

The diagram illustrates the transformation of a flat table into a hierarchical tree structure through three stages, connected by large gray arrows.

Stage 1: Initial Flat Table

1	"Hello World"
---	---------------

Stage 2: Hierarchical Tree Structure

The flat table is transformed into a hierarchical tree structure. The root node is a box with two children:

- Left child: A box with two rows:

1	1
2	3
- Right child: A box with two rows:

1	"Hello"
2	"World"

Below the right child, there is a separate box representing the flattened result of the right child's subtree:

1	"HELLO"
2	"WORLD"

Stage 3: Further Transformation

The hierarchical tree structure is further transformed. The root node is a box with two children:

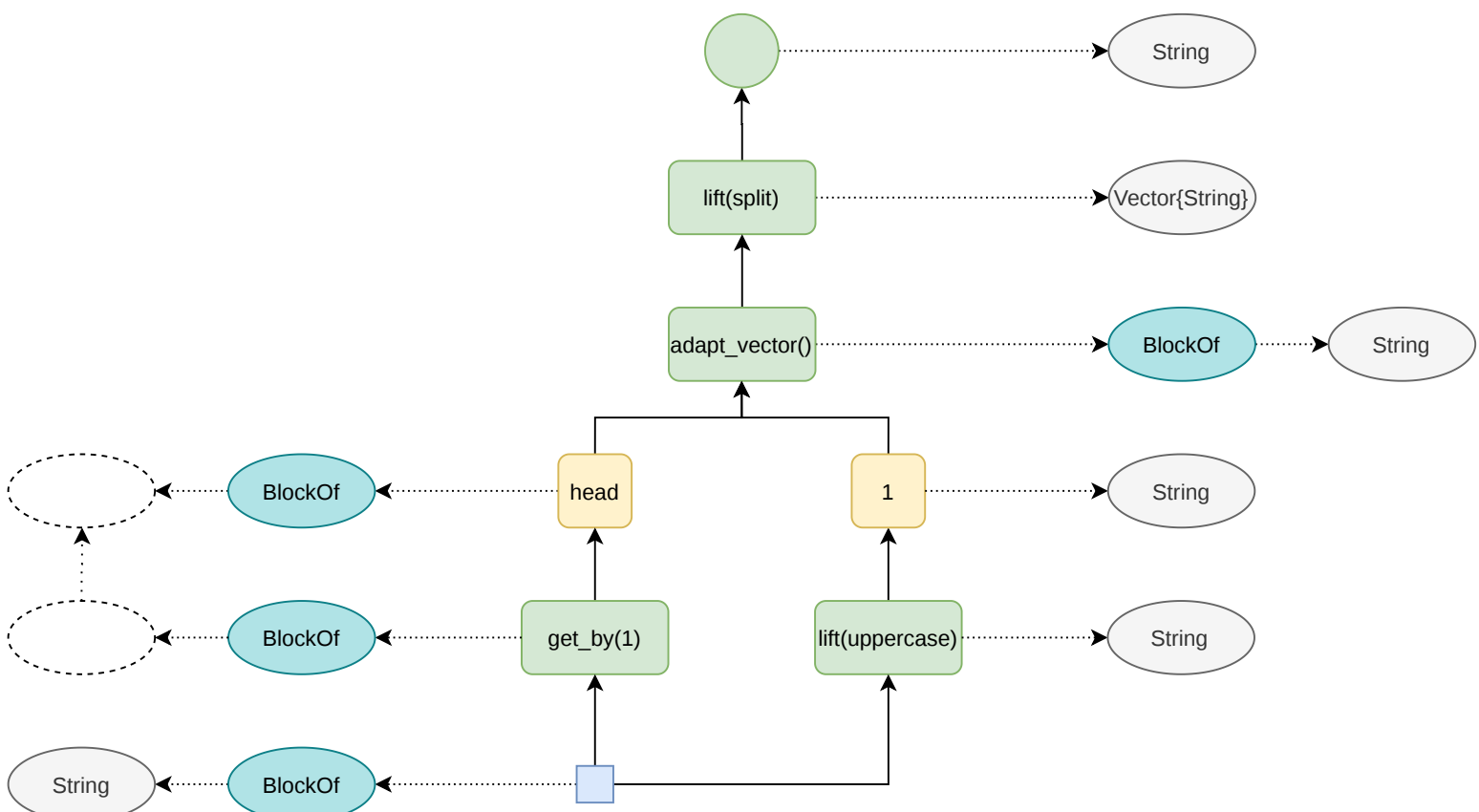
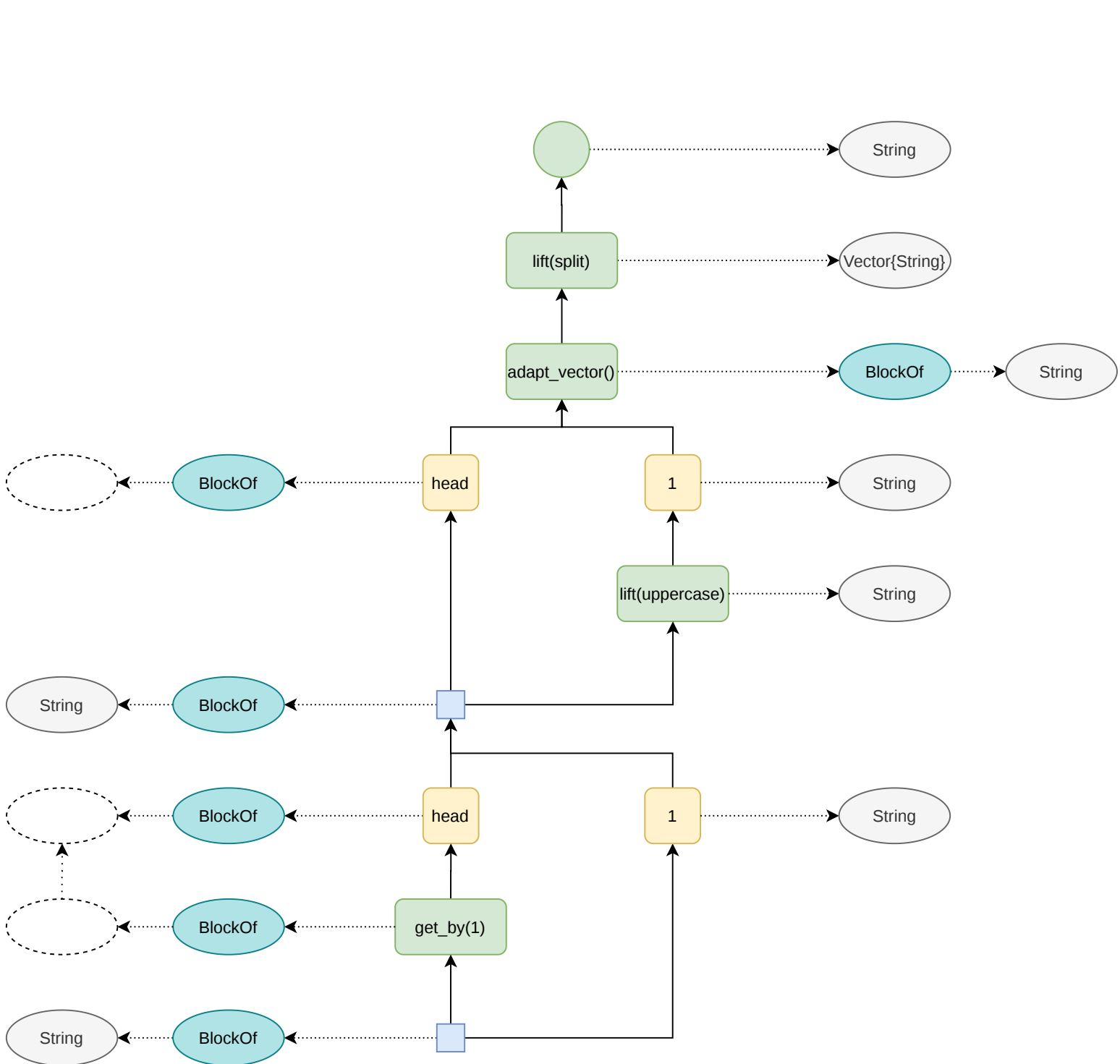
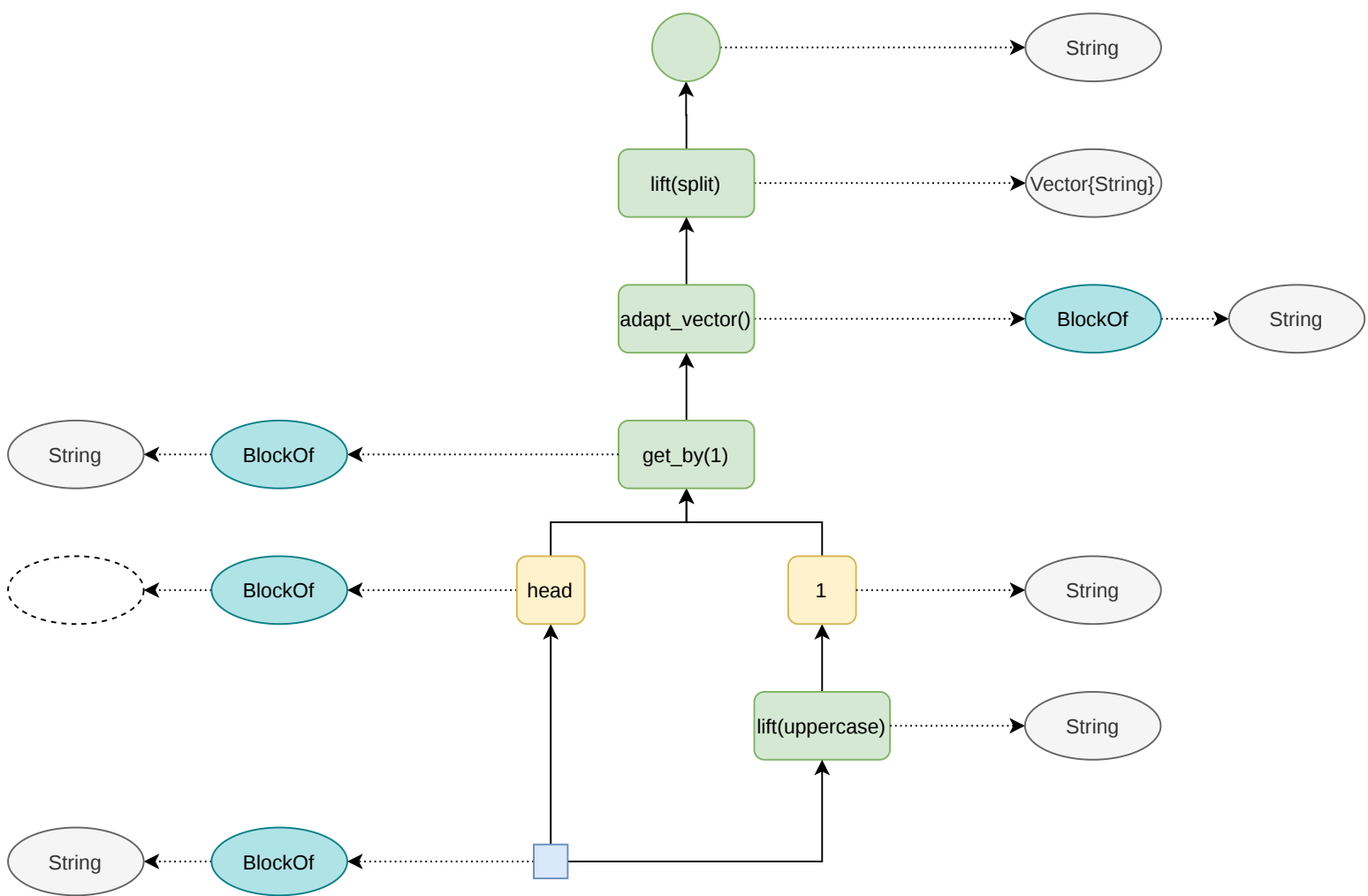
- Left child: A box with two rows:

1	1
2	2
- Right child: A box with one row:

1	"HELLO"
---	---------

Below the right child, there is a separate box representing the flattened result of the right child's subtree:

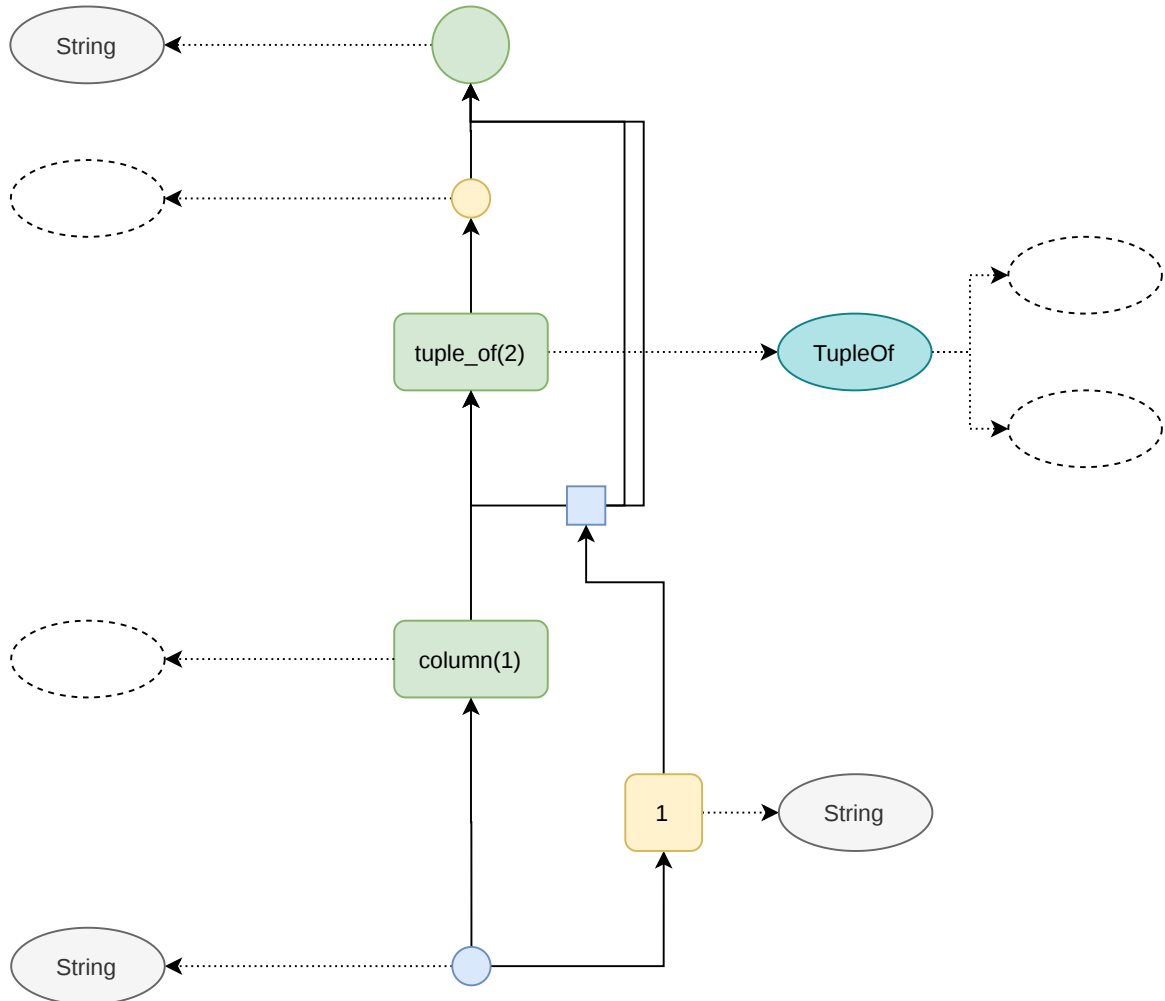
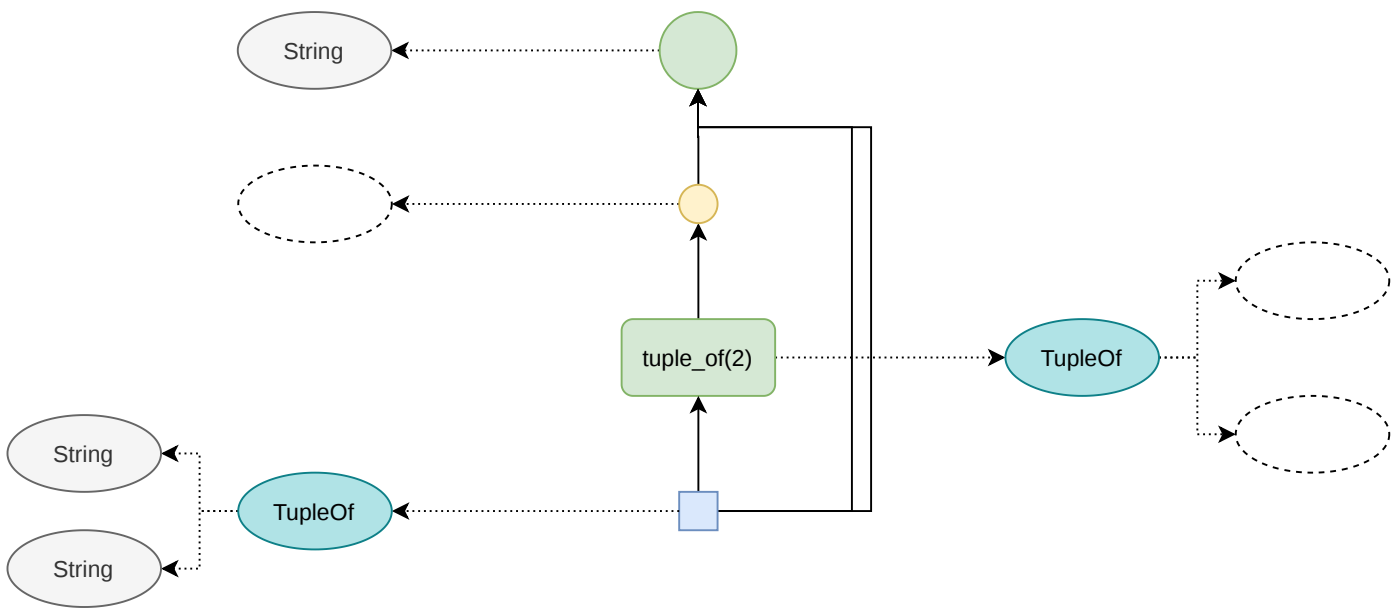
1	1
---	---



wrap()



chain_of(tuple_of(2), column(1))



sieve_by()



chain_of(wrap(), block_length())

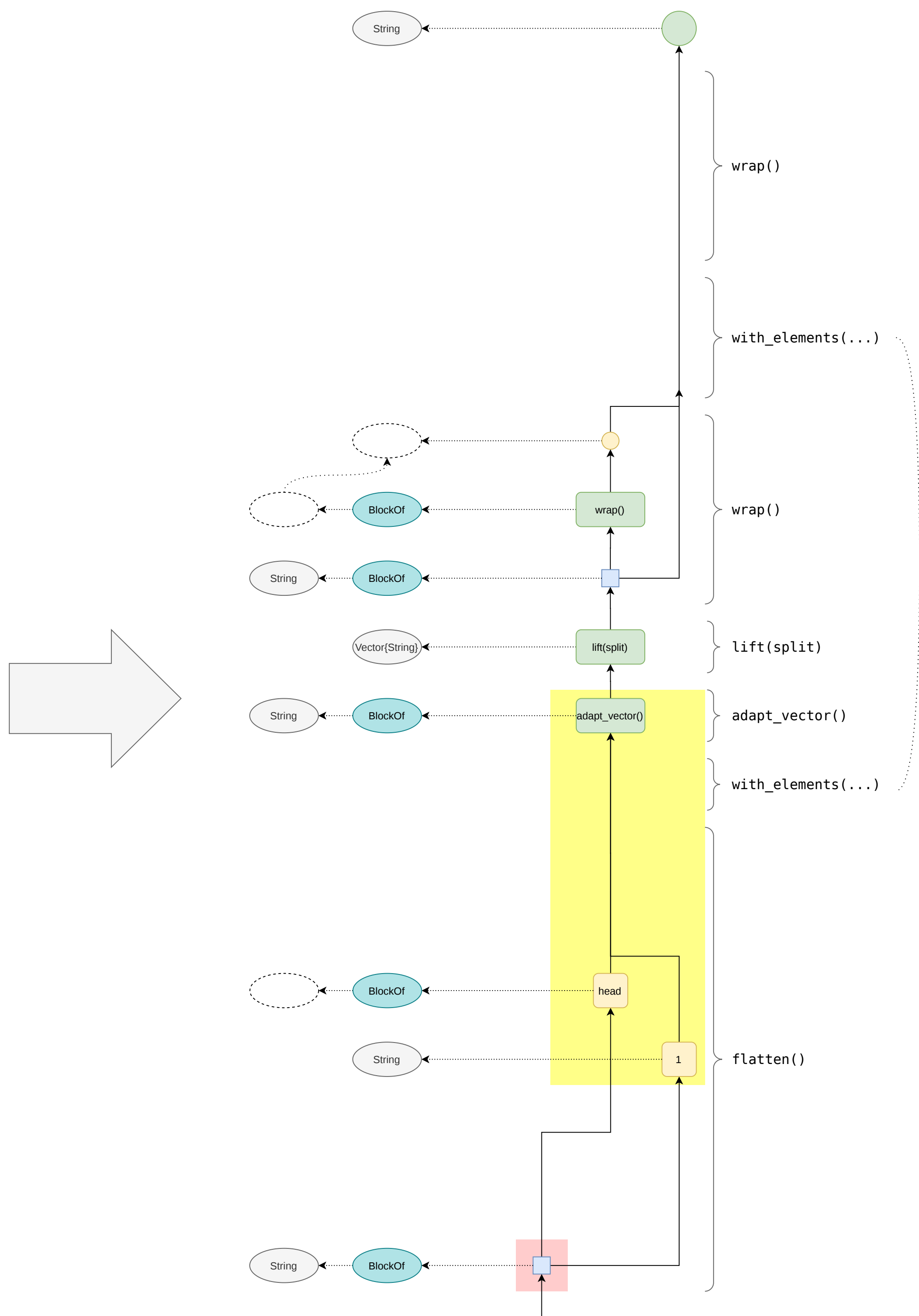
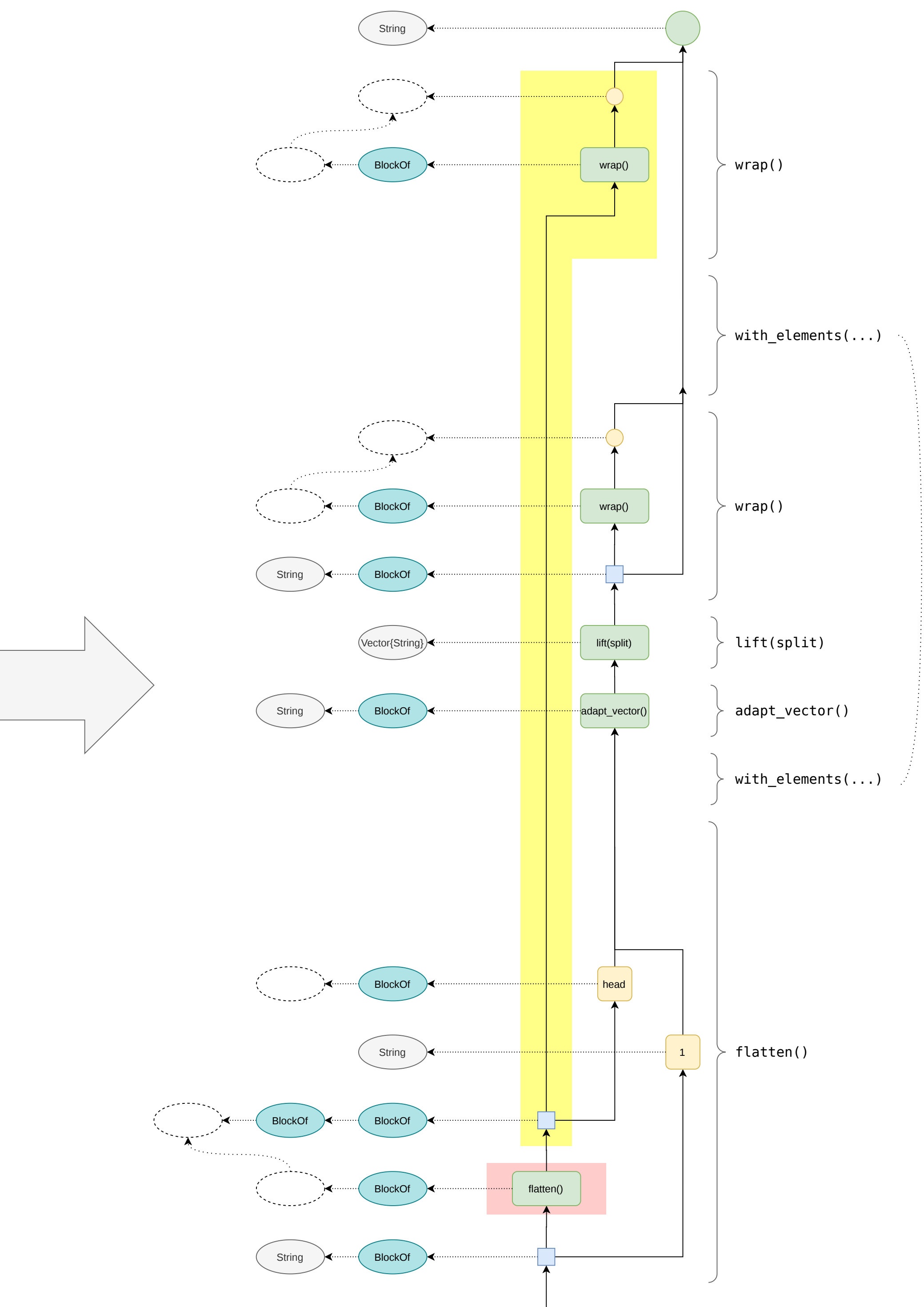


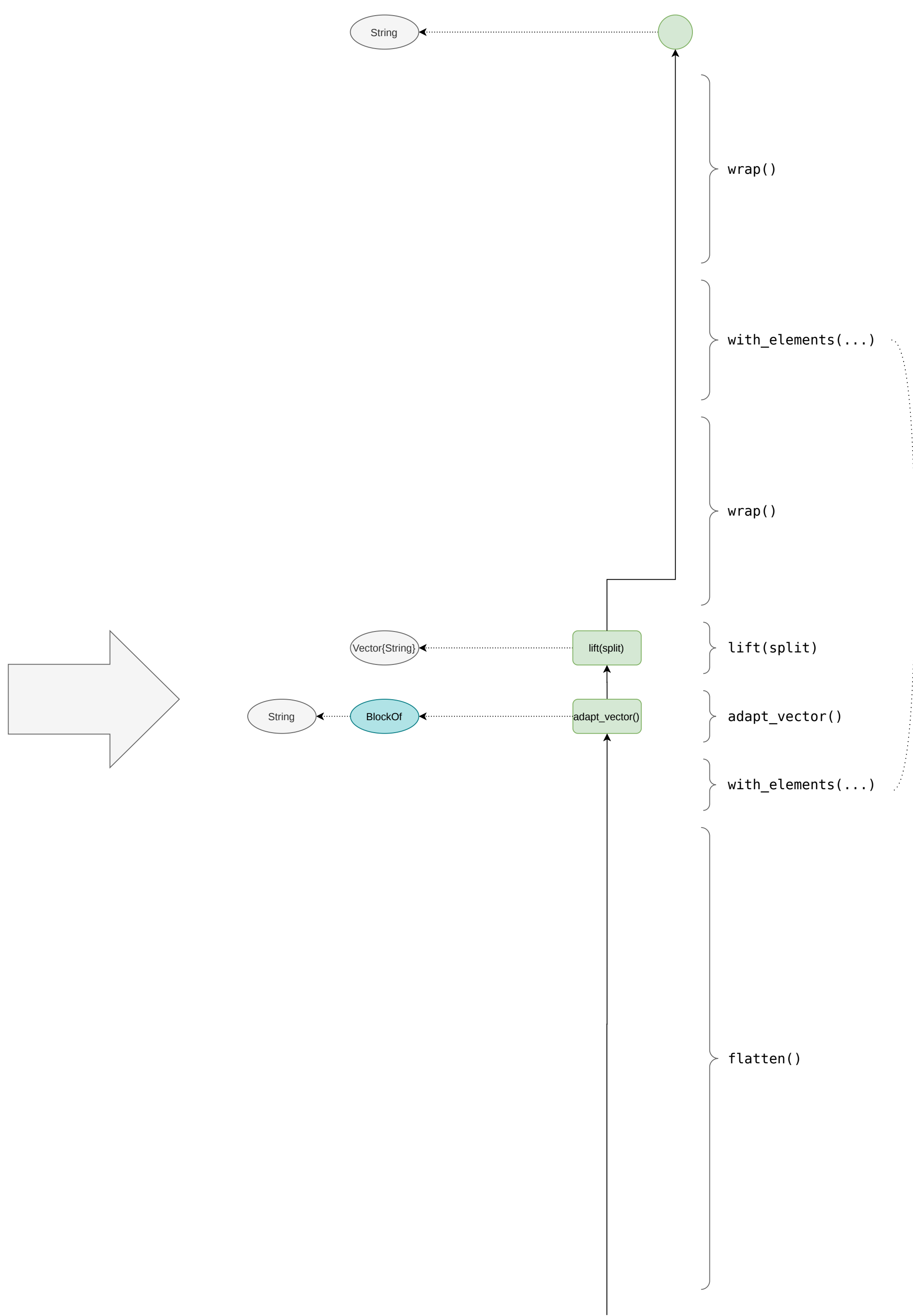
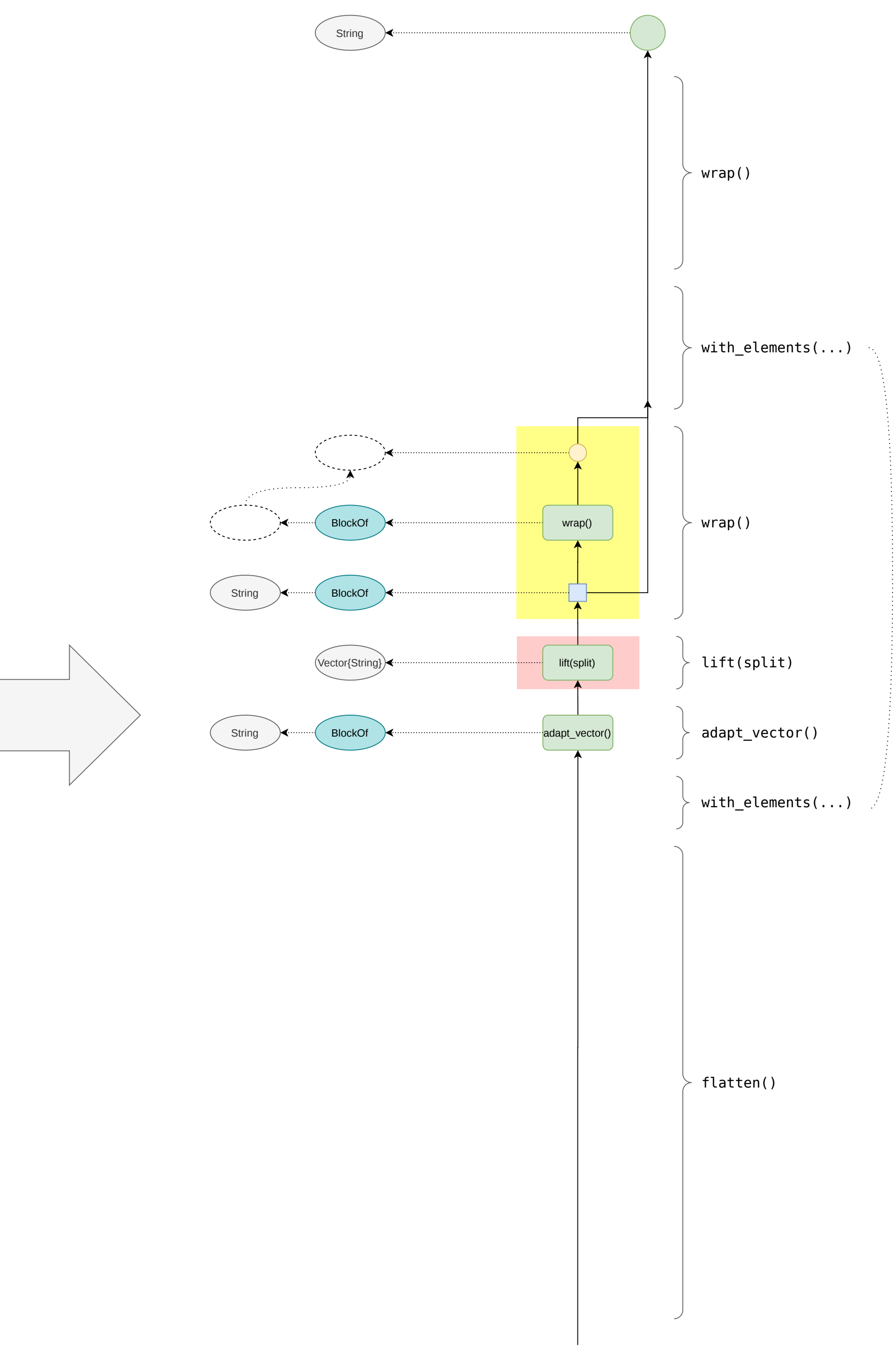
@query "Hello World" split(it)

```
chain_of(
  wrap(),
  with_elements(
    chain_of(
      wrap(),
      lift(split),
      adapt_vector()),
    flatten())
```



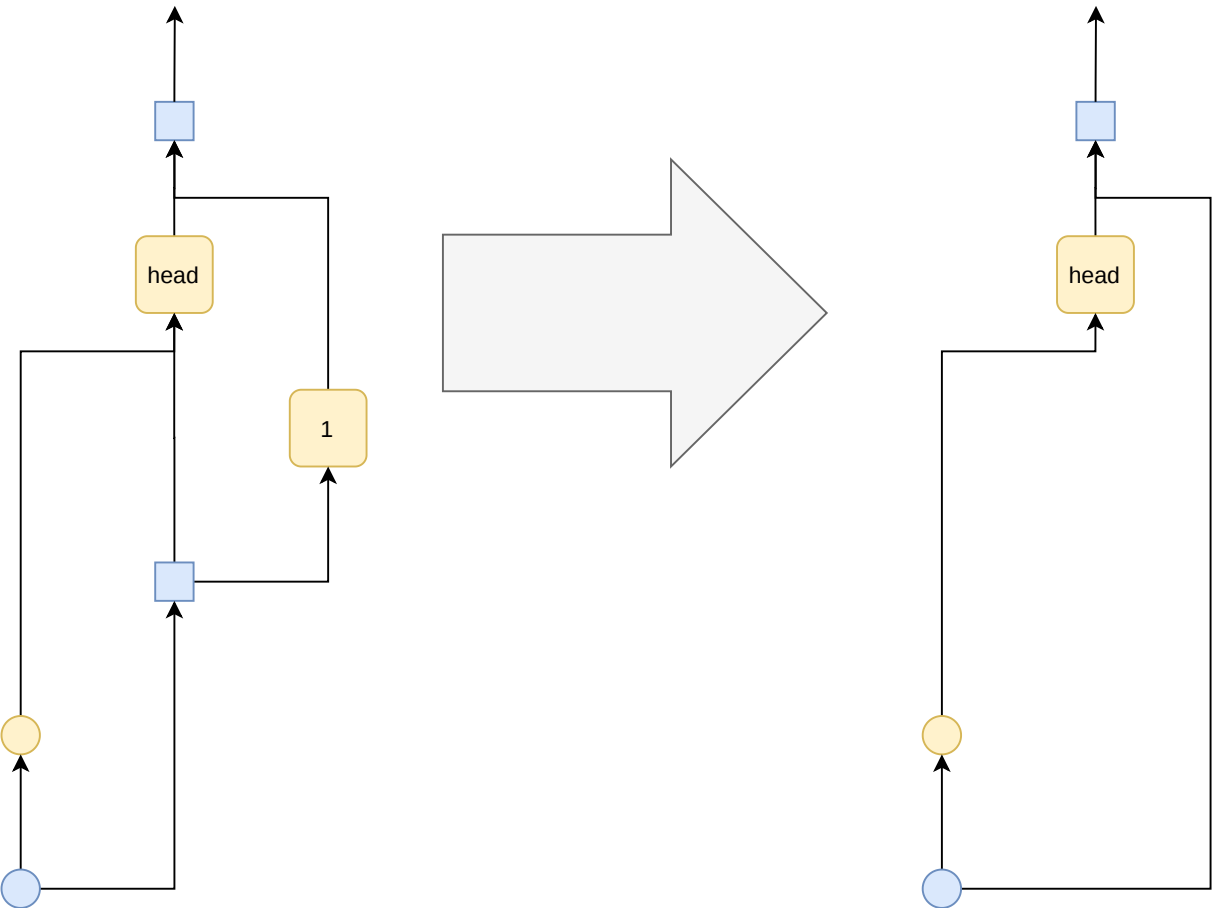
untrace(n::NodeRef, guard::NodeRef)::Tuple{Pipeline,Vector{NodeRef}}





@query "Hello World" split(it)

```
chain_of(
  wrap(),
  with_elements(
    chain_of(
      wrap(),
      lift(split),
      adapt_vector()),
    flatten())
```



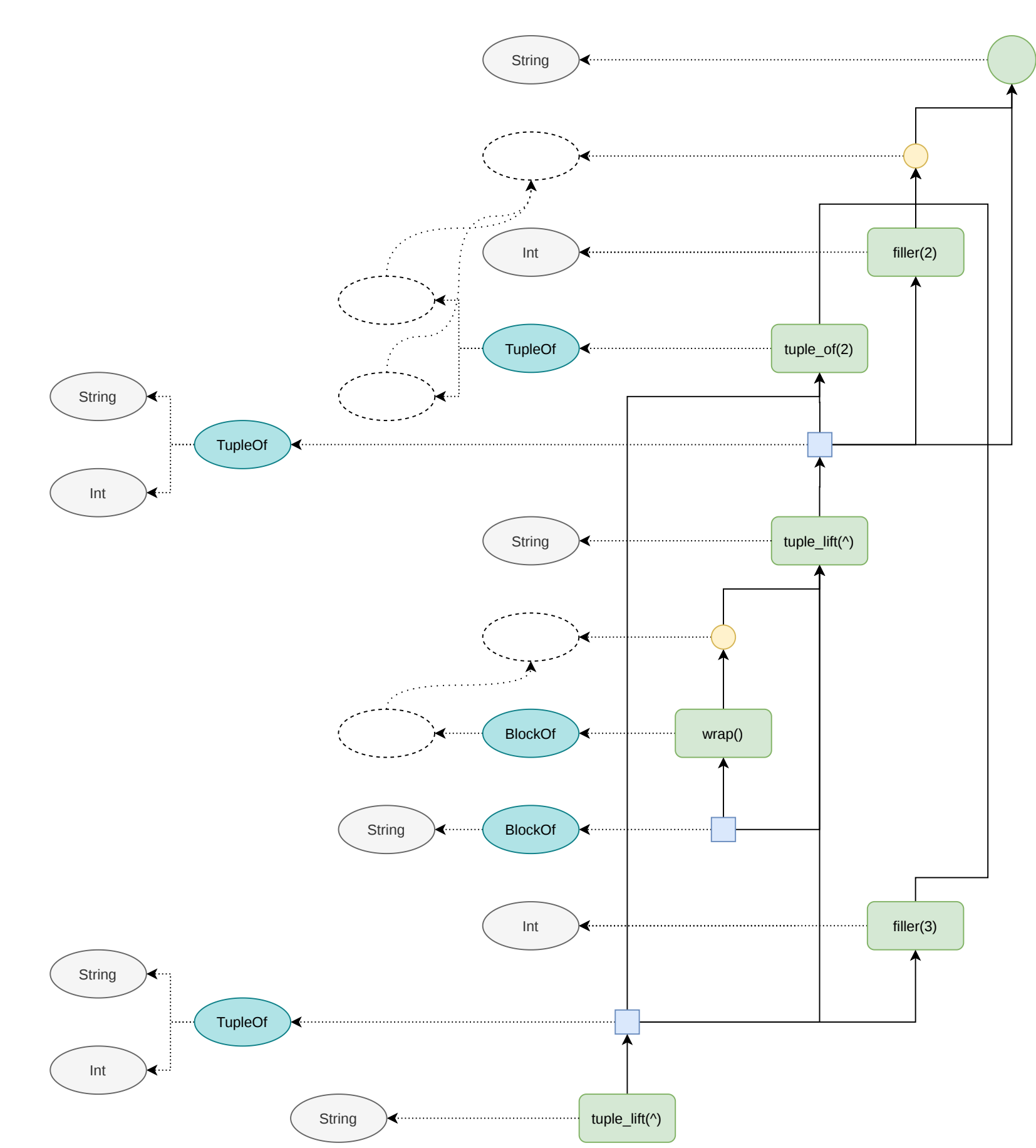




`untrace(n::NodeRef, guard::NodeRef)::Tuple{Pipeline,Vector{NodeRef}}`

`chain_of(f, tuple_of(g, h)) => tuple_of(chain_of(f, g), chain_of(f, h))`

`{it ^ 2, (it ^ 2) ^ 3}`



`untrace(n::NodeRef, guard::NodeRef)::Tuple{Pipeline,Vector{NodeRef}}`

@query "Hello World" split(it)

```
chain_of(
  wrap(),
  with_elements(wrap()),
  with_elements(lift(split)),
  with_elements(adapt_vector()),
  flatten())
```

