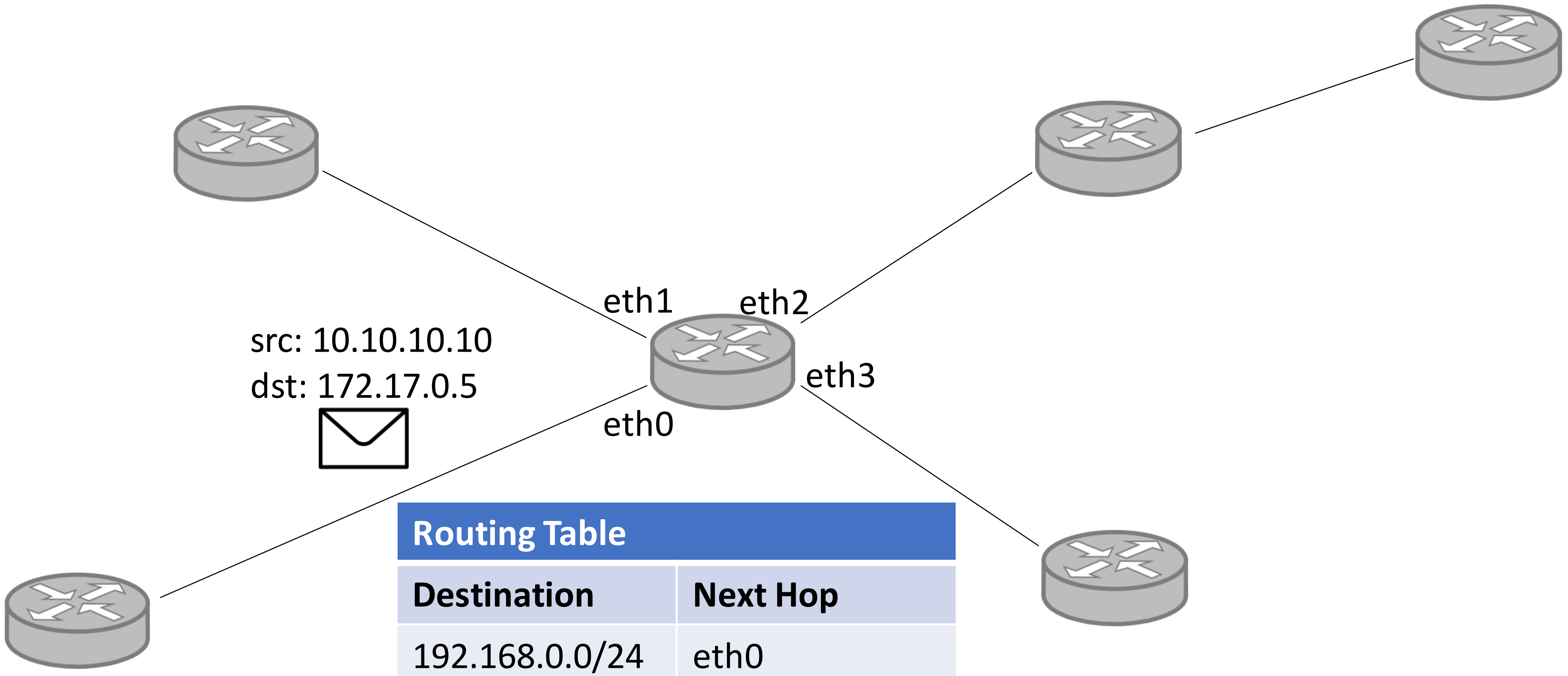
Remote Method Invocation in ICN

Michał Król

City, University of London

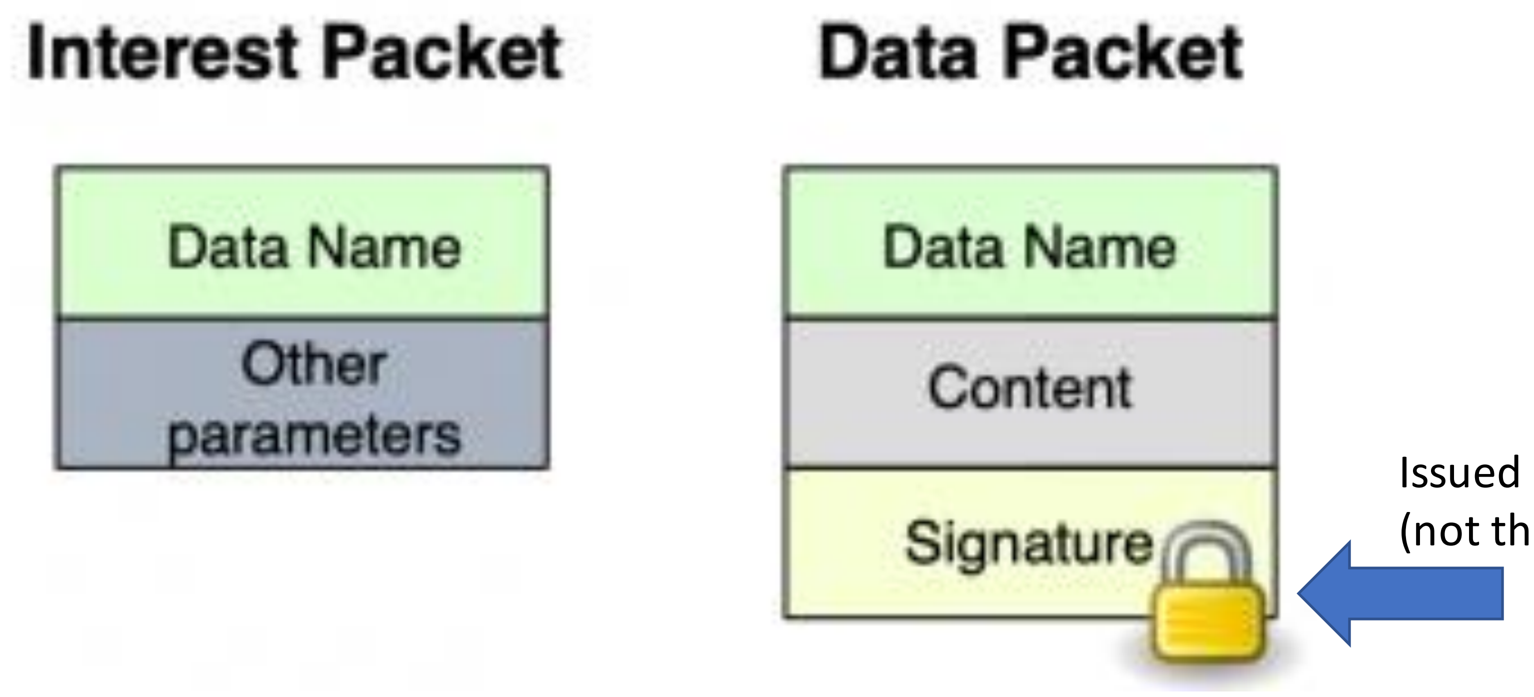
Information-Centric Networks



Routing Table

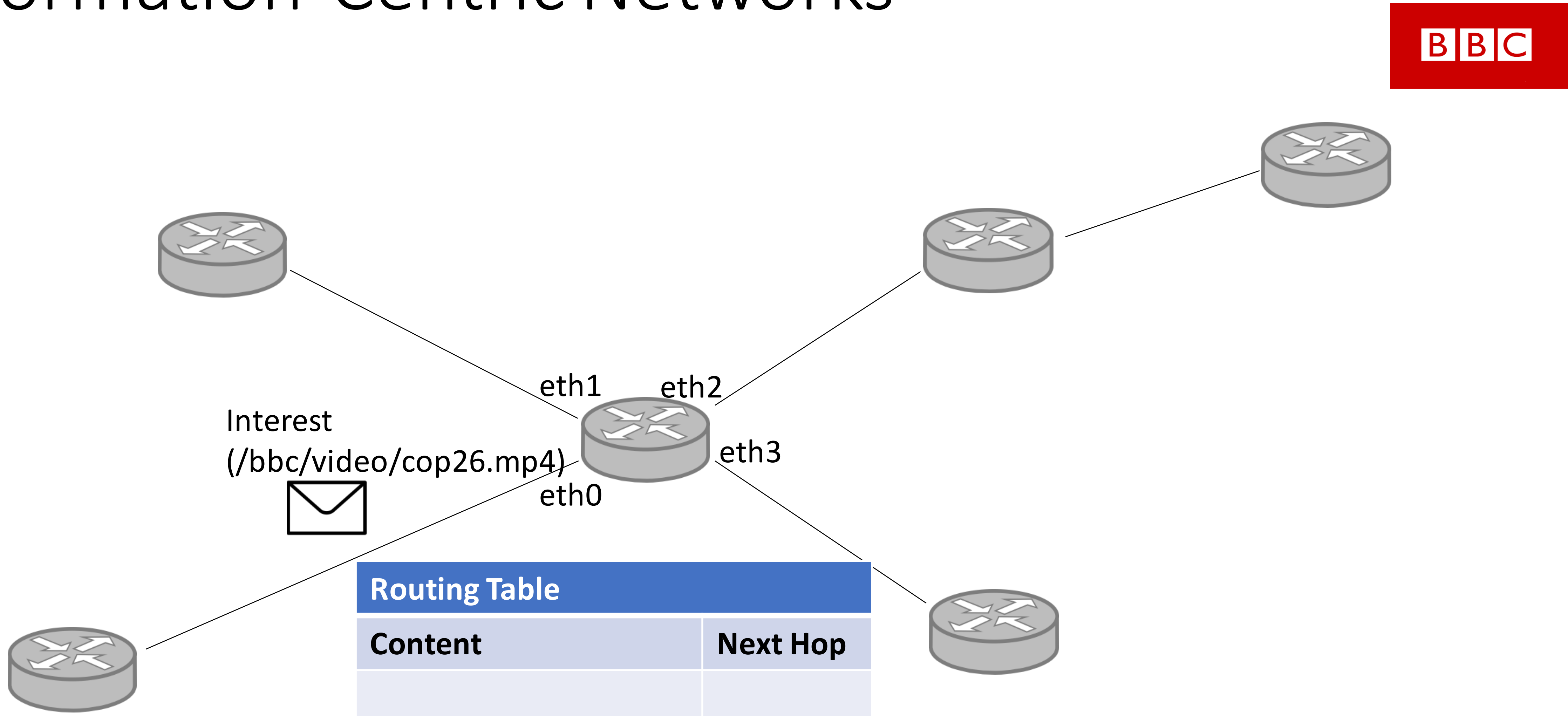
Destination	Next Hop
192.168.0.0/24	eth0
10.0.0.0/8	eth1
172.17.0.1/16	eth2

Information-Centric Networks



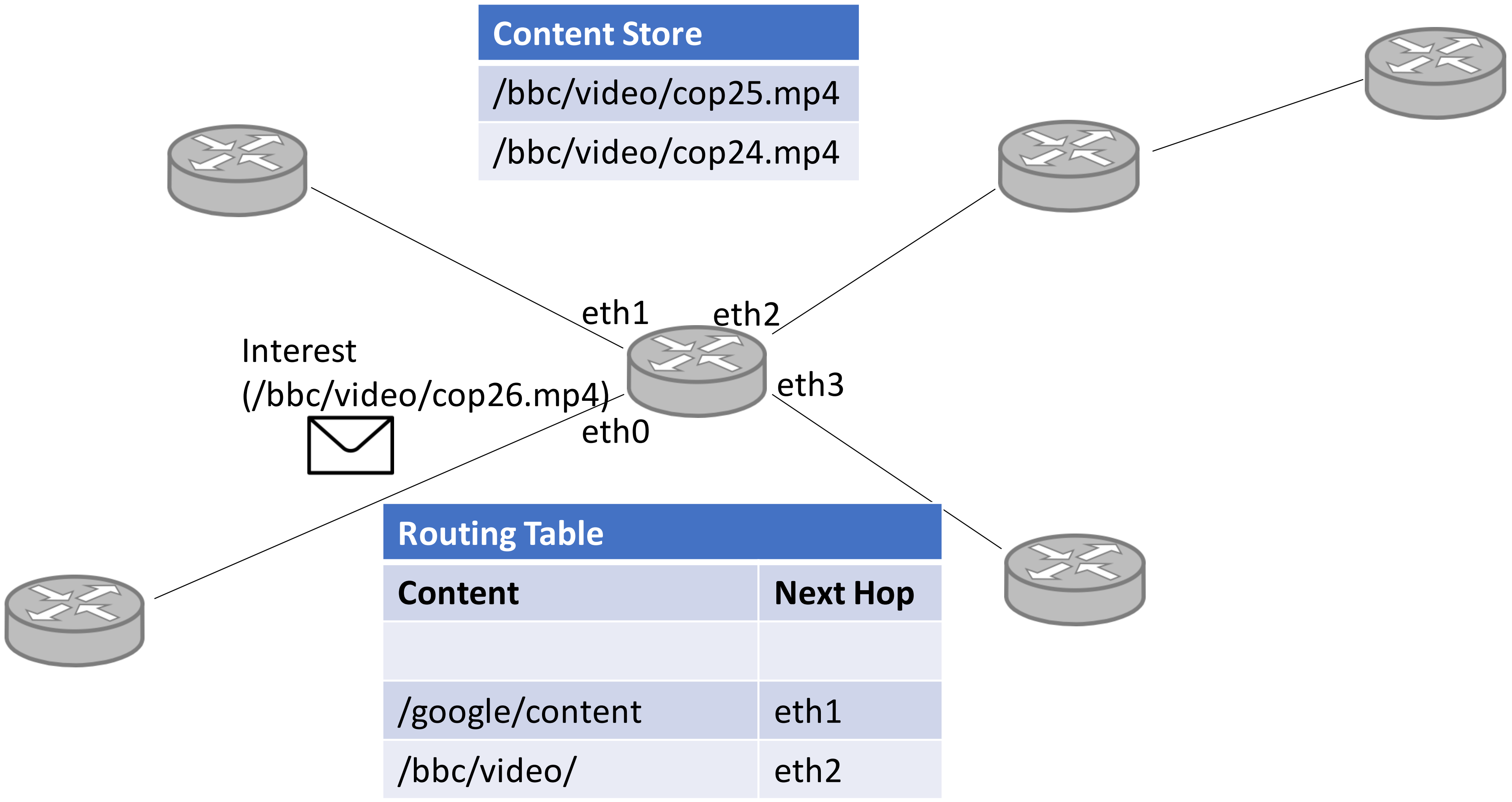
Issued by the producer
(not the sender)

Information-Centric Networks



Routing Table	
Content	**Next Hop**
/google/content	eth1
/bbc/video/	eth2

Information-Centric Networks

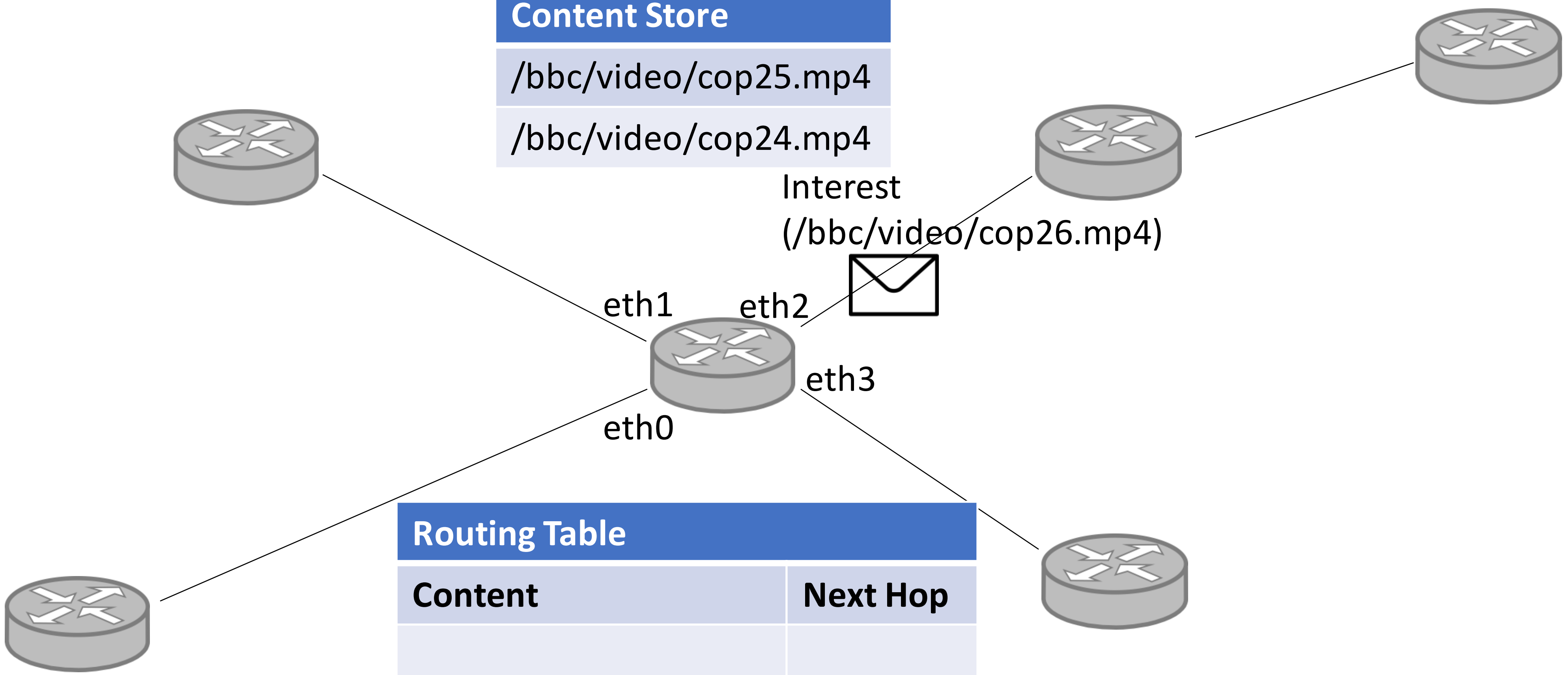


Content Store
/bbc/video/cop25.mp4
/bbc/video/cop24.mp4

Routing Table

Content	Next Hop
/google/content	eth1
/bbc/video/	eth2

Information-Centric Networks

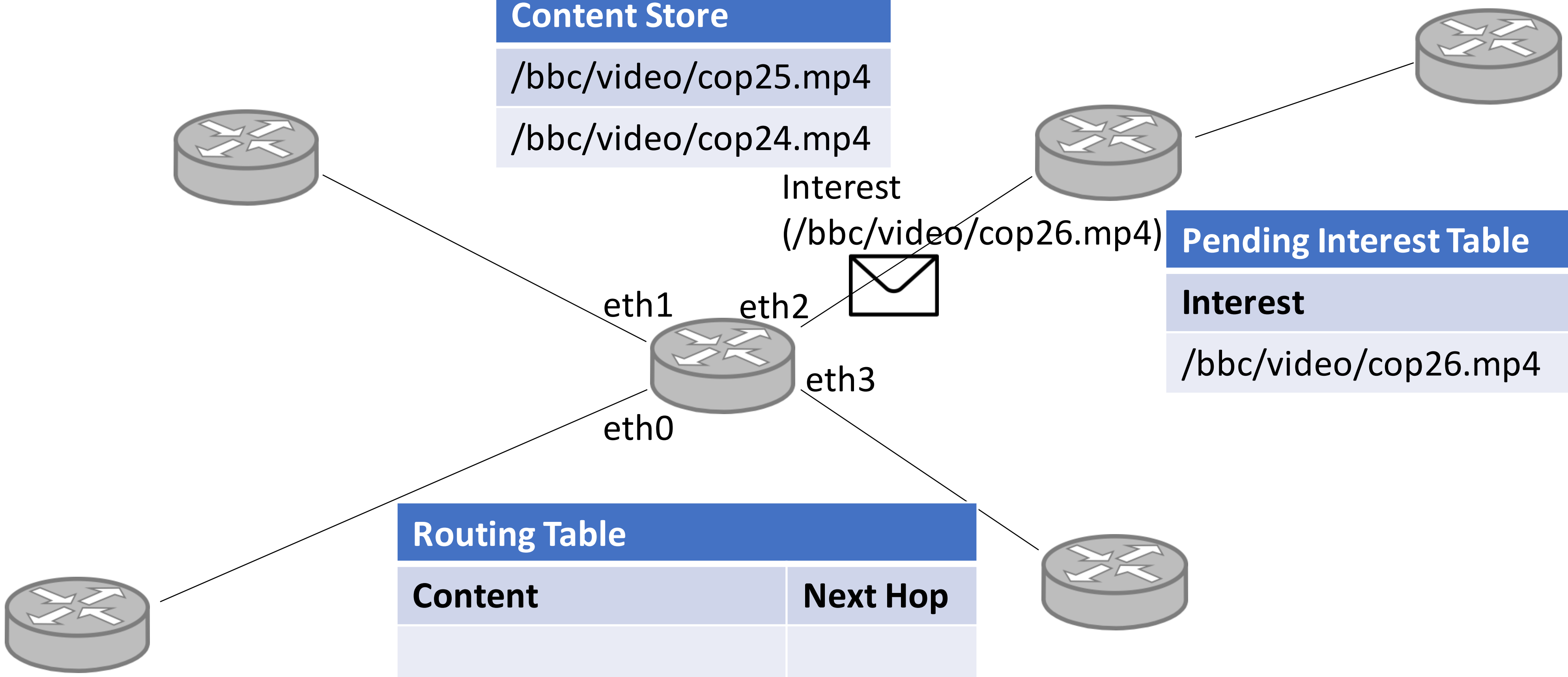


Content Store
/bbc/video/cop25.mp4
/bbc/video/cop24.mp4

Routing Table

Content	Next Hop
/google/content	eth1
/bbc/video/	eth2

Information-Centric Networks



Content Store
/bbc/video/cop25.mp4
/bbc/video/cop24.mp4

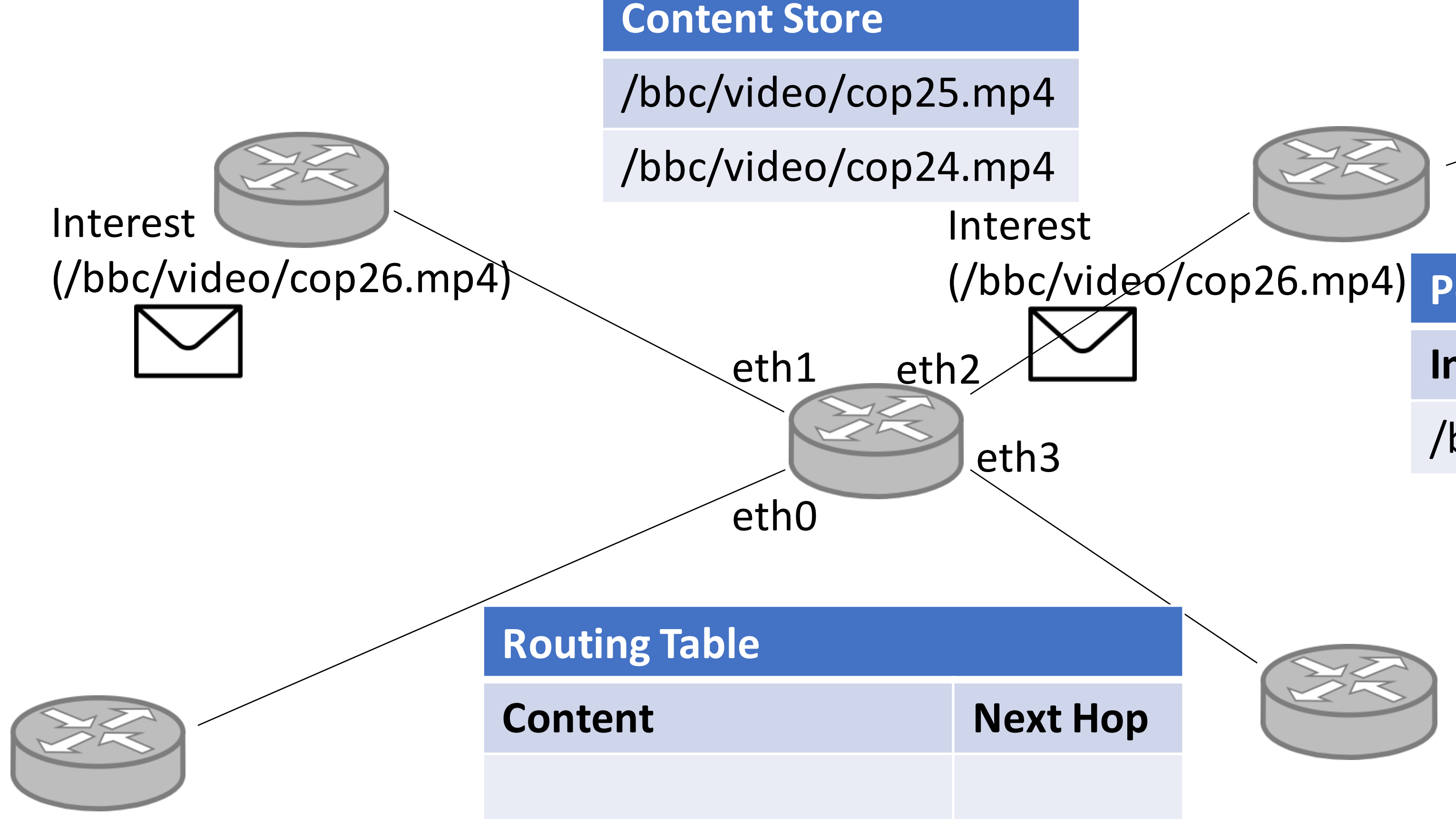
Pending Interest Table

Interest	Interface
/bbc/video/cop26.mp4	eth0

Routing Table

Content	Next Hop
/google/content	eth1
/bbc/video/	eth2

Information-Centric Networks



Content Store
/bbc/video/cop25.mp4
/bbc/video/cop24.mp4

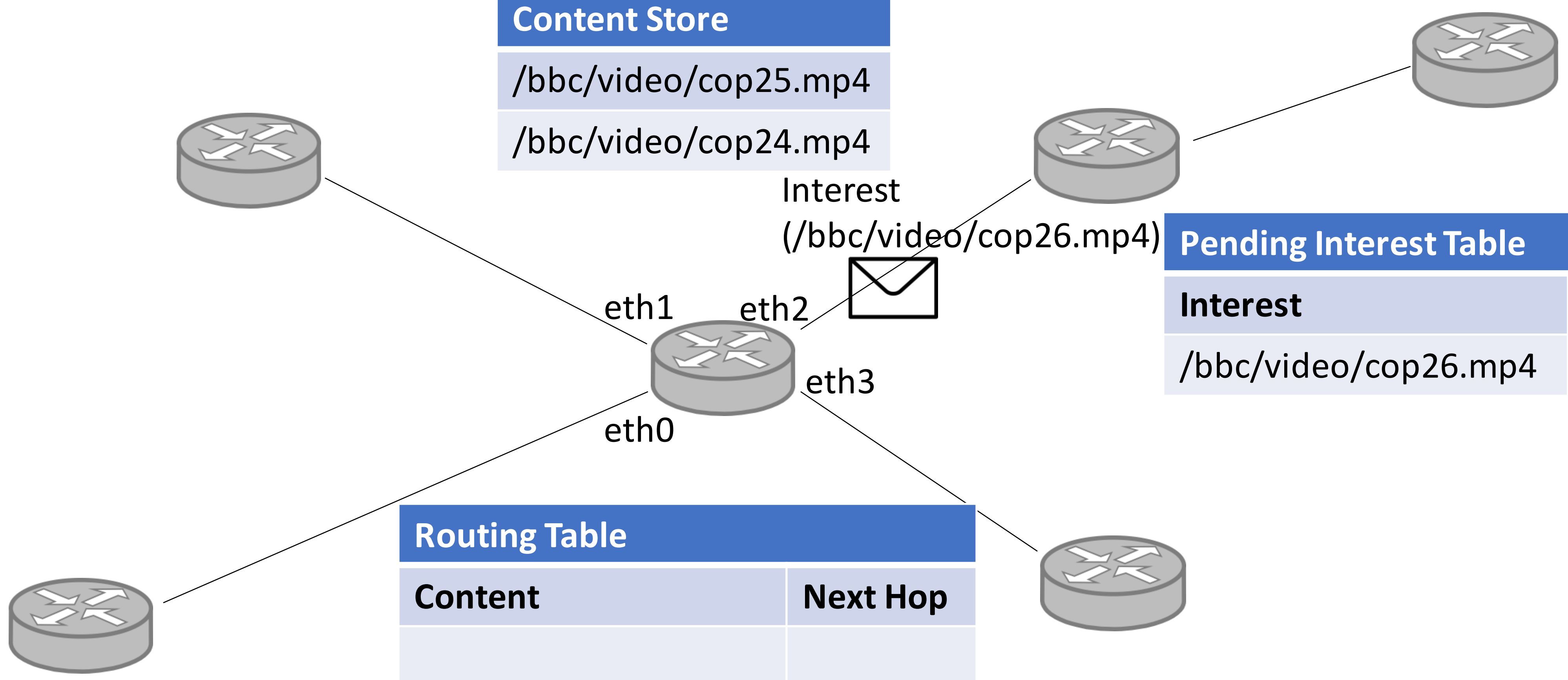
Pending Interest Table

Interest	Interface
/bbc/video/cop26.mp4	eth0

Routing Table

Content	Next Hop
/google/content	eth1
/bbc/video/	eth2

Information-Centric Networks



Content Store
/bbc/video/cop25.mp4
/bbc/video/cop24.mp4

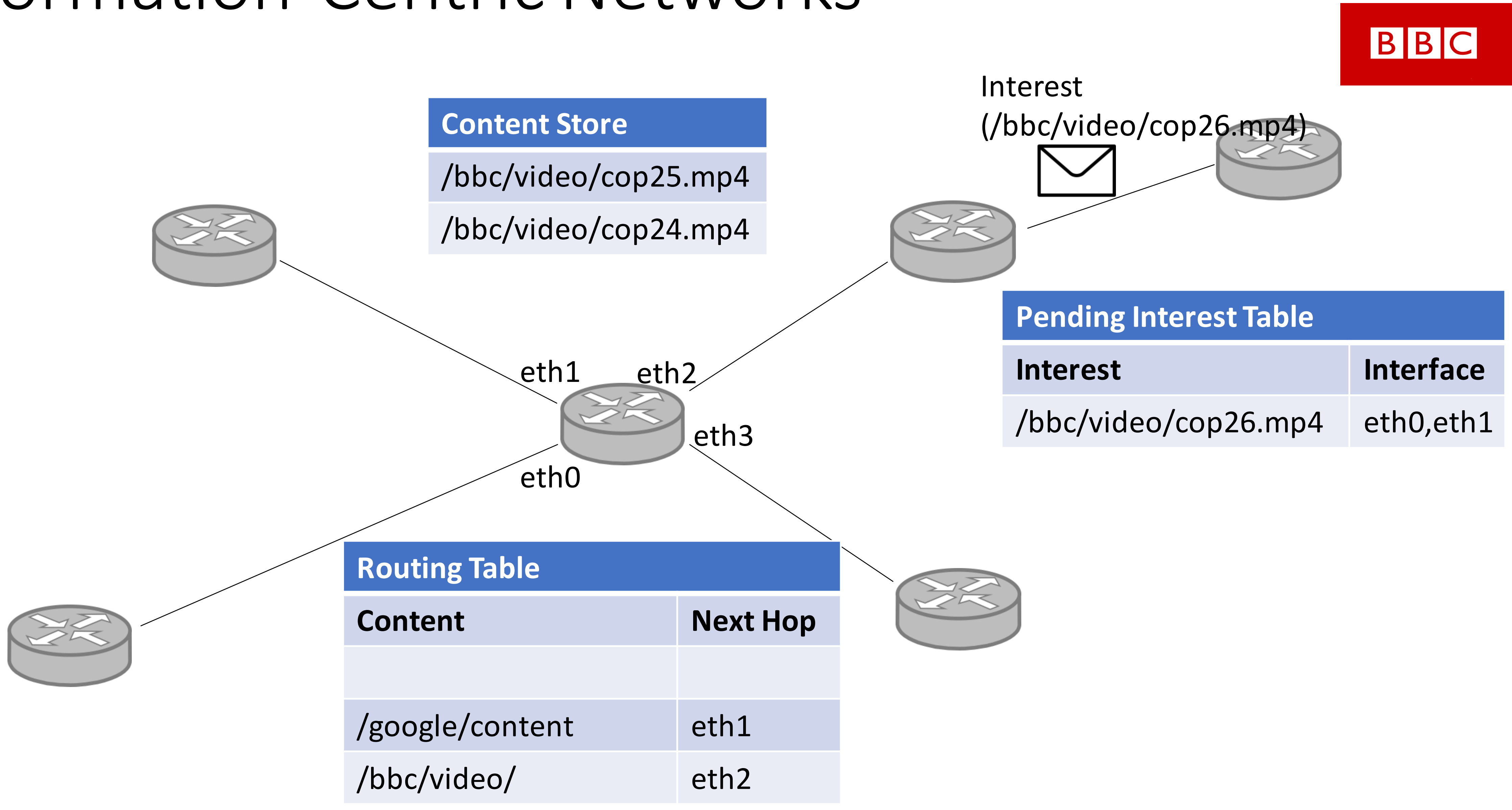
Pending Interest Table

Interest	Interface
/bbc/video/cop26.mp4	eth0,eth1

Routing Table

Content	Next Hop
/google/content	eth1
/bbc/video/	eth2

Information-Centric Networks



Content Store
/bbc/video/cop25.mp4
/bbc/video/cop24.mp4

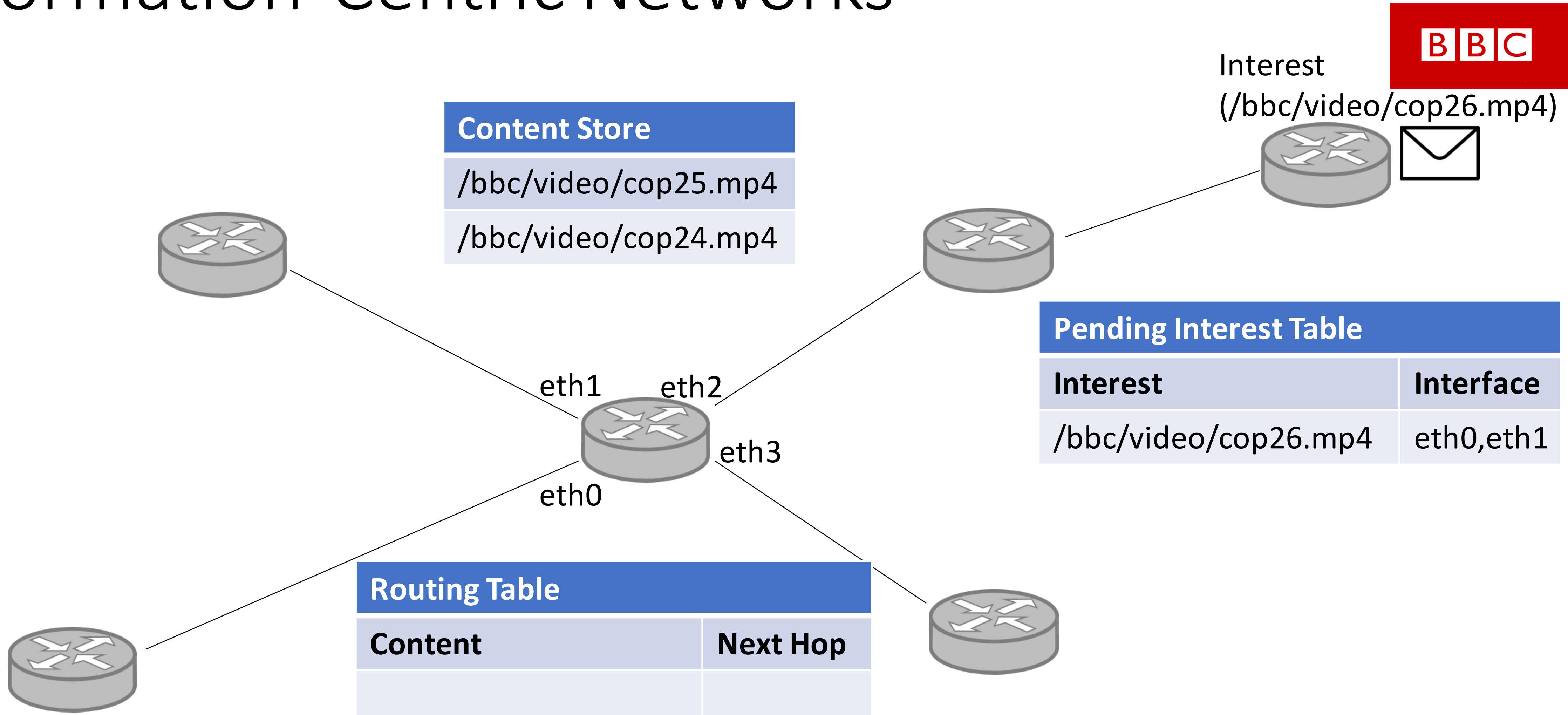
Pending Interest Table

Interest	Interface
/bbc/video/cop26.mp4	eth0,eth1

Routing Table

Content	Next Hop
/google/content	eth1
/bbc/video/	eth2

Information-Centric Networks



Content Store
/bbc/video/cop25.mp4
/bbc/video/cop24.mp4

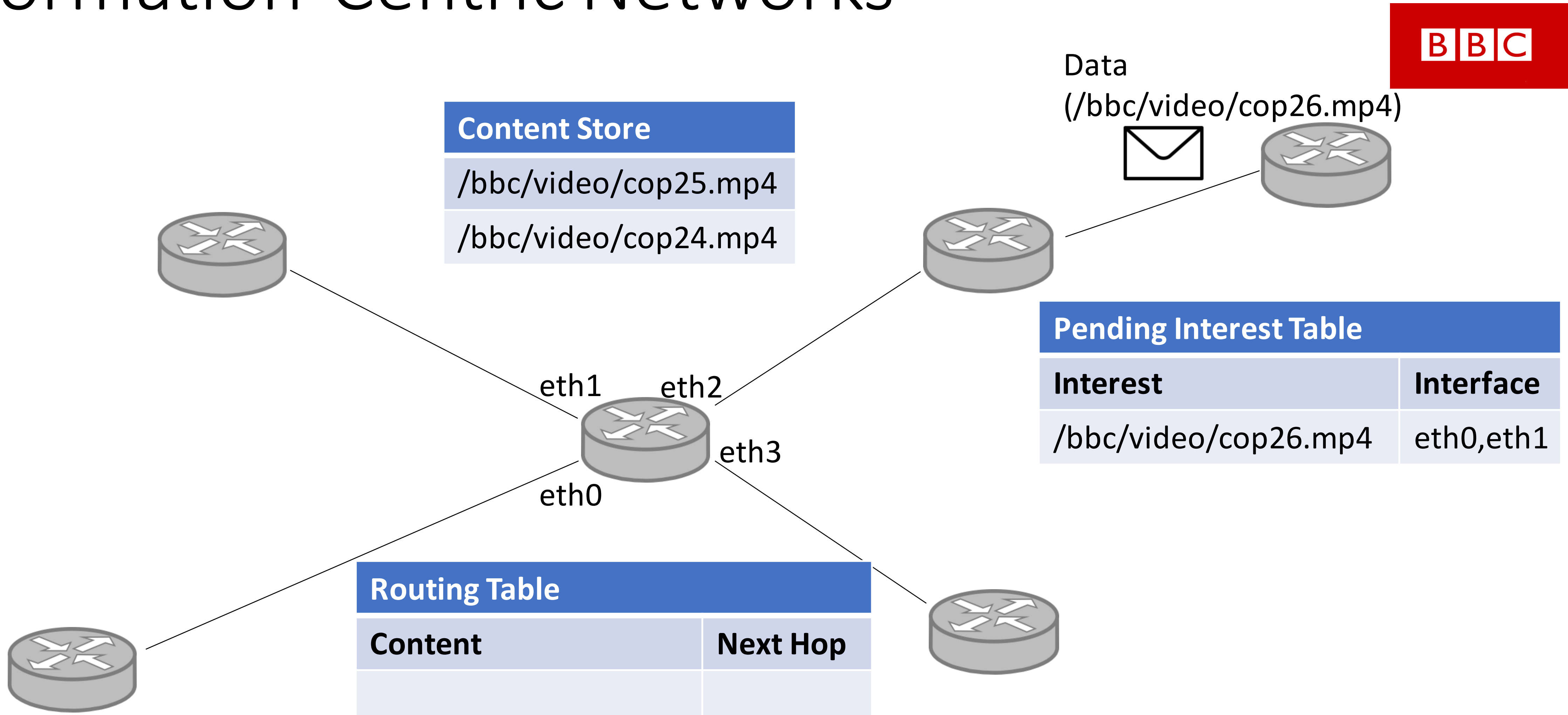
Pending Interest Table

Interest	Interface
/bbc/video/cop26.mp4	eth0,eth1

Routing Table

Content	Next Hop
/google/content	eth1
/bbc/video/	eth2

Information-Centric Networks

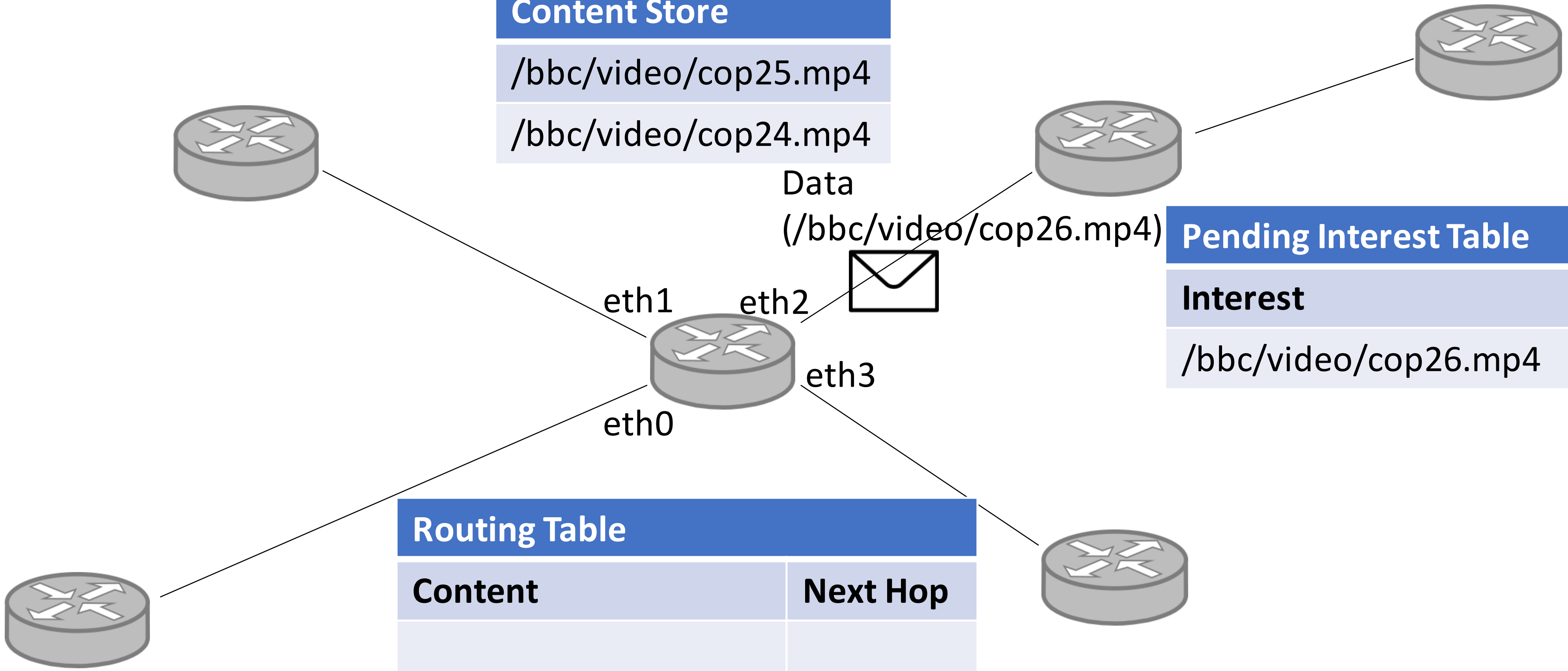


Content Store
/bbc/video/cop25.mp4
/bbc/video/cop24.mp4

Pending Interest Table	
Interest	**Interface**
/bbc/video/cop26.mp4	eth0,eth1

Routing Table	
Content	**Next Hop**
/google/content	eth1
/bbc/video/	eth2

Information-Centric Networks



Content Store
/bbc/video/cop25.mp4
/bbc/video/cop24.mp4

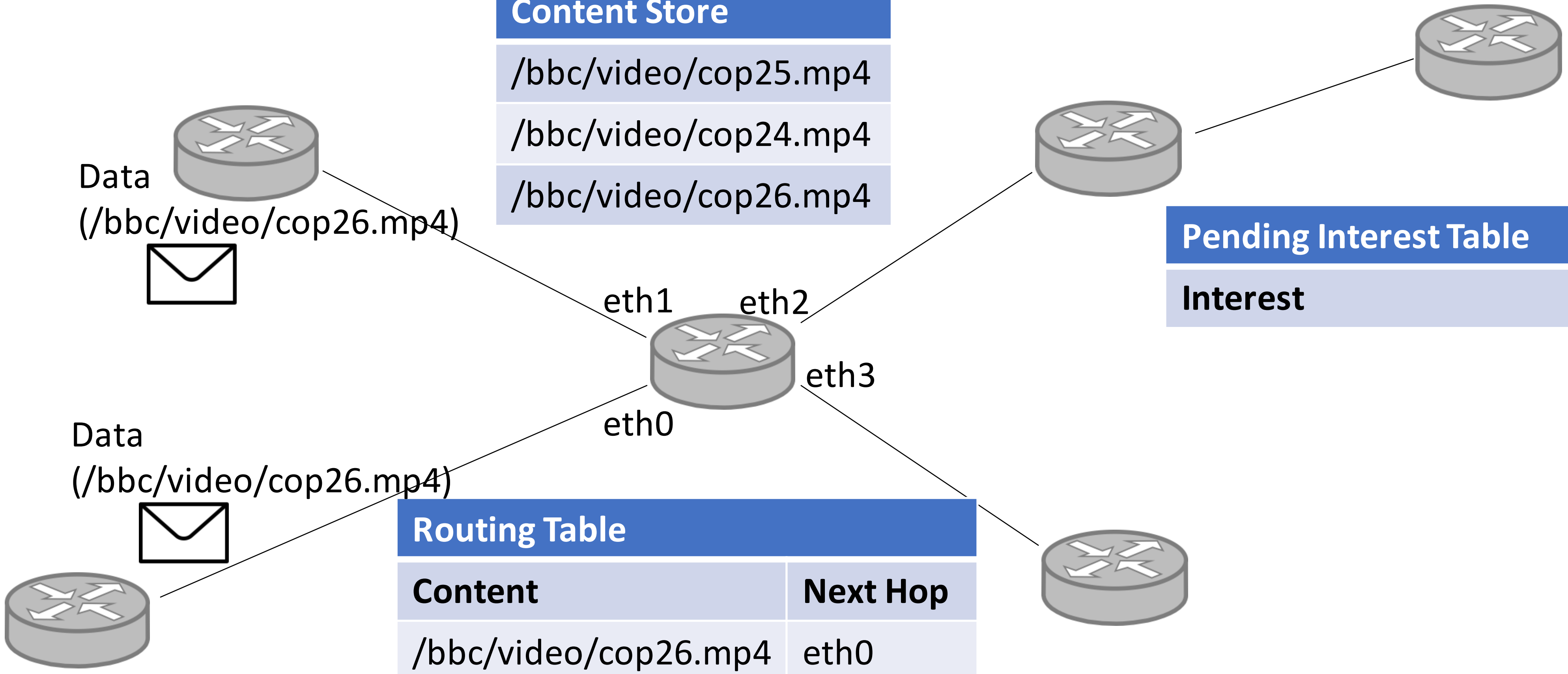
Pending Interest Table

Interest	Interface
/bbc/video/cop26.mp4	eth0

Routing Table

Content	Next Hop
/google/content	eth1
/bbc/video/	eth2

Information-Centric Networks

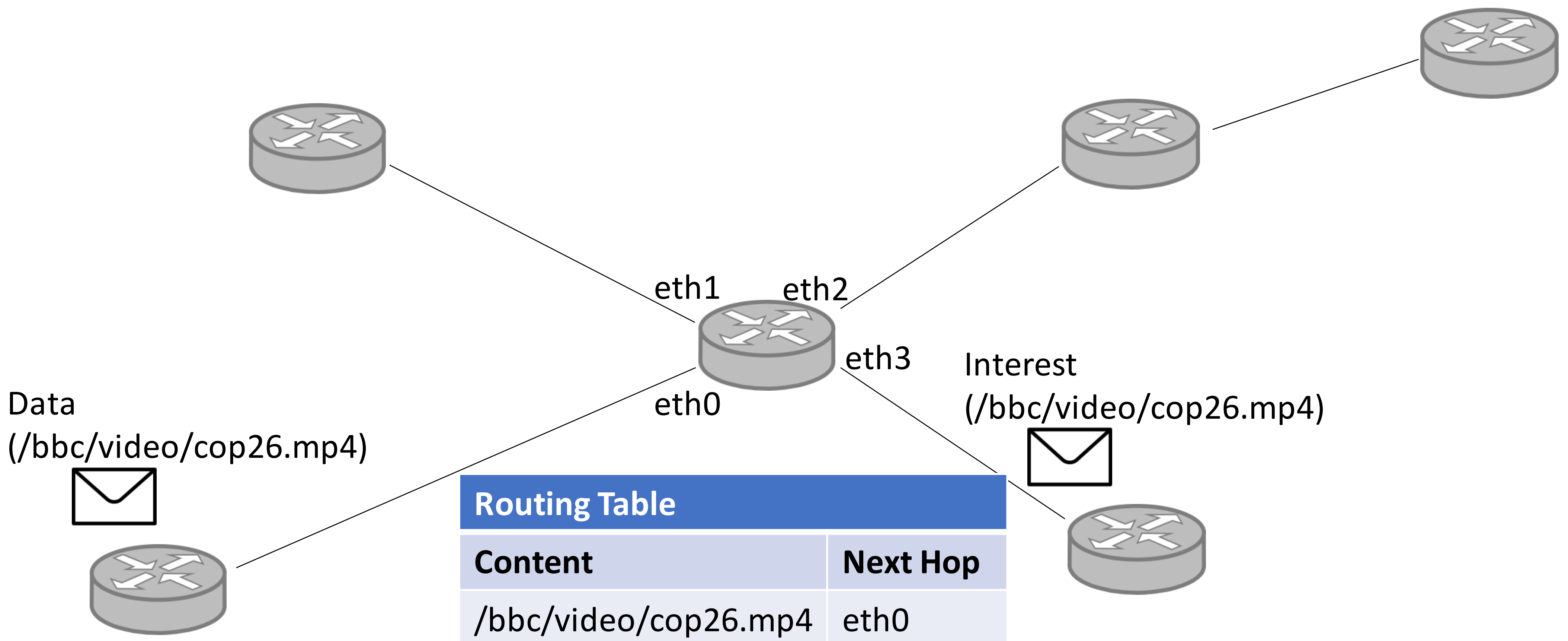


Content Store
/bbc/video/cop25.mp4
/bbc/video/cop24.mp4
/bbc/video/cop26.mp4

Pending Interest Table	
Interest	**Interface**

Routing Table	
Content	**Next Hop**
/bbc/video/cop26.mp4	eth0
/google/content	eth1
/bbc/video/	eth2

Information-Centric Networks



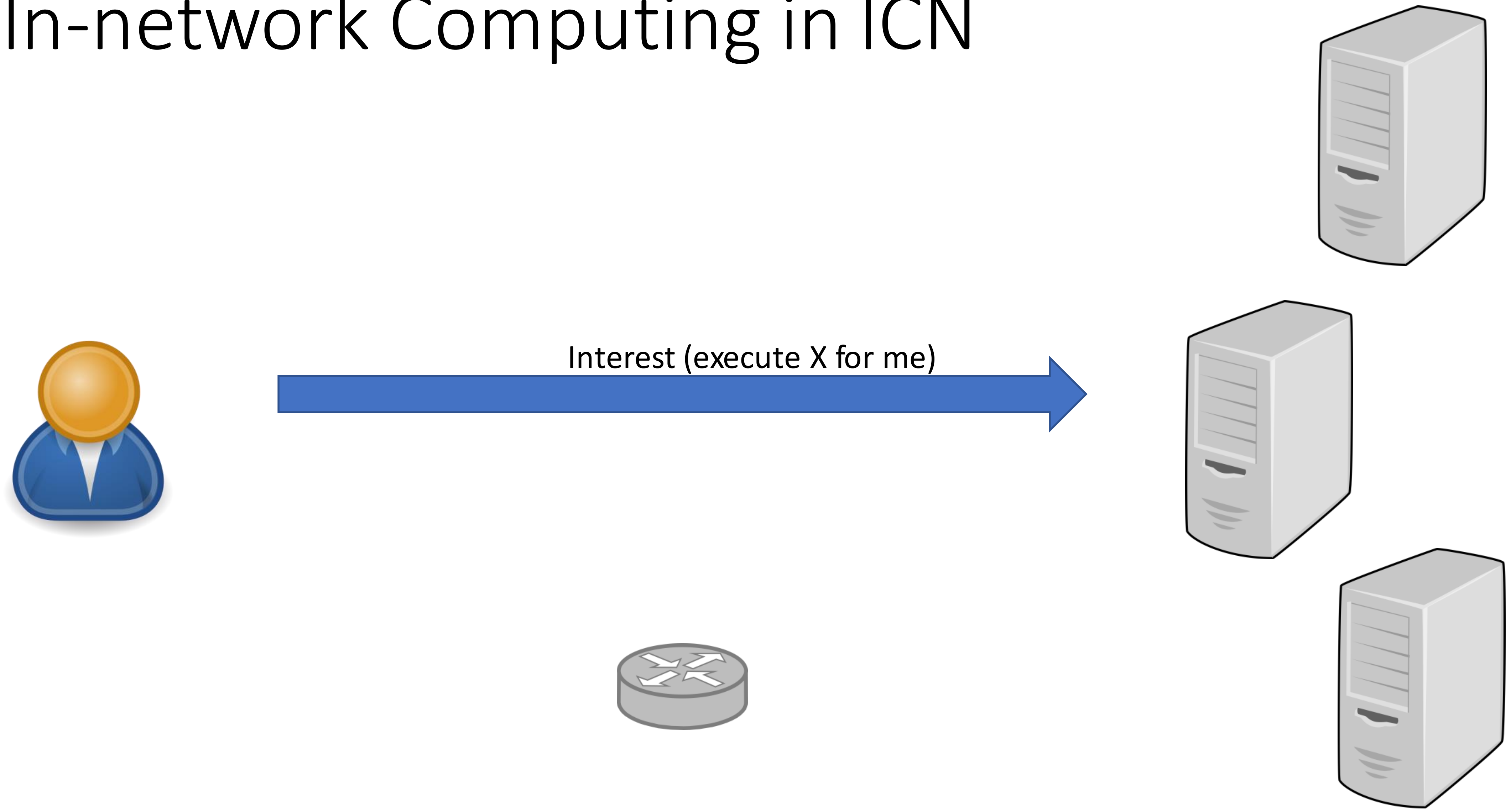
Routing Table

Content	Next Hop
/bbc/video/cop26.mp4	eth0
/google/content	eth1
/bbc/video/	eth2

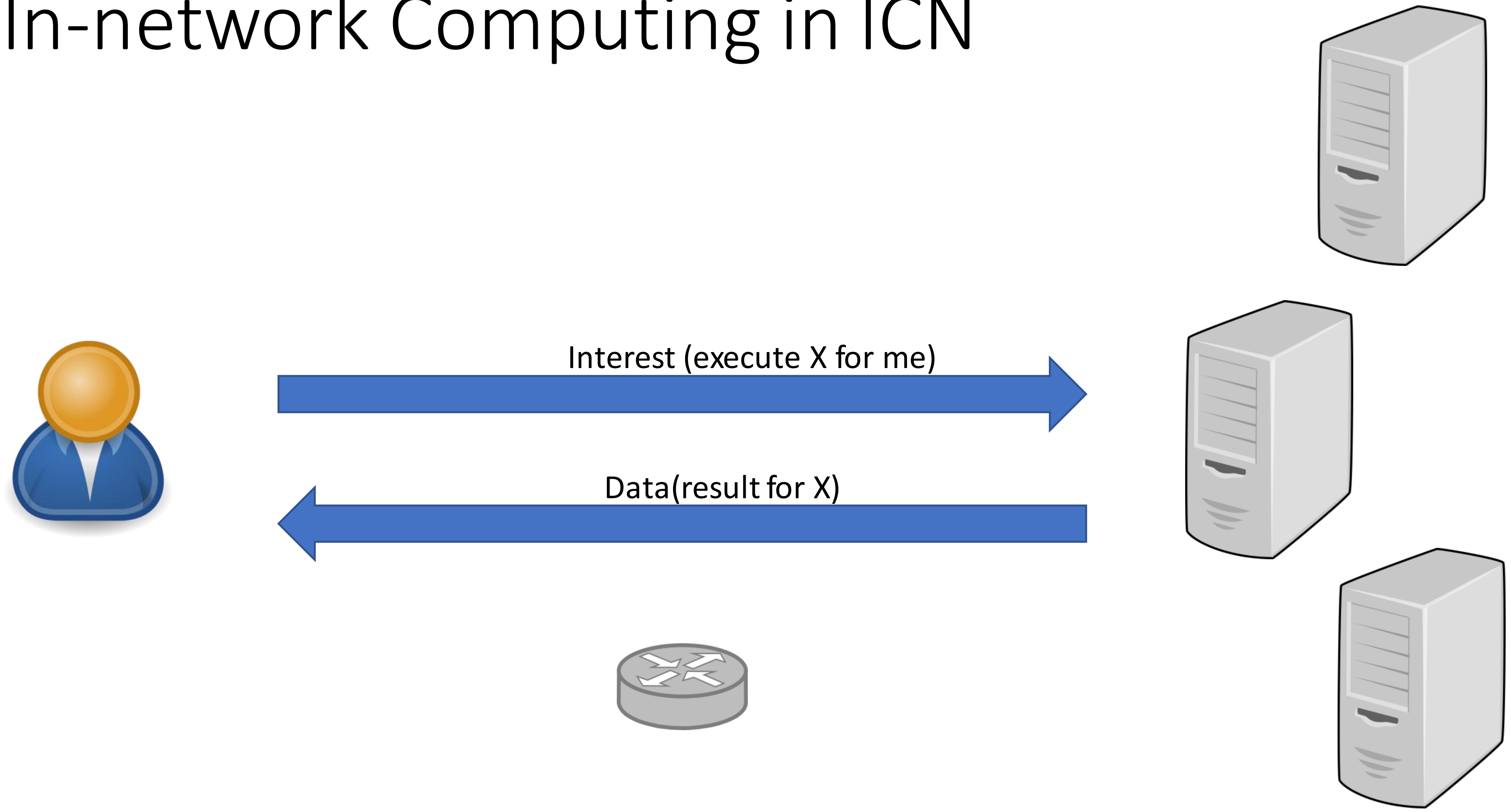
In-network Computing in ICN

- Any node can become a computation node
- Explicit names in the Interest/Data exposed to the network layer
- Easy re-use of (partial) results

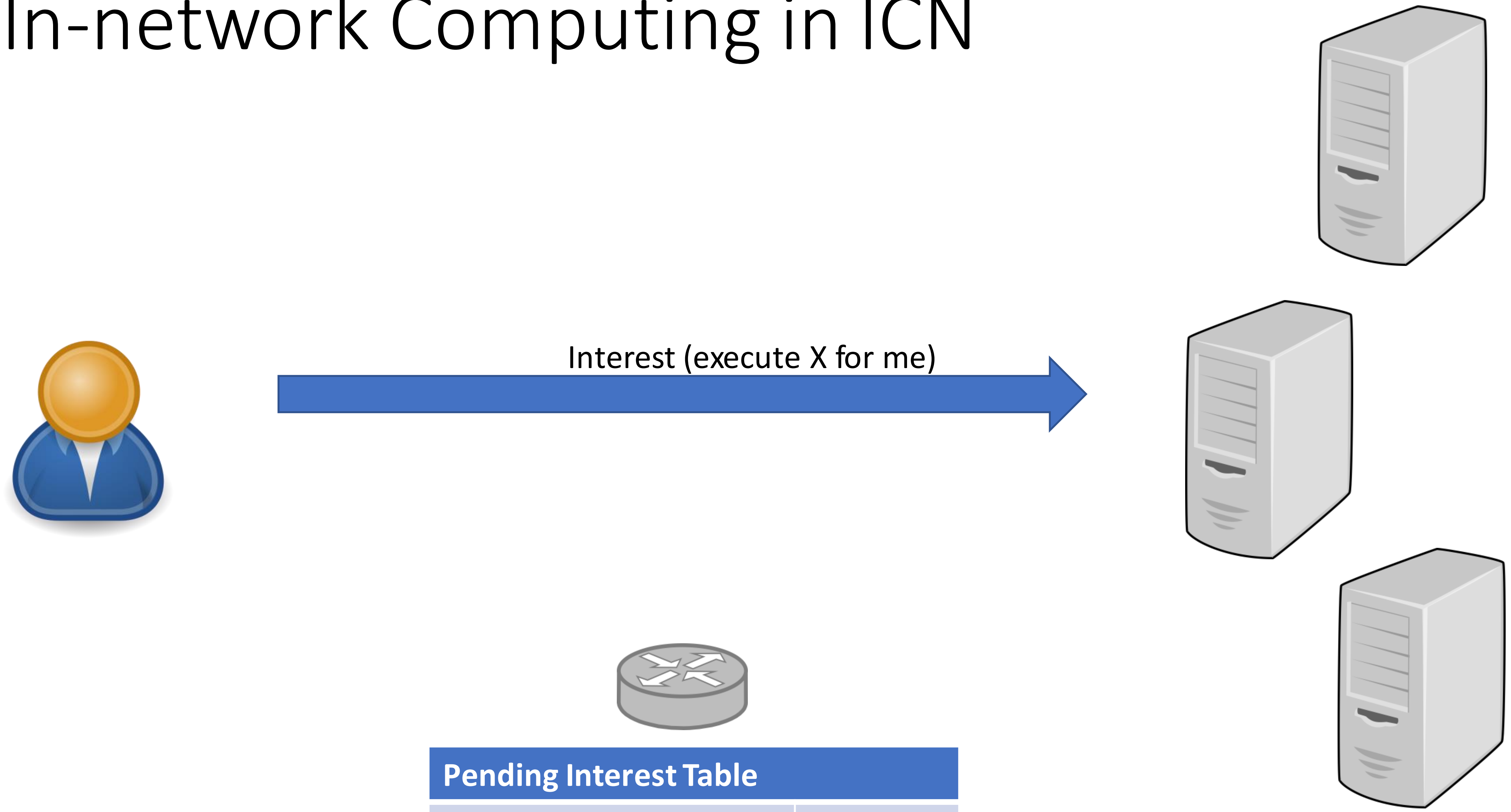
In-network Computing in ICN



In-network Computing in ICN

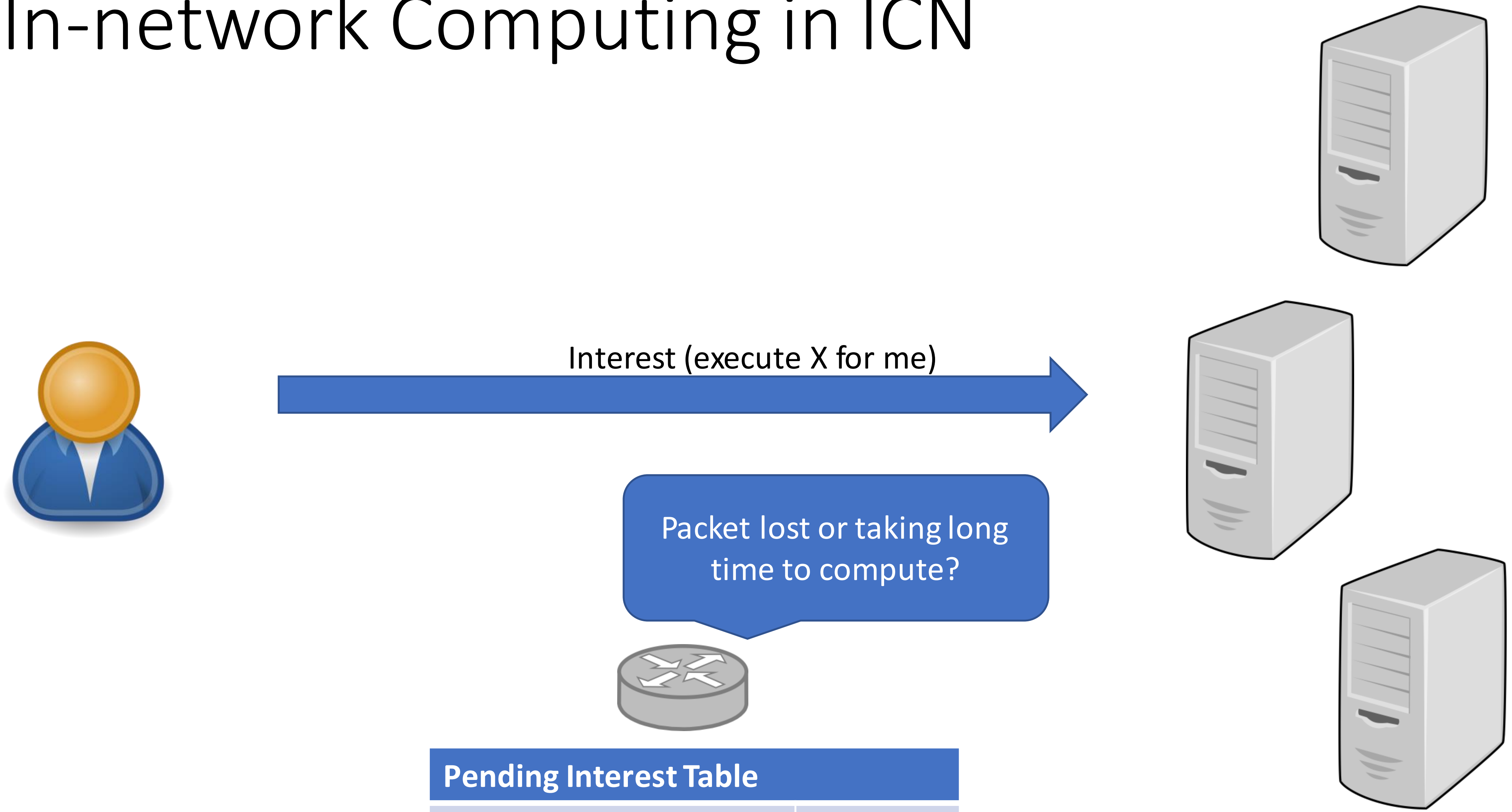


In-network Computing in ICN



Pending Interest Table	
Interest	**Interface**
/execute/X	eth0

In-network Computing in ICN

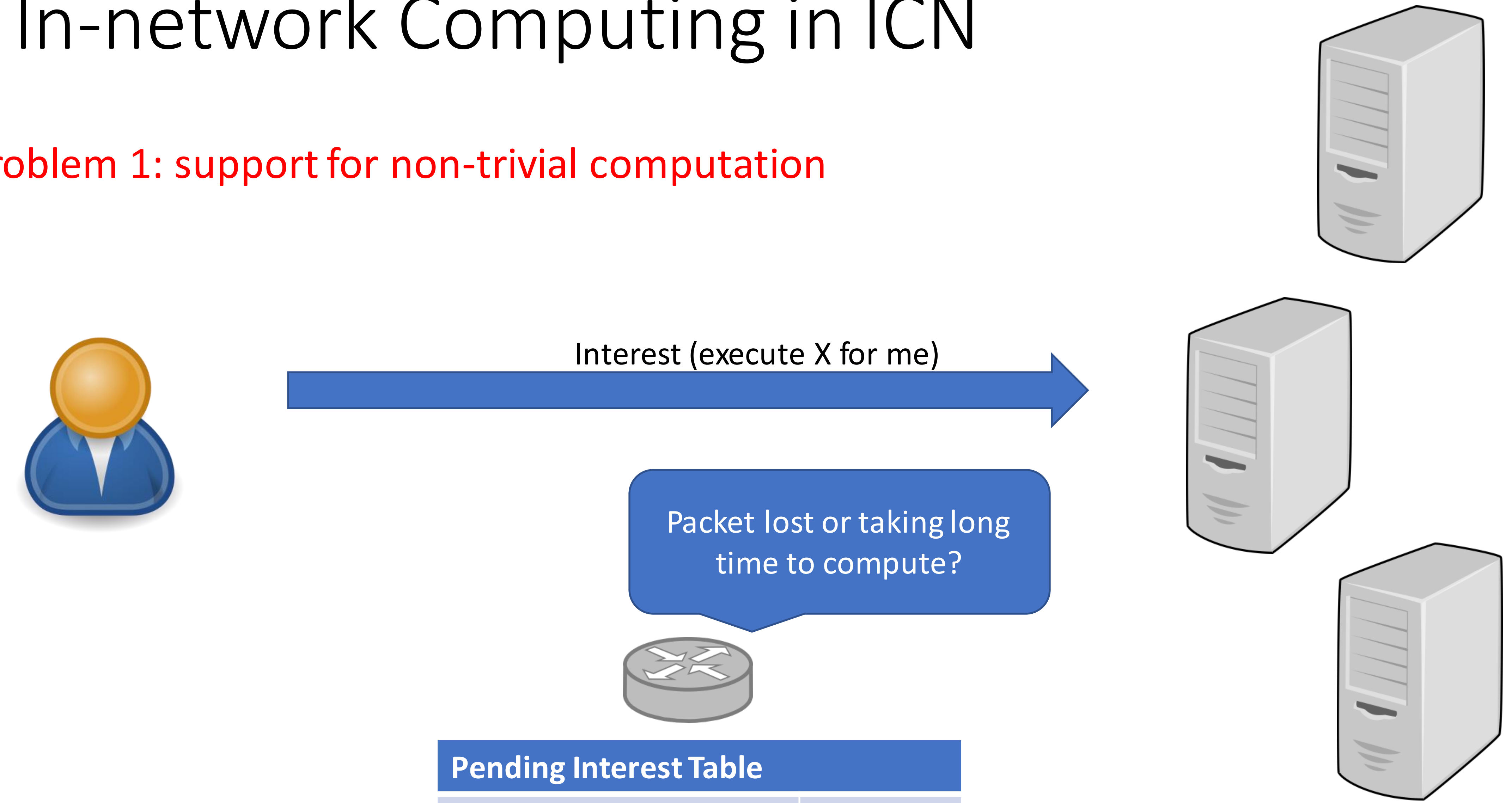

Interest (execute X for me)
Packet lost or taking long time to compute?

Pending Interest Table

Interest	**Interface**
/execute/X	eth0

In-network Computing in ICN

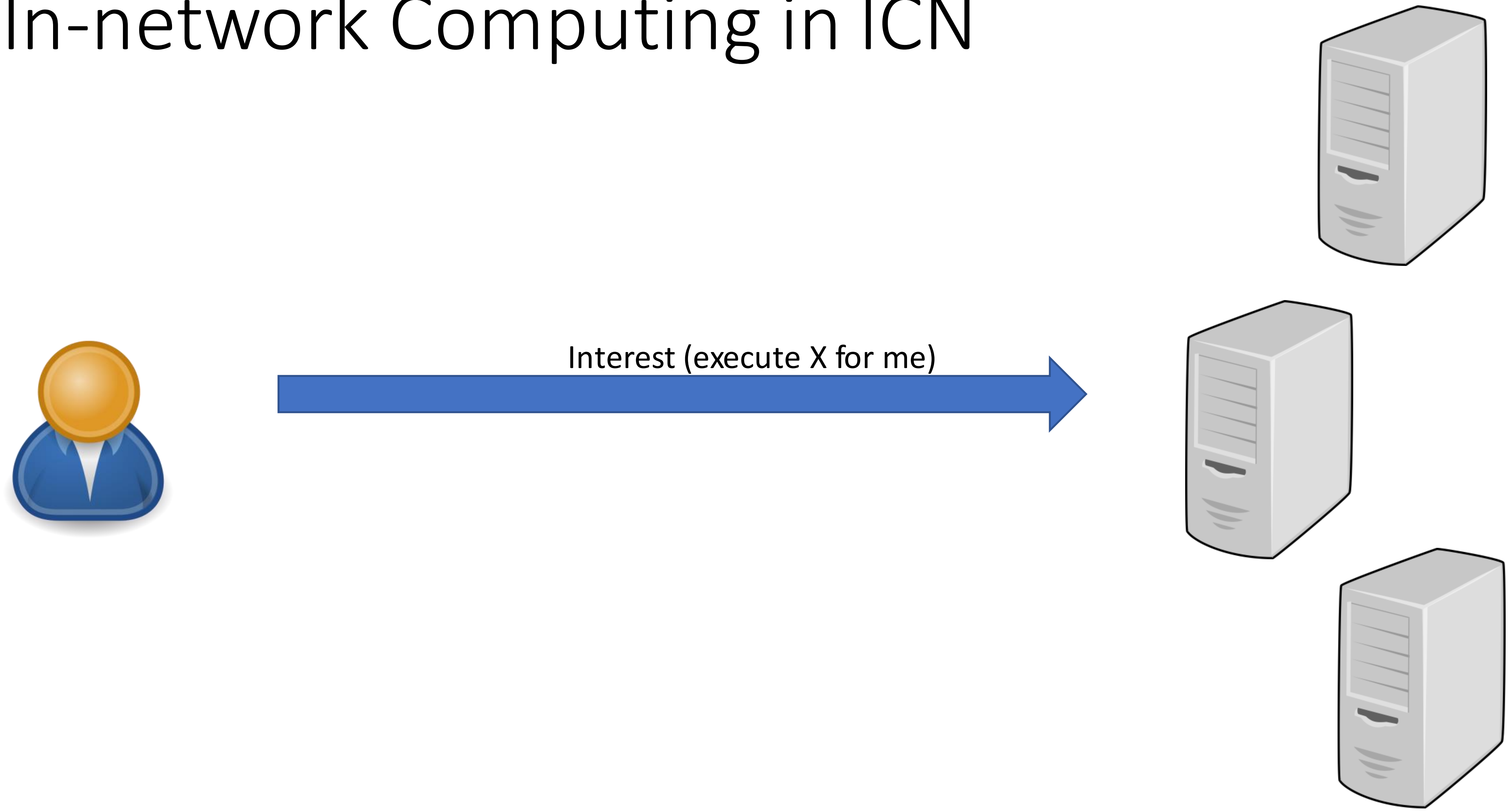
Problem 1: support for non-trivial computation


Interest (execute X for me)
Packet lost or taking long time to compute?

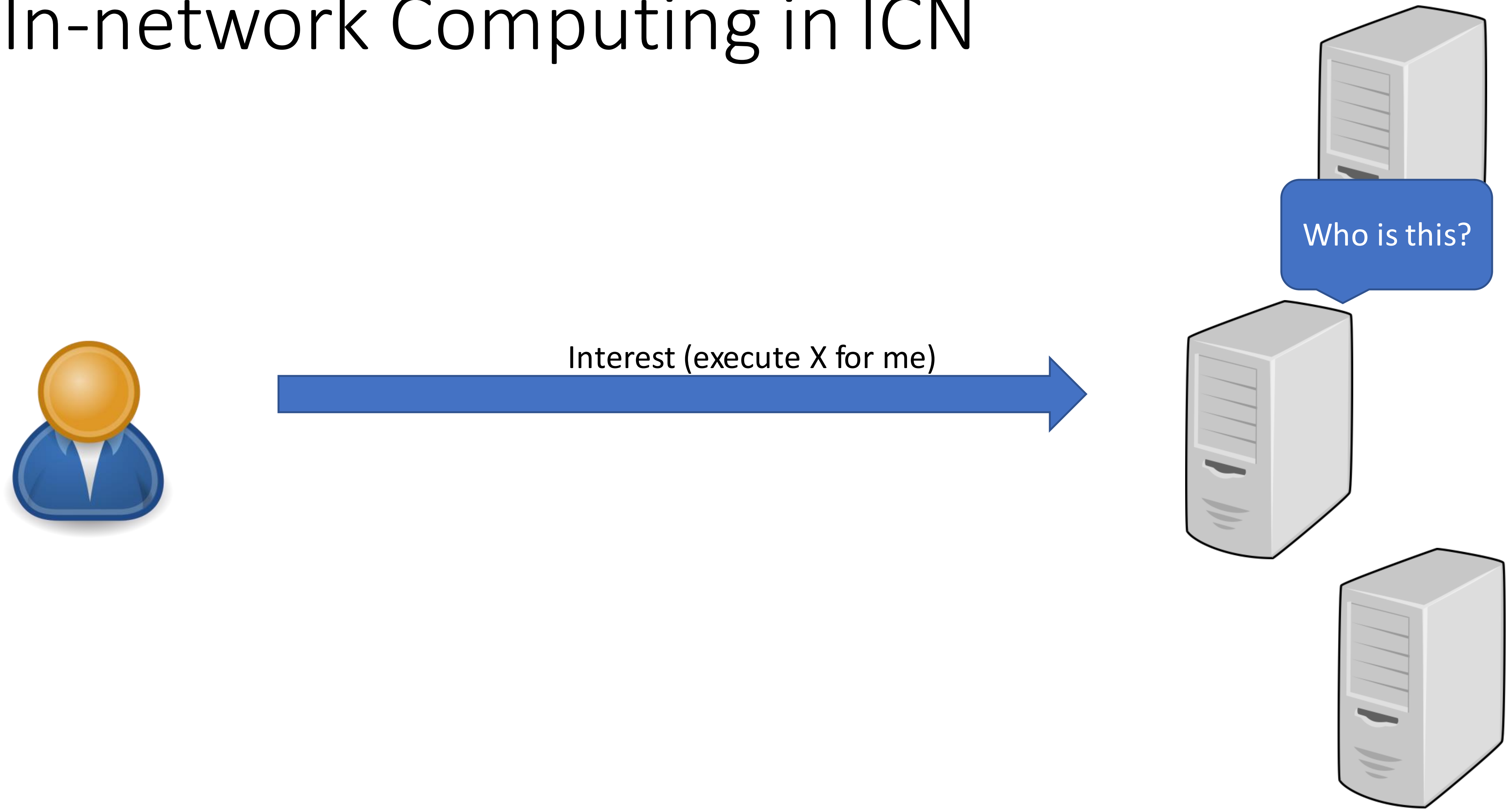
Pending Interest Table

Interest	**Interface**
/execute/X	eth0

In-network Computing in ICN

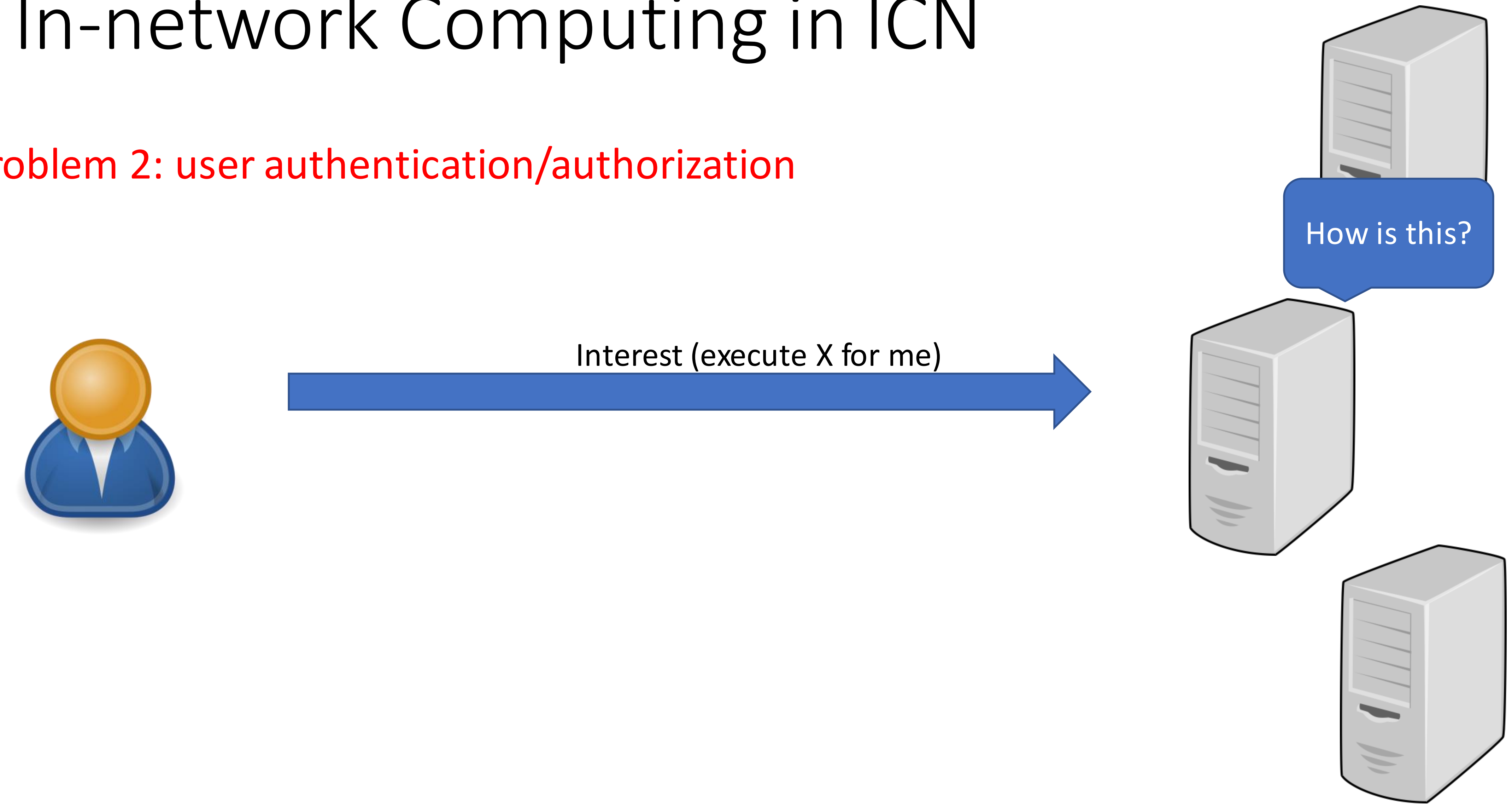


In-network Computing in ICN

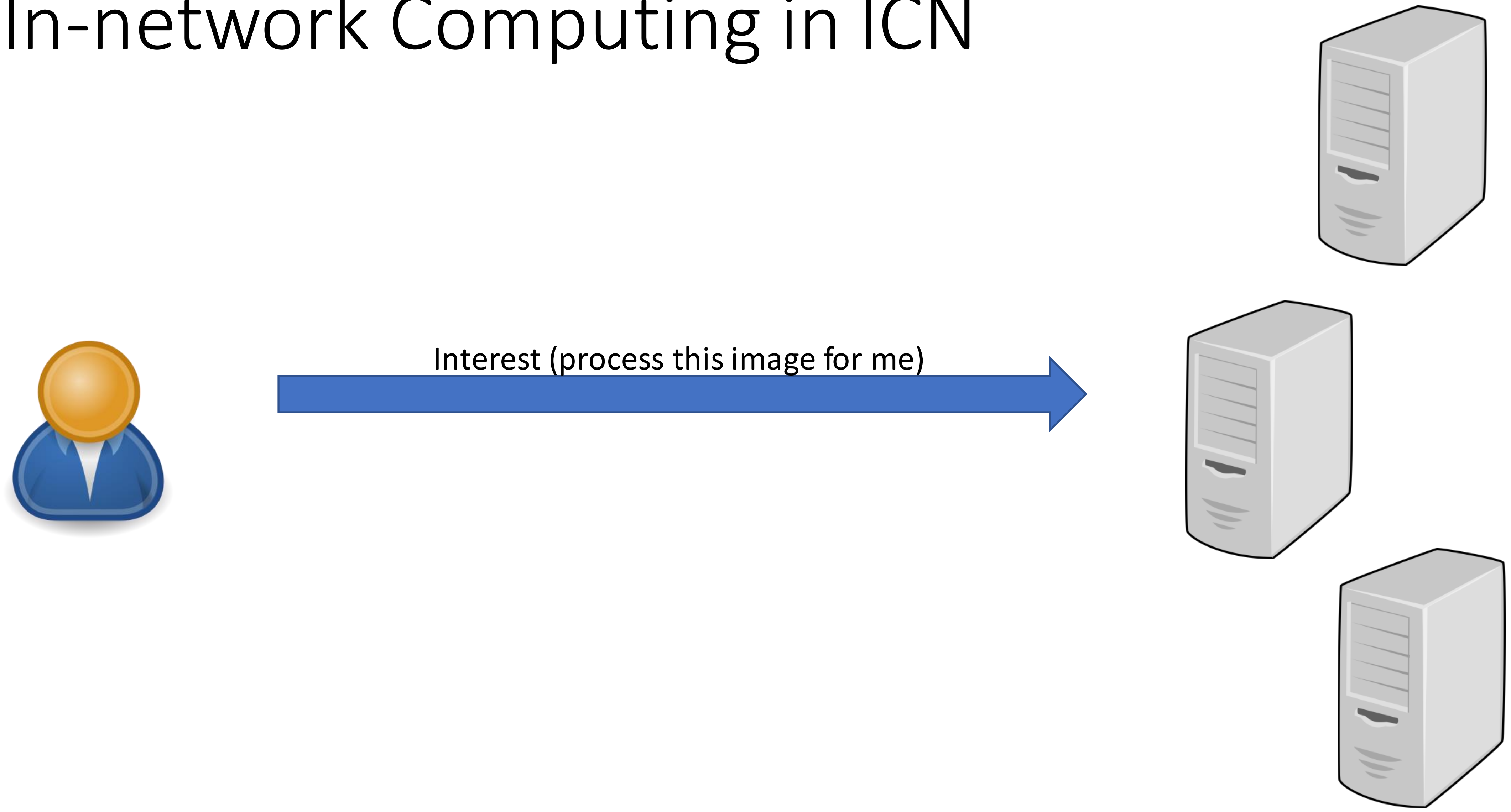


In-network Computing in ICN

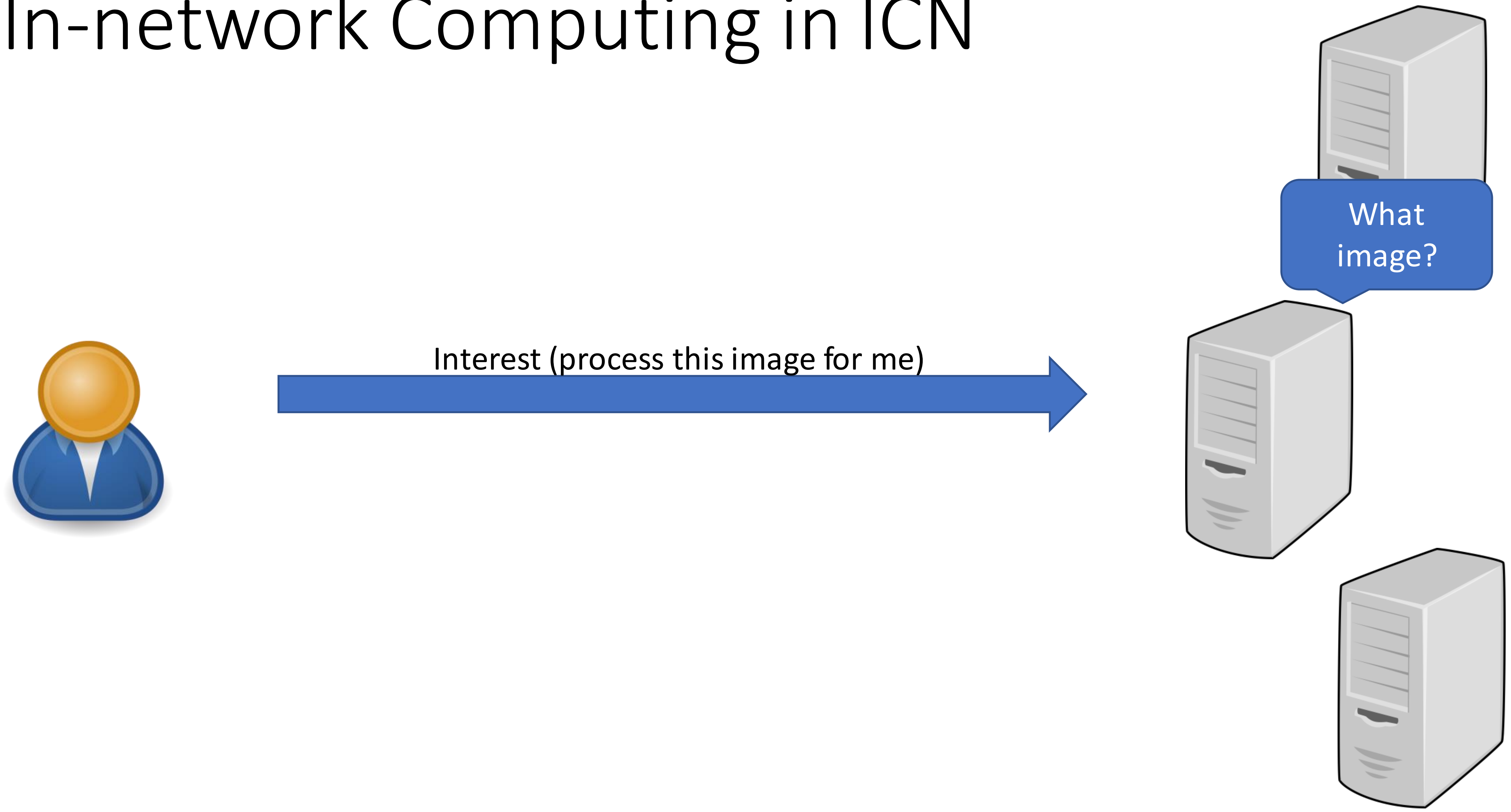
Problem 2: user authentication/authorization



In-network Computing in ICN

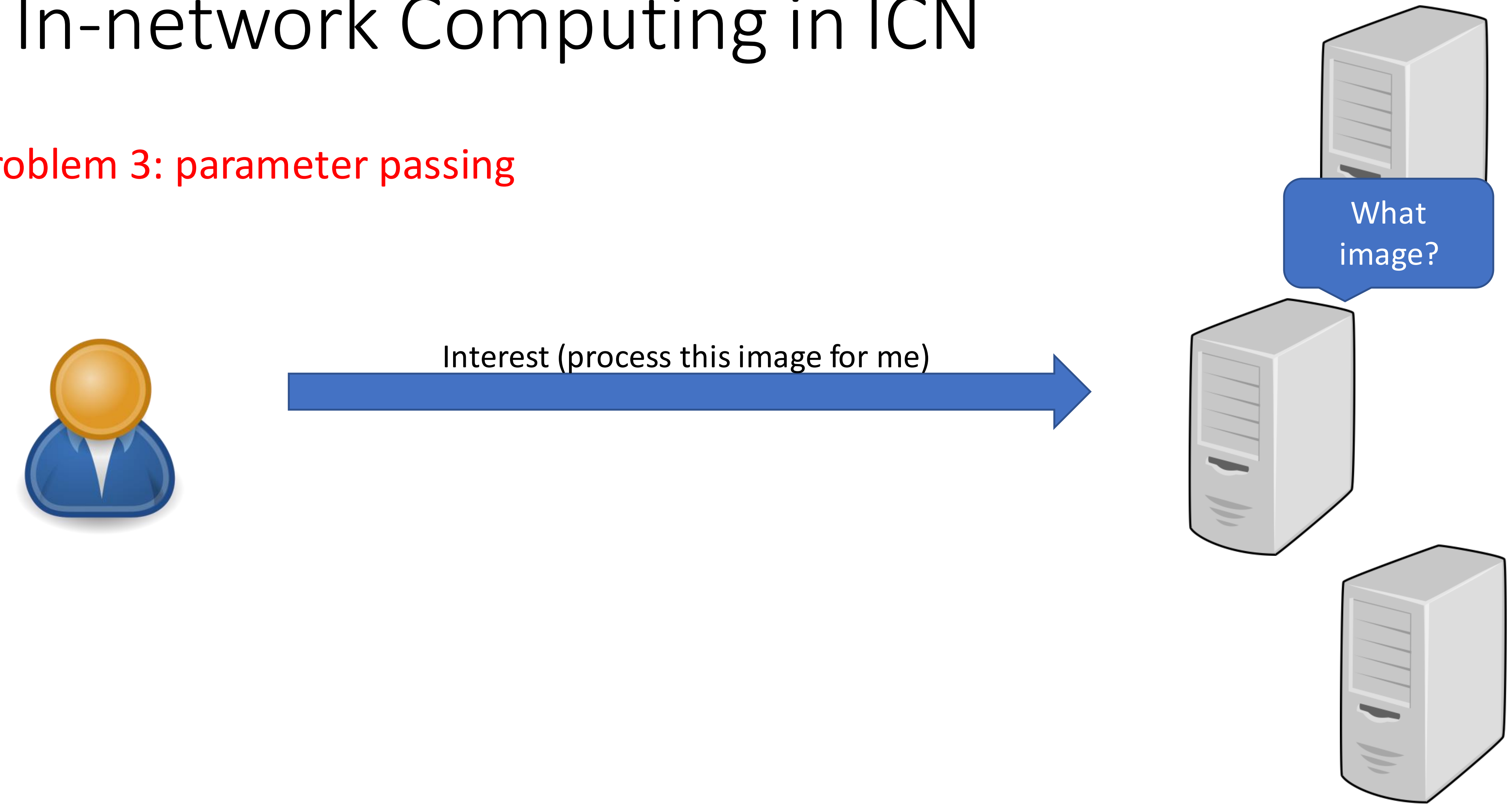


In-network Computing in ICN

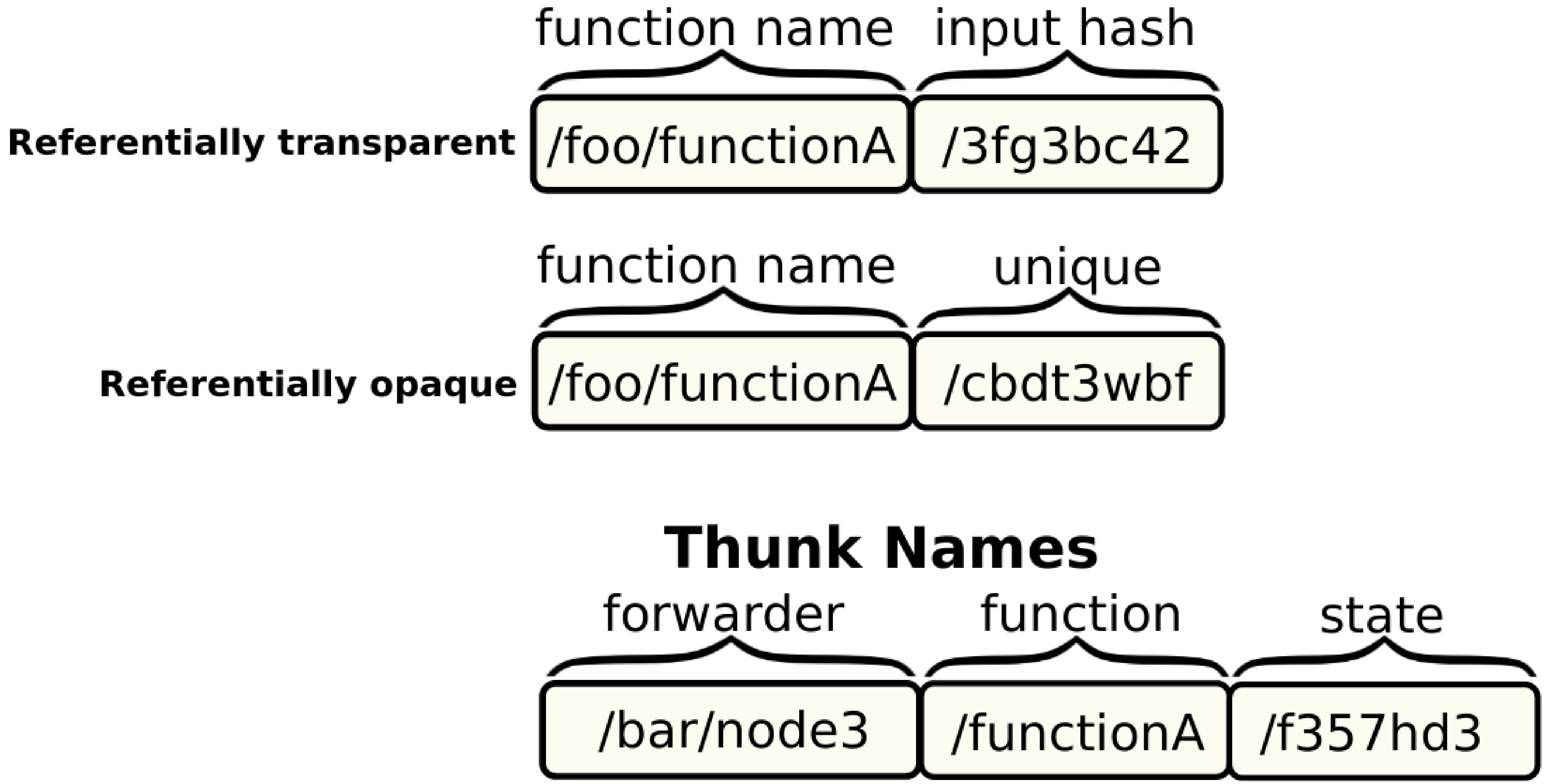


In-network Computing in ICN

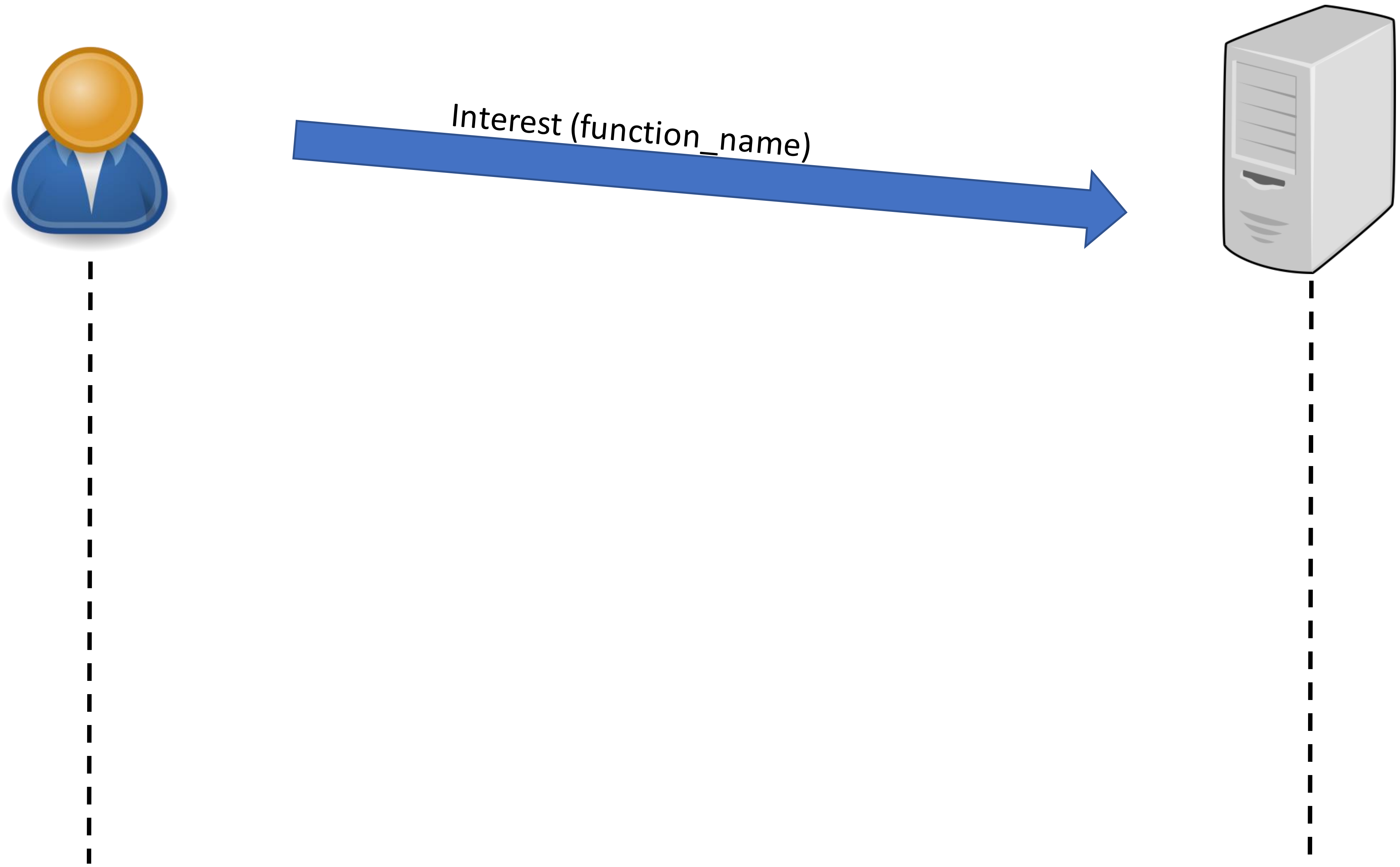
Problem 3: parameter passing



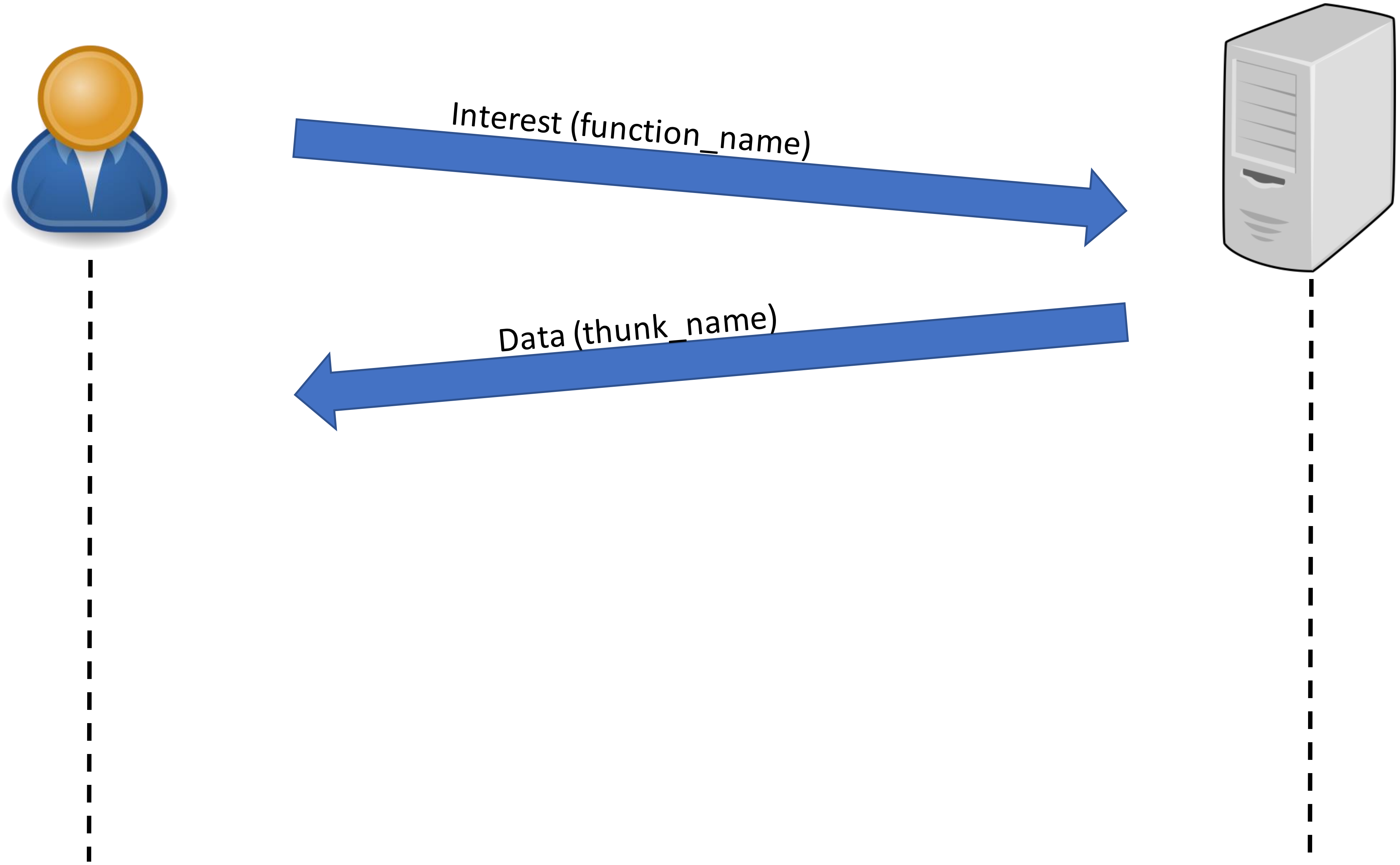
Naming


function name
input hash
Referentially transparent
/foo/functionA
/3fg3bc42
function name
unique
Referentially opaque
/foo/functionA
/cbdt3wbf
Thunk Names
forwarder
function
state
/bar/node3
/functionA
/f357hd3

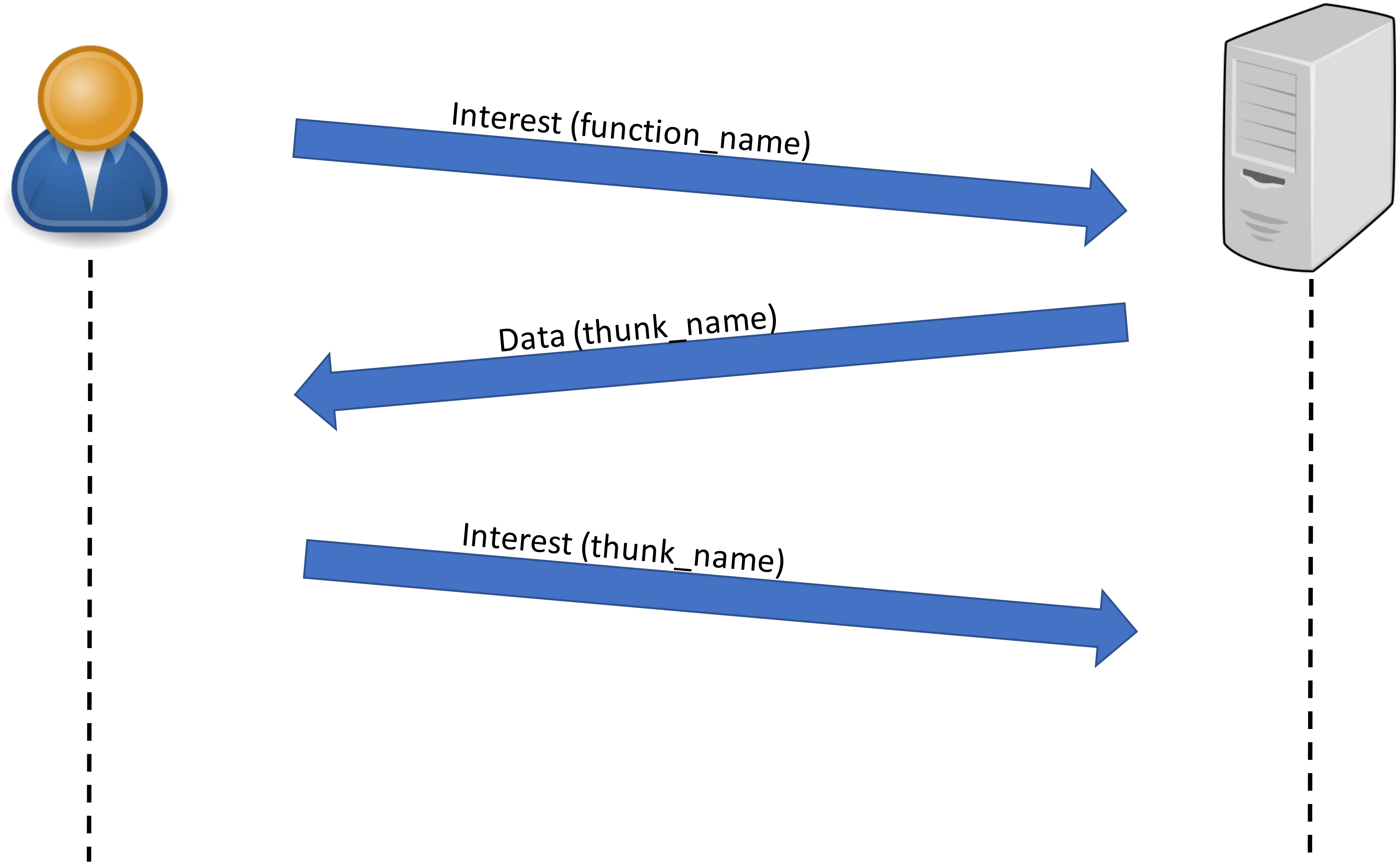
Thunks



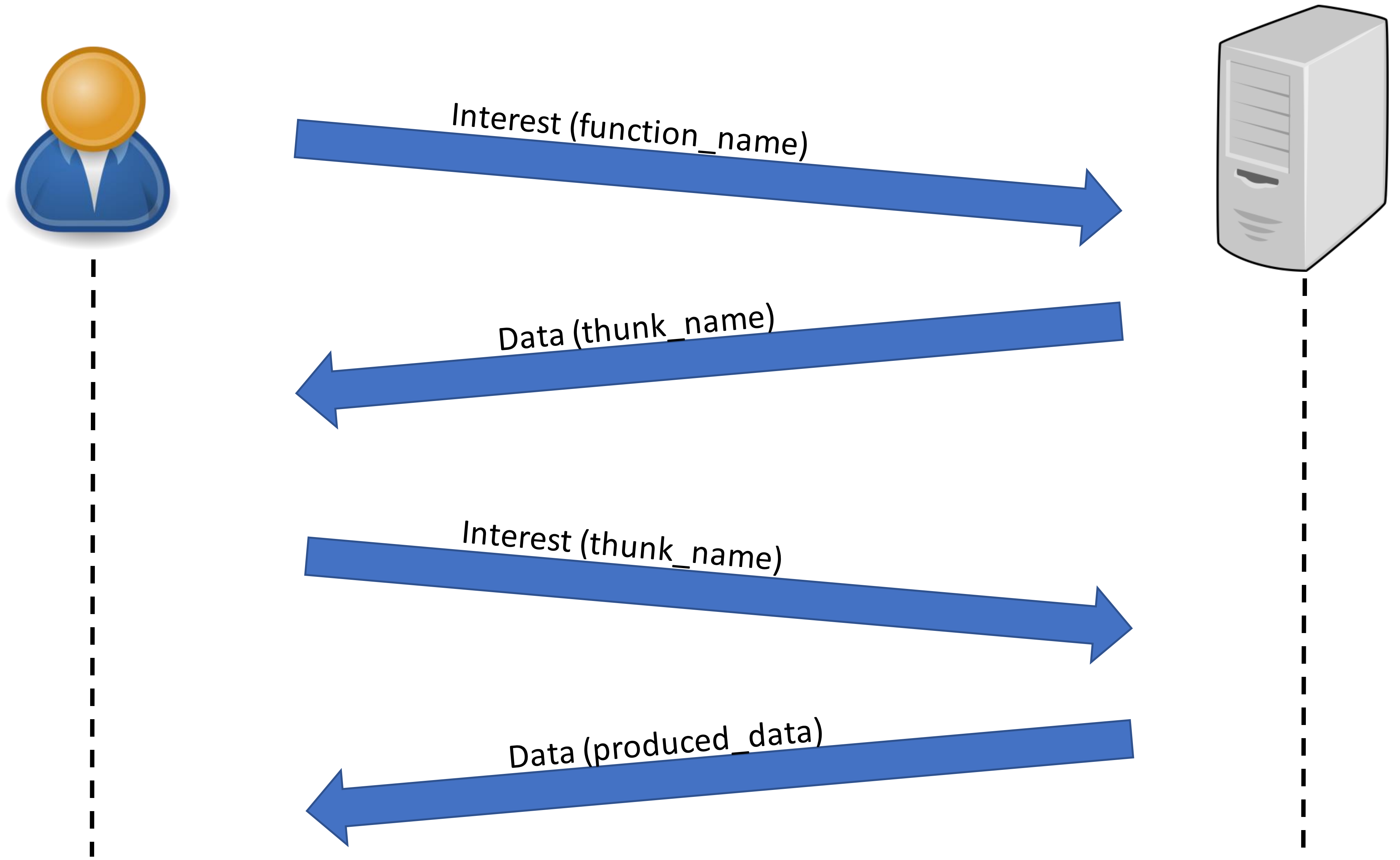
Thunks



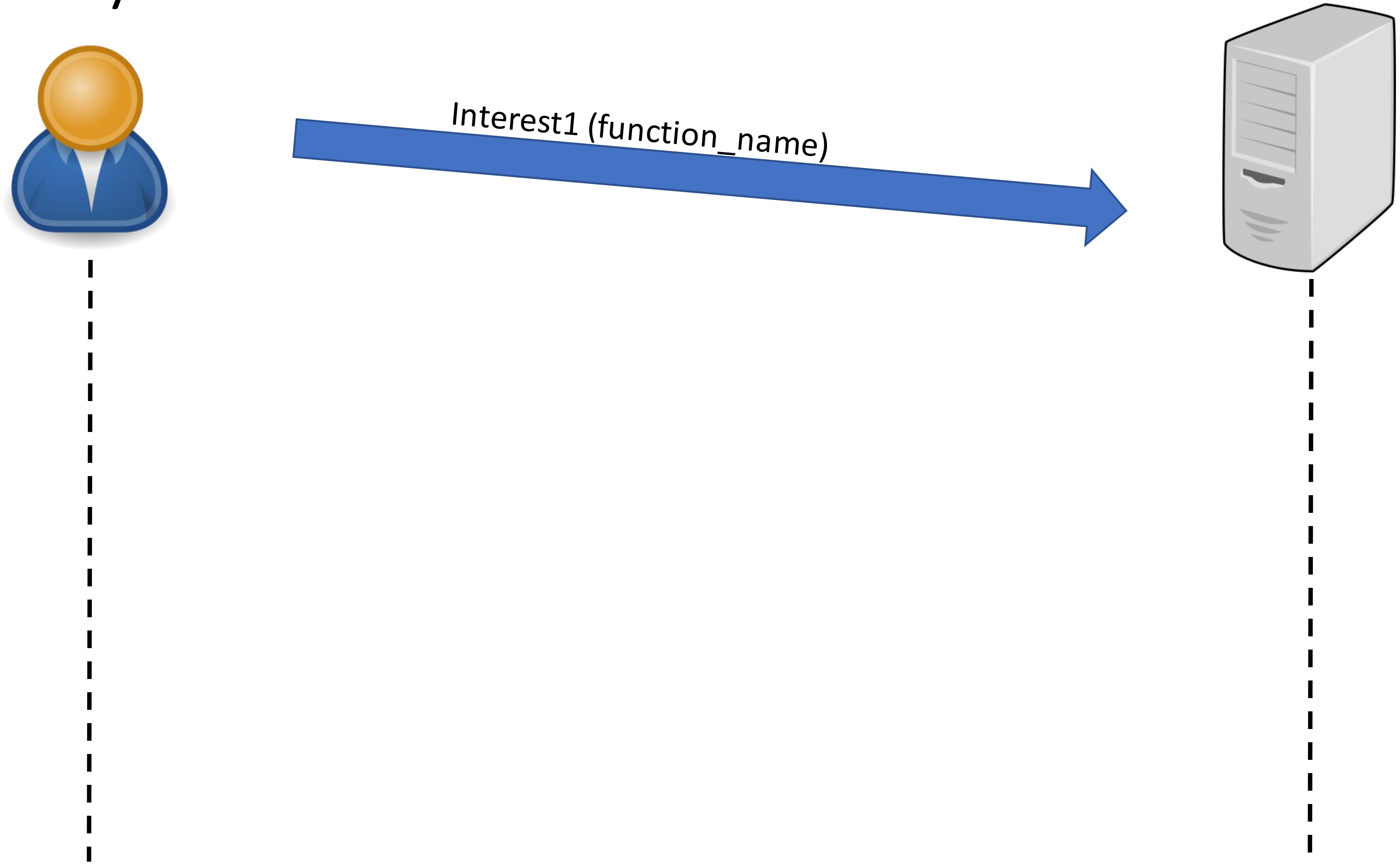
Thunks


Interest (function_name)
Data (thunk_name)
Interest (thunk_name)

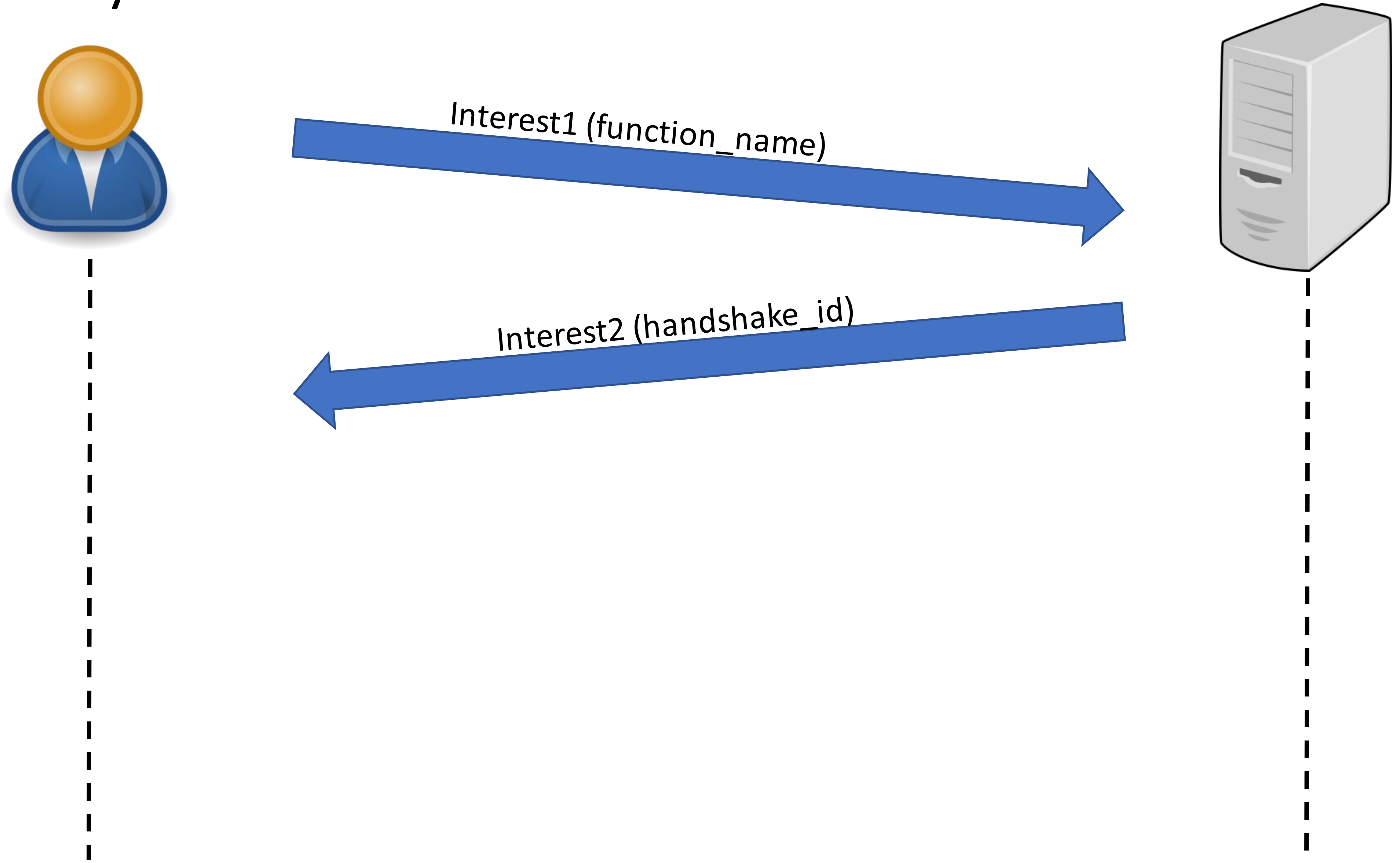
Thunks


Interest (function_name)
Data (thunk_name)
Interest (thunk_name)
Data (produced_data)

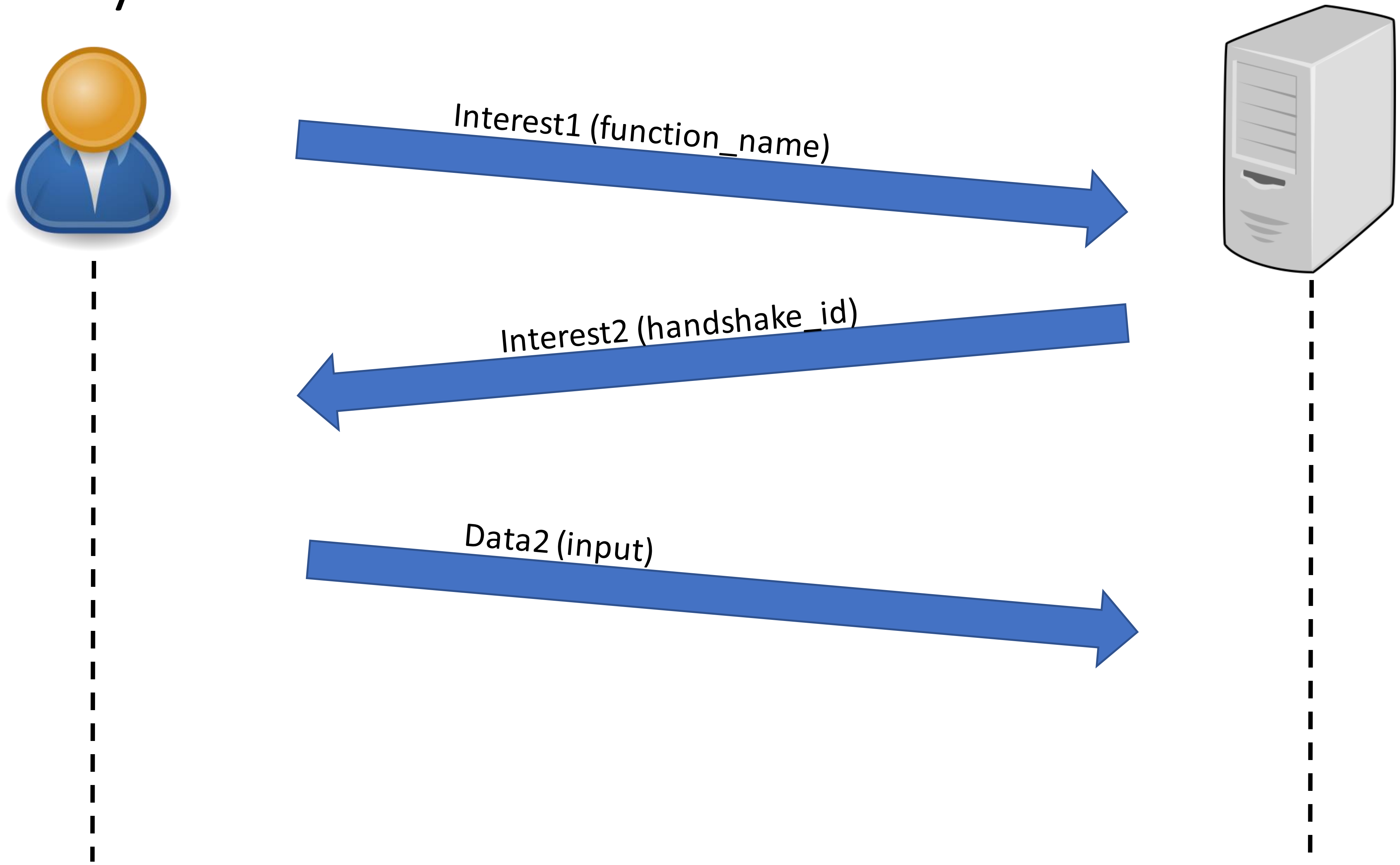
4-way handshake



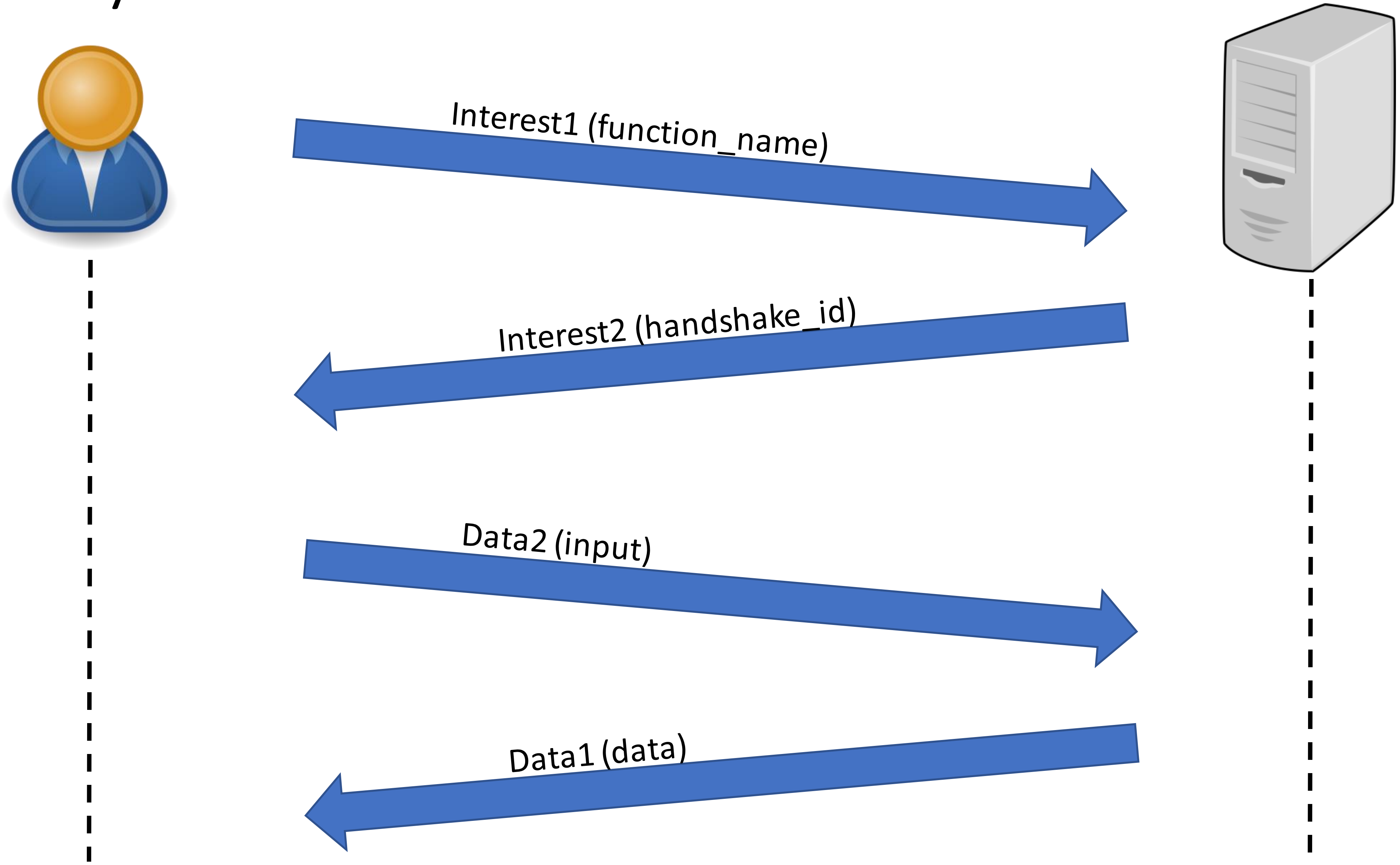
4-way handshake



4-way handshake



4-way handshake



Use case

- Airport health screening system
- Detect people with pulmonary diseases
- Collect and analyze cough audio samples
- Deployed using comodity devices

```
class CoughAnalyzer:
    #class state
    coughs = []
    alert = False


    def addSample(self, sample_f, features_f):
        sample, features =
        coughs.append([sample, features])
        if diseaseDetected(coughs):
            alert = True


def removeSpeech(sample_f):
    sample =
    # remove speech from the sample
    return anonymized_sample


def extractFeatures(sample_f):
    sample =
    # analyze the sample
    return features
########### main ###########
analyzer = CoughAnalyzer()
while True:
    sample_f = recordAudio()
    anonymized_sample_f = removeSpeech(sample_f)
    features_f = extractFeatures(anonimized_sample_f)
    analyzer.addSample(anonymized_sample_f, features_f)
```

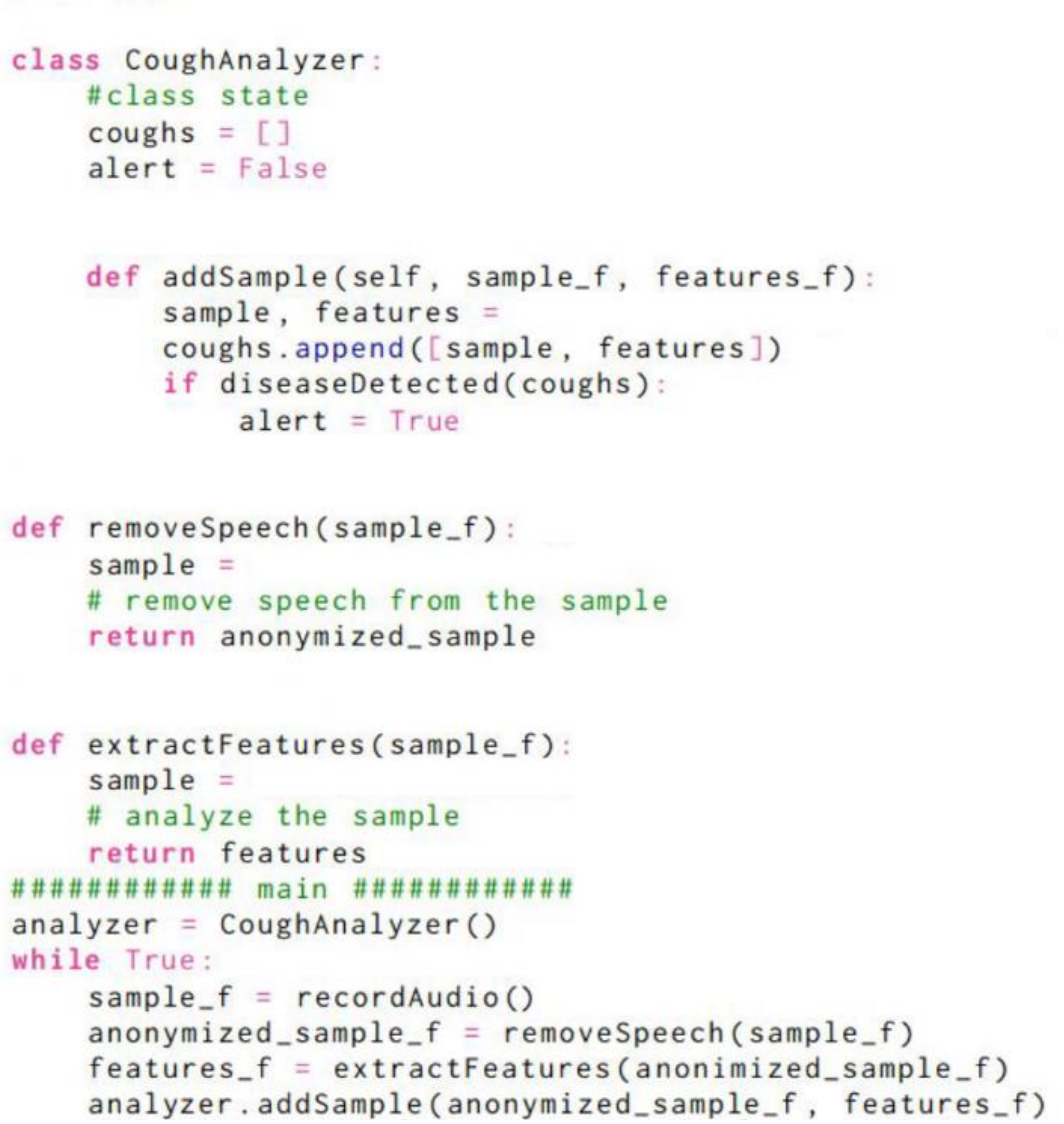
Code

Decorators:

- @cfn.transparent
- @cfn.opaque
- @cfn.actor

Methods:

- cfn.get(future)



```
class CoughAnalyzer:
    #class state
    coughs = []
    alert = False


    def addSample(self, sample_f, features_f):
        sample, features =
        coughs.append([sample, features])
        if diseaseDetected(coughs):
            alert = True


def removeSpeech(sample_f):
    sample =
    # remove speech from the sample
    return anonymized_sample


def extractFeatures(sample_f):
    sample =
    # analyze the sample
    return features
############ main ############
analyzer = CoughAnalyzer()
while True:
    sample_f = recordAudio()
    anonymized_sample_f = removeSpeech(sample_f)
    features_f = extractFeatures(anonimized_sample_f)
    analyzer.addSample(anonymized_sample_f, features_f)
```

Code

Decorators:

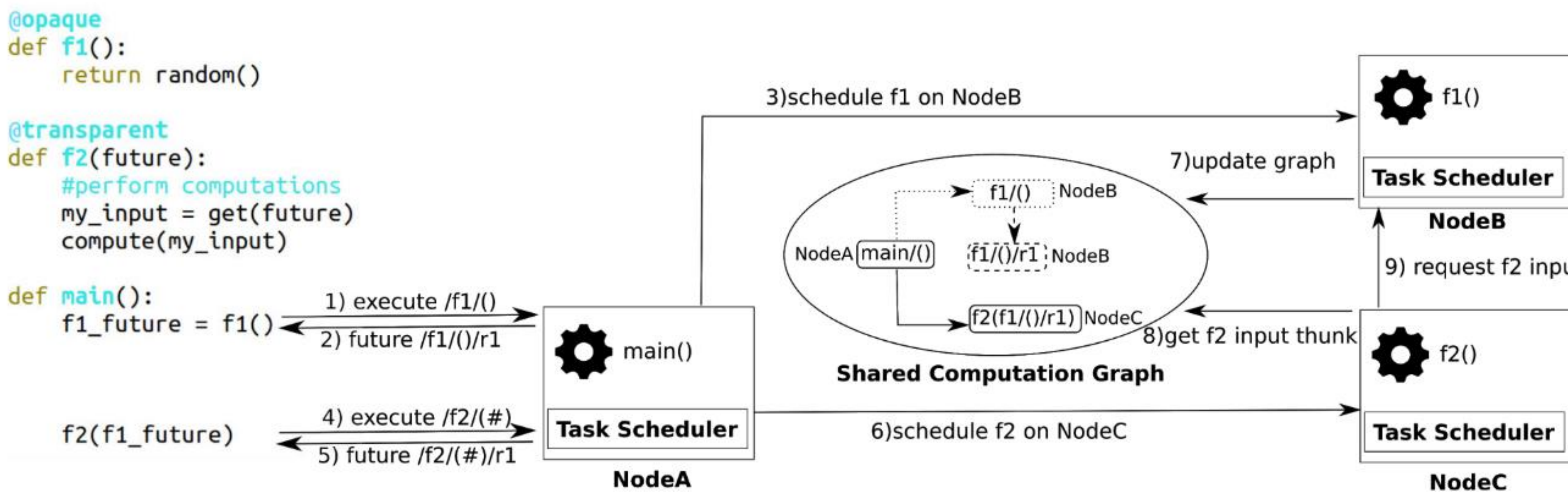
- @cfn.transparent
- @cfn.opaque
- @cfn.actor

Methods:

- cfn.get(future)

```
@cfn.actor
class CoughAnalyzer:
    #class state
    coughs = []
    alert = False

    @cfn.transparent
    def addSample(self, sample_f, features_f):
        sample, features = cfn.get(sample_f, features_f)
        coughs.append([sample, features])
        if diseaseDetected(coughs):
            alert = True

@cfn.opaque
def removeSpeech(sample_f):
    sample = cfn.get(sample_f)
    # remove speech from the sample
    return anonymized_sample

@cfn.transparent
def extractFeatures(sample_f):
    sample = cfn.get(sample_f)
    # analyze the sample
    return features
############ main ############
analyzer = CoughAnalyzer()
while True:
    sample_f = recordAudio()
    anonymized_sample_f = removeSpeech(sample_f)
    features_f = extractFeatures(anonimized_sample_f)
    analyzer.addSample(anonymized_sample_f, features_f)
```

Distributed computations

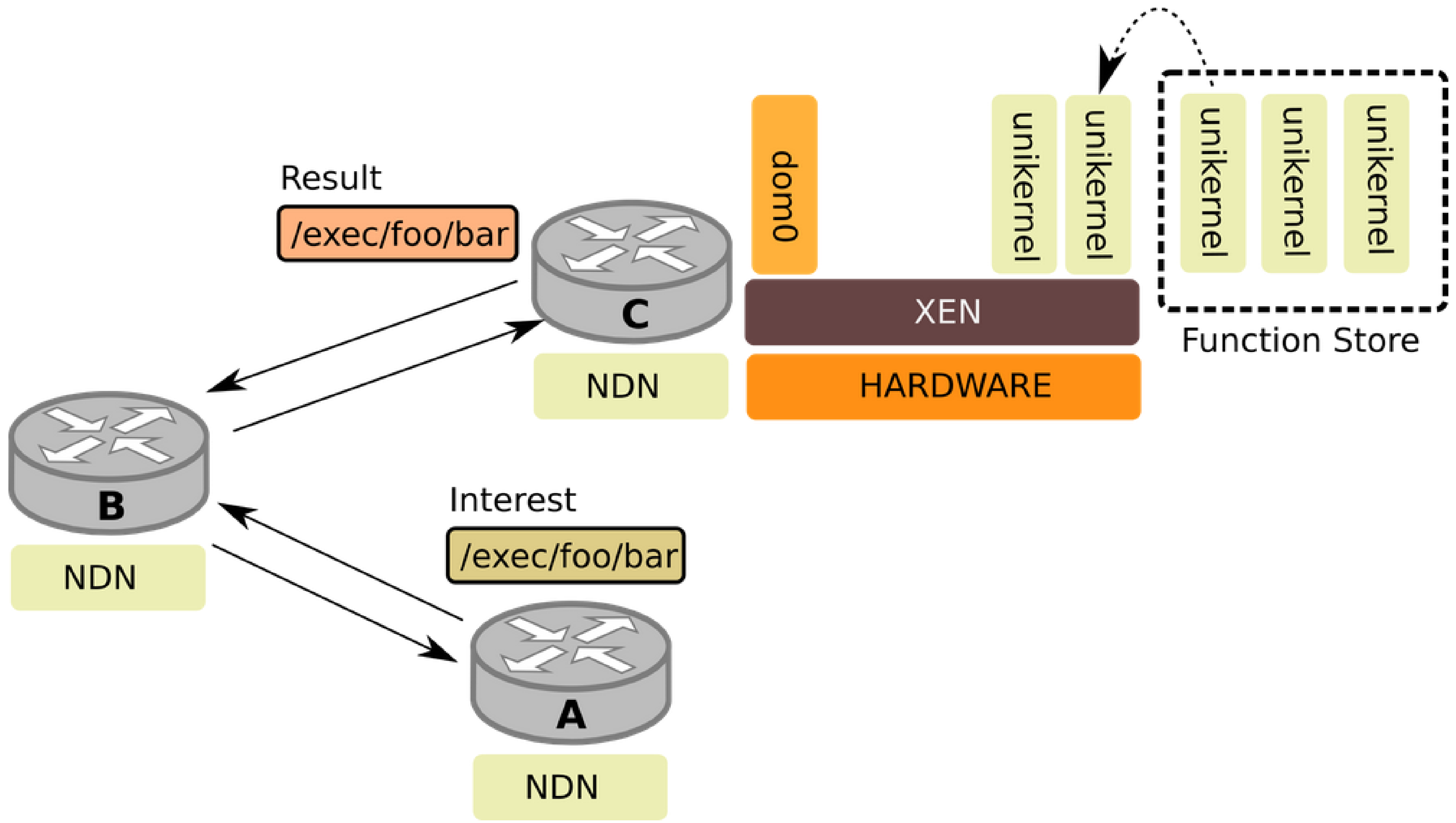


```
@opaque
def f1():
    return random()

@transparent
def f2(future):
    #perform computations
    my_input = get(future)
    compute(my_input)

def main():
    f1_future = f1()


    f2(f1_future)
```

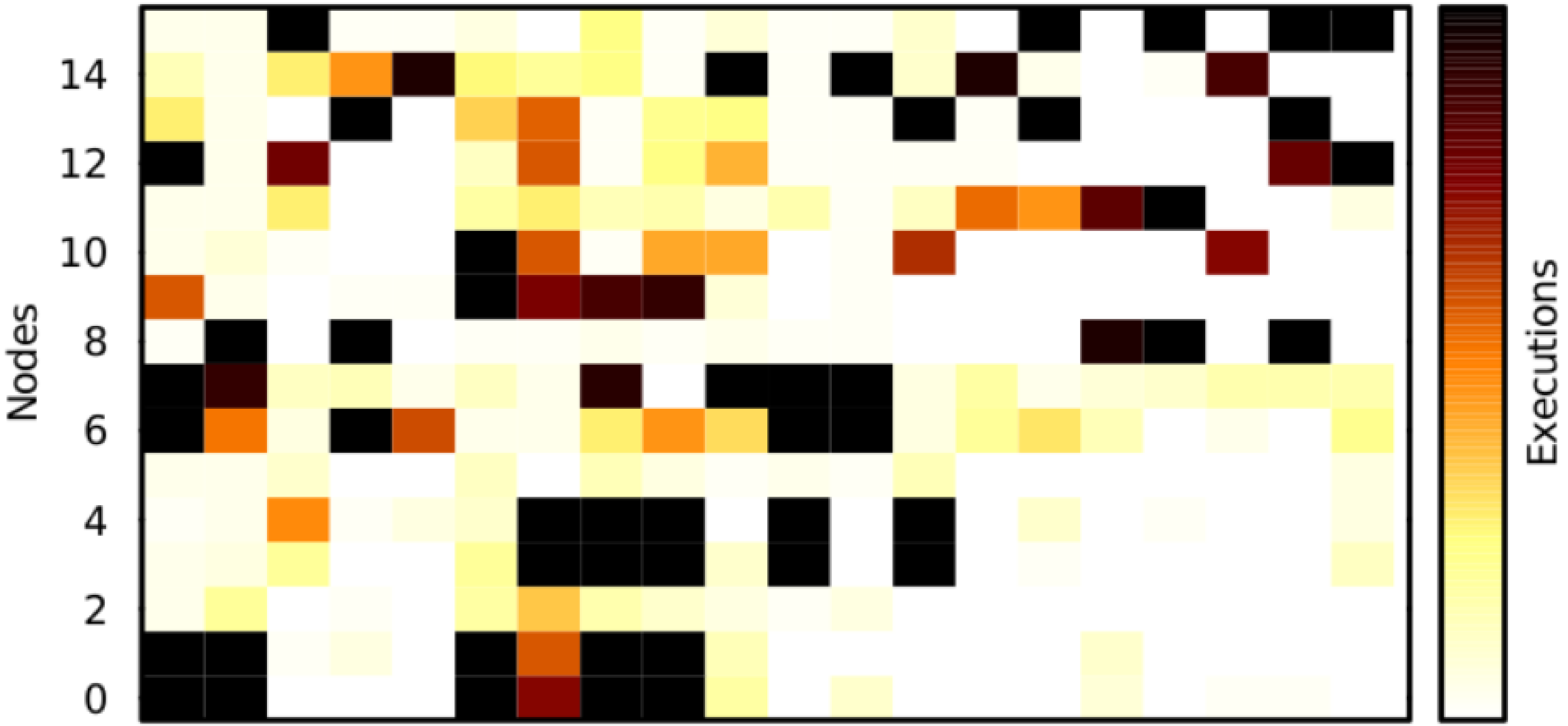
Named Function as a Service

- Function execution environment
- Self-adaptation to the application requirements
- Based on Unikernels

Named Function as a Service



Named Functions as a Service


Nodes
Executions