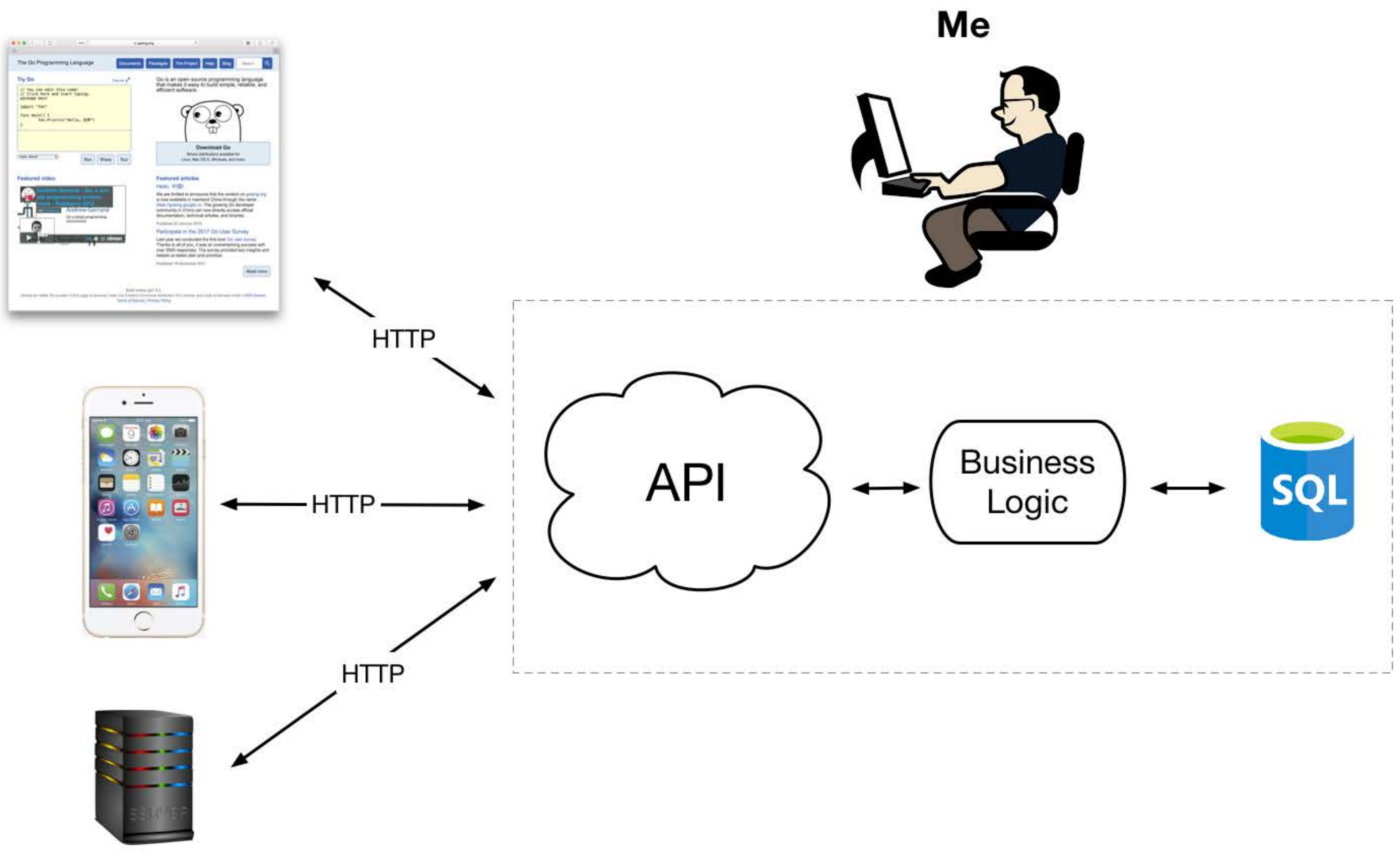
RESTFUL APIS IN GO

API Conference London 2018

Ralf Wirdemann


Me
HTTP
HTTP
HTTP
API
Business Logic
SQL

GO 101

STATICALLY TYPED

```
var i int

s := "Hallo"        // string
f := 3.142          // float

type struct Product {
    Id   int
    Name string
}

p := Product{1, "Schuhe"}
foo(p)
```

SMALL: ONLY 25 KEYWORDS

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

LOOK, I'M FUNCTIONAL

```go
func bar(x int) bool {
    return x == 42
}

func foo(f func (x int) bool) bool {
    return f(42)
}

func main() {
    foo(bar)
}
```

WELL, I'M OO TOO

```
type Figure interface {
    draw()
}

type Rectangle struct {
    size   int
    border int
}

func (this Rectangle) draw() {
}

r := Rectangle{}
r.draw()
```

CROSS PLATFORM-BINARIES

STANDARD LIBRARY

NET/HTTP

Package http provides HTTP client and server implementations.

```go
http.Handle("/foo", fooHandler)

func fooHandler(w http.ResponseWriter, r *http.Request) {
    fmt.Fprintf(w, "Hey, %q", html.EscapeString(r.URL.Path))
}

log.Fatal(http.ListenAndServe(":8080", nil))
```

ENCODING/JSON

Package json implements encoding and decoding of JSON as defined in RFC 4627.

```
type Message struct {
    Name string
    Body string
}

m := Message{"Alice", "Hello"}
b, err := json.Marshal(m)
b == []byte(`{"Name":"Alice","Body":"Hello"}`)

var n Message
err := json.Unmarshal(b, &n)
n == Message{Name: "Alice", Body: "Hello"}
```

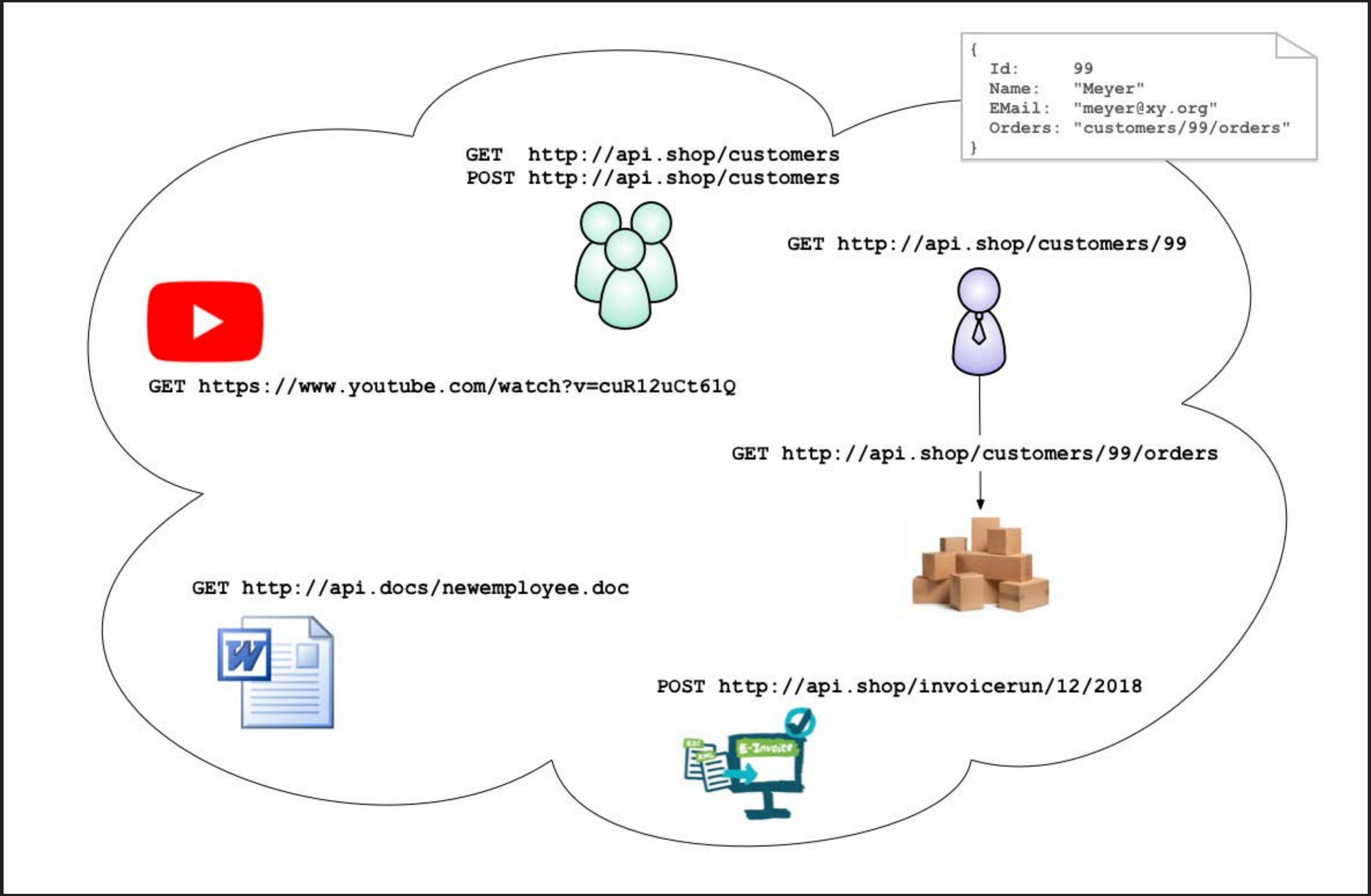
TESTING

Package testing provides support for automated testing of Go packages. It is intended to be used in concert with the “go test” command, which automates execution of any function of the form.

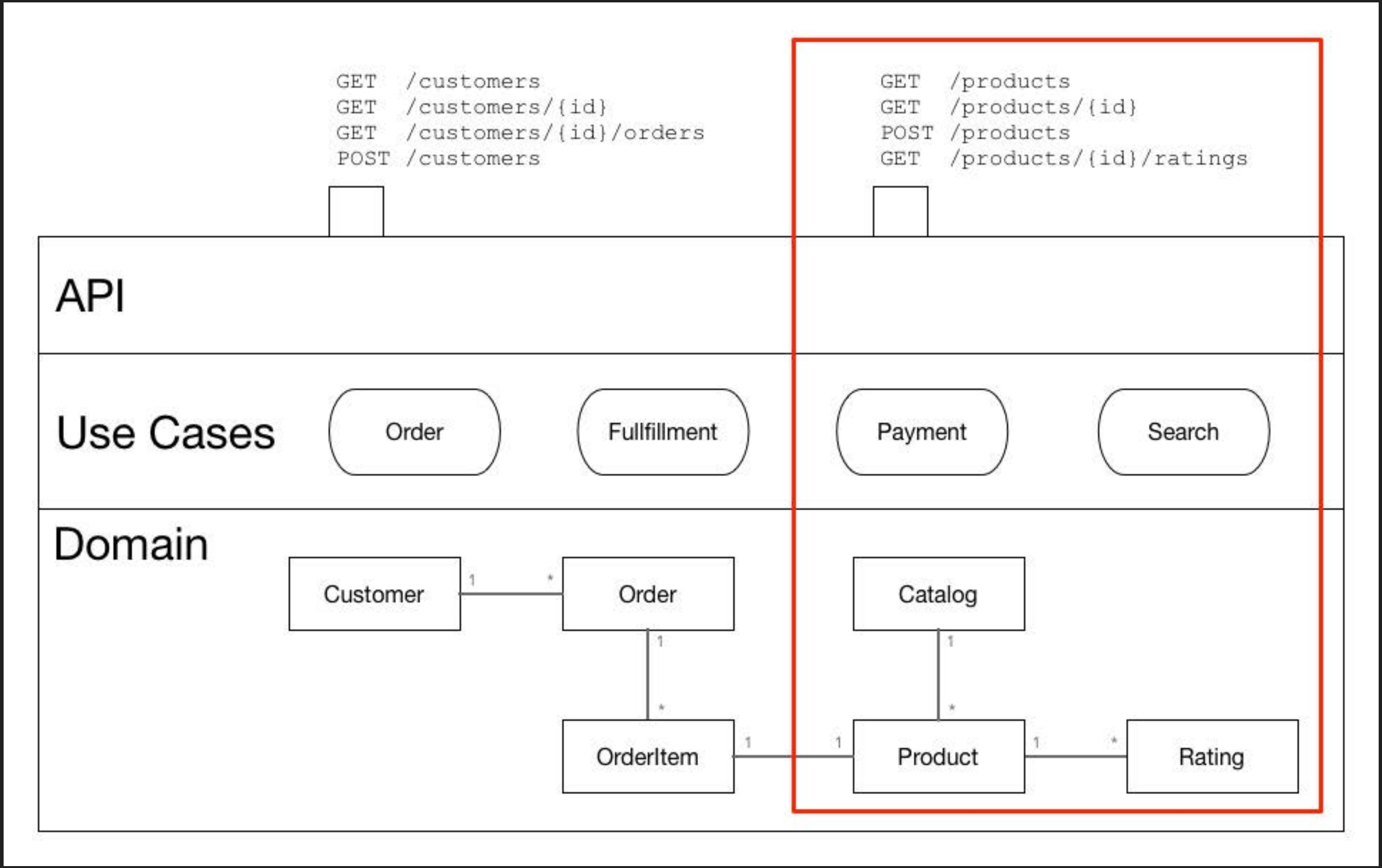
```
func TestAverage(t *testing.T) {
    v := Average([]float64{1,2})
    if v != 1.5 {
        t.Error("Expected 1.5, got ", v)
    }
}

$ go test
PASS
ok      rest-book/chapter11/math      0.032s
```

REST 101


GET http://api.shop/customers
POST http://api.shop/customers
GET http://api.shop/customers/99
GET https://www.youtube.com/watch?v=cuR12uCt61Q
GET http://api.shop/customers/99/orders
GET http://api.docs/newemployee.doc
POST http://api.shop/invoicerun/12/2018
E-Invoice

```
{
  Id:      99
  Name:    "Meyer"
  EMail:   "meyer@xy.org"
  Orders:  "customers/99/orders"
}
```


GET /customers
GET /customers/{id}
GET /customers/{id}/orders
POST /customers
GET /products
GET /products/{id}
POST /products
GET /products/{id}/ratings
API
Use Cases
Order
Fullfillment
Payment
Search
Domain
Customer
Order
OrderItem
Catalog
Product
Rating
1
*
1
*
1
1
1
*
1
*

GO PAIR PROGRAMMING

ralf.wirdemann@kommitment.biz