HIVE CUSTOM FUNCTIONS IN PYTHON

HIVE HAS THE ABILITY TO ALLOW USERS TO DEFINE CUSTOM FUNCTIONS

HIVE CUSTOM FUNCTIONS

FOR EXAMPLE YOU MIGHT WANT TO WRITE

REPLACETEXT()

REPLACE ALL OCCURRENCES OF A STRING IN SOME TEXT

HIVE CUSTOM FUNCTIONS

REPLACETEXT()

YOU CAN IMPLEMENT THE LOGIC FOR THIS CUSTOM FUNCTIONS IN

JAVA

PYTHON

HIVE CUSTOM FUNCTIONS

YOU CAN IMPLEMENT THE LOGIC FOR THESE CUSTOM FUNCTIONS IN

JAVA

PYTHON

IN JAVA THERE IS A SET OF CLASSES THAT CAN BE USED TO IMPLEMENT CUSTOM FUNCTIONS

HIVE CUSTOM FUNCTIONS

YOU CAN IMPLEMENT THE LOGIC FOR THESE CUSTOM FUNCTIONS IN

JAVA

PYTHON

OTHERWISE YOU CAN USE A PYTHON SCRIPT TO DEFINE THE FUNCTION

HIVE CUSTOM FUNCTIONS PYTHON

OTHERWISE YOU CAN USE A PYTHON SCRIPT TO DEFINE THE FUNCTION

THE SCRIPT WILL BE RUN USING STREAMING API FROM HADOOP

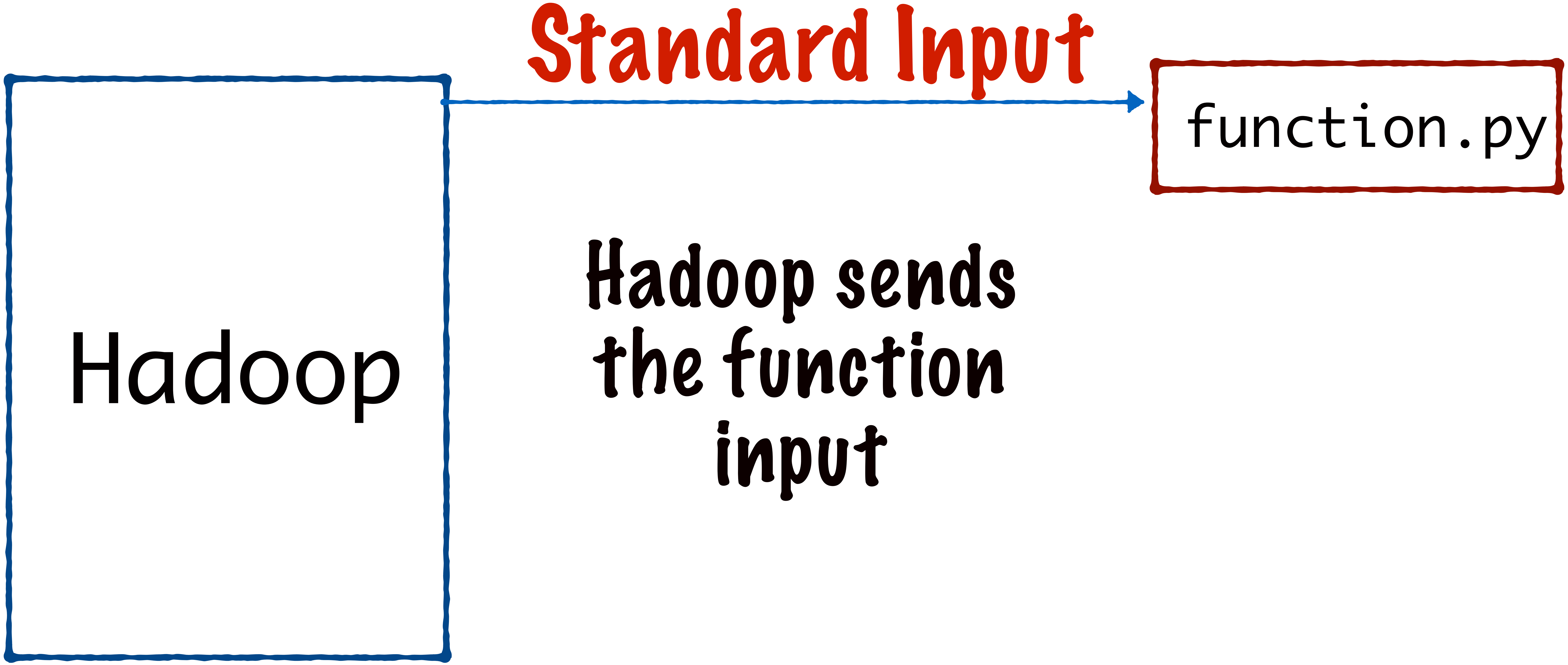
Hadoop Streaming API

The Streaming API uses **Standard Input/Output** to communicate with your program

Hadoop Streaming API

Let's say we implemented a function in Python

Hadoop Streaming API



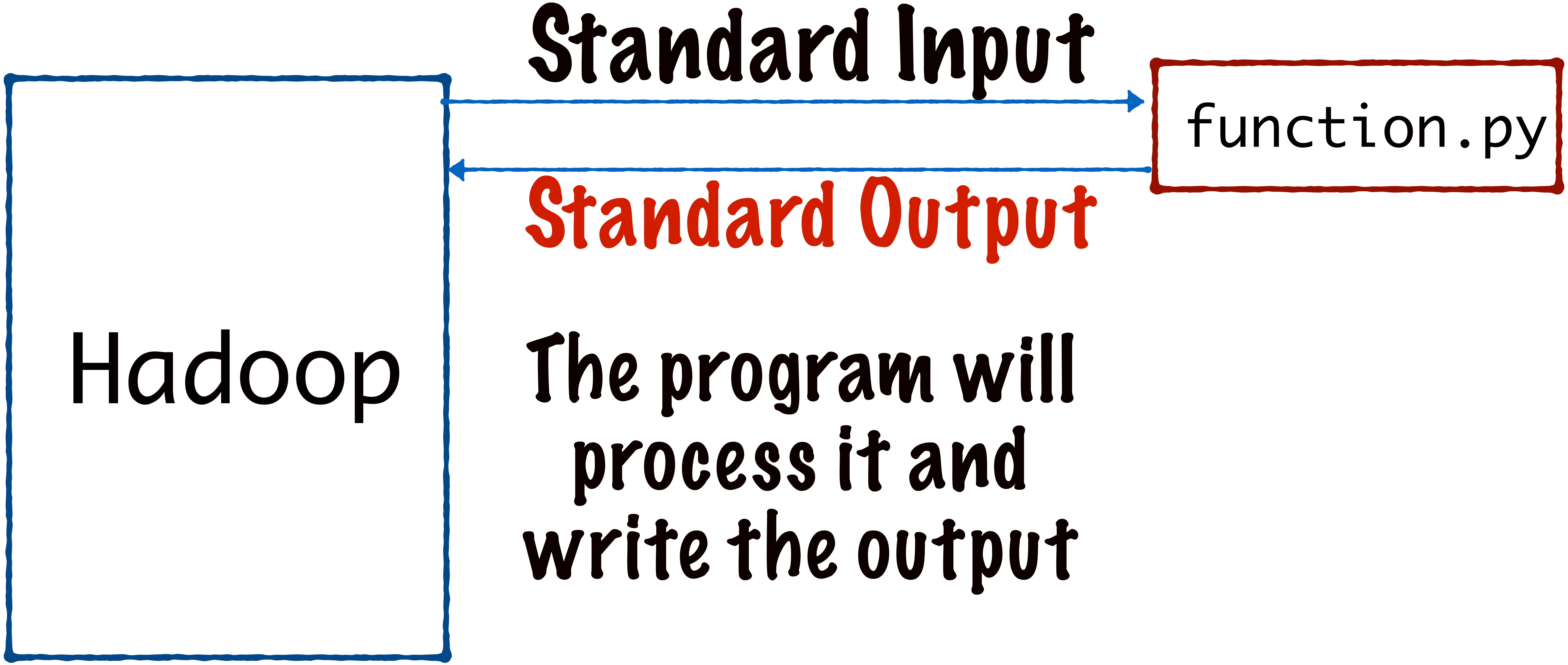
Standard Input

Hadoop

function.py

Hadoop sends the function input

Hadoop Streaming API



Hadoop

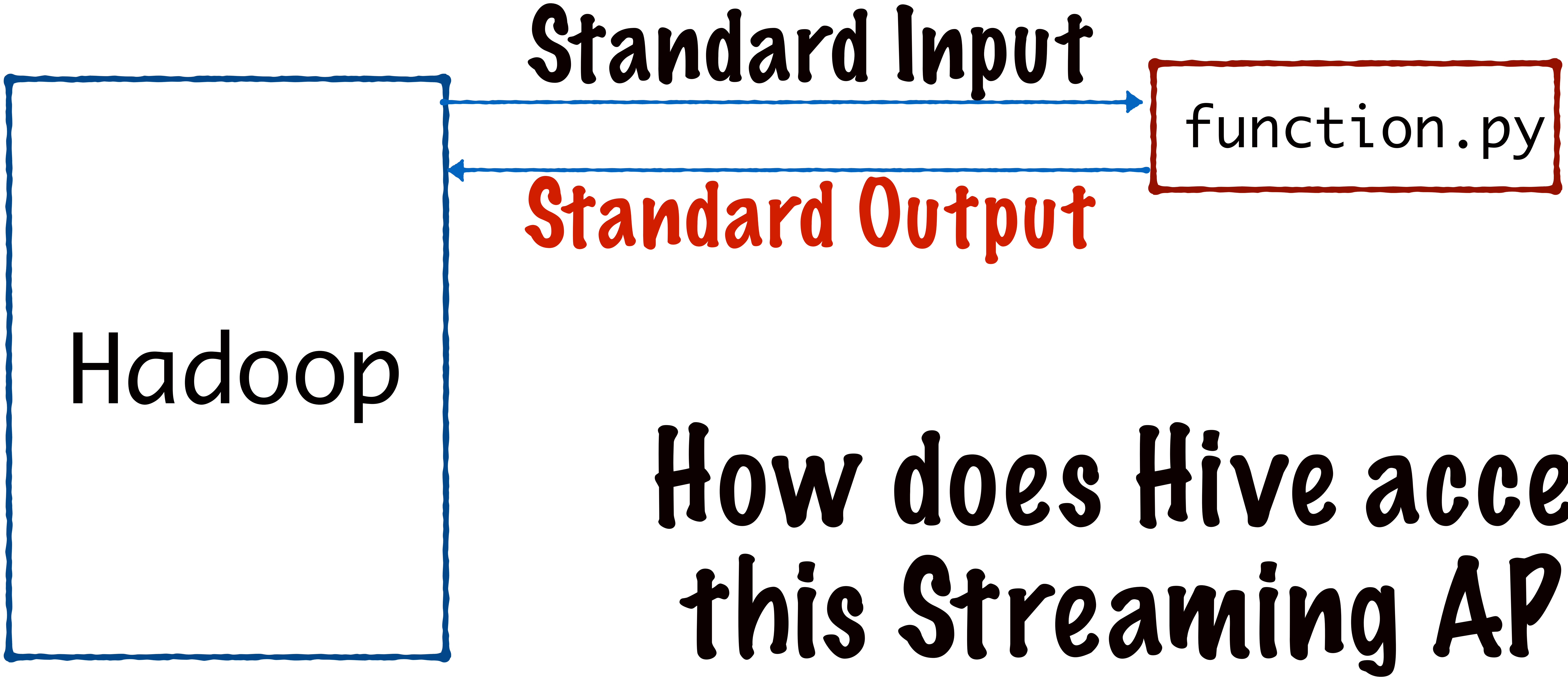
Standard Input

function.py

Standard Output

The program will process it and write the output

Hadoop Streaming API



Hadoop

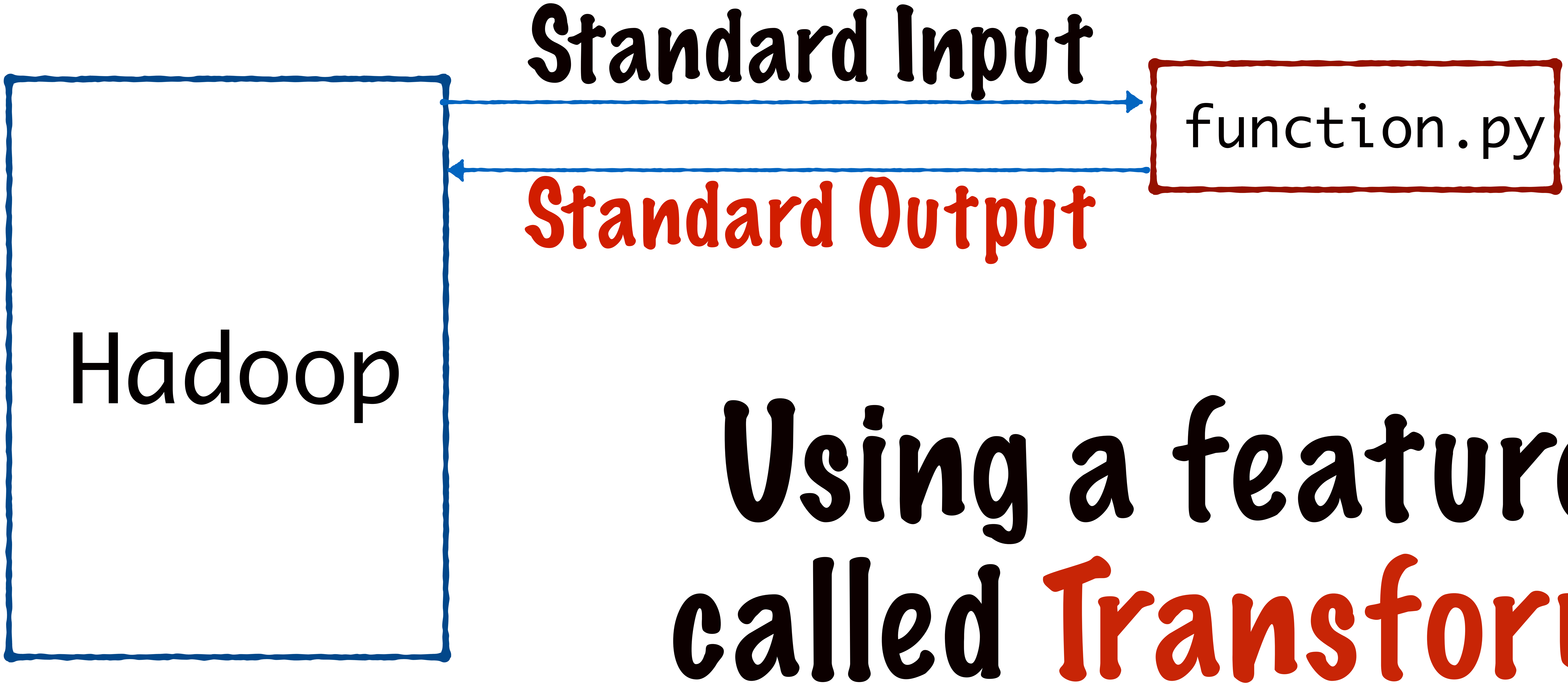
Standard Input

function.py

Standard Output

How does Hive access this Streaming API?

Hadoop Streaming API



Using a feature called Transform

Transform

function.py

```
SELECT TRANSFORM(firstname,lastname) USING
‘python function.py’ as isLonger from
employees;
```

Each row with first name, last name will be passed to the script function.py

Transform

function.py

```
SELECT TRANSFORM(firstname,lastname) USING
‘python function.py’ as isLonger from
employees;
```

The script will process it and return a row

Transform

function.py

Let's write a function that will compare the lengths of first name, last name of 2 employees

The function returns true if last name longer than first name

function.py

```
import sys
for line in sys.stdin:
(firstname,lastname)=line.split(‘\t’)
    if len(firstname)>len(lastname):
        print "TRUE"
```

```
SELECT TRANSFORM(firstname,lastname)
USING ‘python function.py’ as
isLonger from employees
```

function.py

```
import sys
for line in sys.stdin:
(firstname,lastname)=line.split(‘\t’)
    if len(firstname)>len(lastname):
        print "TRUE"
```

```
SELECT TRANSFORM(firstname,lastname) USING
‘python function.py’ as isLonger from
employees;
```

function.py

```
import sys
for line in sys.stdin:
```

Each row with first name, last name is passed to the script over standard input

function.py

```
import sys
for line in sys.stdin:
(firstname,lastname)=line.split(‘\t’)
    if len(firstname)>len(lastname):
        print "TRUE"
```

The row needs to be split to extract the first name , last name strings

function.py

```
import sys
for line in sys.stdin:
(firstname,lastname)=line.split(‘\t’)
    if len(firstname)>len(lastname):
        print "TRUE"
```

The row delimiter is always tab

function.py

```
import sys
for line in sys.stdin:
(firstname,lastname)=line.split(‘\t’)
    if len(firstname)>len(lastname):
        print "TRUE"
```

This is a property of the Hadoop Streaming api

```
function.py

import sys
for line in sys.stdin:
(firstname,lastname)=line.split(‘\t’)
    if len(firstname)>len(lastname):
        print "TRUE"
```

check the condition and print the result to the Standard Output

Transform

function.py

Once you have the function script you need to register it to hive before using it

```
add FILE /Users/
swethakolalapudi/function.py;
```