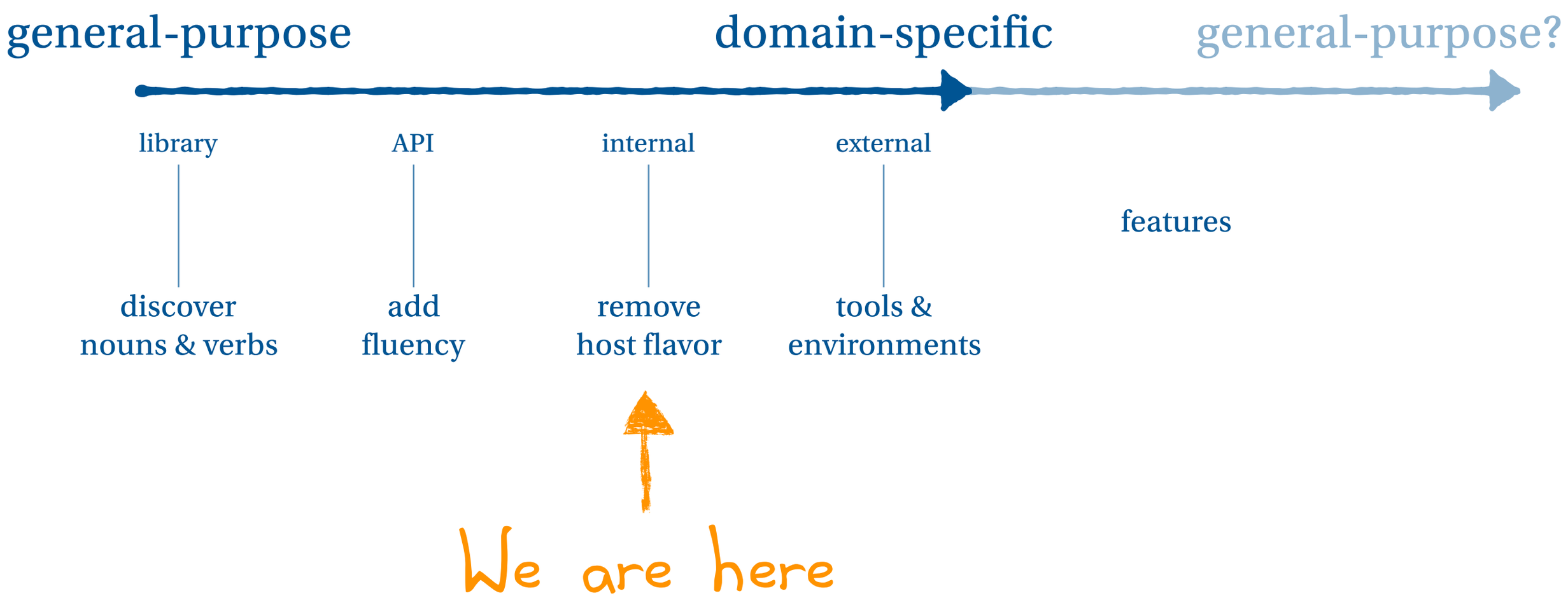
The evolution of a DSL?



Simple techniques for adding fluency

Most general-purpose languages support these features.

names

including Unicode

```
sin(0)
```

ASK: If the DSL supports Unicode, how will the user write programs?

whitespace

```
computer();
  processor();
    cores(2);
  disk();
    size(150);
```

function composition

```
computer(
  processor(
    cores(2)
  ),
  disk(
    size(150)
  )
);
```

method chaining

```
computer()
  .processor()
    .cores(2)
  .disk()
    .size(150)
  .end();
```

Techniques for hiding the host language

These features tend to be language-specific. Some languages support this ability more than others.

infix operators	`set1 union set2` `salaries map giveRaise`
pre- and postfix operators	`~1` `i++`

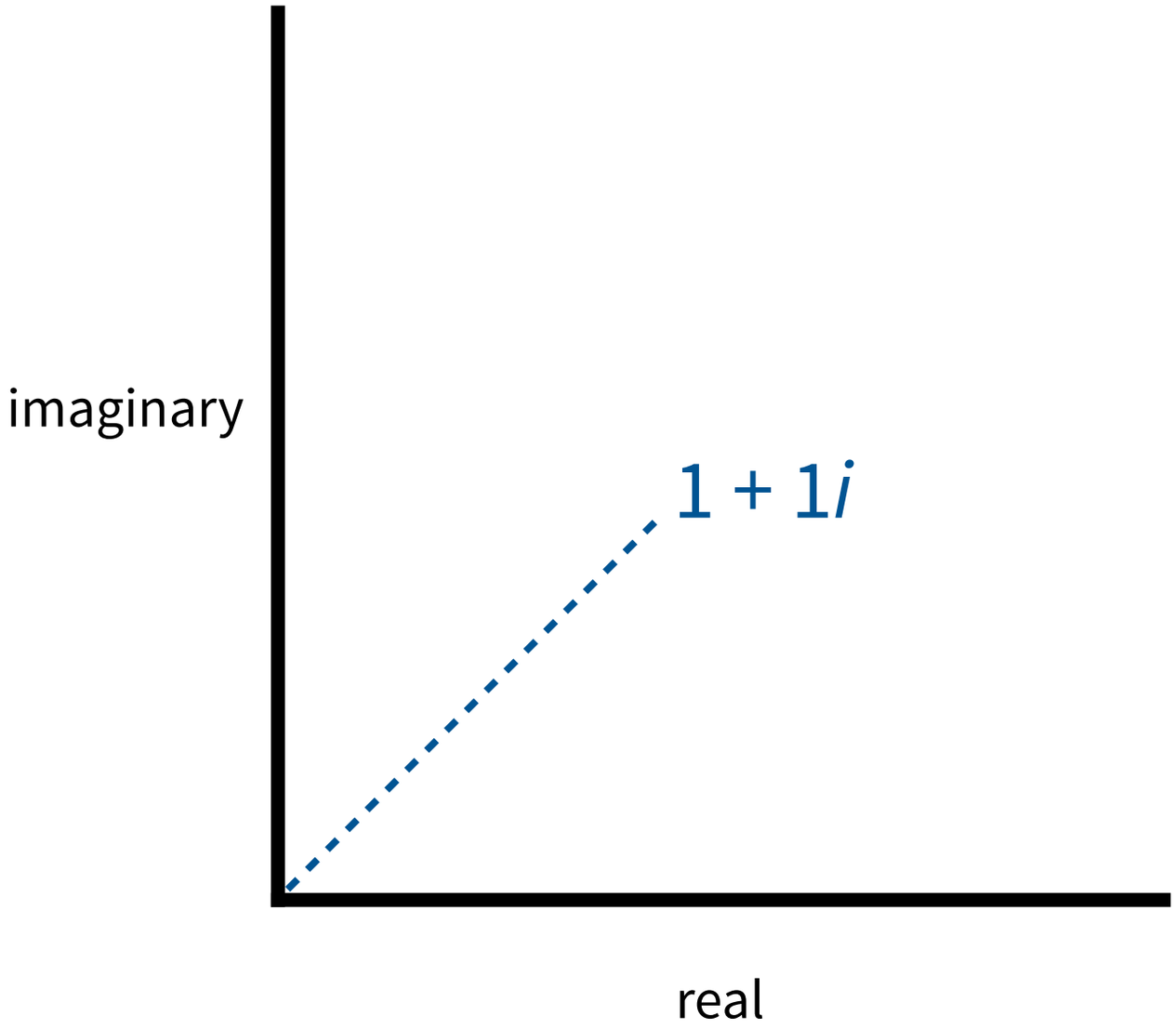
(re-)defining operators	`set1 + set2`
	`set1 ∪ set2`
	Different host languages gives us different

closures
i.e., by-name parameters in Scala
	`test("An empty Set should have size 0") {`
	`  assert(Set.empty.size == 0)`
	`}`
	Useful for defining new **control-flow structures**

literal extension	`3 little pigs`

Is this a DSL?

Complex numbers



$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

$$(a + bi) * (c + di) = (ac - bd) + (ad + bc)i$$