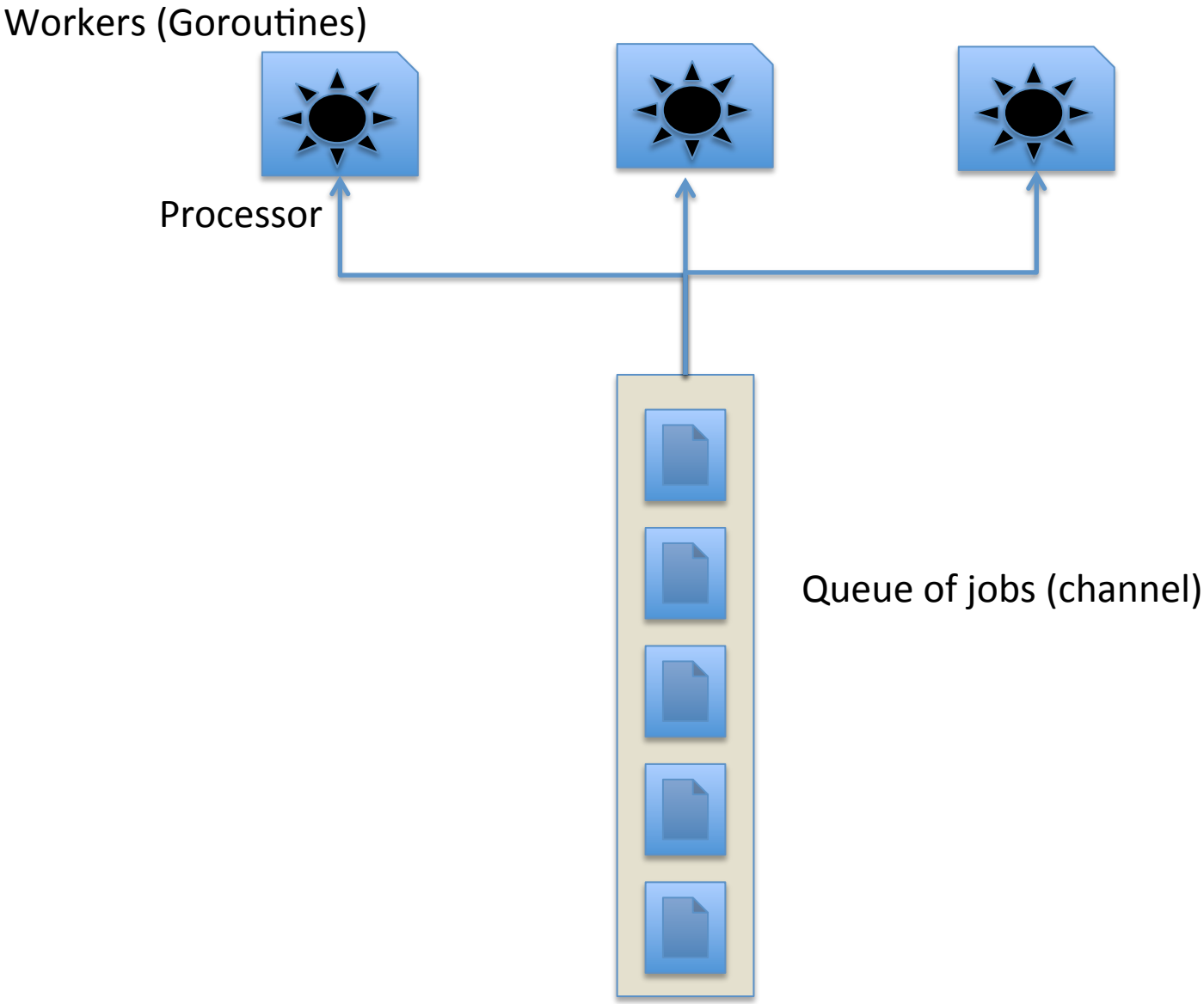
Go Benefits

- Fast compile times
- 'Lightweight' syntax (25 keywords, 13 builtin functions)
- Easy concurrency using "Goroutines" & "channels"
- Solid standard library
- Compile/link to a single binary
- Easy compile across:
  - OS: darwin, dragonfly, freebsd, linux, netbsd, openbsd, plan9, solaris, windows,
  - Architectures: 386, amd64, amd64 , arm, arm64, ppc64, ppc64le

Sample Program

- Create three workers that execute on a backlog of jobs
- Create a common queue of 25 jobs to be processed by the set of workers
- Each job can execute one of two types of tasks:
  - "Email" a message to a recipient
  - "Text" a message to a recipient
  - Task mix is: ~ 70% texting, 30% emailing
- Each job takes 3 seconds to execute
- Print out a log and summary

Overview



Go Language Features

- Fire up “Goroutines” (***workers***) to work on jobs concurrently
- Use a “channel” (***job queue***) to safely communicate to a Goroutine
- Functions as variables (functions are “first class”citizens) (***tasks***)
- Use a “waitgroup” to manage the processing of concurrently running Goroutines
- Compile to Mac and Windows binaries
  - Windows compile with one command:
  - “GOOS=windows GOARCH=386 go build -o techedge.exe”