And Then There Were More:

Secure Communication for More Than Two Parties

Carnegie Mellon University

THE UNIVERSITY OF UTAH

Microsoft Research

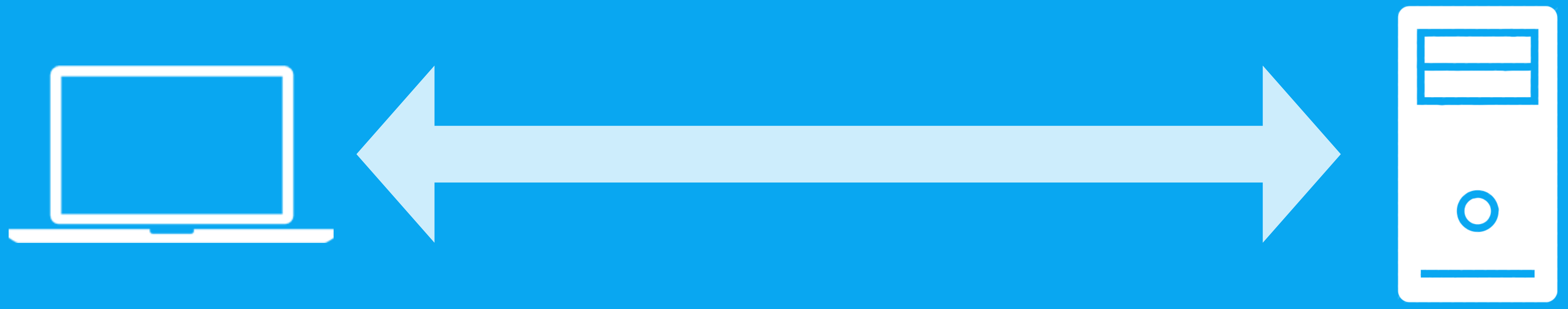
David Naylor
Carnegie Mellon

Richard Li
University of Utah

Christos Gkantsidis
Microsoft Research

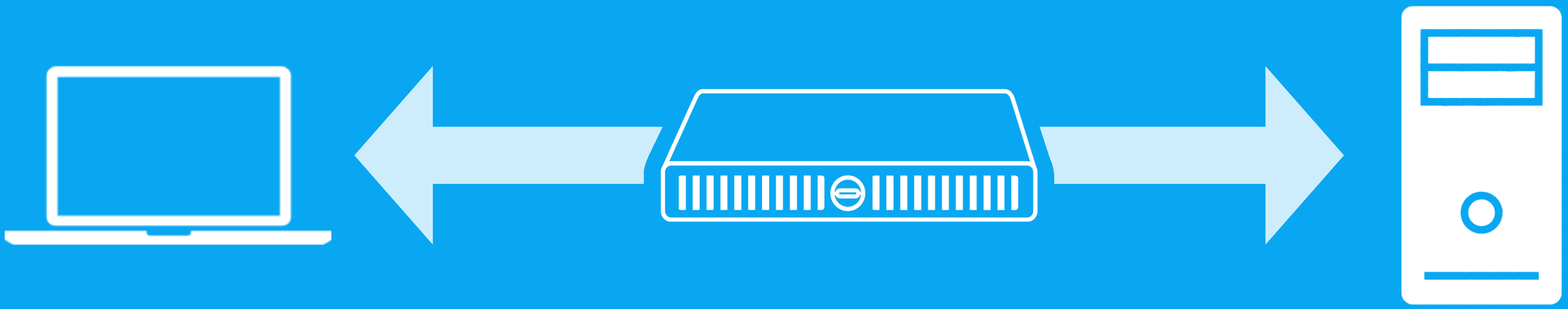
Thomas Karagiannis
Microsoft Research

Peter Steenkiste
Carnegie Mellon



In most networks,

# middleboxes ≈ # routers



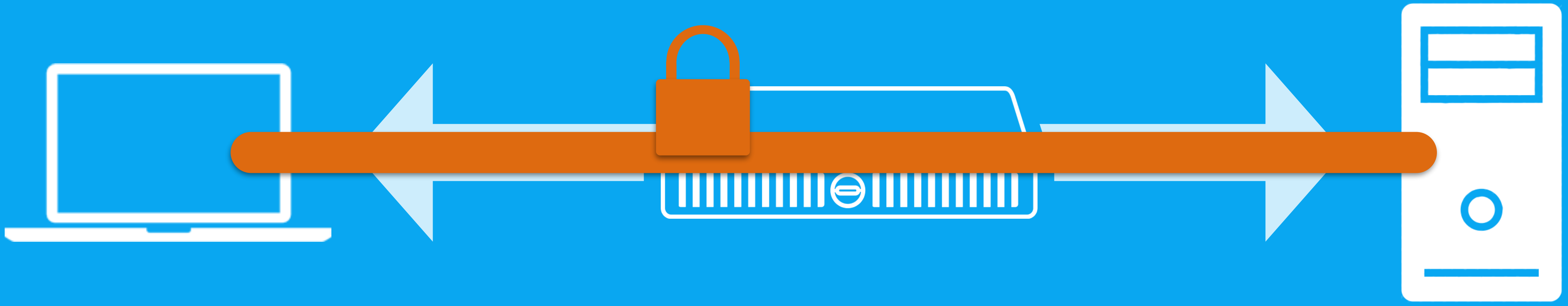
Web Cache
Compression Proxy
Intrusion Detection System

Virus Scanner
Parental Filter
Load Balancer

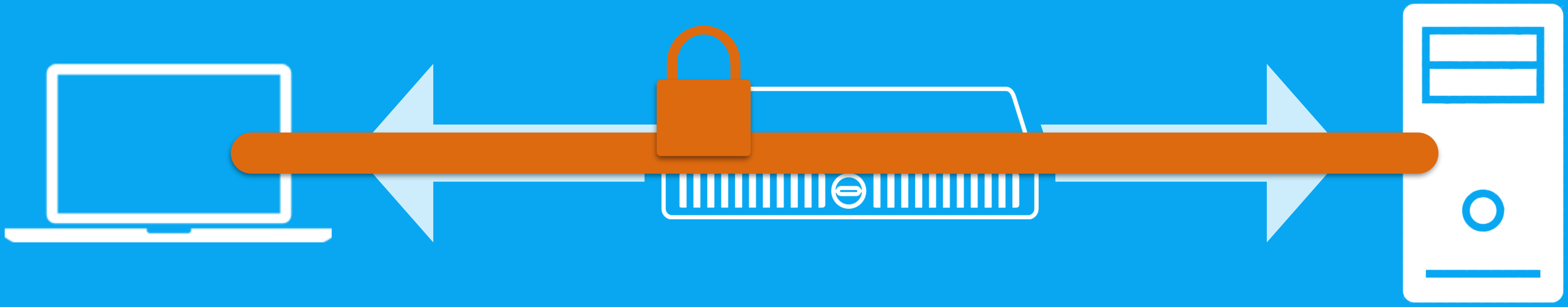
[Making Middleboxes Someone Else's Problem. *SIGCOMM '12*]

In most networks,

# middleboxes ≈ # routers



Encryption blinds middleboxes.



Encryption blinds middleboxes.

Goal: Encryption + Middleboxes

Goal: Encryption + Middleboxes

1

Design Space

For secure, multi-entity communication protocols

2

mbTLS

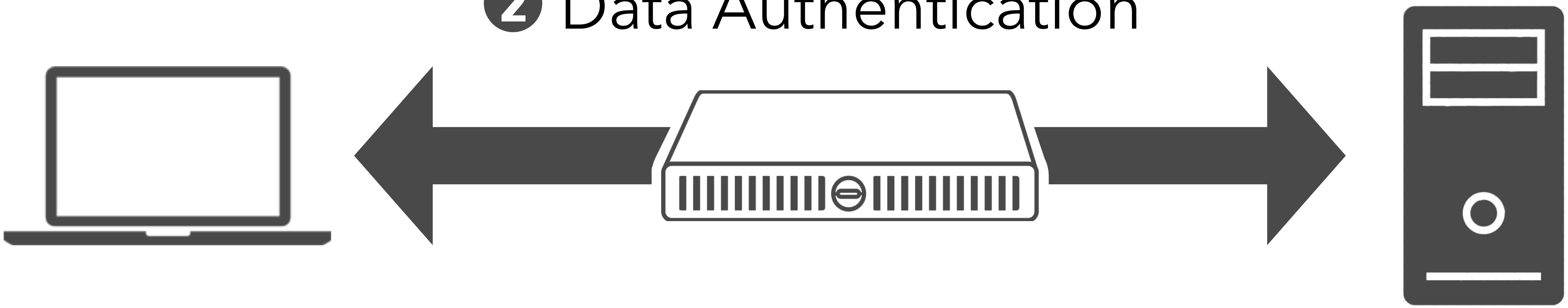
A deployable protocol for outsourced middleboxes.

There’s a **big** design space for ***secure***, ***multi-entity*** *communication protocols*

There’s a **big** design space for *secure, multi-entity* *communication protocols*

1. Extend TLS Security Properties
2. New Security Properties
3. Other Properties

① Extend TLS Security Properties

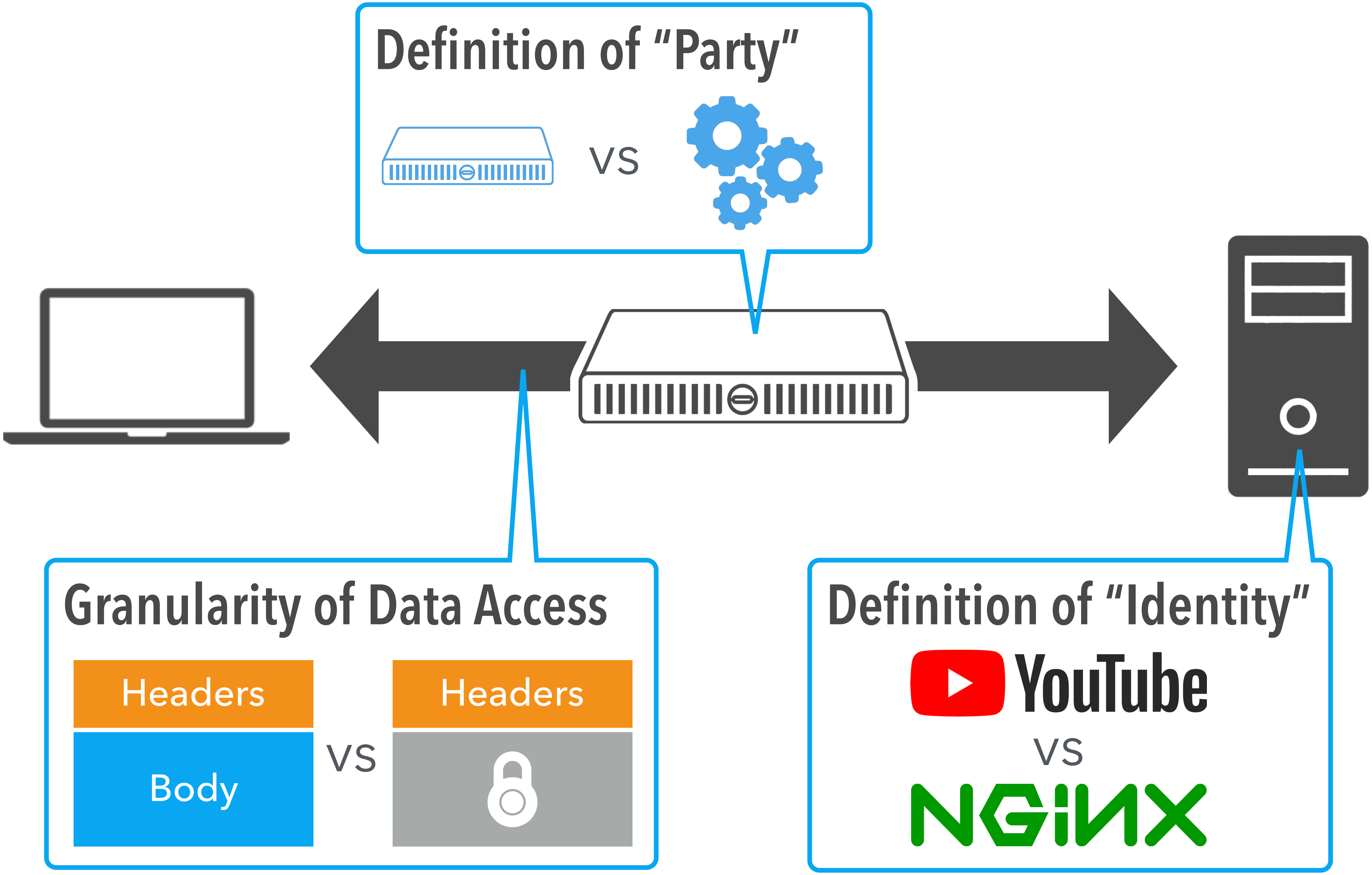


❶ Data Secrecy

❷ Data Authentication

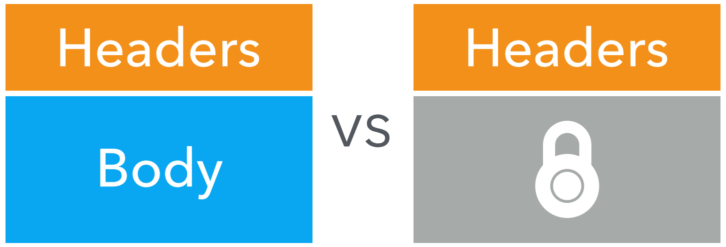
❸ Entity Authentication

① Extend TLS Security Properties

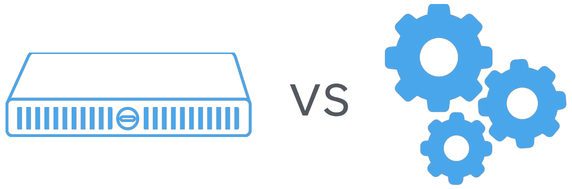


1 Extend TLS Security Properties

Granularity of Data Access



Definition of "Party"



Definition of "Identity"

YouTube

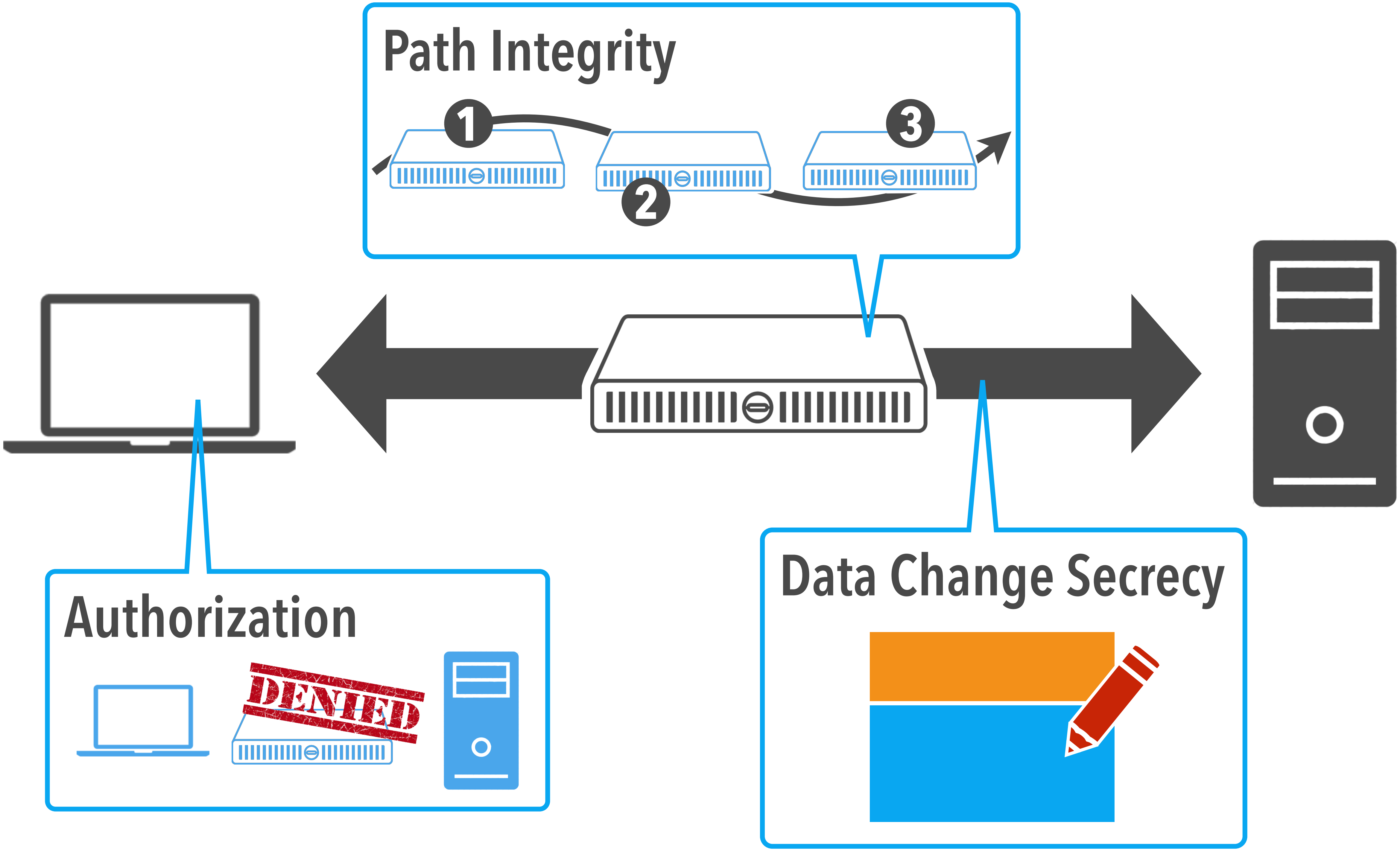
VS

NGINX

2 New Security Properties

3 Other Properties

2 New Security Properties



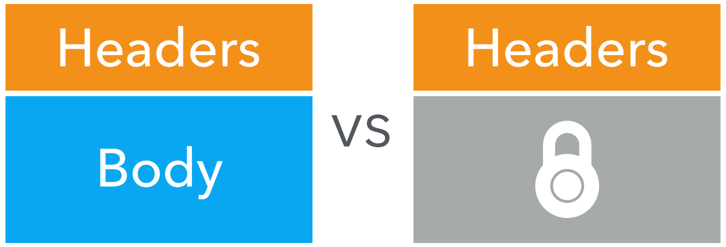
Path Integrity

Authorization

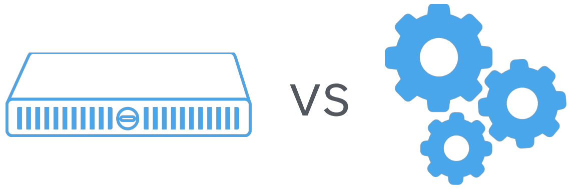
Data Change Secrecy

1 Extend TLS Security Properties

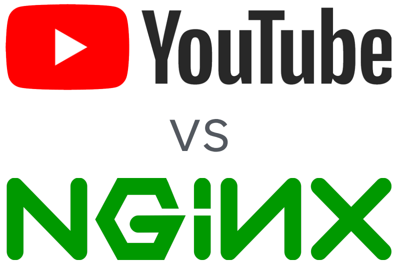
Granularity of Data Access



Definition of "Party"

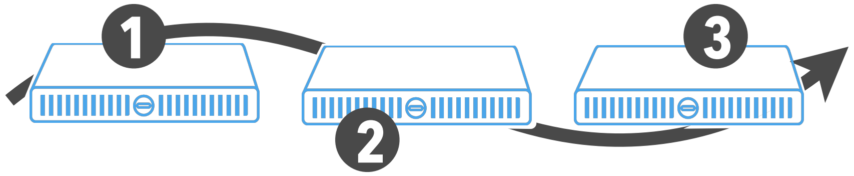


Definition of "Identity"



2 New Security Properties

Path Integrity



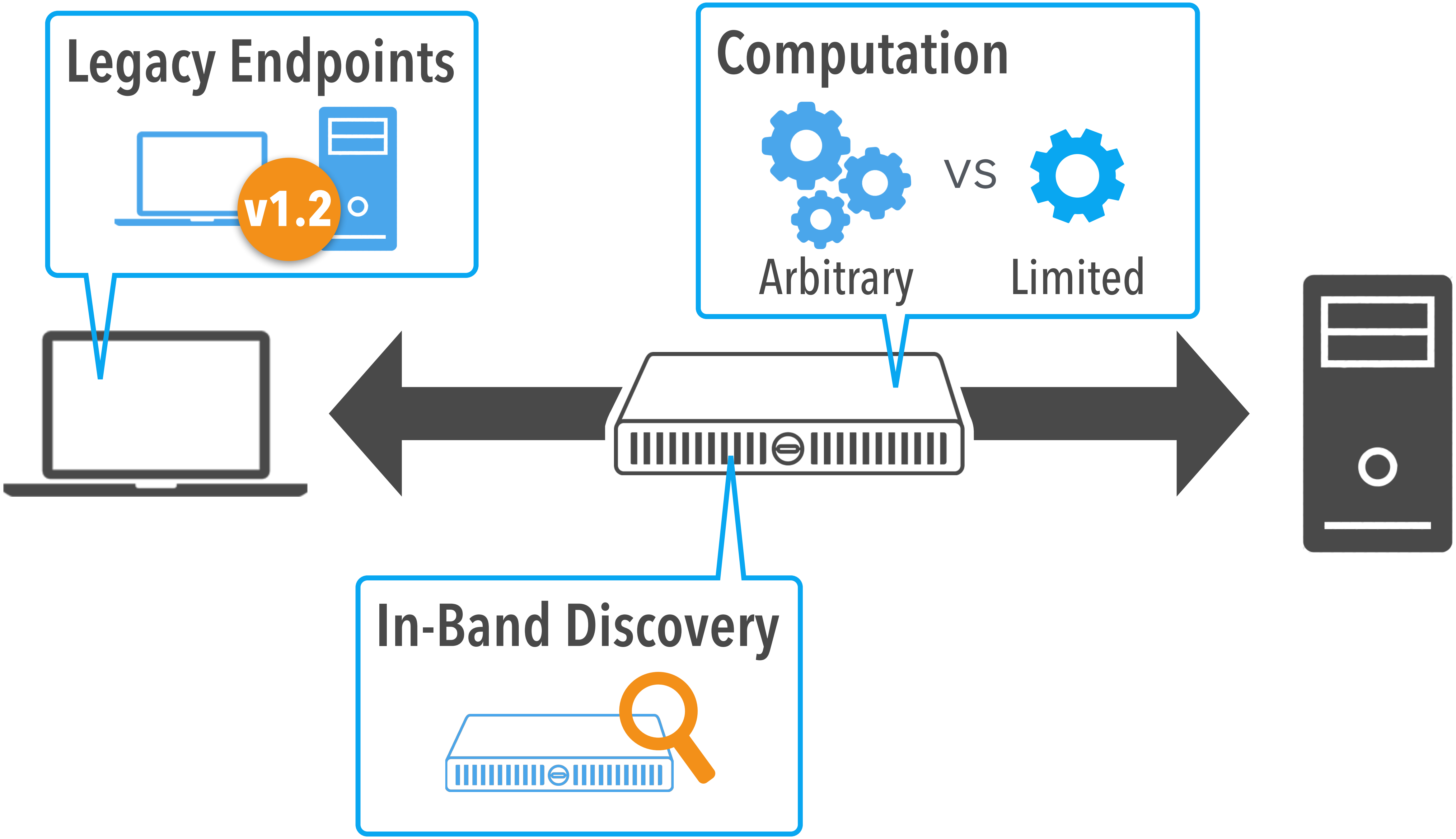
Data Change Secrecy

Authorization

3 Other Properties

3

Other Properties



Legacy Endpoints

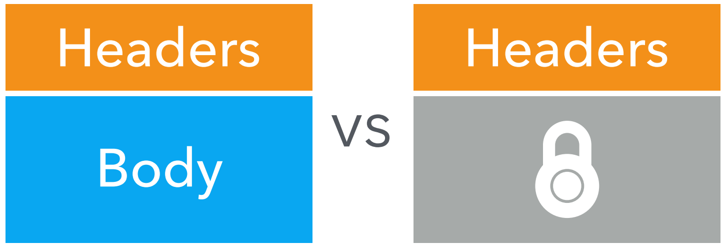
Computation

Arbitrary vs Limited

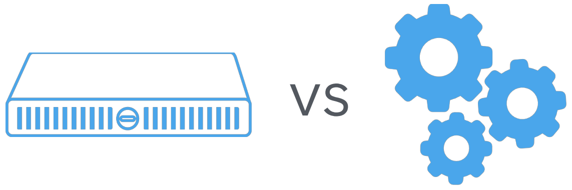
In-Band Discovery

1 Extend TLS Security Properties

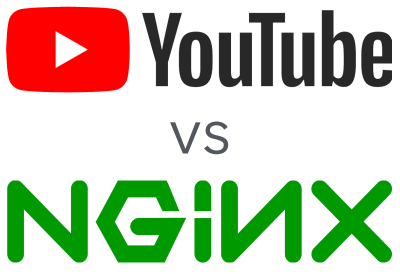
Granularity of Data Access



Definition of “Party”

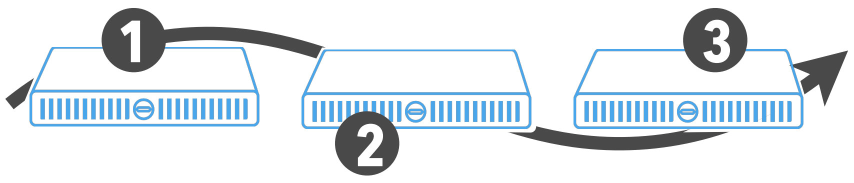


Definition of “Identity”



2 New Security Properties

Path Integrity



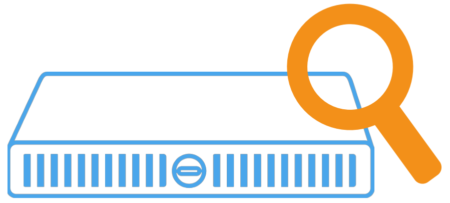
Data Change Secrecy

Authorization

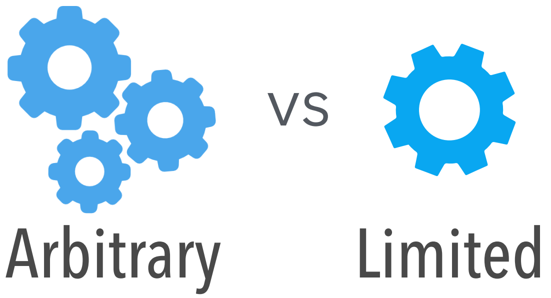
3 Other Properties

Legacy Endpoints

In-Band Discovery



Computation



There's a **big** design space for *secure*, *multi-entity* *communication protocols*

1. Extend TLS Security Properties
2. New Security Properties
3. Other Properties

There's a **big** design space for *secure, multi-entity communication protocols*

There is no one-size-fits-all solution.

There’s a **big** design space for ***secure, multi-entity*** *communication protocols*

There is no one-size-fits-all solution.

Supporting one property often precludes another.

Supporting one property often precludes another.

TLS interception with custom root certificates

Supports

two legacy endpoints

Prevents

endpoint authentication (owner or code)

YouTube

vs

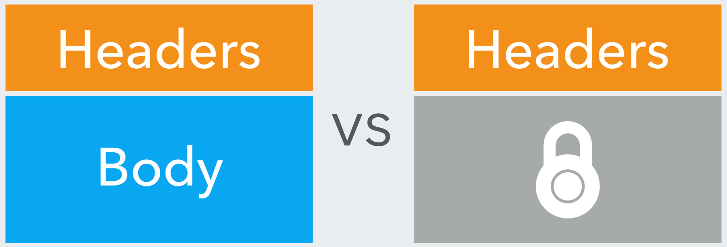
NGINX

Supporting one property often precludes another.

Multi-Context TLS (mcTLS) [SIGCOMM '15]

Supports

fine-grained data access



Prevents

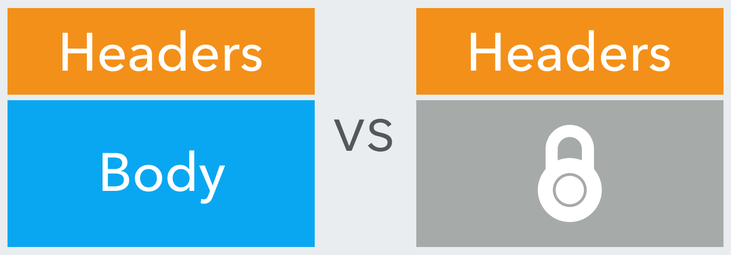
legacy support

Supporting one property often precludes another.

BlindBox [SIGCOMM '15]

Supports

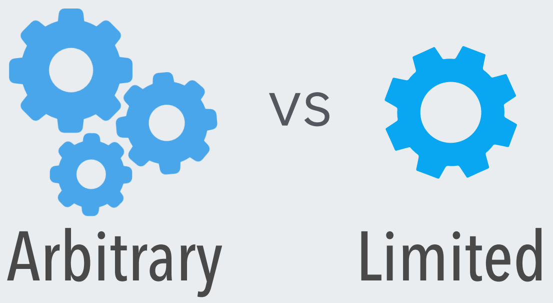
functional crypto
(minimal data access)



Headers
Body
vs
Headers

Prevents

arbitrary computation



Arbitrary
vs
Limited

There's a **big** design space for ***secure, multi-entity*** *communication protocols*

There is no one-size-fits-all solution.

Supporting one property often precludes another.

There's a **big** design space for ***secure***, ***multi-entity*** *communication protocols*

There is no one-size-fits-all solution.

Supporting one property often precludes another.

Goal: Encryption + Middleboxes

1

Design Space

For secure, multi-entity communication protocols

2

mbTLS

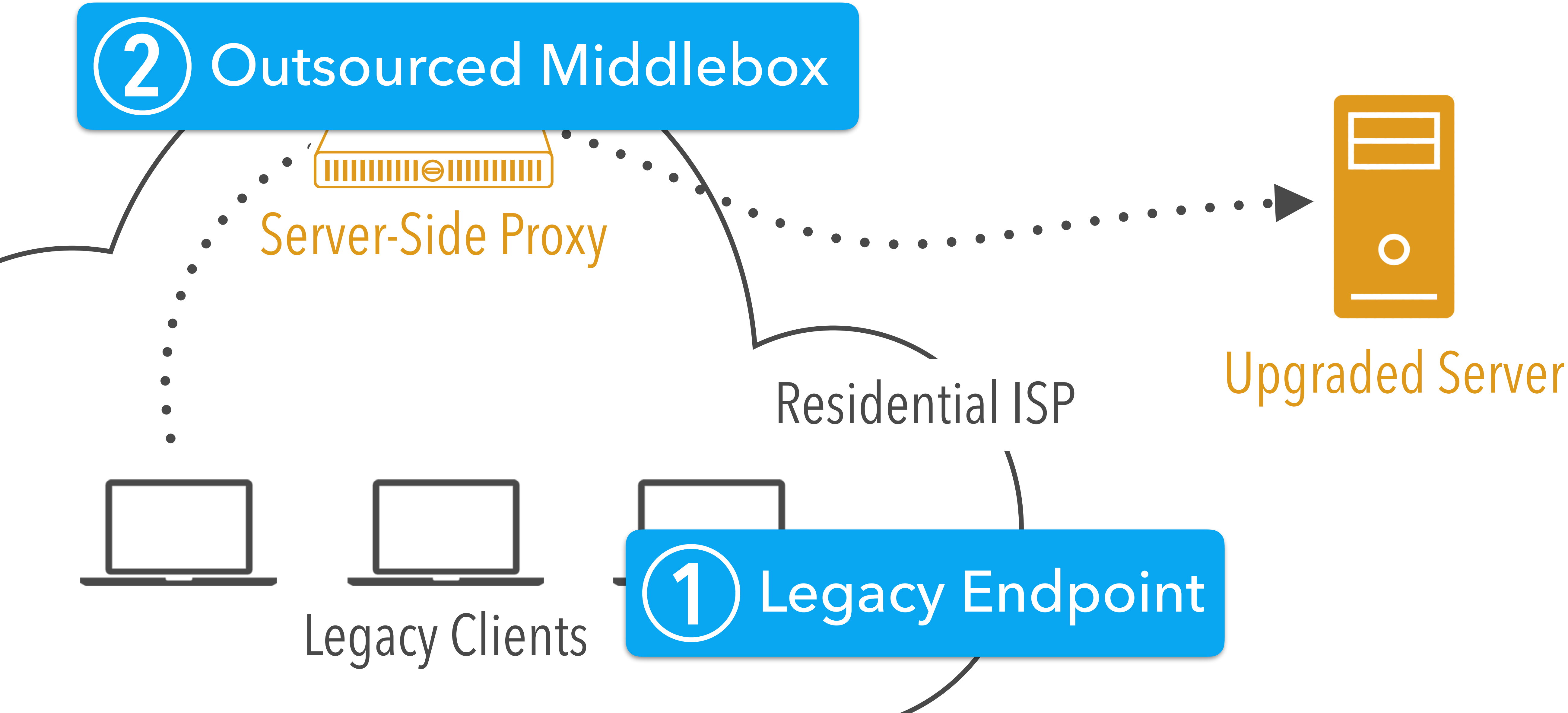
A deployable protocol for outsourced middleboxes.

mbTLS targets two common-case, real-world needs

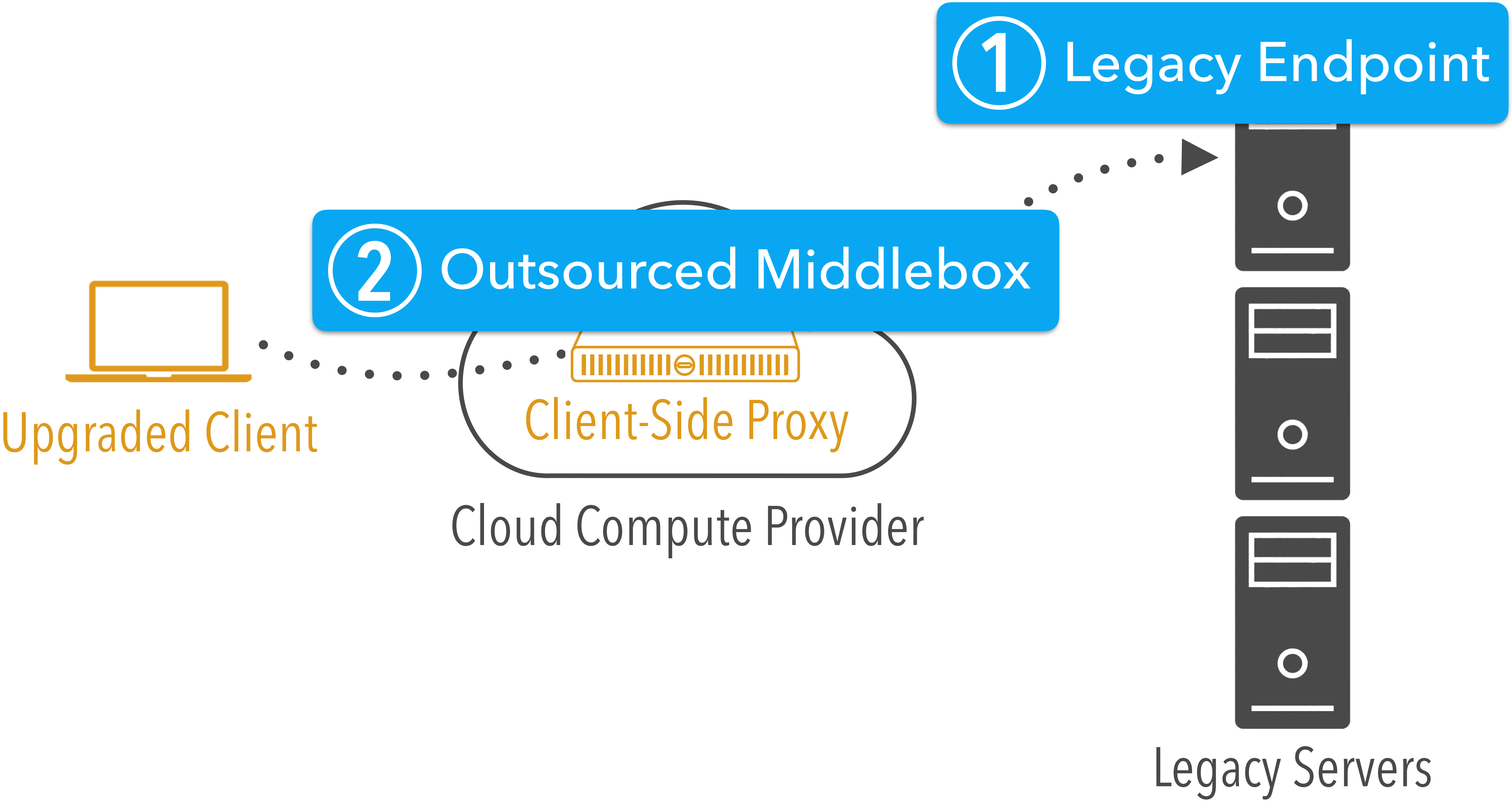
1. **Immediate deployability**
   Interoperate with one legacy endpoint

2. **Protection for outsourced middleboxes**
   Protect session data from middlebox infrastructure
   (in addition to traditional network attackers)

mbTLS targets two common-case, real-world needs



mbTLS targets two common-case, real-world needs



mbTLS targets two common-case, real-world needs

1. **Immediate deployability**

   Interoperate with one legacy endpoint

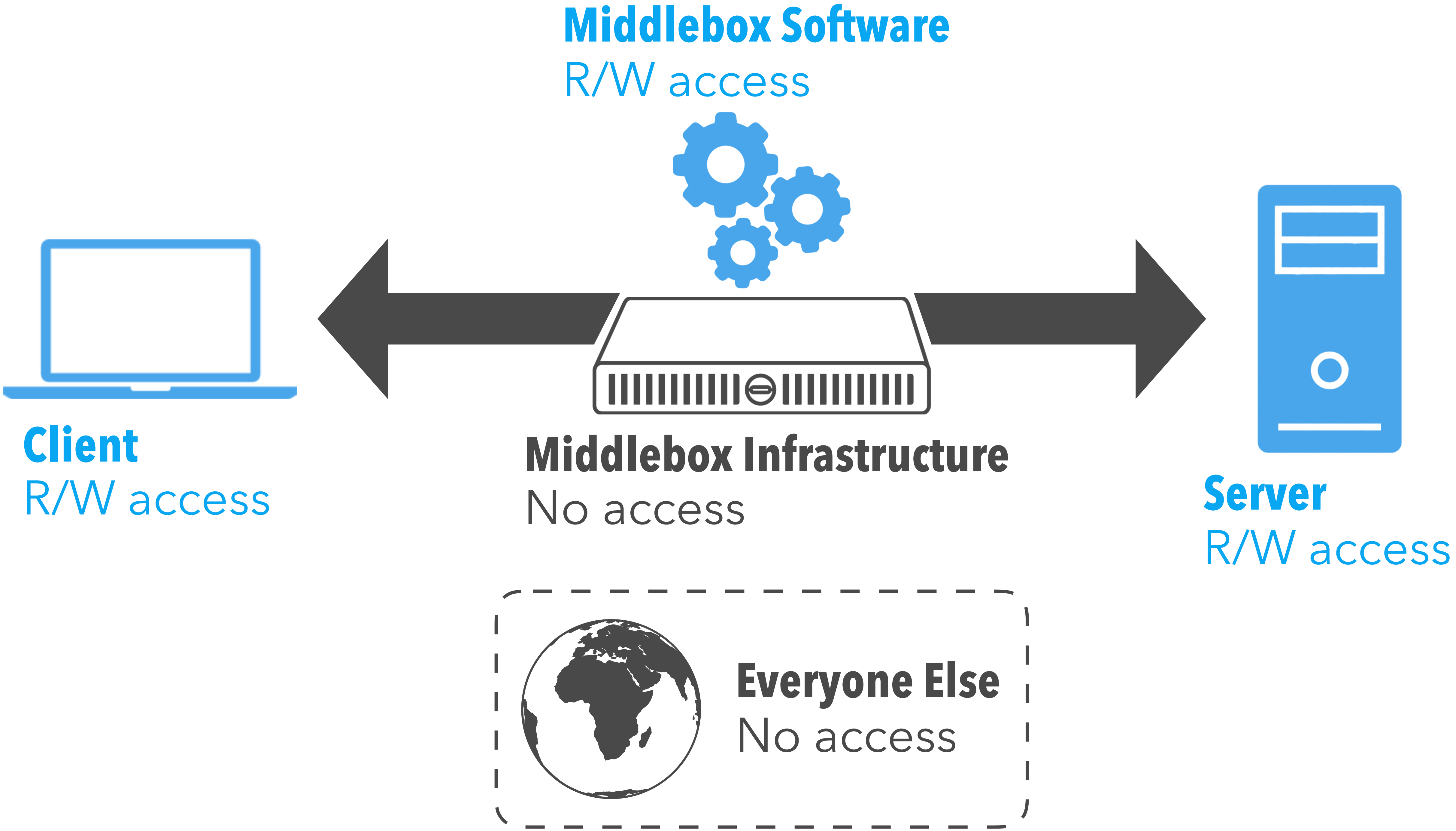
2. **Protection for outsourced middleboxes**

   Protect session data from middlebox infrastructure
   (in addition to traditional network attackers)

2 Protection for outsourced middleboxes

Protect session data from middlebox infrastructure

(in addition to traditional network attackers)



Middlebox Software
R/W access

Client
R/W access

Middlebox Infrastructure
No access

Server
R/W access

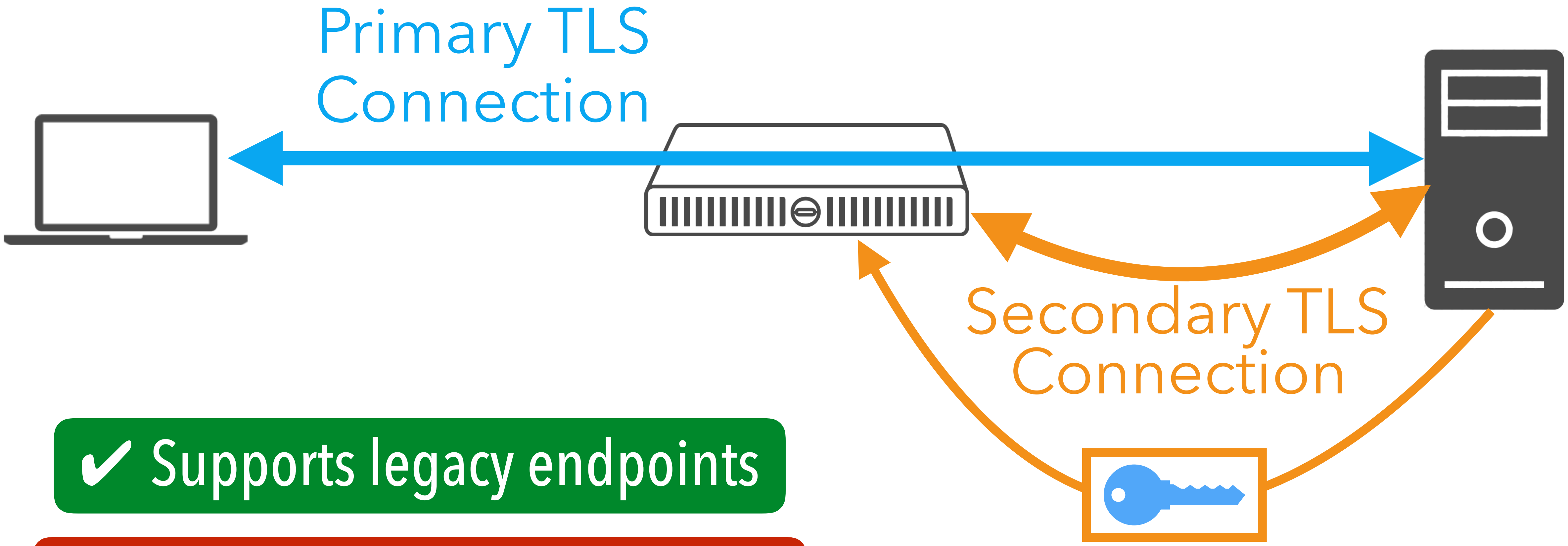
Everyone Else
No access

mbTLS targets two common-case, real-world needs

1. **Immediate deployability**
   Interoperate with one legacy endpoint

2. **Protection for outsourced middleboxes**
   Protect session data from middlebox infrastructure
   (in addition to traditional network attackers)

A first approach: pass primary session key over secondary TLS session



✔ Supports legacy endpoints

✗ Data and keys visible in RAM

An aside:

Intel SGX

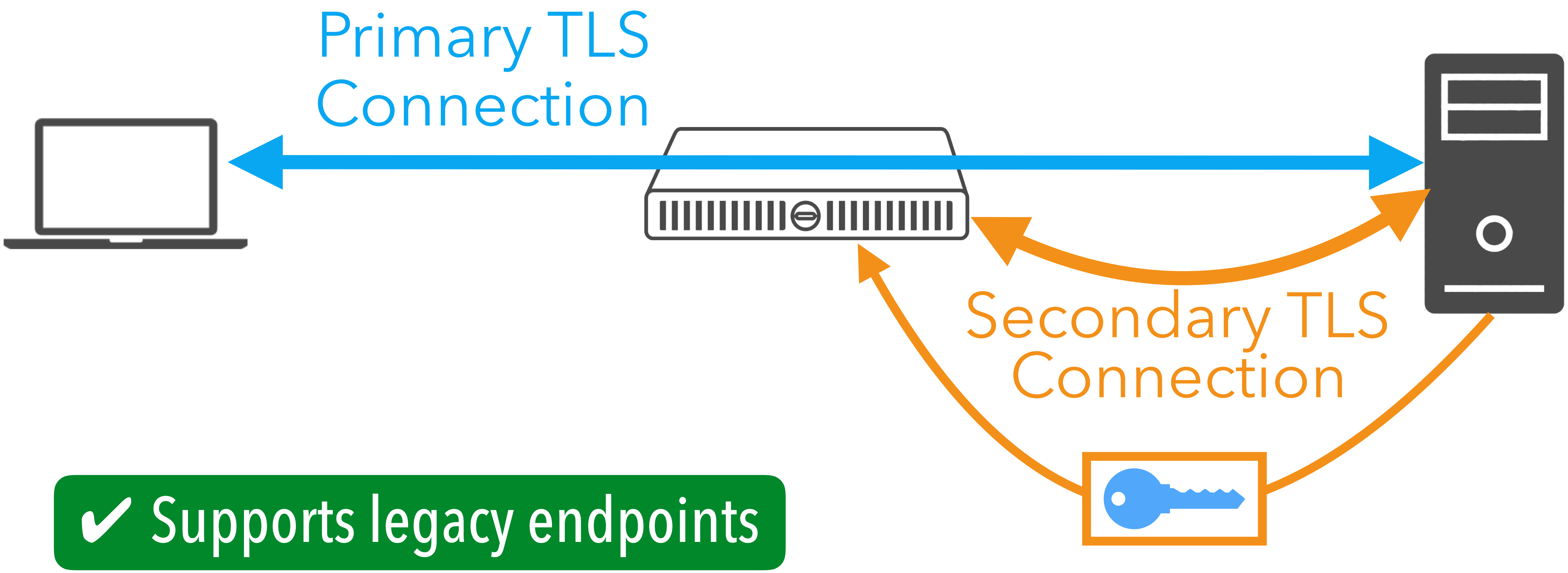
❶ **Secure Execution Environment**

Program code, data, and stack encrypted.

❷ **Remote Attestation**

Prove to remote party that ① is working.

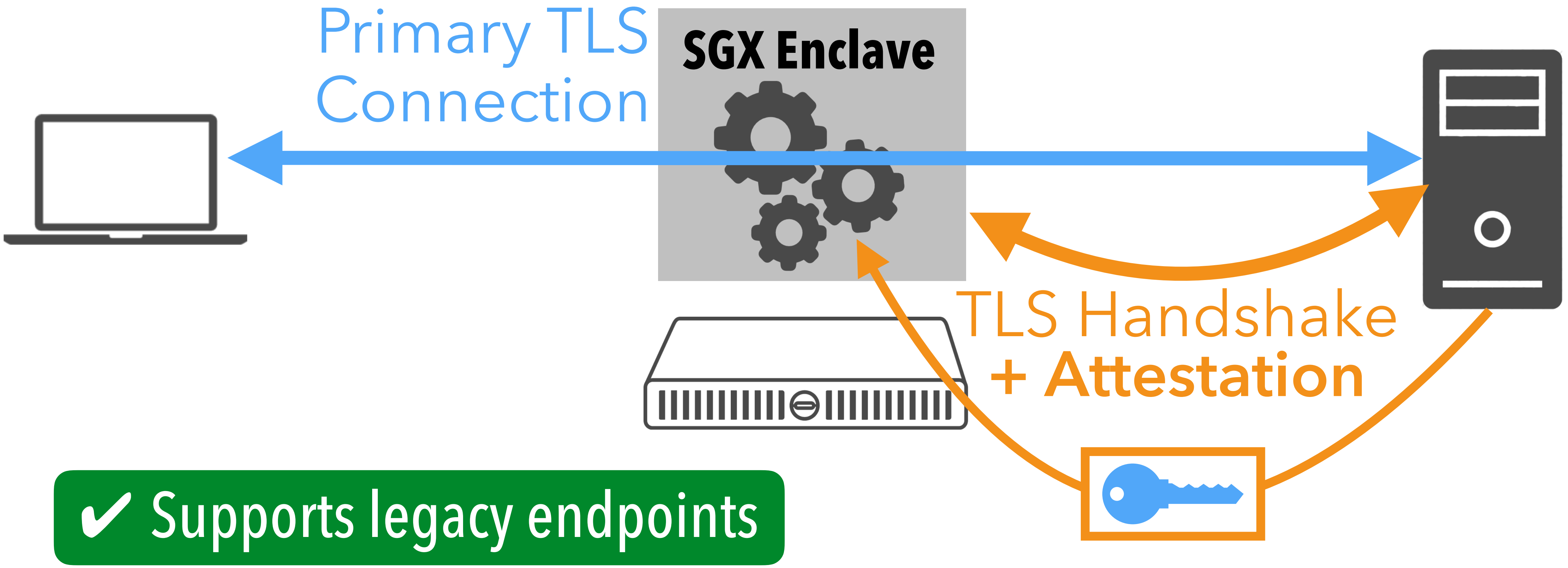
A first approach: pass primary session key over secondary TLS session



✔ Supports legacy endpoints

✗ Data and keys visible in RAM

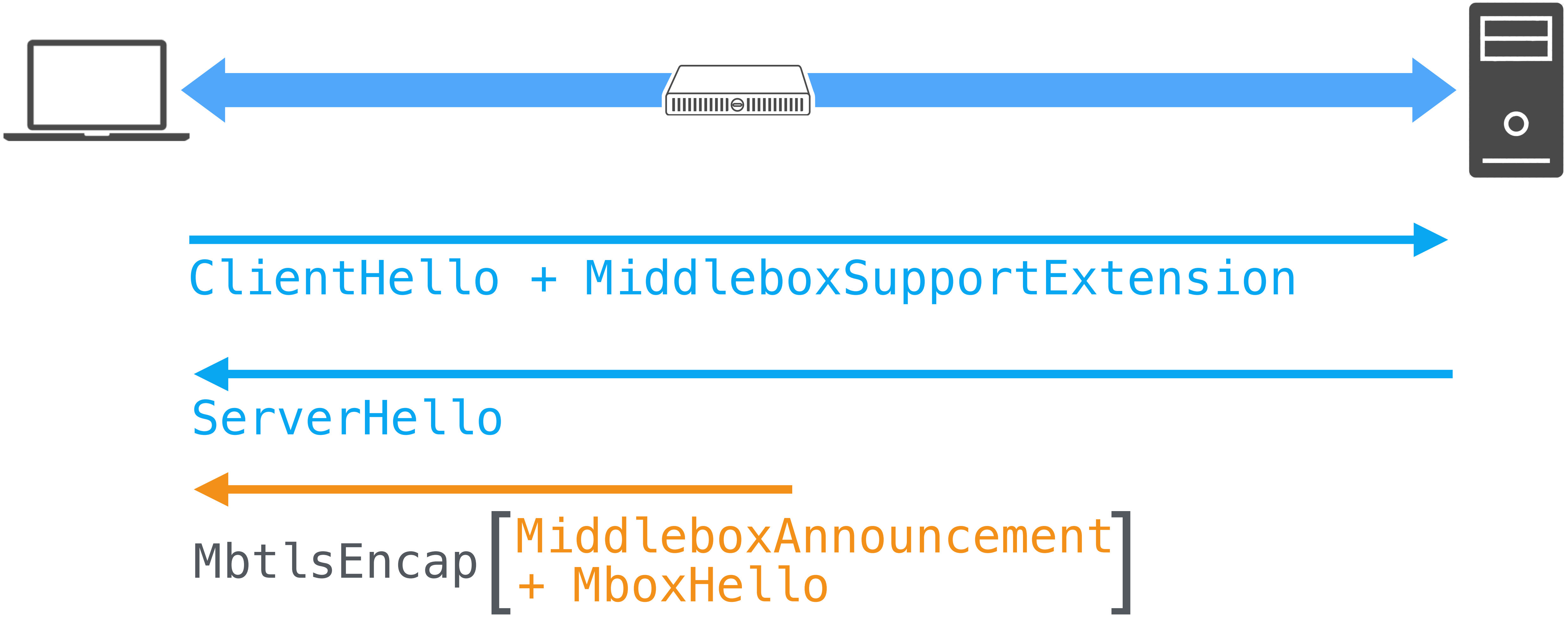
mbTLS protects session data and keys using SGX



✔ Supports legacy endpoints

✔ Data and keys encrypted in RAM

On-path middleboxes can be discovered "on-the-fly"

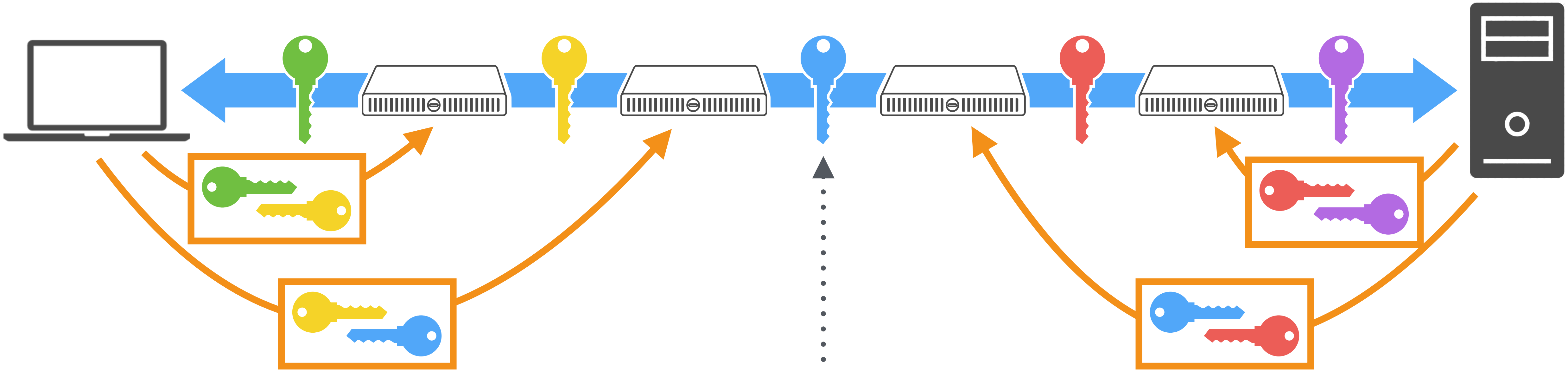


ClientHello + MiddleboxSupportExtension

ServerHello

MbtlsEncap[MiddleboxAnnouncement + MboxHello]

Per-hop keys provide path integrity and data change secrecy



Original session key "bridges" client- and server-side middleboxes.

Evaluation

1. **What overheads does mbTLS introduce?**

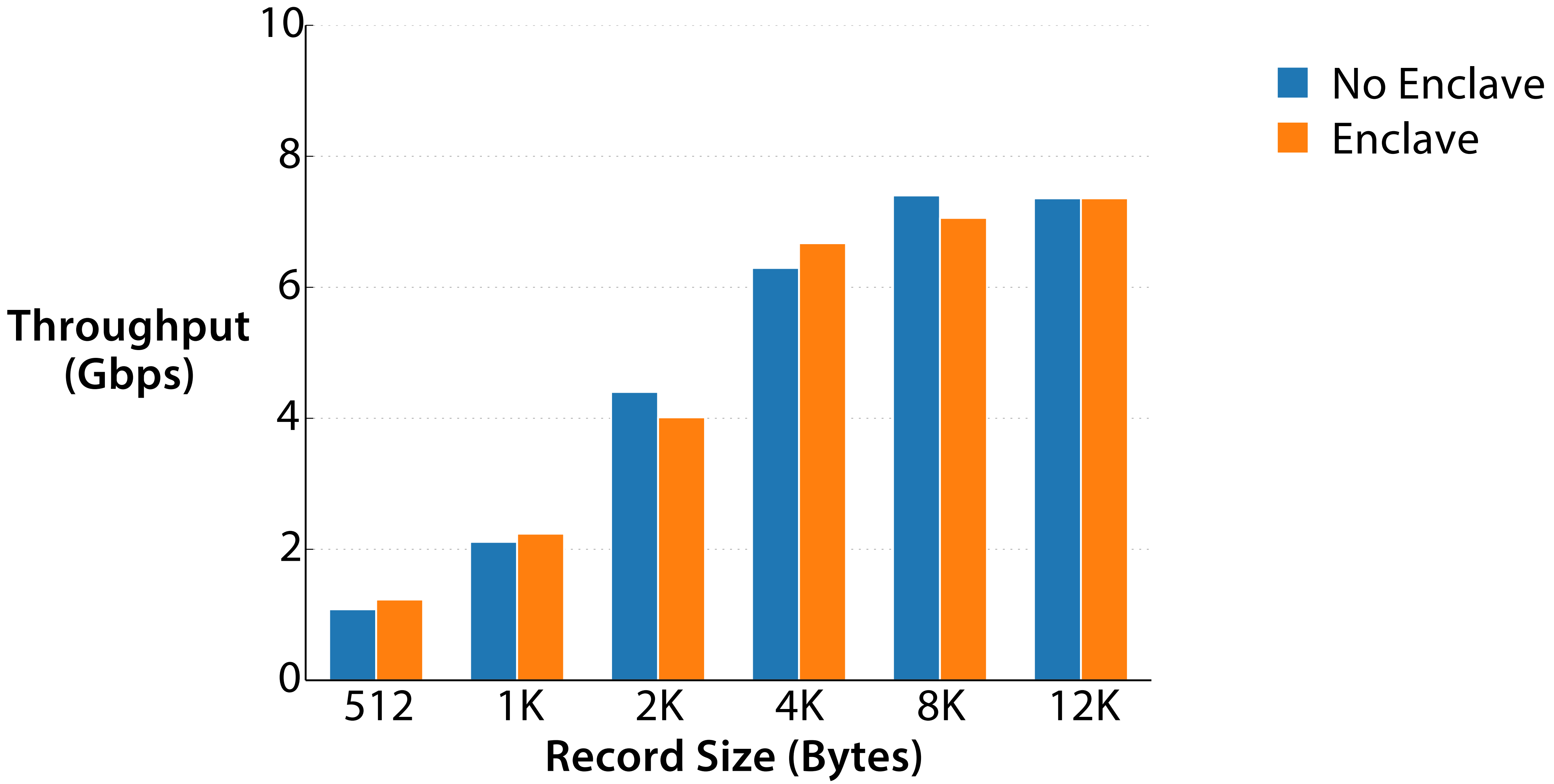
   From SGX?

   From crypto?

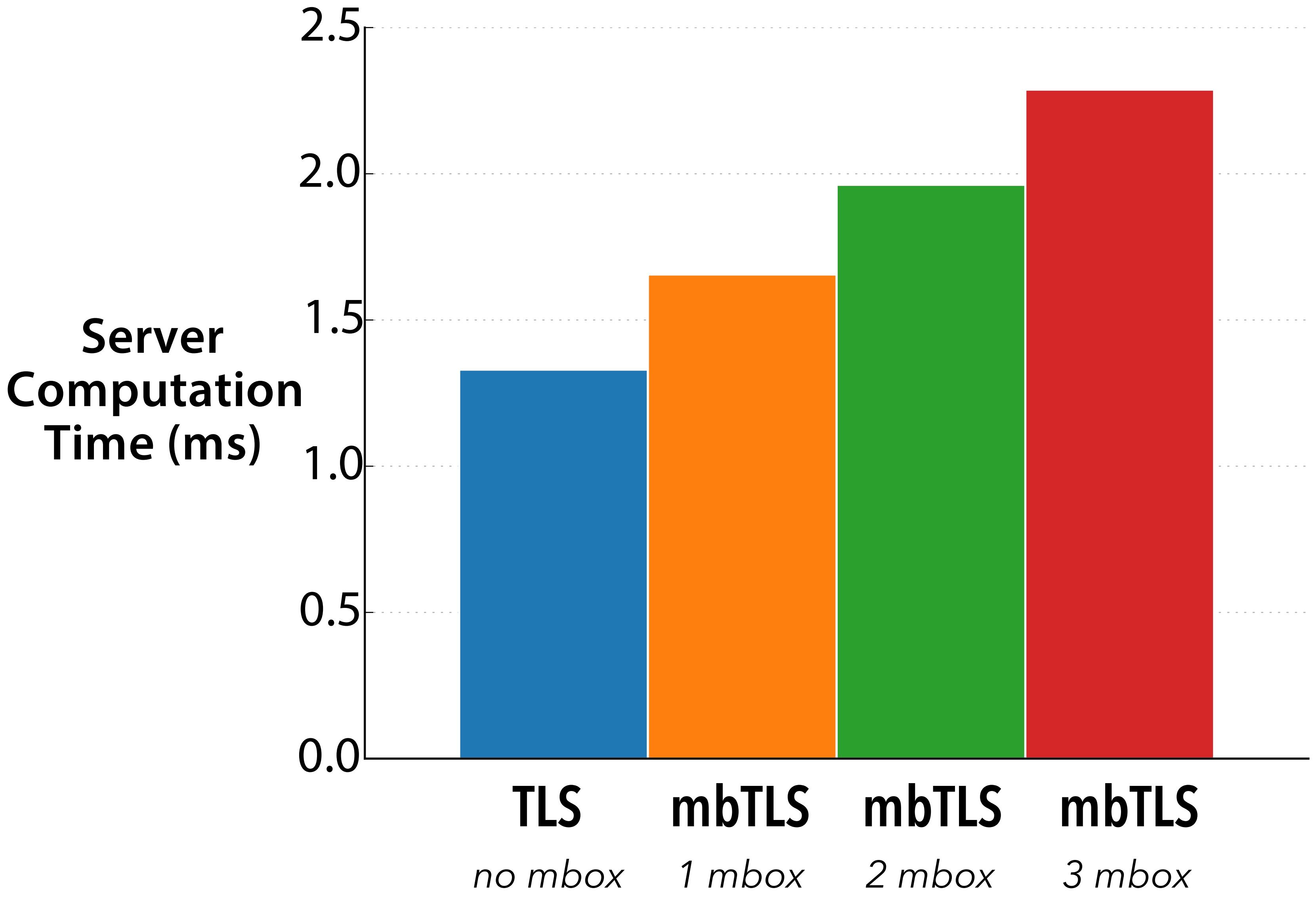
2. **Is mbTLS immediately deployable?**

   Will existing network devices drop mbTLS handshake messages?

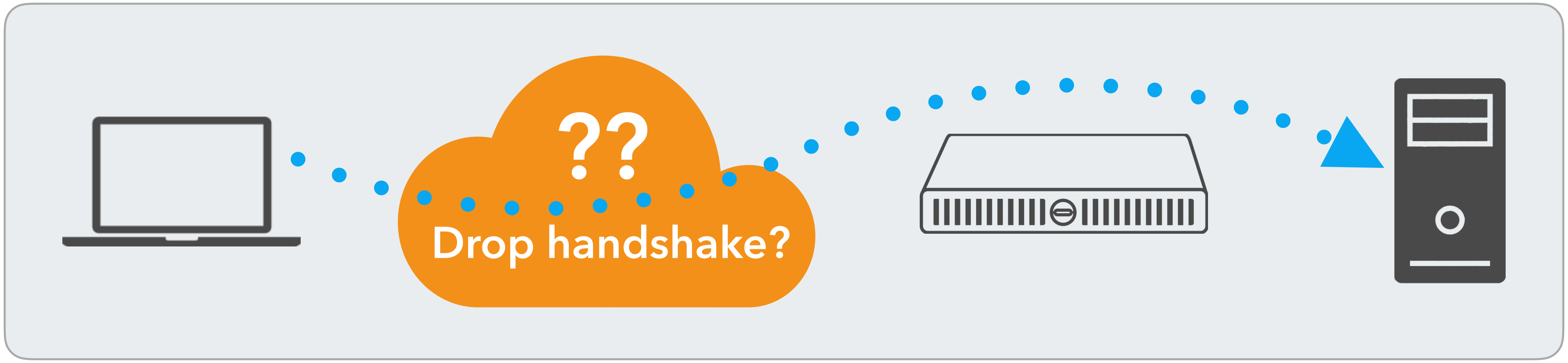
SGX doesn’t have much impact on I/O+compute-intensive workloads



mbTLS adds some handshake CPU overhead on the server



mbTLS' handshake protocol changes are deployable today



No handshakes were dropped.

6 enterprise networks

11 university networks

56 hosting networks

34 residential networks

35 colocation networks

19 data center networks

2 mobile networks

1 public network

77 unlabeled networks

And Then There Were More:

Secure Communication for More Than Two Parties

Carnegie Mellon University

THE UNIVERSITY OF UTAH

Microsoft Research

David Naylor
Carnegie Mellon

Richard Li
University of Utah

Christos Gkantsidis
Microsoft Research

Thomas Karagiannis
Microsoft Research

Peter Steenkiste
Carnegie Mellon