

QFaaS: Accelerating and Securing Serverless Cloud Networks with QUIC

Kaiyu Hou¹, **Sen Lin**¹, Yan Chen¹, Vinod Yegneswaran²

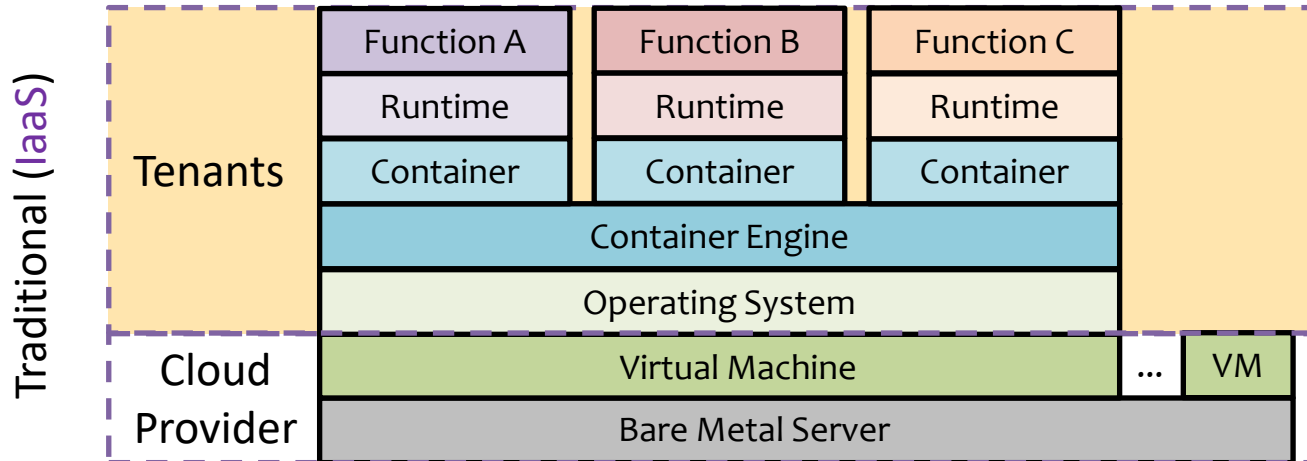
¹Northwestern University, ²SRI International

WHAT IS SERVERLESS COMPUTING

What is Serverless Computing

Serverless Computing

- Function as a Service (FaaS)
- Providers: IaaS + OS + Runtime
- Tenants: Stateless Functions



What is Serverless Computing

Serverless Computing

- Function as a Service (FaaS)
- Providers: IaaS + OS + Runtime
- Tenants: Stateless Functions

HTTP Request

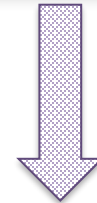


```
@requires_authorization
def somefunc(param1='', param2=0):
    r'''A docstring'''
    if param1 > param2: # interesting
        print 'Gre\ater'
    return (param2 - param1 + 1 + 0b101) or None

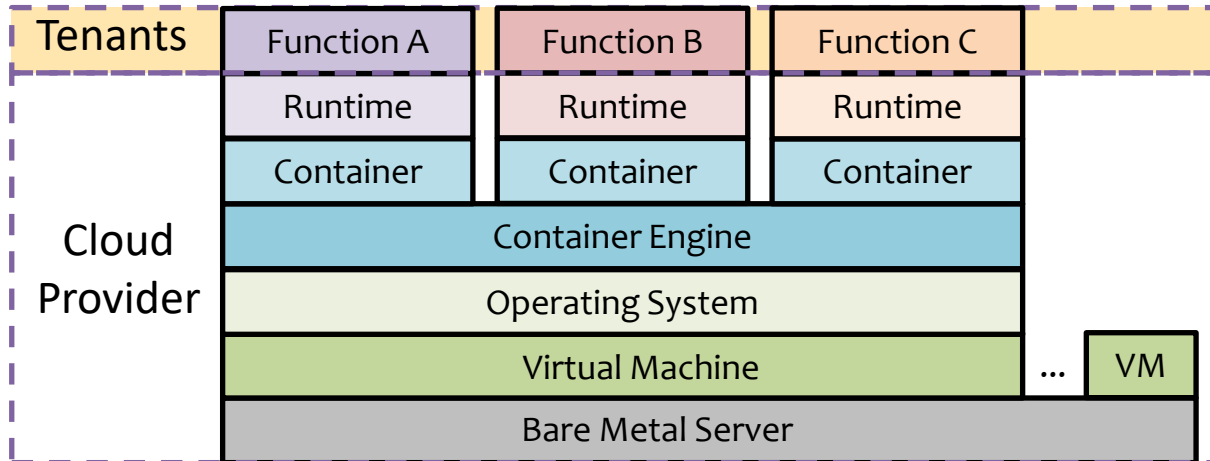
class SomeClass:
    pass

>>> message = '''interpreter
... prompt'''
```

HTTP Response



Serverless (FaaS)



BENEFITS AND COST OF SERVERLESS COMPUTING

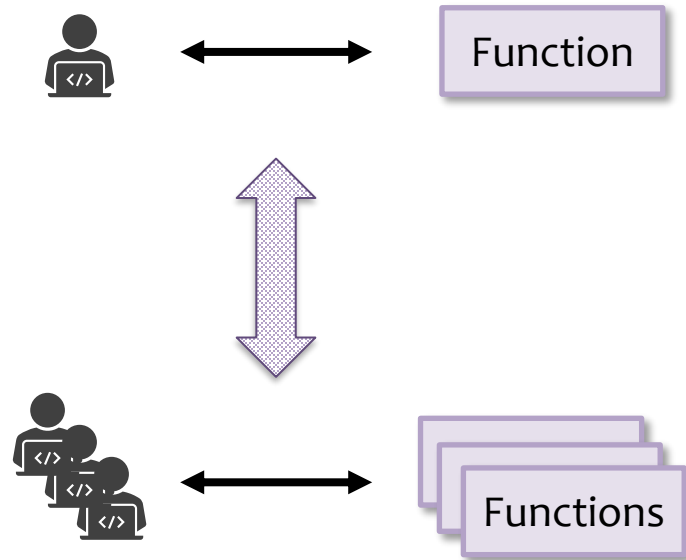
Benefits of Serverless Computing

Pros

Agile Auto-Scaling

- Cloud providers can **quickly** and **automatically** scale up/down function instances in response to burst requests

Bill-Based-on-Usage



Benefits of Serverless Computing

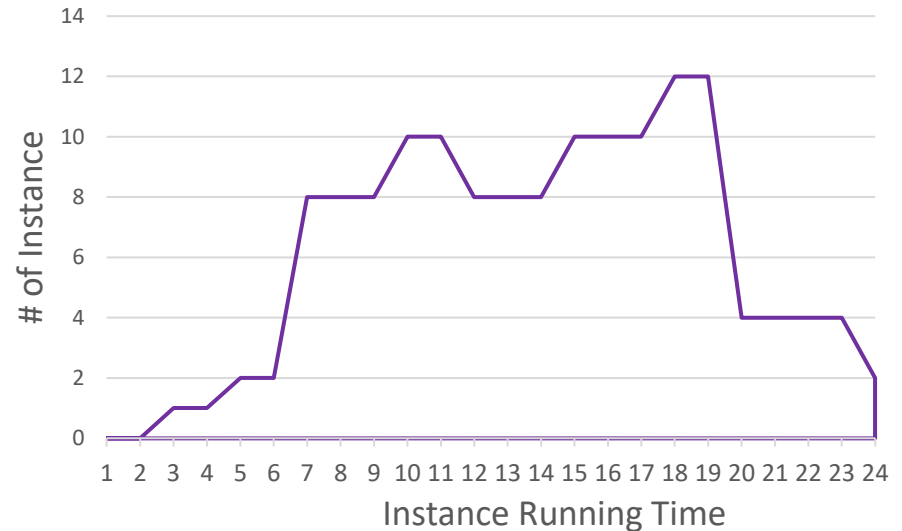
Pros

Agile Auto-Scaling

- Cloud providers can **quickly** and **automatically** scale up/down function instances in response to burst requests

Bill-Based-on-Usage

- Auto-scaling fixes over-provision and under-provision problems
- Tenants only **pay** for the actual function **execution** time



Benefits of Serverless Computing

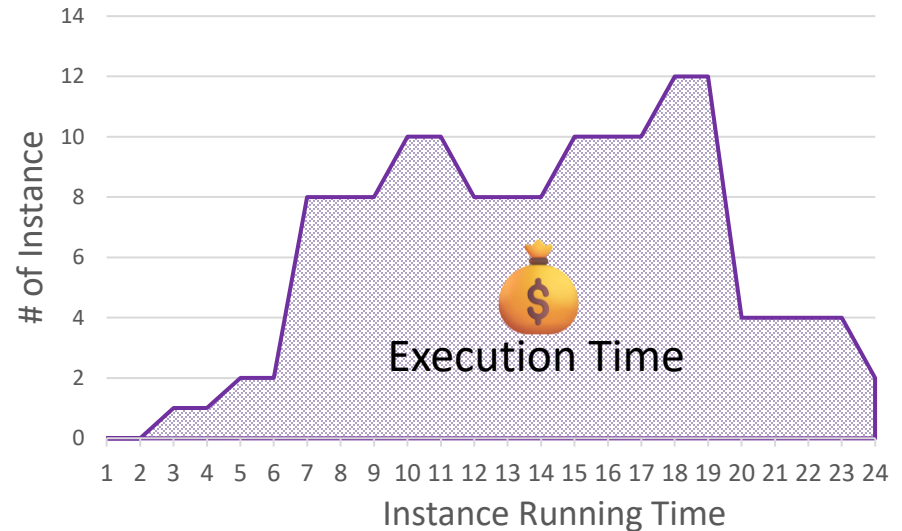
Pros

Agile Auto-Scaling

- Cloud providers can **quickly** and **automatically** scale up/down function instances in response to burst requests

Bill-Based-on-Usage

- Auto-scaling fixes over-provision and under-provision problems
- Tenants only **pay** for the actual function **execution** time



Benefits of Serverless Computing

Pros

Agile Auto-Scaling

- Cloud providers can **quickly** and **automatically** scale up/down function instances in response to burst requests

Bill-Based-on-Usage

- Auto-scaling fixes over-provision and under-provision problems
- Tenants only **pay** for the actual function **execution** time

“Serverless Computing is expected to become the dominant cloud computing paradigm.”

-- A Berkeley view on serverless computing.

Cost of Benefits

Pros

Agile Auto-Scaling

- Cloud providers can **quickly** and **automatically** scale up/down function instances in response to burst requests

Bill-Based-on-Usage

- Auto-scaling fixes over-provision and under-provision problems
- Tenants only **pay** for the actual function **execution** time



Cons

Increased Latency

- Cold-start latency
- Connection establishment latency

Cost of Benefits

Cons

Increased Latency

- Cold-start latency
- Connection establishment latency

Cost of Benefits

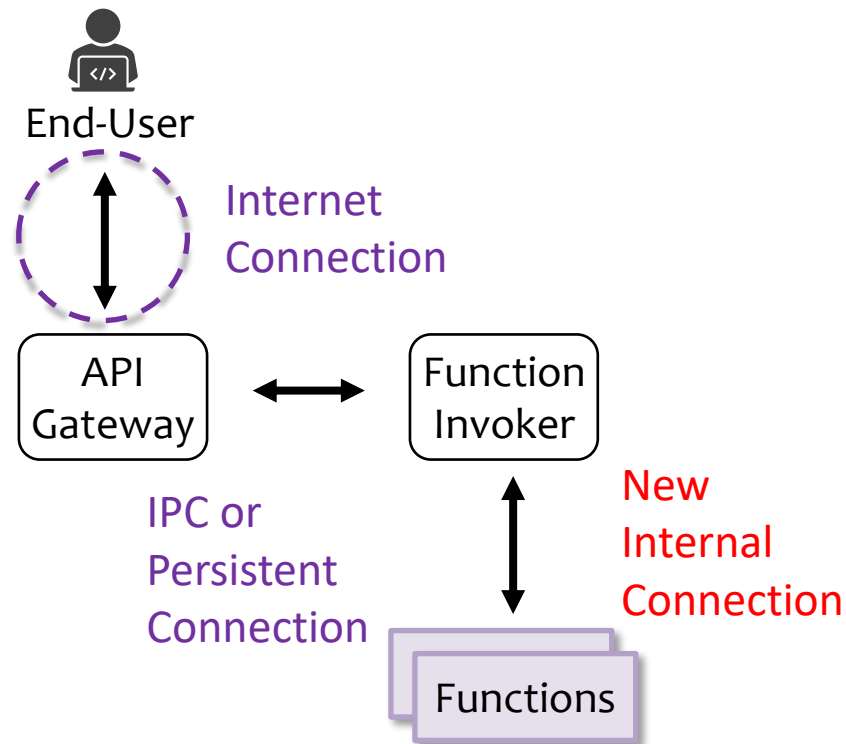
Cons

Increased Latency

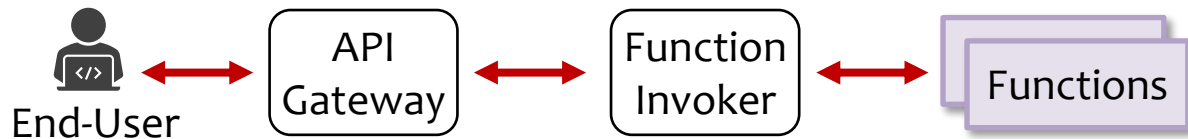
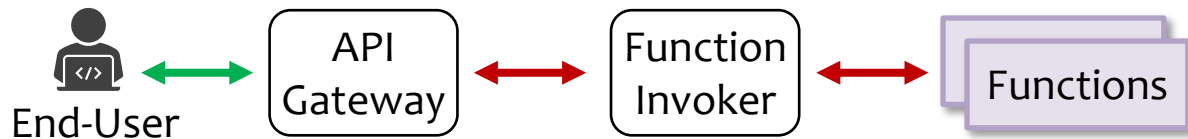
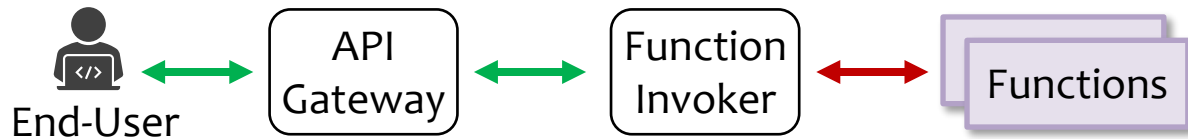
- Cold-start latency
- Connection establishment latency

Inevitable

Function instances are ephemeral under auto-scaling.



Cloud Providers Exchange Security for Low-Latency



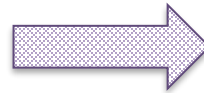
Cloud Providers Exchange Security for Low-Latency

Zero Trust

Any entities, even the internal ones, should not be trusted by default

Best Practice

Major Providers: dedicate **full encryption** to all internal connections



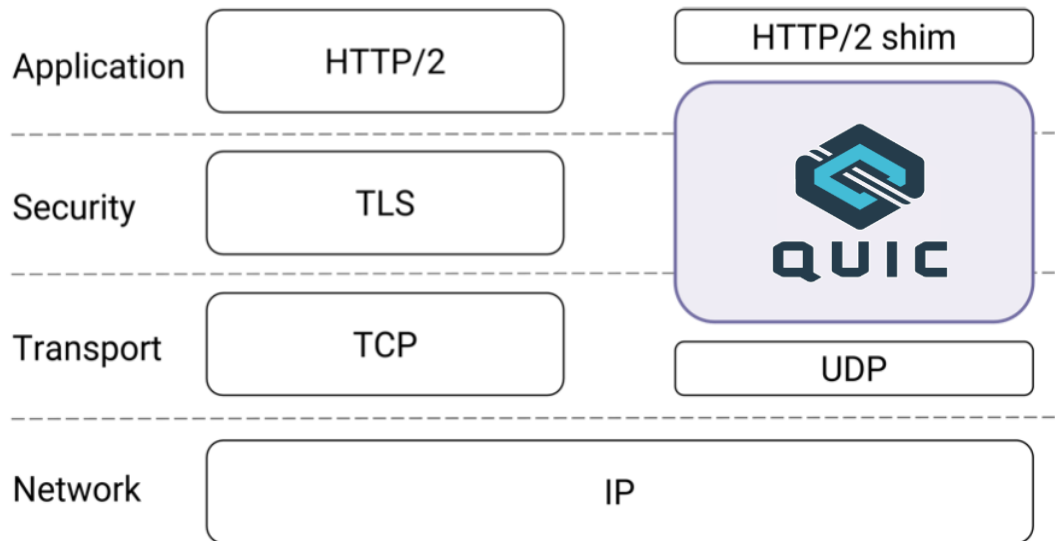
QFaaS

Security

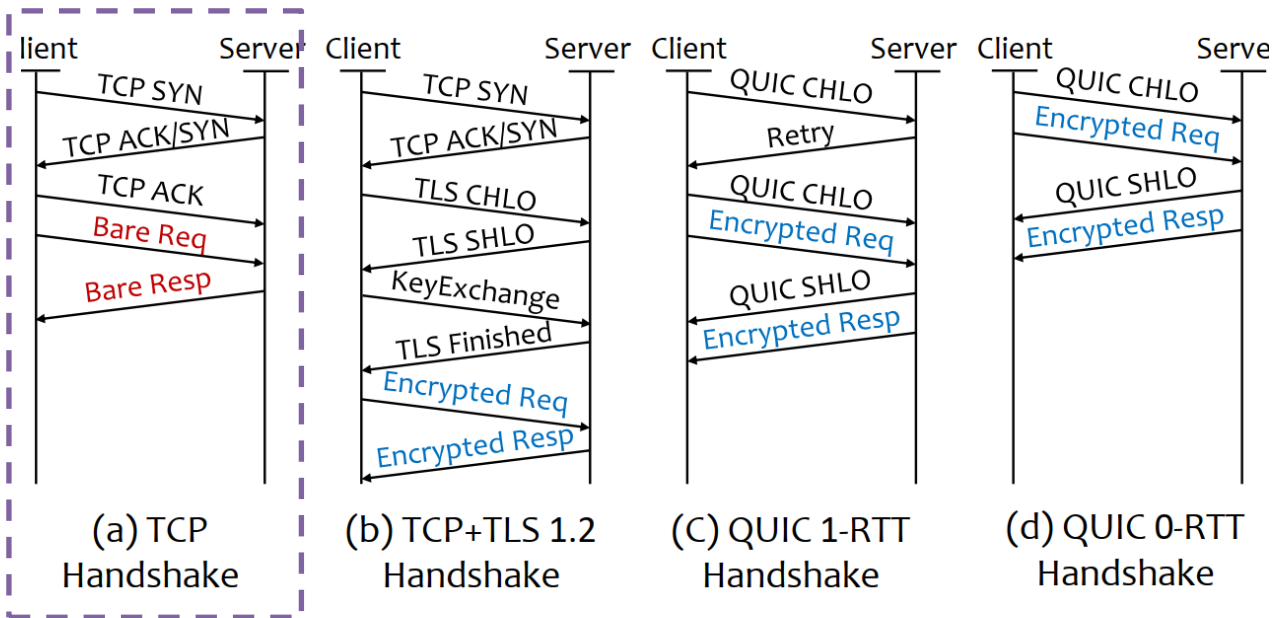
Low-Latency

QFaaS DESIGN

QFaaS: Fuse Serverless with QUIC



QFaaS: Fuse Serverless with QUIC

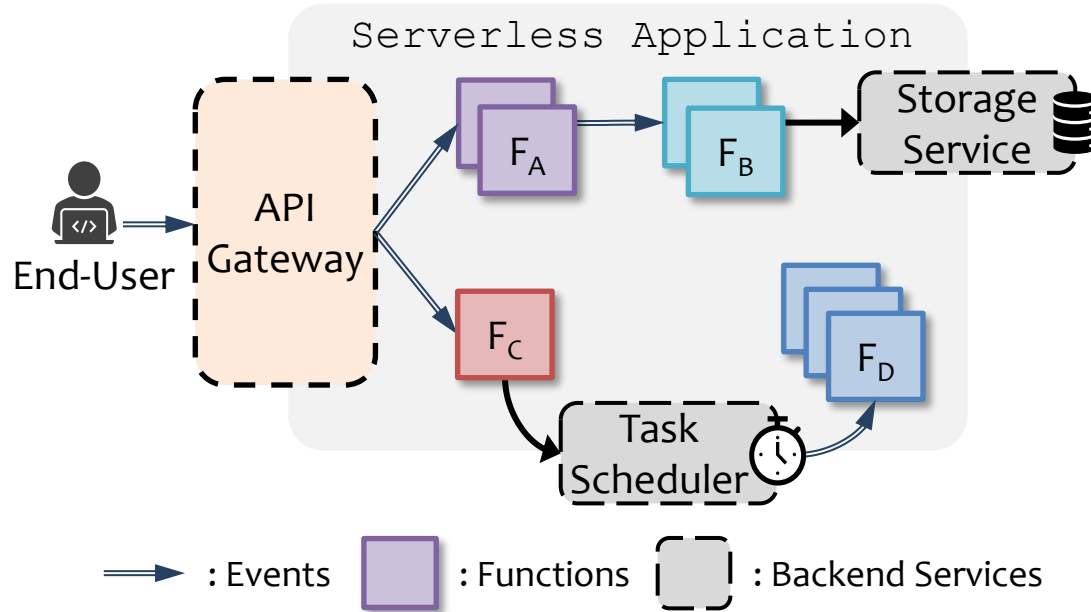


System Goal

Add QUIC into the serverless platform without any tenants' code modification

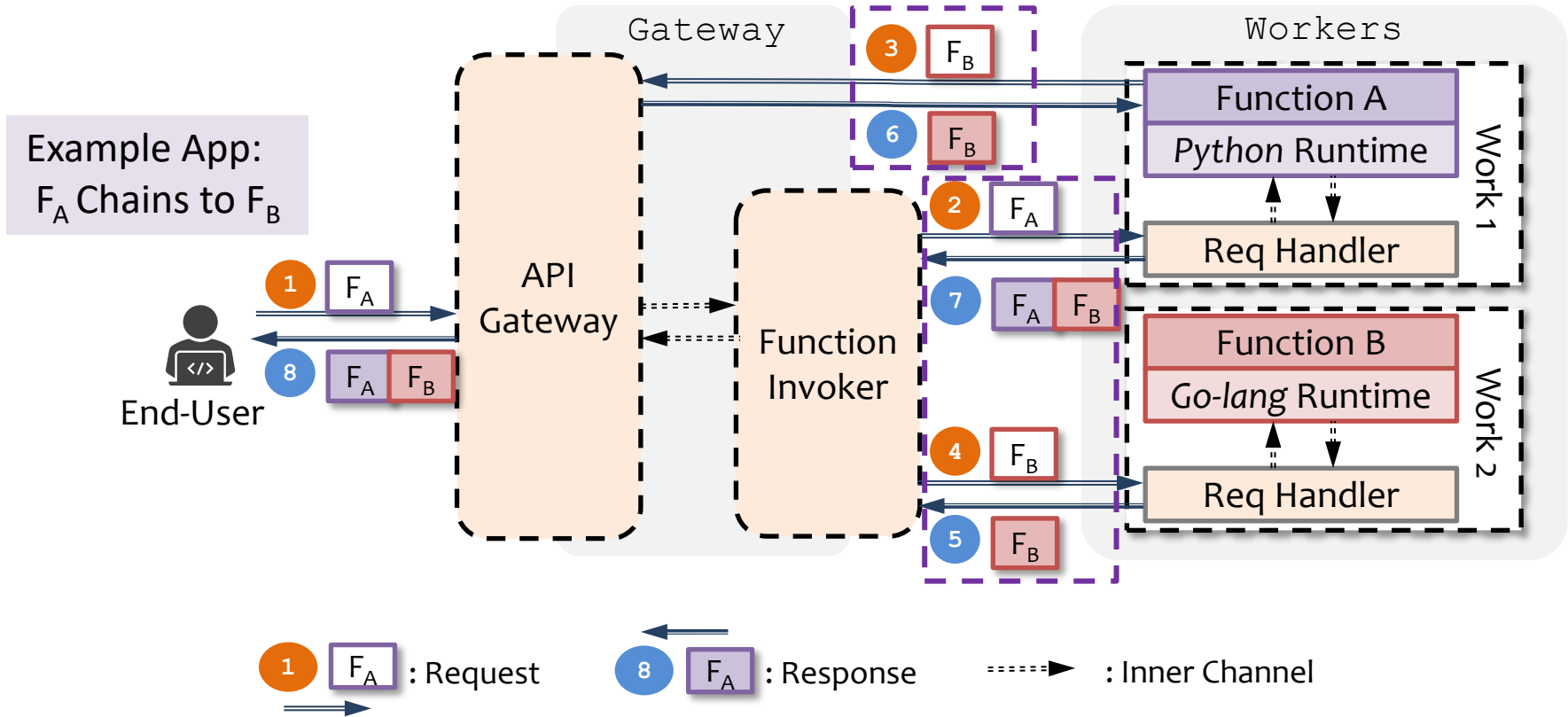
Scheme	TCP	TCP + TLS 1.2	QUIC 1-RTT	QUIC 0-RTT
New Session	1	3	1	-
Recover Session	1	2	1	0

Model Serverless Computing: Logical Model

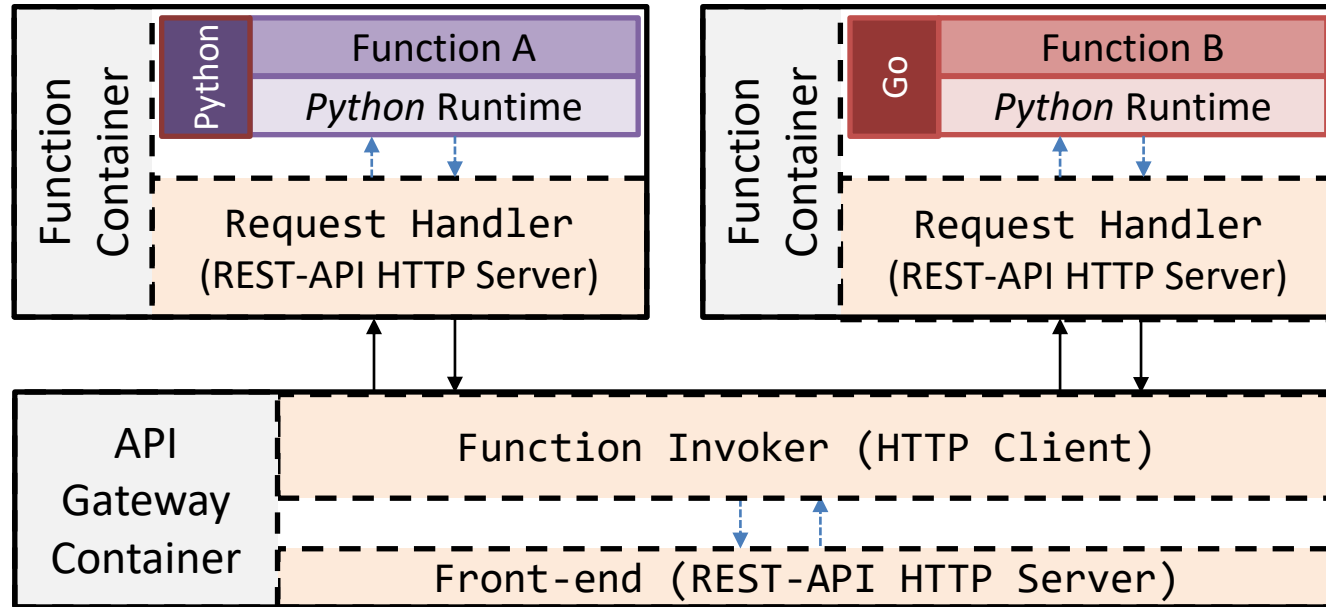


- ✗ User-Function Connection
- ✗ Function-Function Connection

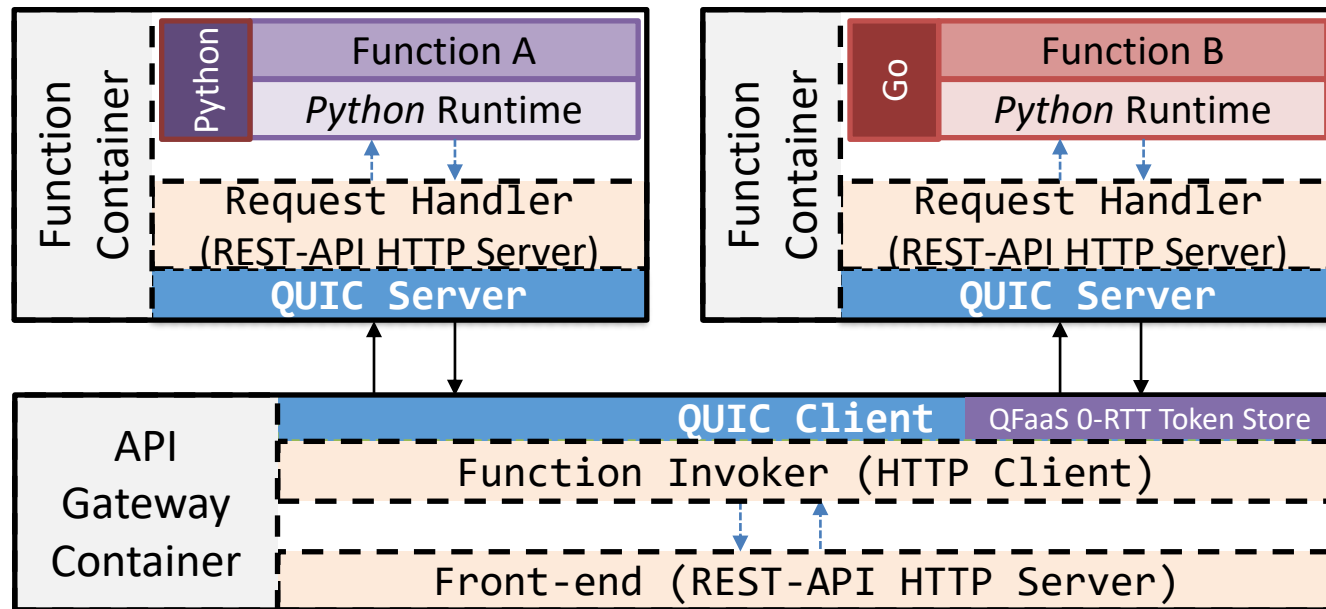
Model Serverless Computing: Network Model



QFaaS System Design



QFaaS System Design



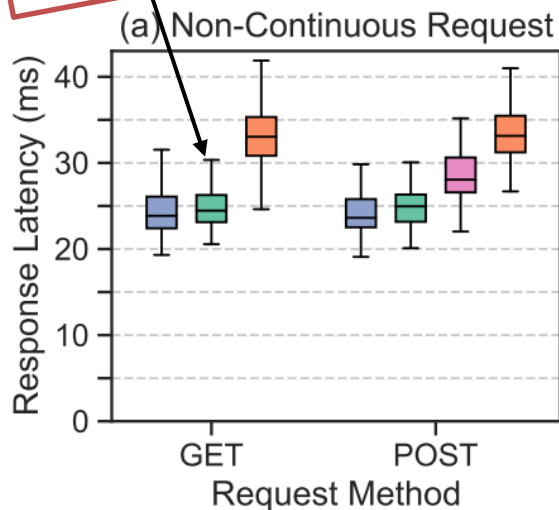
Totally Transparent

QFaaS EVALUATION

Benefits of QFaaS on Single Functions

0-RTT Mode

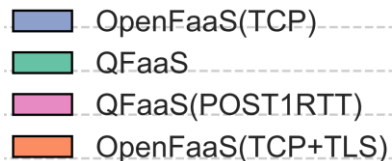
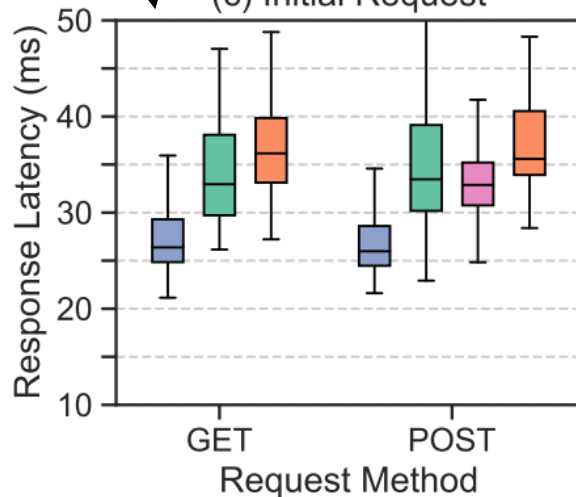
28%
Reduction



1-RTT Mode

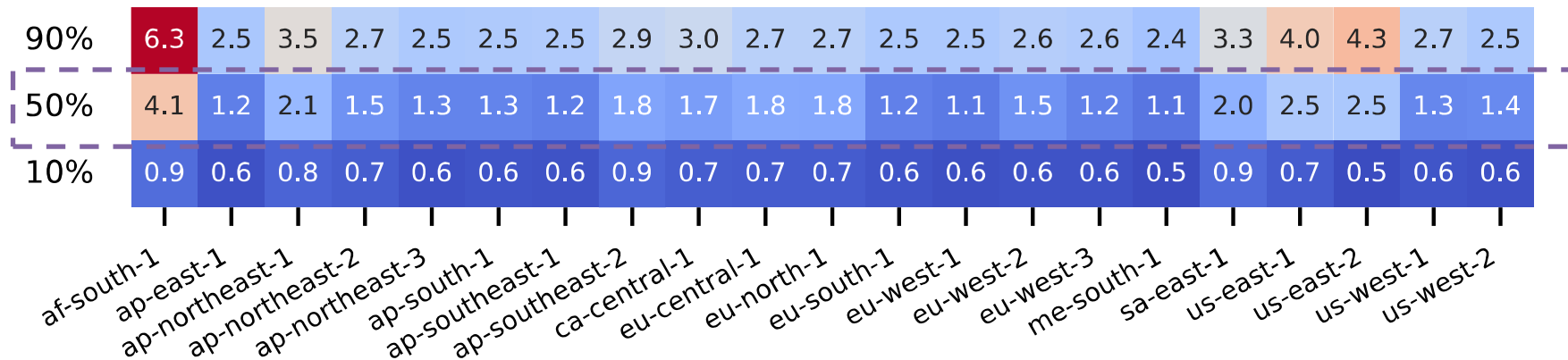
11%
Reduction

(c) Initial Request



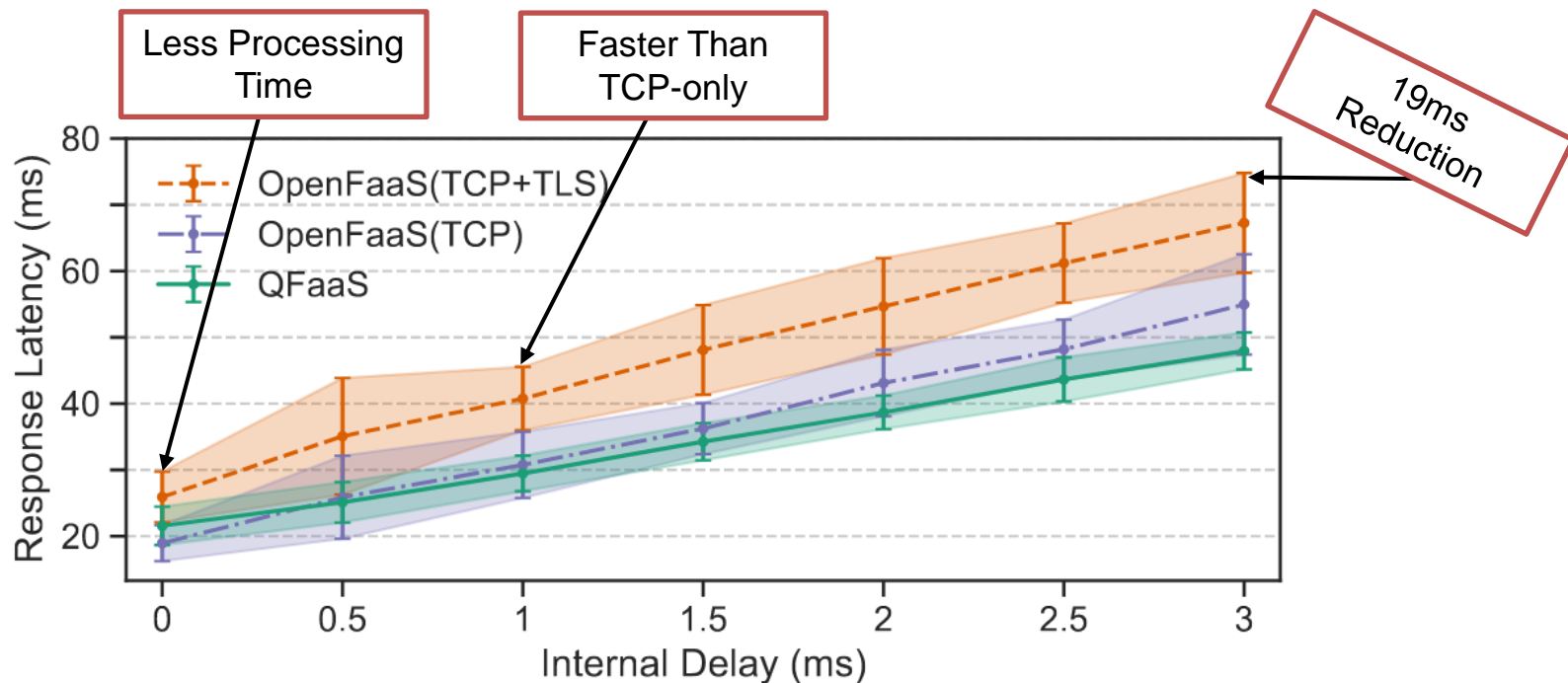
QFaaS is faster than OpenFaaS(TCP+TLS) in both 0-RTT and 1-RTT modes

Benefits of QFaaS under Variant Intra-Cloud Latency



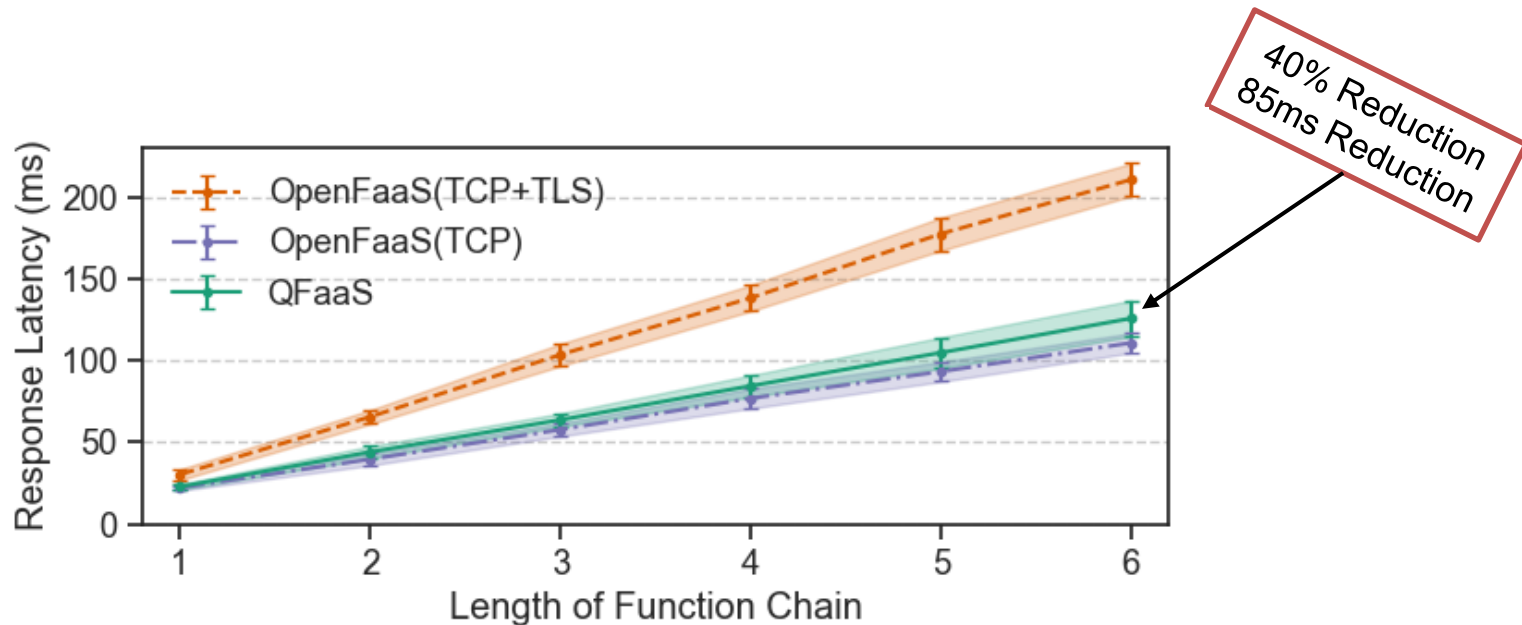
AWS Intra-Cloud Latency (Past Year)

Benefits of QFaaS under Variant Intra-Cloud Latency



QFaaS can not only save transmission latency but also processing latency
The advantage of QFaaS is enlarged as the inter-cloud latency increases

Benefits of QFaaS Function Chain Library



QFaaS performance is aligned with OpenFaaS(TCP+TLS)

The advantage of QFaaS is enlarged as the function chain length increases

Conclusion

Network Abstraction

A clear network abstraction for serverless applications to identify potential network bottlenecks

Design

A new system design to accelerate and secure serverless networks which requires no tenant code modification

Implementation

Open-source implementation of the QFaaS prototype
<https://github.com/qfaas-project>

Benefits

QFaaS reduce response latency of single functions by 28%, chained functions by 40%, and ~50ms in real-world serverless applications. QFaaS is even faster than insecure TCP-only platforms

Thank You