

Deep Dive into Go Packages

OVERVIEW OF A PACKAGE



Michael Van Sickle

@vansimke



Course Overview

Overview of a Package

Working within a Package

Preparing a Package to Be
Used

Using Other Packages



Overview



Elements of a package

- On the filesystem
- In a source file

Types of packages



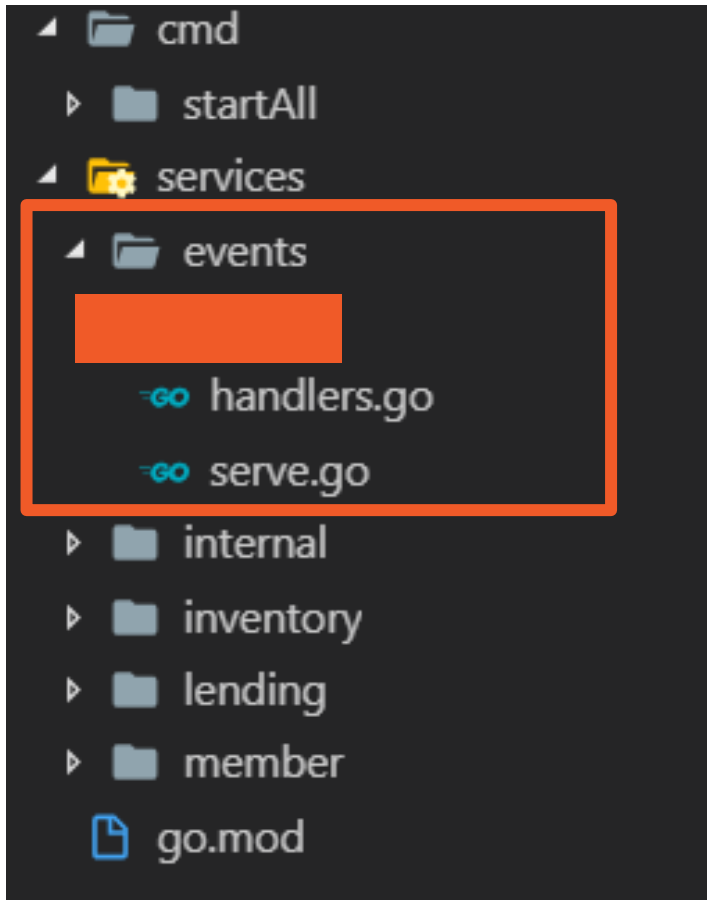
Packages

A package ... is constructed from one or more source files that together declare constants, types, variables and functions belonging to the package and which are accessible in all files of the same package.

<https://golang.org/ref/spec#Packages>

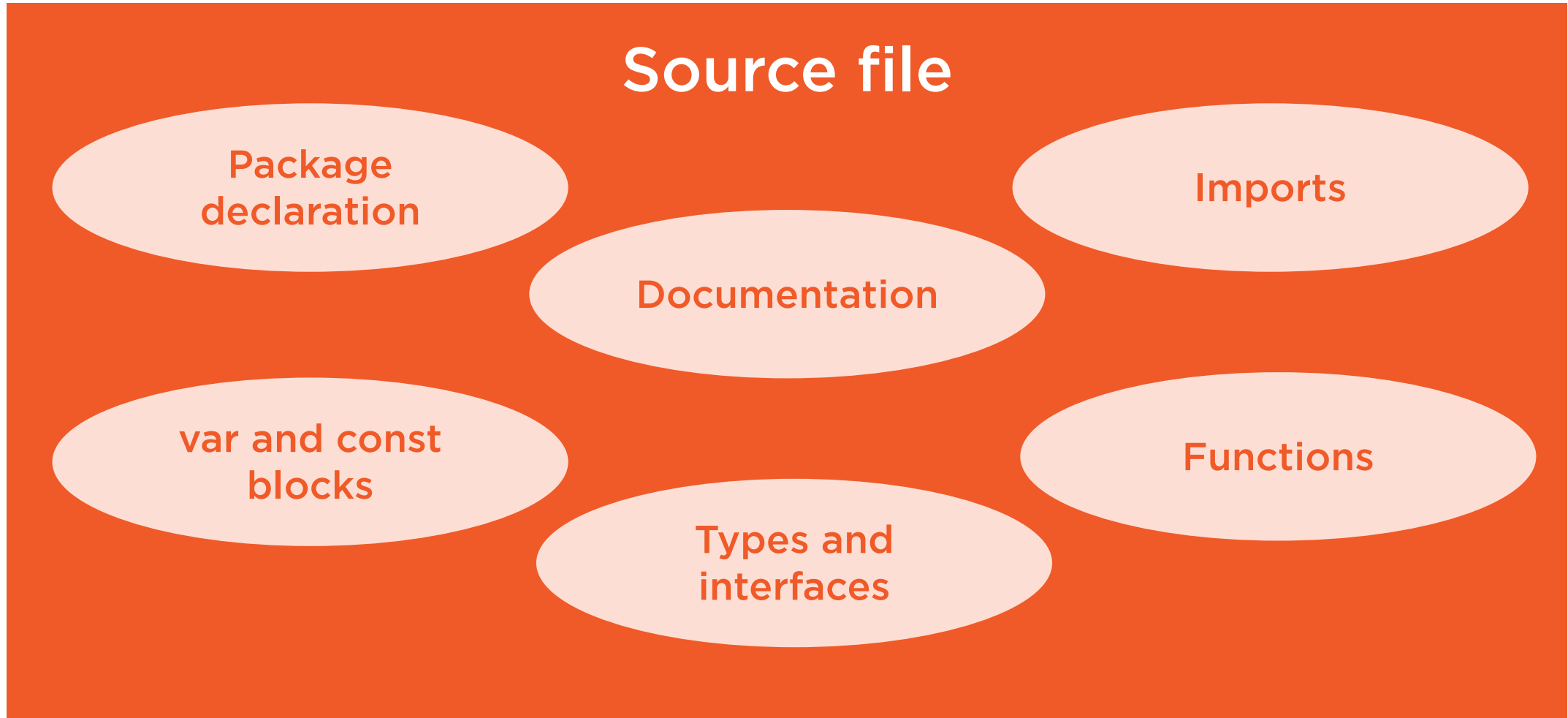


Packages Elements on the Filesystem



All Go source files in a single directory
- Subdirectories are excluded

Package Elements



Library Packages

Consumed by another package

package services

...



Library Packages

Consumed by another package

Name must match directory name

// .../lm/**services**

package services

...



Library Packages

```
// .../lm/services  
package services
```

...

Consumed by another package

Name must match directory name

Should provide a focused set of
related features



Main Packages

Application entry point

Contains a main() function

```
package main
```

```
func main() {
```

```
    ...
```

```
}
```



Main Packages

```
// ../lm/cmd/launchall
```

```
package main
```

```
func main() {
```

```
    ...
```

```
}
```

Application entry point

Contains a main() function

Can be in any directory



Main Packages

```
// ../lm/cmd/launchall
```

```
package main
```

```
func main() {
```

```
    ...
```

```
}
```

Application entry point

Contains a main() function

Can be in any directory

Focus on app setup and
initialization



Summary



Elements of a package

- On the filesystem
- In a source file

Types of packages

- library
- main