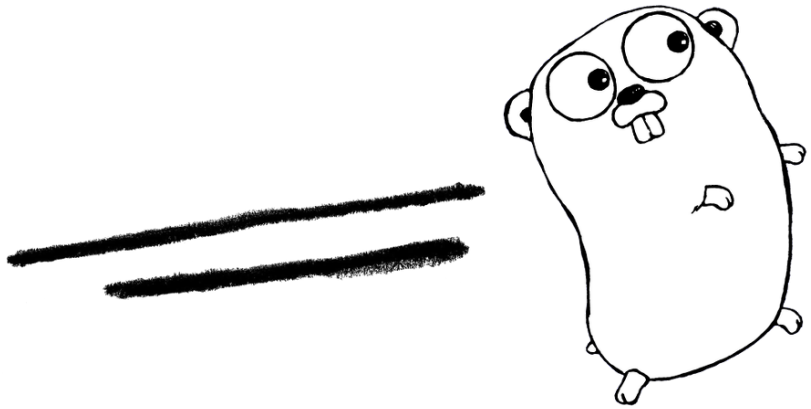
Why Write in Go?

Context around the Go Programming Language

Grant Griffiths
Software Engineer - Predix PCE PaaS
Digital Technology Leadership Program

What is Go?

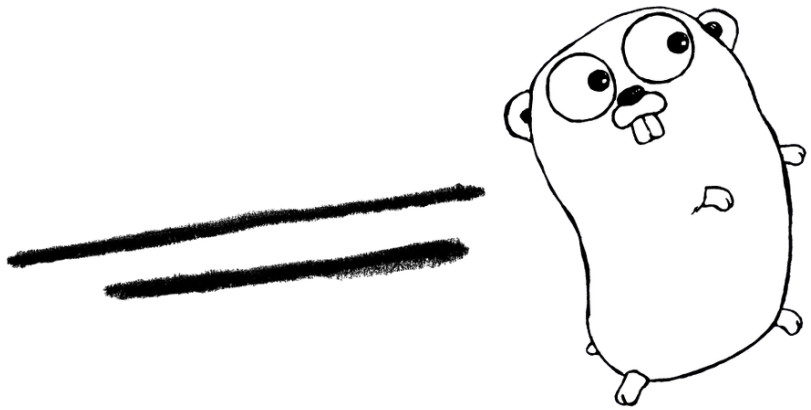
- General-purpose language
- Strongly typed
- Garbage collected
- Explicit support for concurrency
- Compiles to executable binary
- Open source, by Google



Why was the language created

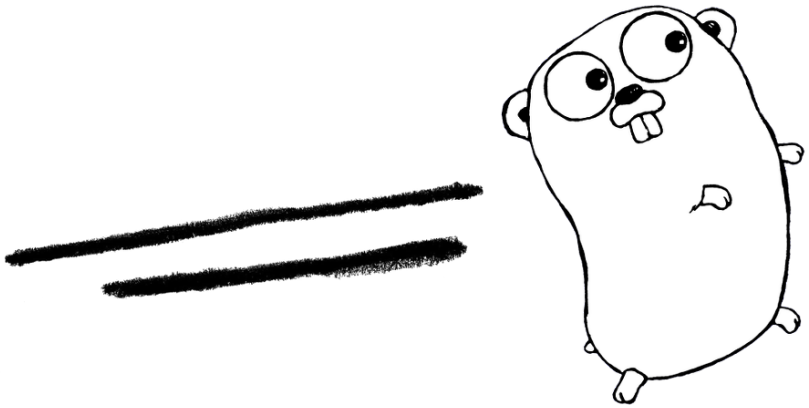
Motivation:
- Software Infrastructure scale issues @ Google
- Frustrating large C/C++ codebases
- Long build times
- Bad tooling

Goals:
- Support for networked/multicore computing
- Simple and elegant, yet high performance
- Quick compilation time for large codebases



Historical Context

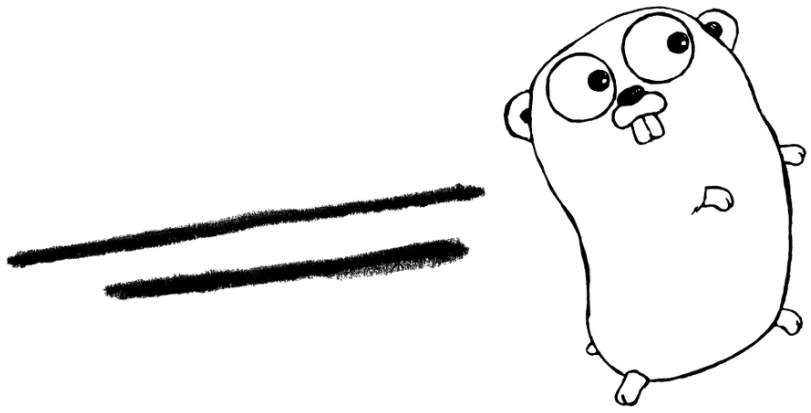
- Authors: Robert Griesemer, Rob Pike, and Ken Thompson
- Whiteboarding started September 2007 while waiting for large C++ builds to run
- Russ Cox joined in late 2008. He helped move the language and libraries from prototype to reality.
- Go 1.0 was released in March 2012
- Became open sourced on November 10, 2009



Ok, but why should I write in Go?

But really, why write in Go?

- Great for building microservices
- Small memory/disk overhead
- Great tooling for development and testing
- Simple, easy to learn - 25 keywords
- Native support for concurrency
- Many major projects written in Go
- Amazing community and Mascot



Building Microservices

```
package main

import (
    "fmt"
    "log"
    "net/http"
    "os"
)

func main() {
    http.HandleFunc("/", hello)
    err := http.ListenAndServe(":"+os.Getenv("PORT"), nil)
    if err != nil {
        log.Fatal("ListenAndServe:", err)
    }
}

func hello(w http.ResponseWriter, req *http.Request) {
    fmt.Fprintln(w, "hello, world!")
}
```

Small memory/disk overhead

- Go vs. Java memory/disk usage
- Deployed two hello world microservices in Go and Java both exposing a REST API

```
     state     since                  cpu    memory         disk           details
#0   running   2017-05-24T00:54:50Z   0.0%   5.8M of 64M    12.5M of 1G
```

```
     state     since                  cpu    memory         disk           details
#0   running   2017-05-24T01:12:48Z   0.1%   315.7M of 512M 134.9M of 1G
```

Tooling

Built in tools such as...

- go fmt (https://golang.org/cmd/gofmt/), Standardized auto formating
- golint (https://github.com/golang/lint), Linter
- goimports (https://godoc.org/golang.org/x/tools/cmd/goimports), Auto imports
- go test (https://golang.org/pkg/testing), Built in testing
- pprof (https://golang.org/pkg/net/http/pprof), Profiling and benchmarking
- many community ones

Easy to learn

Designed to be simple and readable

Only 25 keywords

```
break        default      func         interface    select
case         defer        go           map          struct
chan         else         goto         package      switch
const        fallthrough  if           range        type
continue     for          import       return       var
```

and 5 secret ones...

```
notwithstanding
thetruthofthematter
despiteallobjections
whereas
insofaras
```

Concurrency - Goroutines

```
package main

import "fmt"

func f(from string) {
    for i := 0; i < 3; i++ {
        fmt.Println(from, ":", i)
    }
}

func main() {
    // Direct function call
    f("direct")

    // Called in a Goroutine (logical thread)
    go f("goroutine")
}
```

Run

Concurrency - Channels

```
package main

import "fmt"

func main() {
    // Channels are the pipes that connect concurrent goroutines
    messages := make(chan string)

    // Send a message to our new goroutine's channel
    go func() {
        messages <- "ping"
    }()

    // Read our message from our main channel
    msg := <-messages
    fmt.Println(msg)
}
```

Run

Concurrency - Much mroe

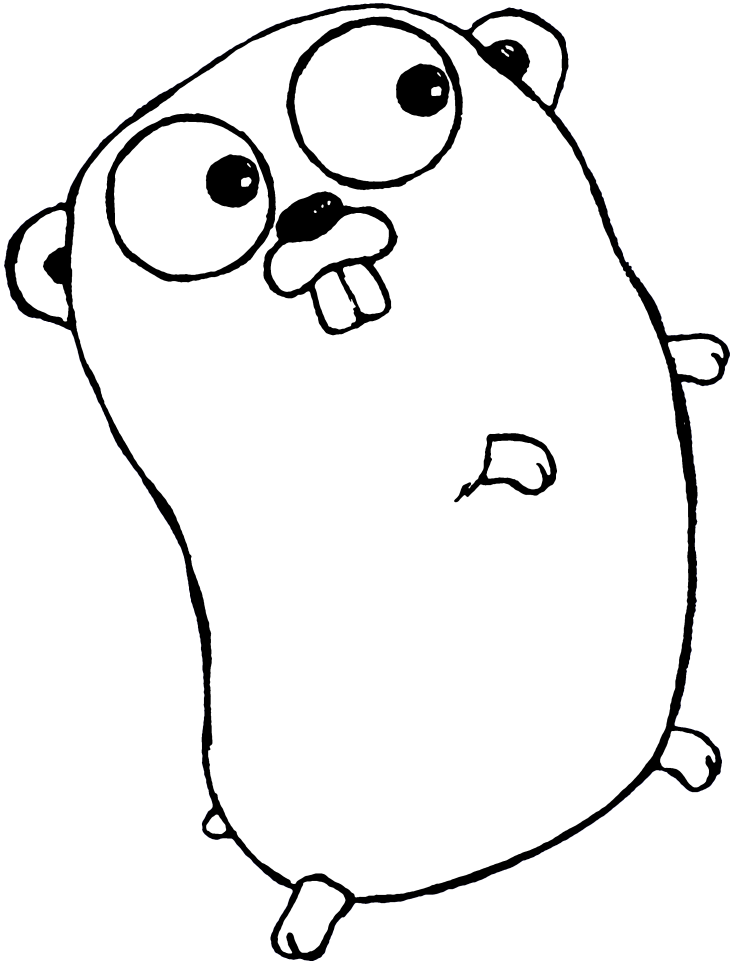
- Wait groups
- Mutex
- Buffered channels
- etc

Major projects in Go

- Kubernetes
- Docker
- Etcd
- Terraform/Vault/Consul/etc
- CoreOS
- Prometheus
- InfluxDB
- Cloud Foundry
- Go
- many more

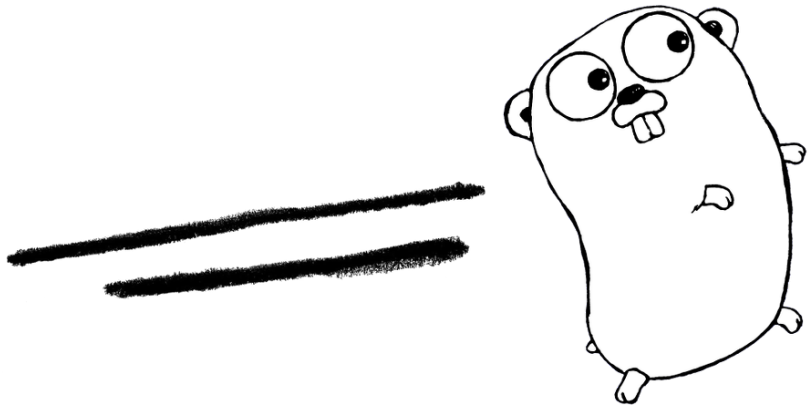
The Go Gopher

By Renée French

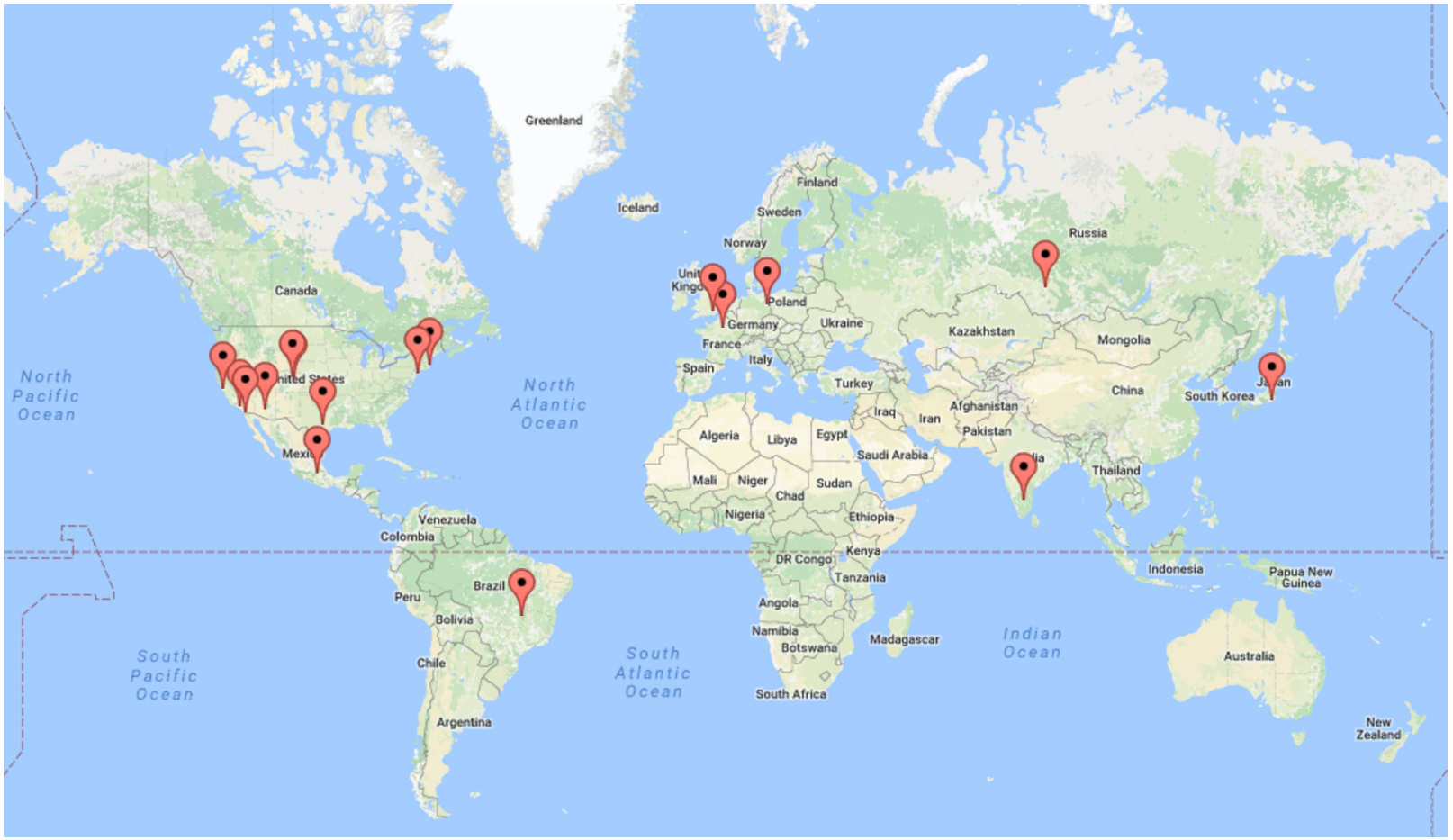


The community

- Very unique and friendly community
- More slides borrowed from State of Go - May 2017 by Francesc

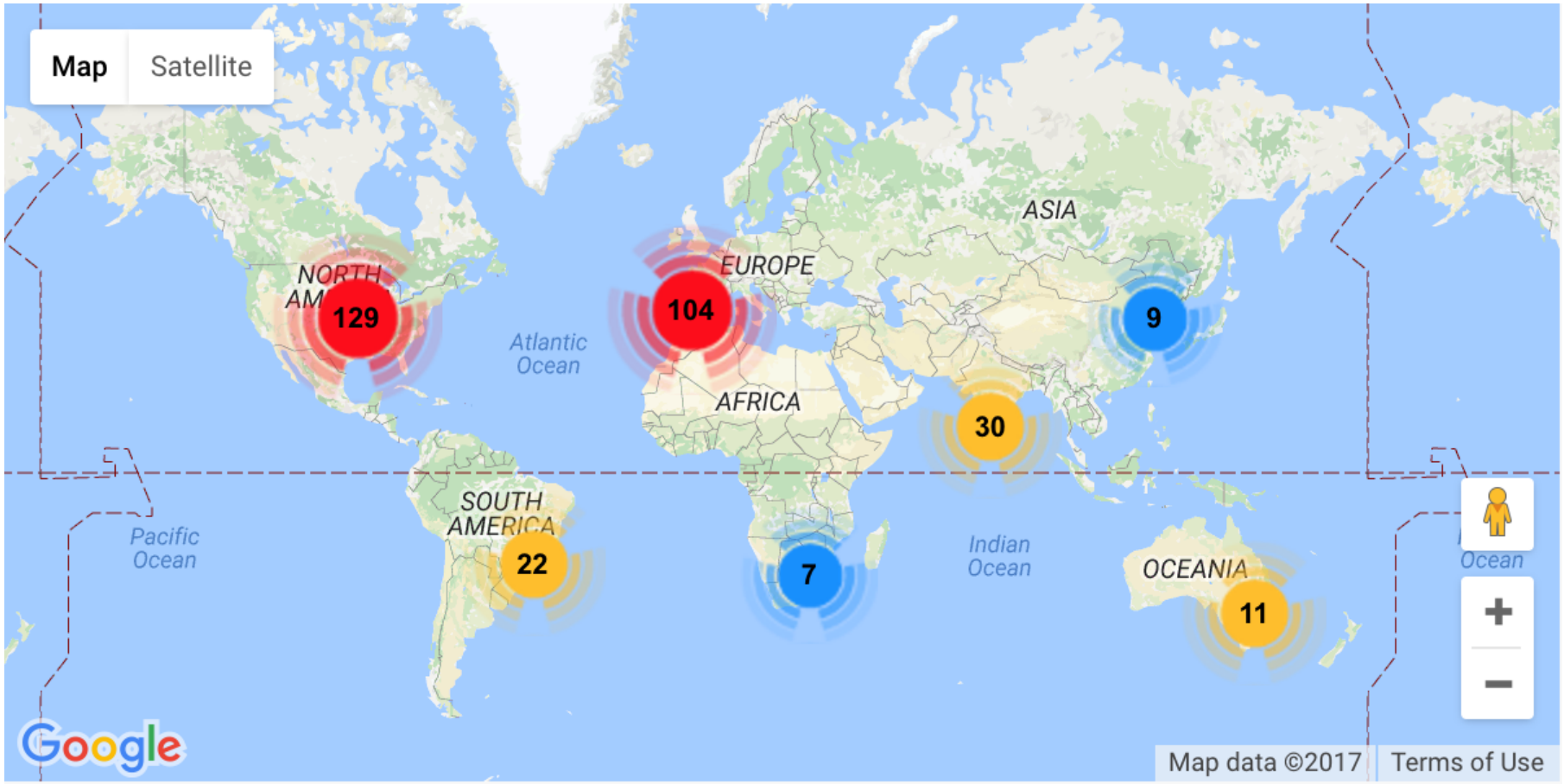


Women Who Go



16 chapters already! www.womenwhogo.org (http://www.womenwhogo.org)

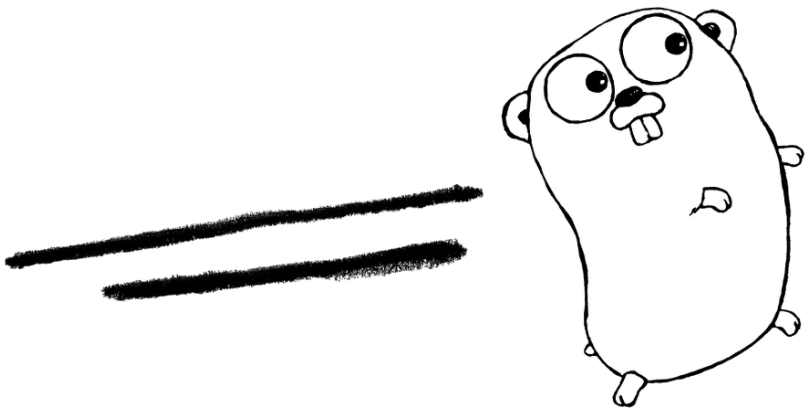
Go meetups



Gophers all around the world! go-meetups.appspot.com (http://go-meetups.appspot.com)

Conferences in 2017:

- GopherCon India (http://www.gophercon.in/), Feb 25-25th
- Gophercon Denver (https://gophercon.com/), Jul 12-15th
- Golang UK (http://golanguk.com/), August 16th-18th
- dotGo (http://2017.dotgo.eu/), Nov 6th
- Golang UK (https://golanguk.com), Aug 16th
- GopherCon Brasil (https://2017.gopherconbr.org/en), Nov 17th



Where to get started

Self learning

- Go by Example (https://gobyexample.com)
- Effective Go (https://golang.org/doc/effective_go.html)

Team Training

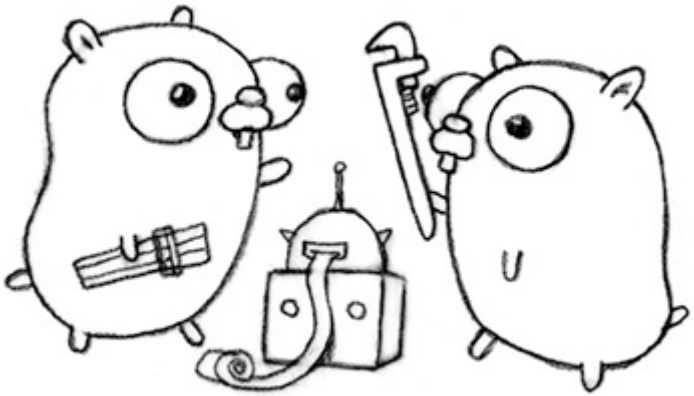
- Ultimate Go (https://www.ardanlabs.com/ultimate-go)
- Gopher Academy (https://gopheracademy.com)

Misc

- Just for Func (https://www.youtube.com/playlist?list=PL64wiCrrxh4Jisi7OcCJIUpguV_f5jGnZ), video series by Francesc Campoy
- Go Time (https://changelog.com/gotime), podcast on Go
- Operation Go (http://gocode.io), fun game to learn Go

But wait... should I write in Go?

- You decide!
- Pick the right tool for the job
- Facts over opinions - a lot of clutter on the internet



Thank you

Grant Griffiths
Software Engineer - Predix PCE PaaS
Digital Technology Leadership Program
grant.griffiths@ge.com (mailto:grant.griffiths@ge.com)