Tracing as a Service

stephan.brandauer@it.uu.se

Problem:

We have no idea what we’re doing

Process:
Invent a Runtime Optimisation

(Idealised)

Process: Invent a Runtime Optimisation

(Idealised)

1. Find a common operation in programs.

Process: Invent a Runtime Optimisation

(Idealised)

1. Find a common operation in programs.
2. Implement a program transformation to speed up the operation.

Process:
Invent a Runtime Optimisation

(Idealised)

1. Find a common operation in programs.
2. Implement a program transformation to speed up the operation.
3. Measure the performance improvement.

4. PLDI, here I come!

4. PLDI, here I come!

(It’s similar for
language abstractions)

- In reality, it’s really hard to find ‘**common**’ features of ‘**typical**’ programs.
- A better process might be to convince yourself that there’s a problem before fixing it.

Enter: Spencer

I am an **online service**.

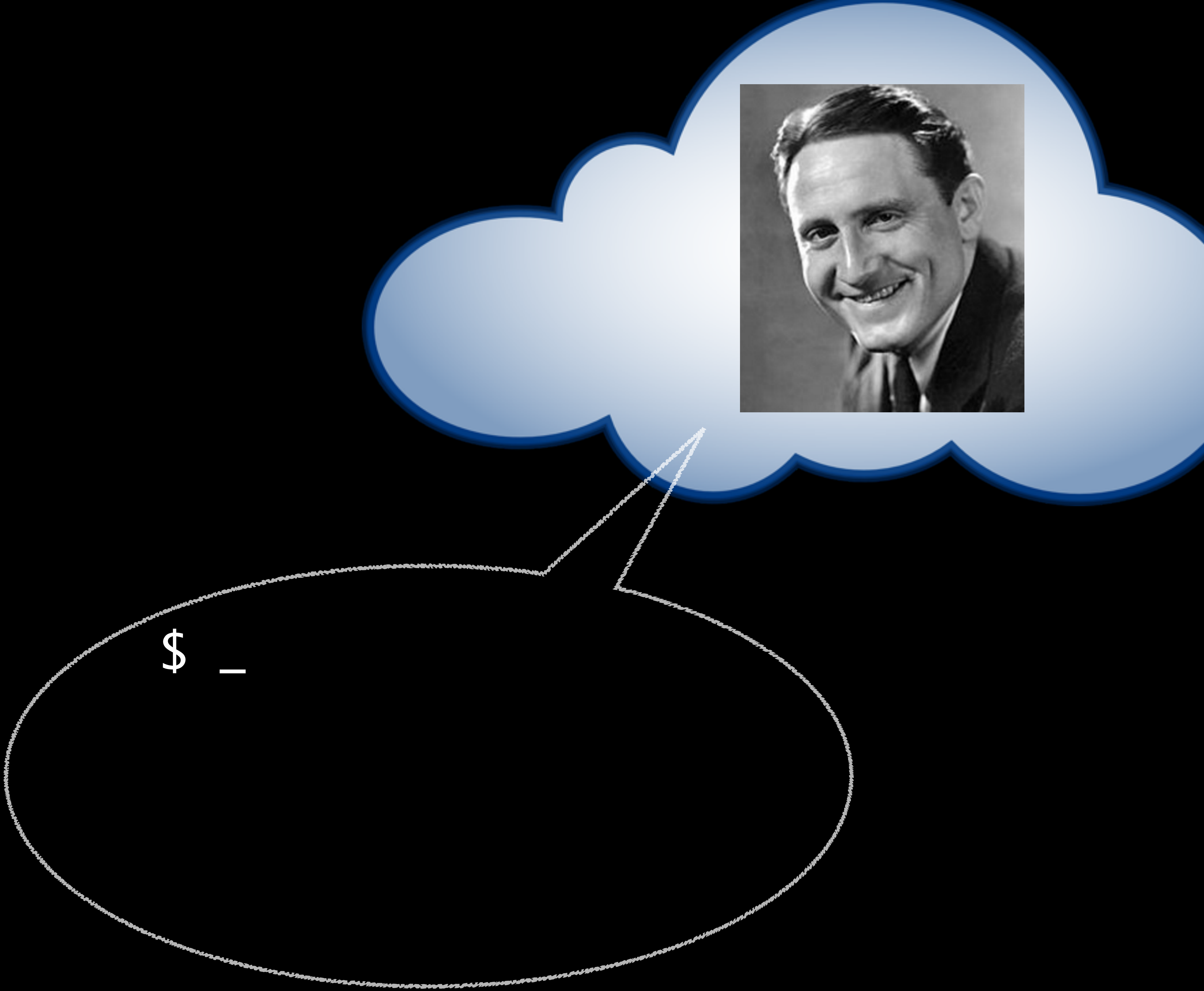
I analyse traces of a **wide range of programs** running on the JVM.

These traces can be hundreds of gigabytes large.
But these traces are pre-recorded.
You don't need a large hard drive, or set up a tool chain.

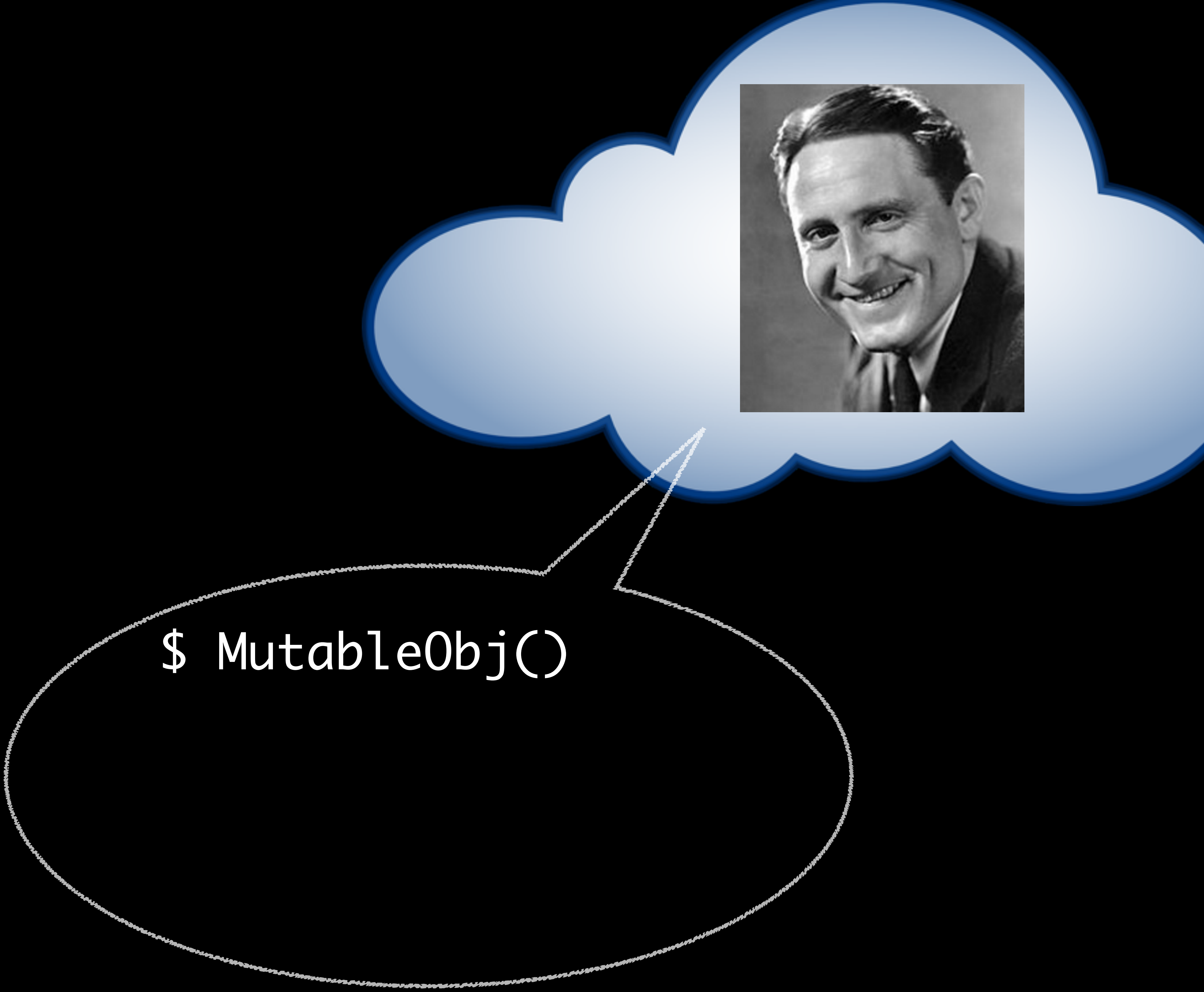
I accept **queries as expressions written in a DSL** that uses some primitive queries, and combinators to combine them into more interesting questions.

I use **caching to avoid recomputation** of old questions.
Go on, ask me a question!

```
$ _
```



```
$ _
```



```
$ MutableObj()
```

```
$ MutableObj()
computing..
```

```
$ MutableObj()
computing..
Objects {1, 17, 23, ..}
are mutable!
```

I cached this. Ask me again!

```
$ _
```

```
$ _
```

```
$ MutableObj()
```

```
$ MutableObj()
computing..
Objects {1, 17, 23, ..}
are mutable!
```

Spencer’s DSL

Query	Meaning

Spencer’s DSL

Query	Meaning
MutableObj()	Set of all objects that were mutated.

Spencer's DSL

Query	Meaning
MutableObj()	Set of all objects that were mutated.
Not(MutableObj()) == ImmutableObj()	Objects that were not mutated.

Spencer's DSL

Query	Meaning
MutableObj()	Set of all objects that were mutated.
Not(MutableObj()) == ImmutableObj()	Objects that were not mutated.
Deeply(ImmutableObj())	Immutable objects from which you could only reach other immutable objects.

Spencer's DSL

Query	Meaning
MutableObj()	Set of all objects that were mutated.
Not(MutableObj()) == ImmutableObj()	Objects that were not mutated.
Deeply(ImmutableObj())	Immutable objects from which you could only reach other immutable objects.
Deeply(UniqueObjs())	Tree-shaped data structures.

Spencer’s DSL

Spencer’s DSL

Spencer’s DSL

Query	Meaning
ProportionByAlloctionSite (Deeply(ImmutableObj())	For each allocation site, proportion of objects that were immutable.
ClassProperty(Deeply(ImmutableObj()))	All classes that had only deeply immutable objects.
ImmutableObj() vs UniqueObj()	Four sets: objects that were 1. immutable & unique 2. !immutable & unique 3. immutable & !unique 4. !immutable & !unique

Status

- Tracing tool implementation: nearly done
- Analysis DSL: needs some changing, but essentially useful
- Web interface: work in progress

Questions?