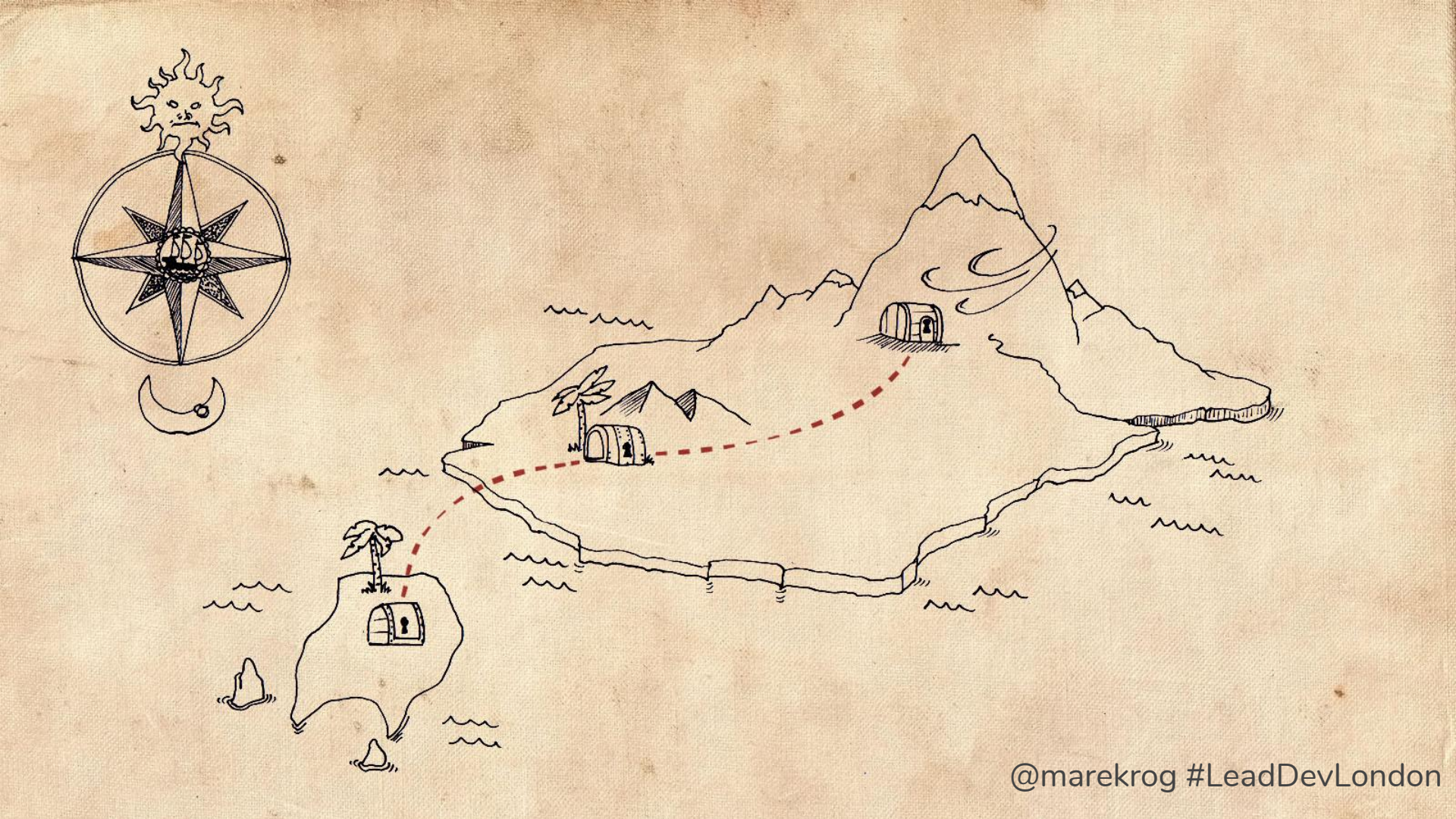
Functional programming **for everyone**

Marek Rogala
@marekrog

Appsilon
DATA SCIENCE

{# LEAD DEV}



Immutability Island

mutable

```
quantities = {
  “bread”: 2,
  “eggs”: 6
}

update(quantities,
       “eggs”, 12)
```

side effect

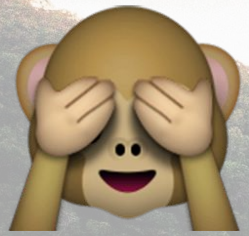
immutable

```
quantities = {
  “bread”: 2,
  “eggs”: 6
}

updated_quantities = update(
  quantities, “eggs”, 12)
```

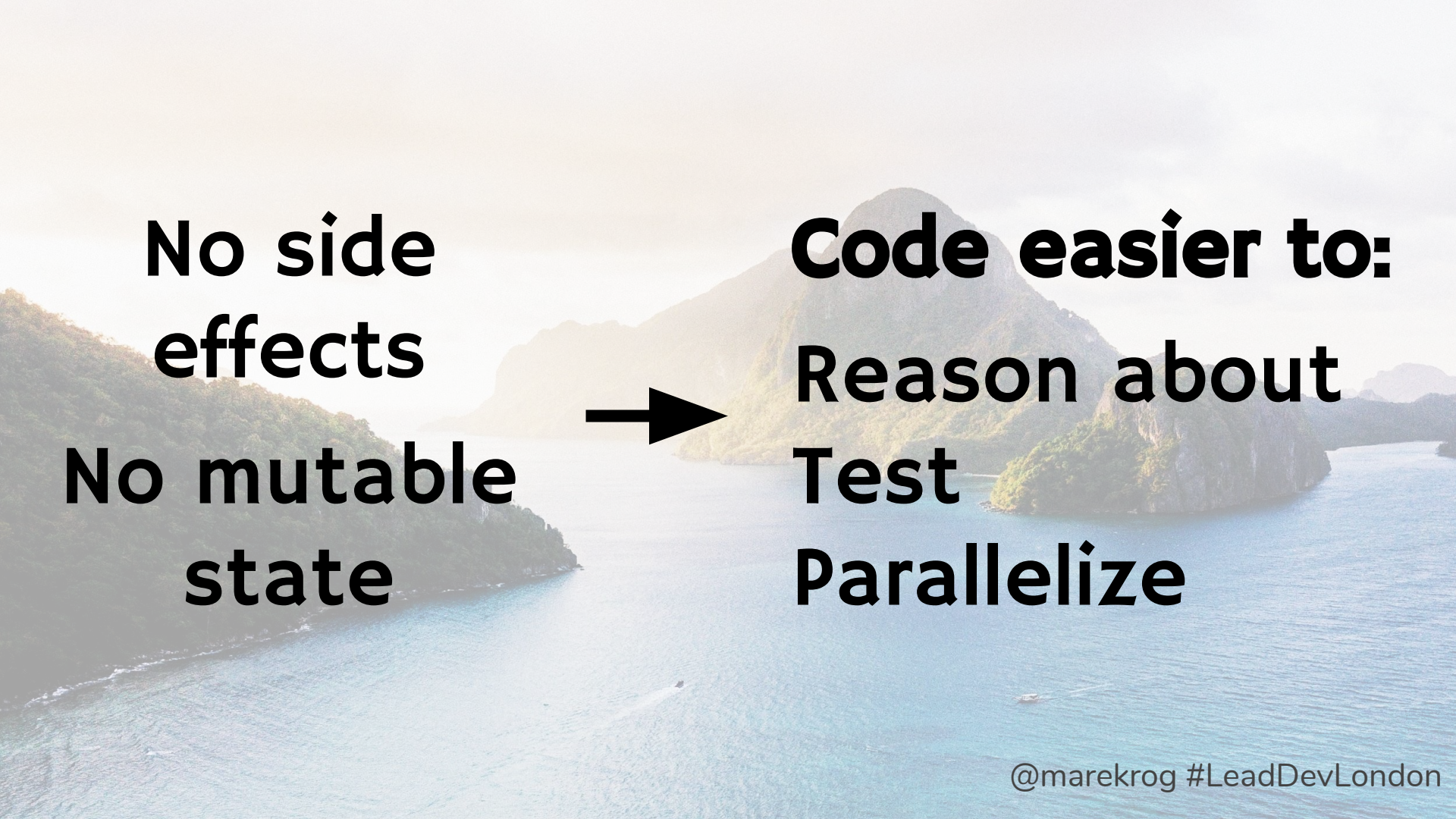
no side effects

No talking to outside world



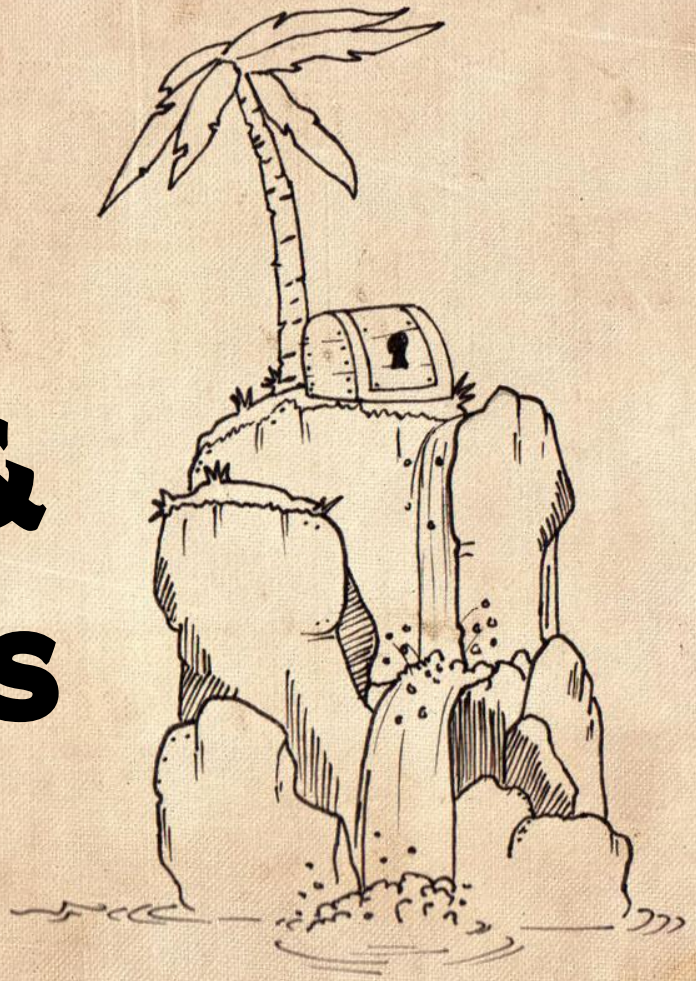
No hidden inputs

No surprises


No side effects
No mutable state
Code easier to:
Reason about
Test
Parallelize

benefits of

immutability & no side effects



Declarative desert

declarative: what

```
soup = boiled(
  chopped(peeled(tomatoes)) +
  water
)
```

imperative: how

```
soup = new Soup()
tomatoes.peel()
tomatoes.chop()
soup.add(water,
  tomatoes)
soup.boil()
```

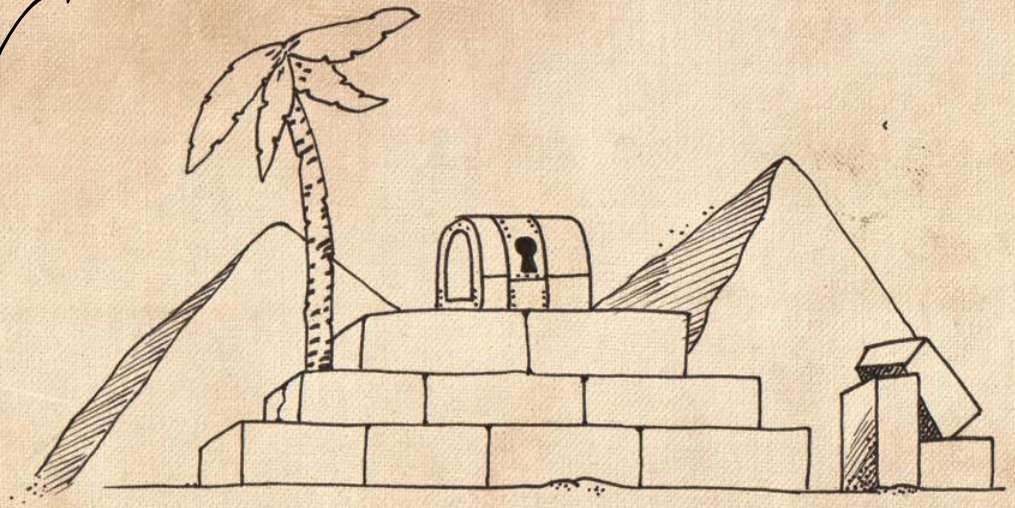
Declarative programming

Code easier to:

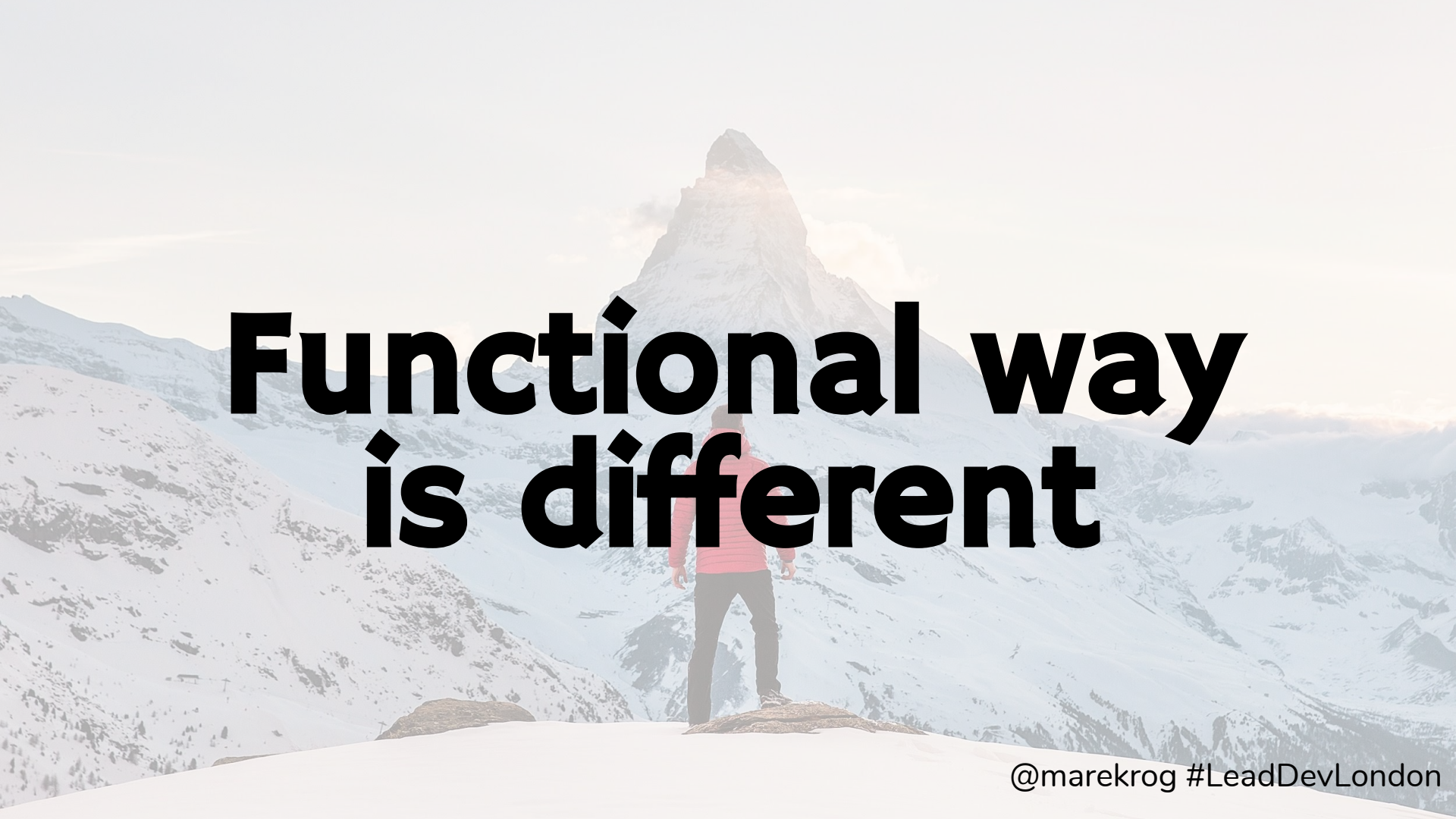
Understand
Optimize
Reuse

design around

what

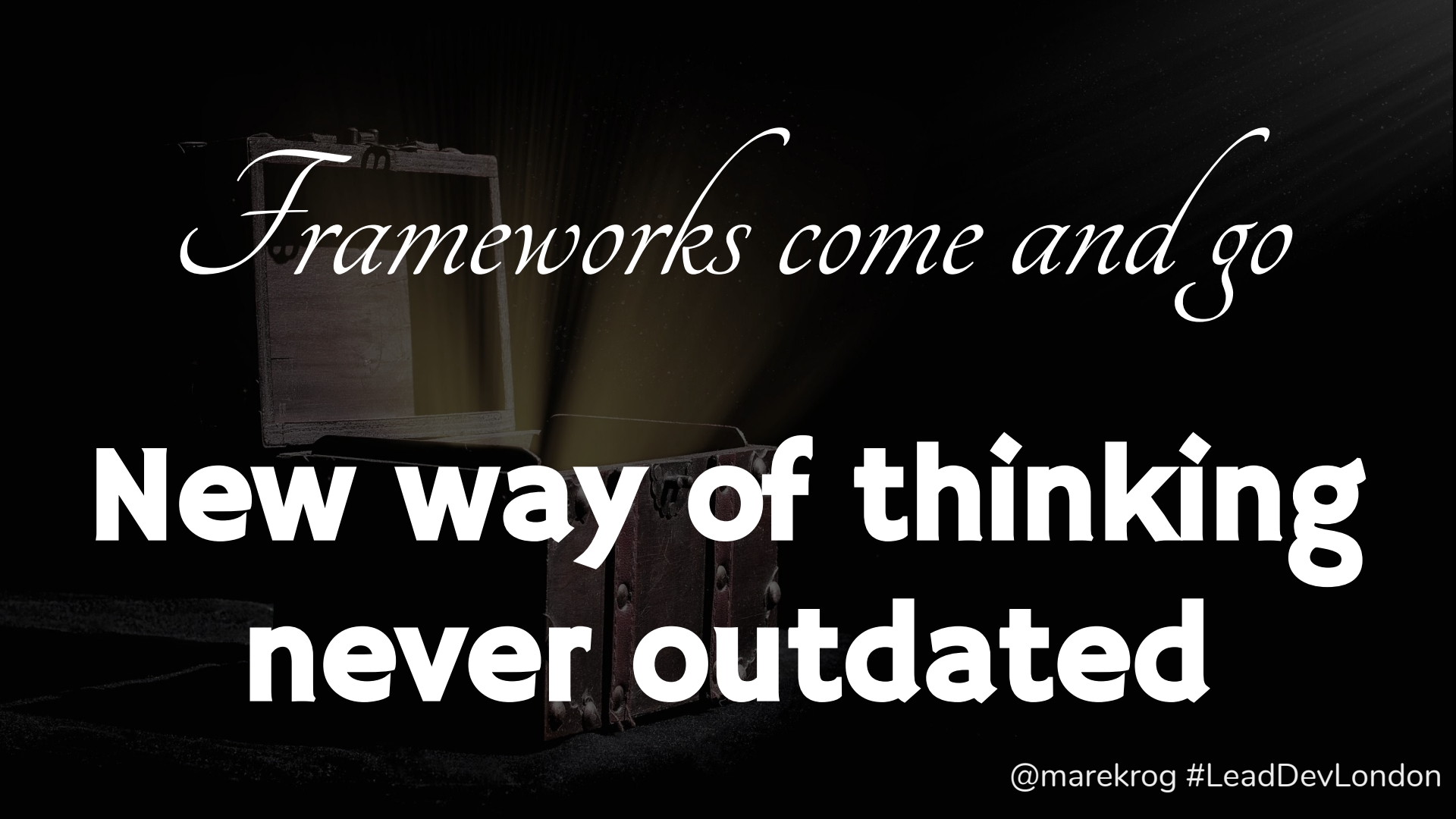


Perspective Peak


Functional way
is different

new

perspective on programming


Frameworks come and go
New way of thinking never outdated
@marekrog #LeadDevLondon

@marekrog #LeadDevLondon

Idea & space

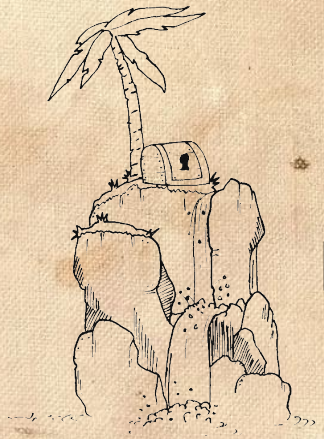
Talk relevant gains

Set expectations

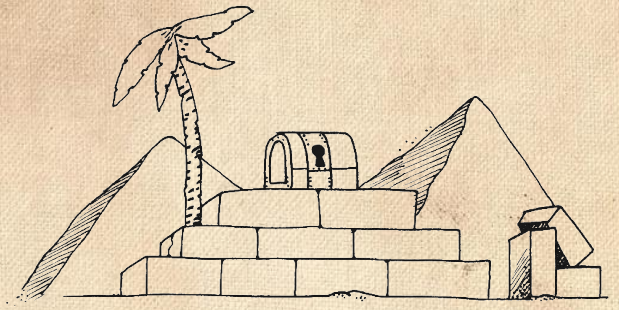
Share resources

1. Why functional programming matters (talk + paper)
2. Functional programming. Simplified (book)
3. So you want to be a functional programmer (articles)
4. Learn you a haskell (online book)
5. Functional Programming in Scala (MOOC)
6. Community (Twitter, Gitter channels)
7. Contribute to friendly Open Source projects, e.g. Scalaz

@marekrog #LeadDevLondon



immutability &
no side effects



designing around *what*

new way of thinking