How Can Go Help You?

Michael Van Sickle

@vansimke

Overview

Philosophy and values

Primary use-cases

Philosophy and Values

- Simplicity
- Network aware and concurrent apps
- Out-of-the-box experience
- Cross-platform
- Backward compatibility

Simplicity

```
i := 1

println(i++)    // ???

println(++i)    // ???
```

Problem:
Increment and decrement expressions are easily misinterpreted

Simplicity

```
i := 1

i++

println(i) // 2

i++

println(i) // 3
```

Solution:
Increment and decrements are statements in Go

`for i := 0; i < 5; i++ ...`	◀ loop with incrementor
`for i < 5 ...`	◀ loop till condition
`for ...`	◀ infinite loop
`for user := range users ...`	◀ loop over collection

All loops in Go
are for-loops!

Network Aware and Concurrent Apps

net and net/http packages

Create web servers using only standard library

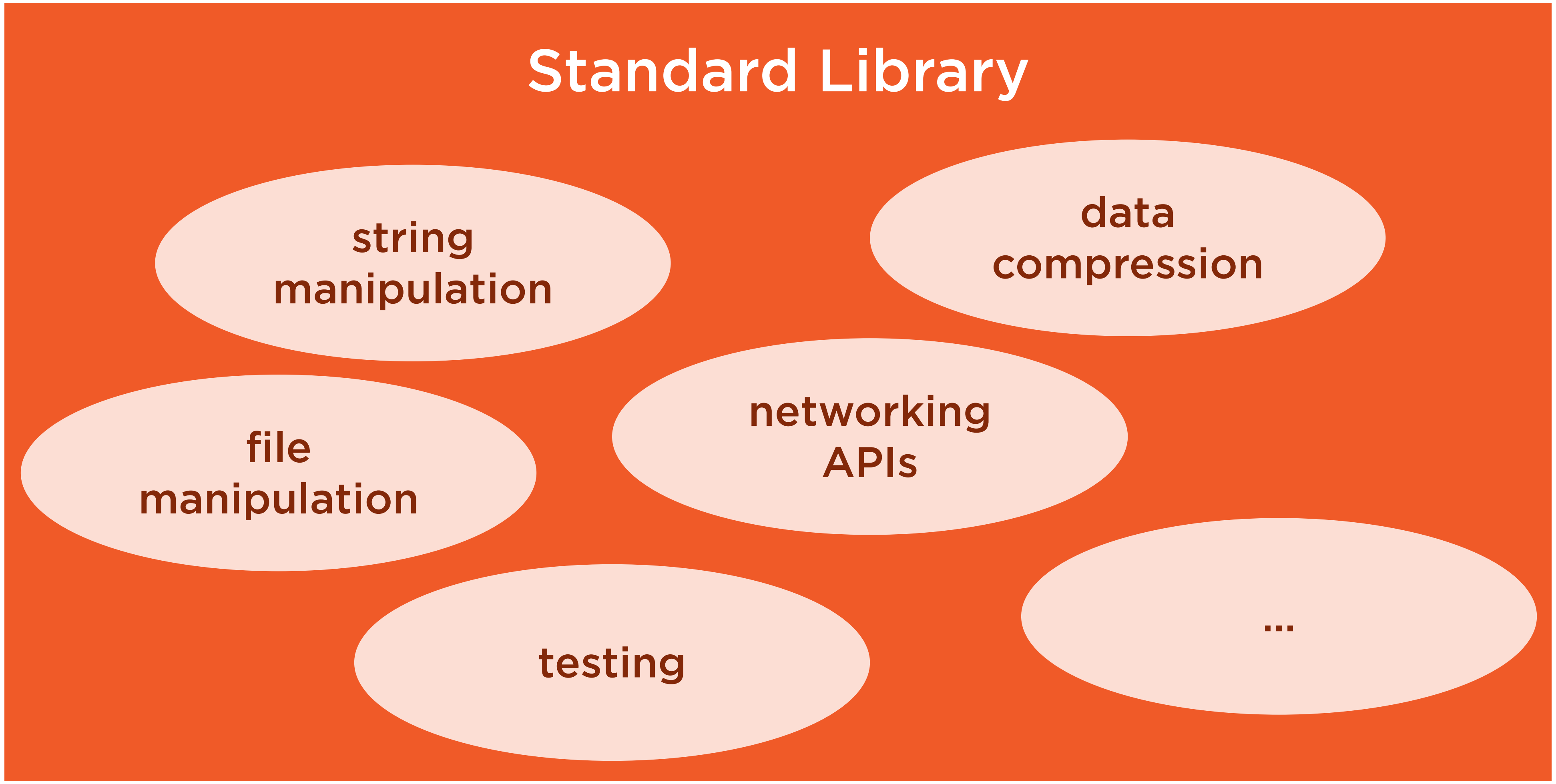
goroutines

Start thousands of concurrent tasks with minimal resources

channels

Safely communicate between concurrent tasks

Out-of-the-box Experience

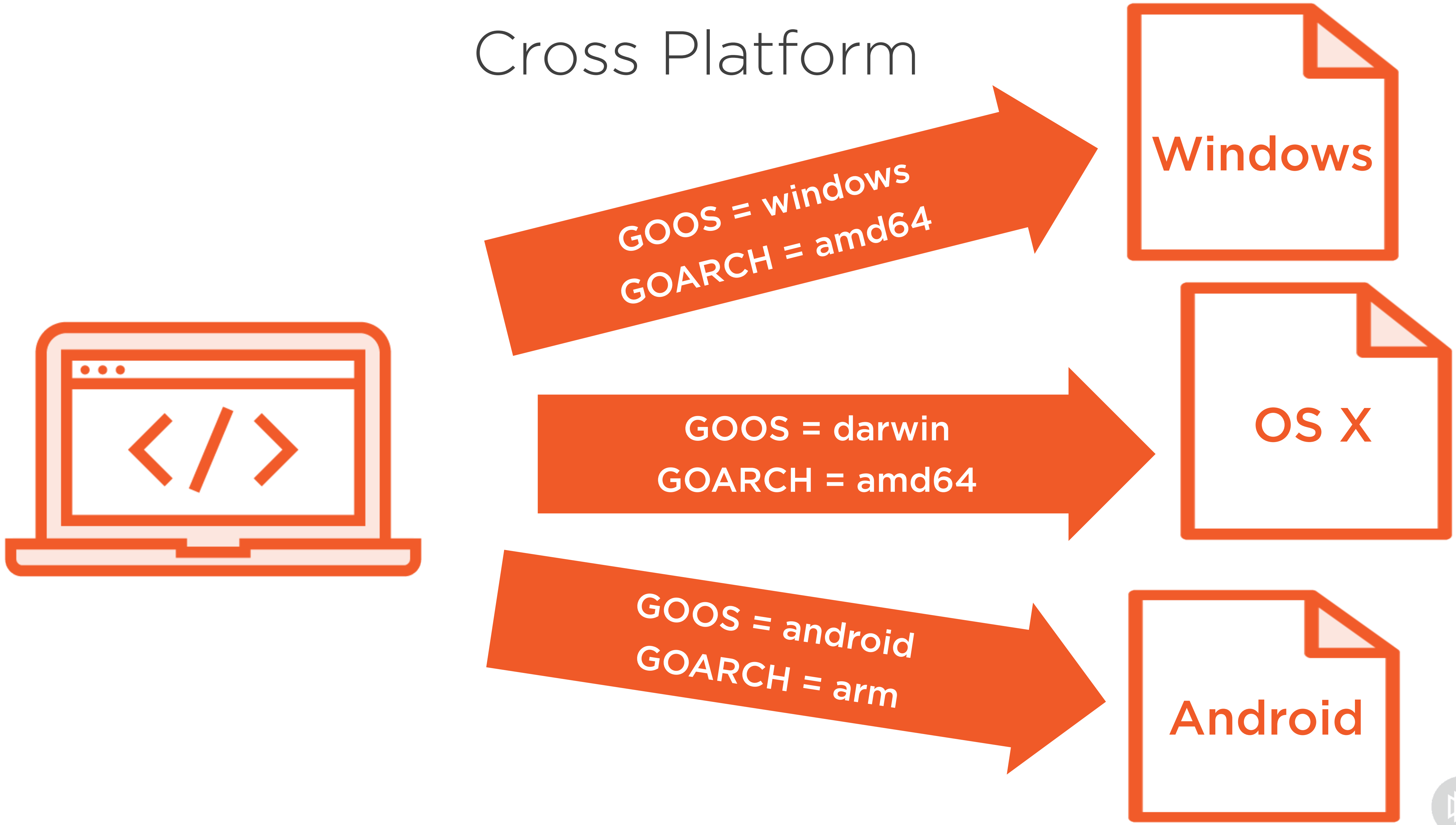


Out-of-the-box Experience

Go CLI

- project initialization
- build
- code generation
- retrieve dependencies
- test
- profiling
- documentation
- report language bugs

Cross Platform



Backward Compatibility

"It is intended that programs written to the Go 1 specification will continue to compile and run correctly, unchanged, over the lifetime of that specification."

https://golang.org/doc/go1compat

Backward Compatibility Exceptions

Security	Unspecified behavior
Spec errors	Bugs

Primary Use Cases

Web services

Web applications

DevOps

GUI / Thick-client

Machine learning

. . .

Primary Use Cases

Web services	Web applications	DevOps
GUI / Thick-client	Machine learning	. . .

Summary

Philosophy and values

- simplicity
- batteries included tooling
- backward compatibility

Primary use-cases

- web services / microservices
- devops