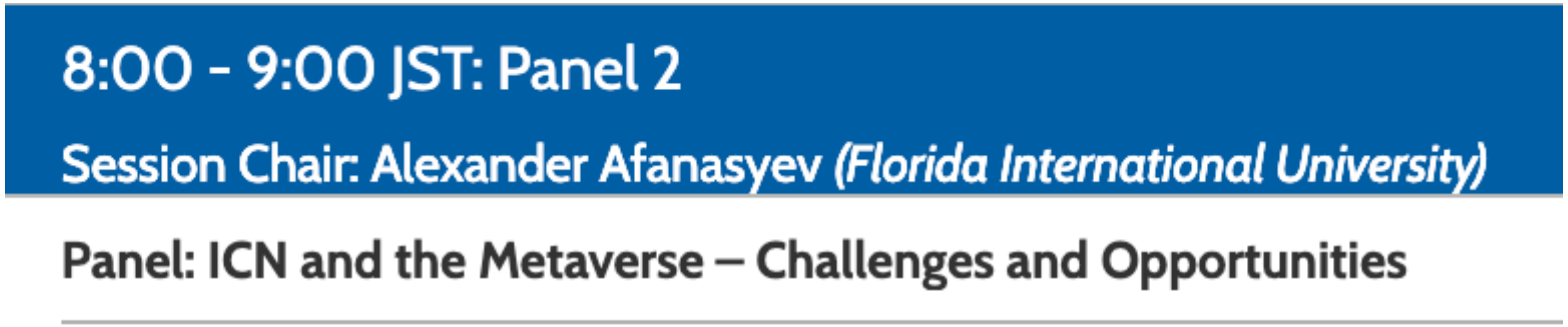
RESTFul Information-Centric Networking

Dirk Kutscher and Dave Oran

ICNRG @ IETF-115
2022-11-08

Background

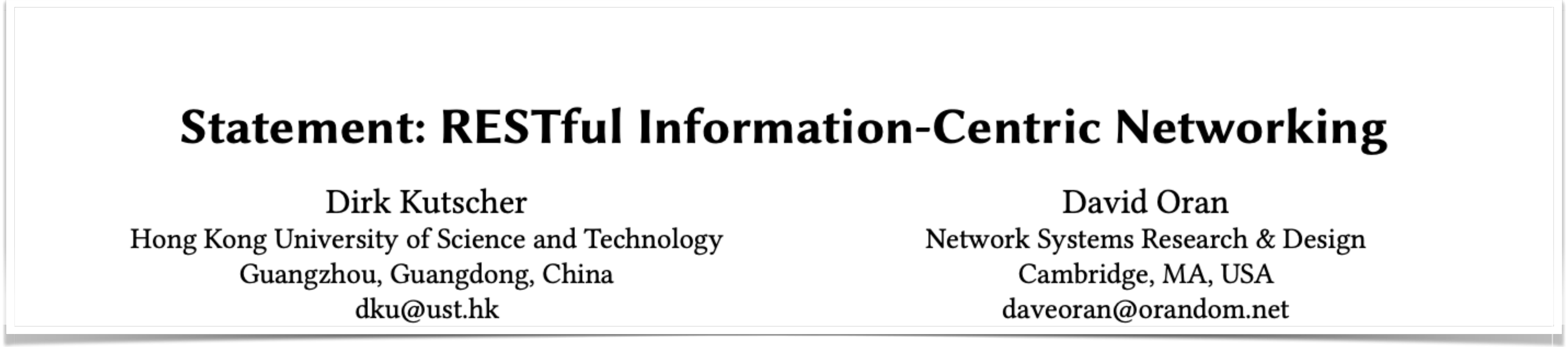
ACM ICN-2022



8:00 - 9:00 JST: Panel 2

Session Chair: Alexander Afanasyev *(Florida International University)*

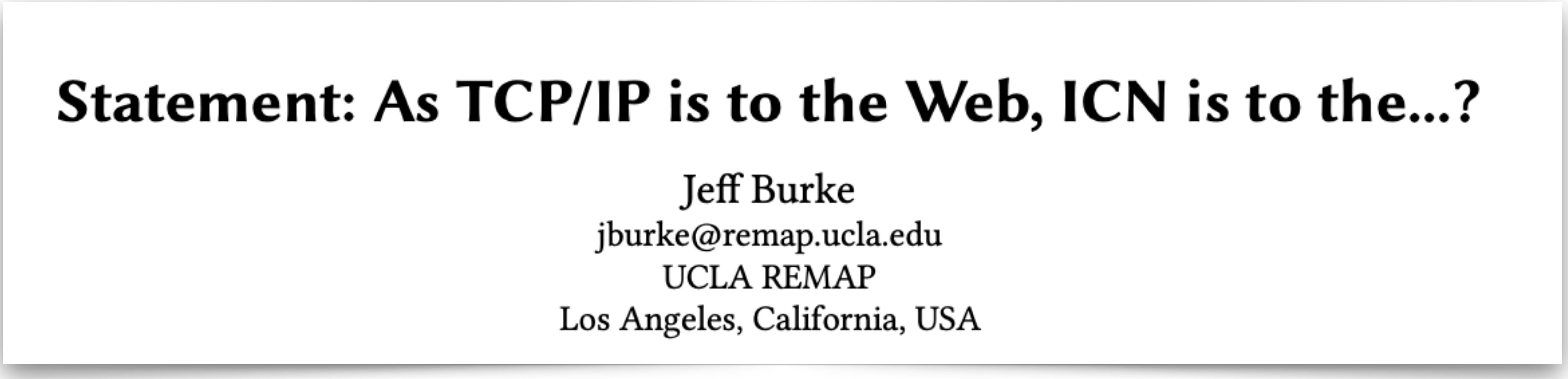
Panel: ICN and the Metaverse – Challenges and Opportunities



Statement: RESTful Information-Centric Networking

Dirk Kutscher
Hong Kong University of Science and Technology
Guangzhou, Guangdong, China
dku@ust.hk

David Oran
Network Systems Research & Design
Cambridge, MA, USA
daveoran@orandom.net

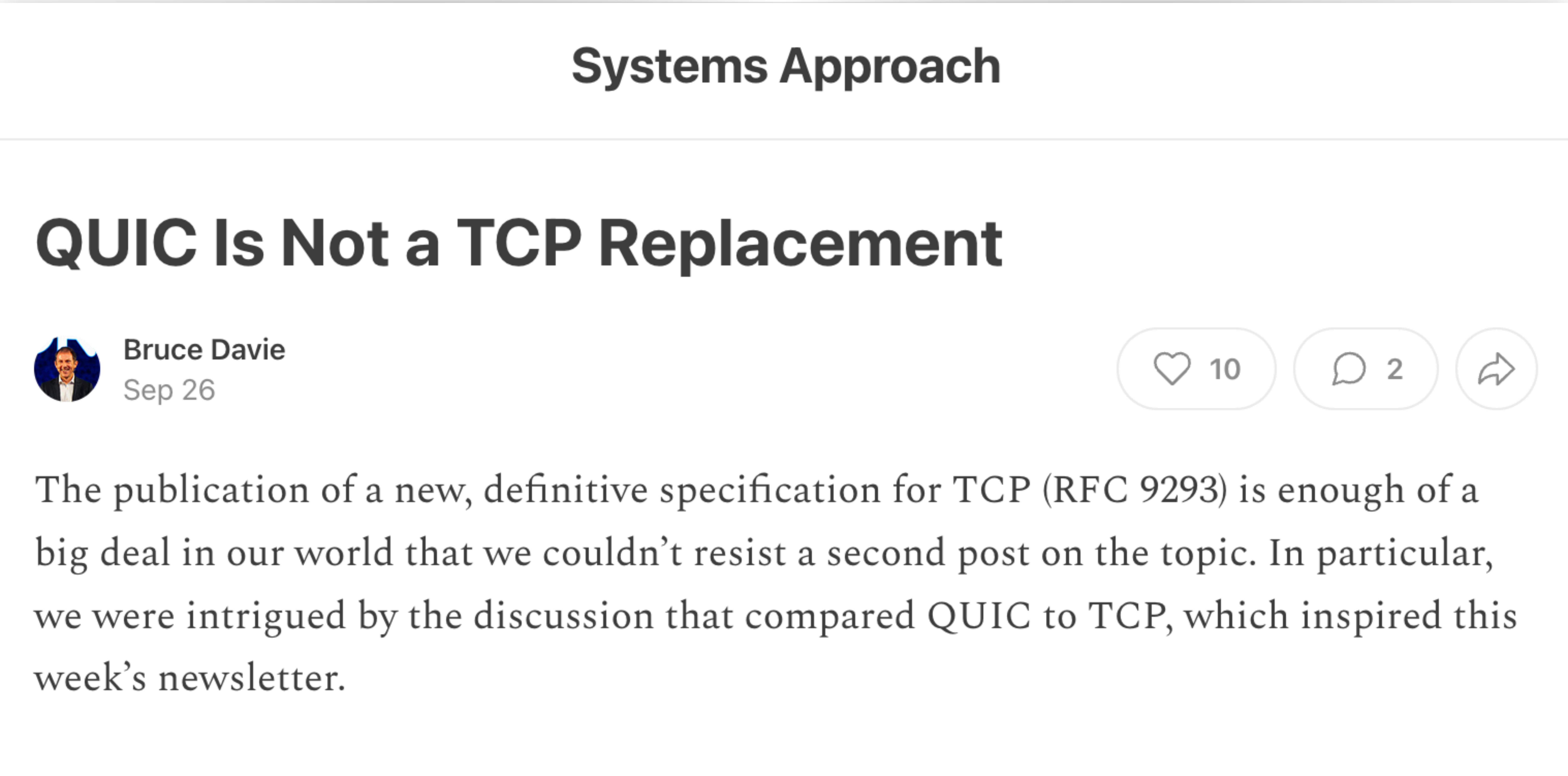


Statement: As TCP/IP is to the Web, ICN is to the...?

Jeff Burke
jburke@remap.ucla.edu
UCLA REMAP
Los Angeles, California, USA

More Background

Internet Protocols for Efficient RPC Communication



Systems Approach

QUIC Is Not a TCP Replacement

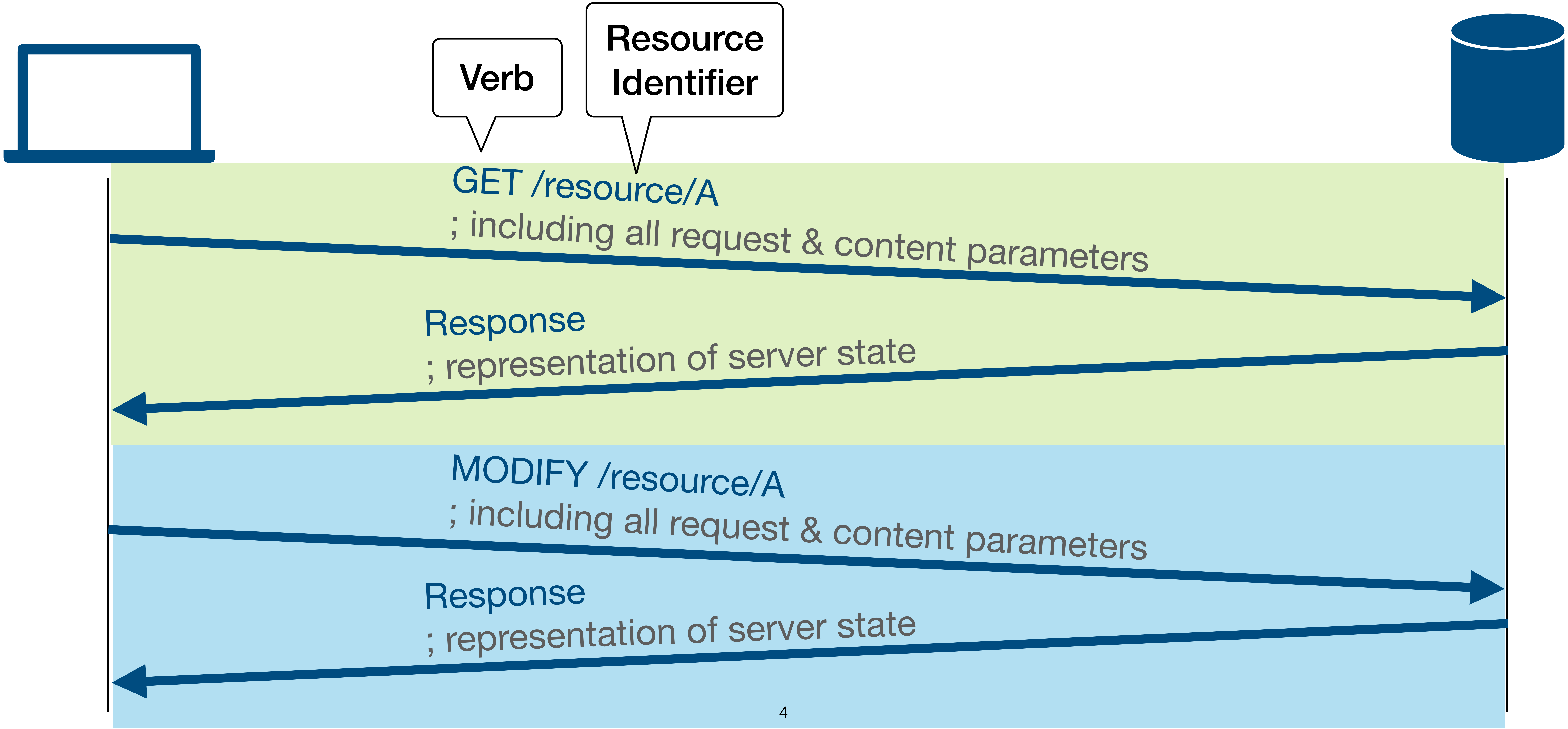
Bruce Davie
Sep 26

The publication of a new, definitive specification for TCP (RFC 9293) is enough of a big deal in our world that we couldn't resist a second post on the topic. In particular, we were intrigued by the discussion that compared QUIC to TCP, which inspired this week's newsletter.

https://systemsapproach.substack.com/p/quic-is-not-a-tcp-replacement

Representational State Transfer

Theory: Stateless Requests



Verb

Resource Identifier

GET /resource/A
; including all request & content parameters

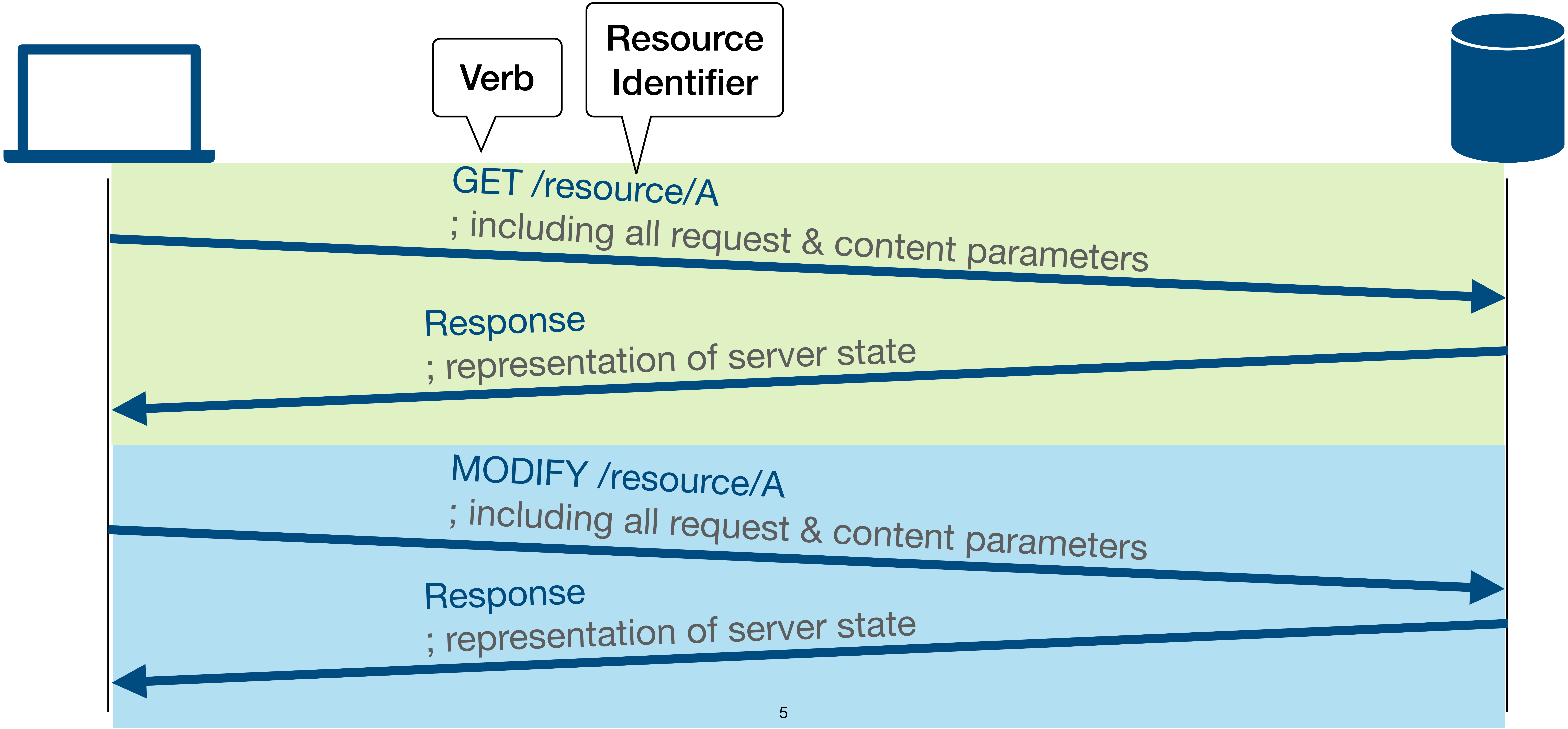
Response
; representation of server state

MODIFY /resource/A
; including all request & content parameters

Response
; representation of server state

Representational State Transfer

Reality: Not So Stateless Requests (Cookies)



Verb

Resource Identifier

GET /resource/A
; including all request & content parameters

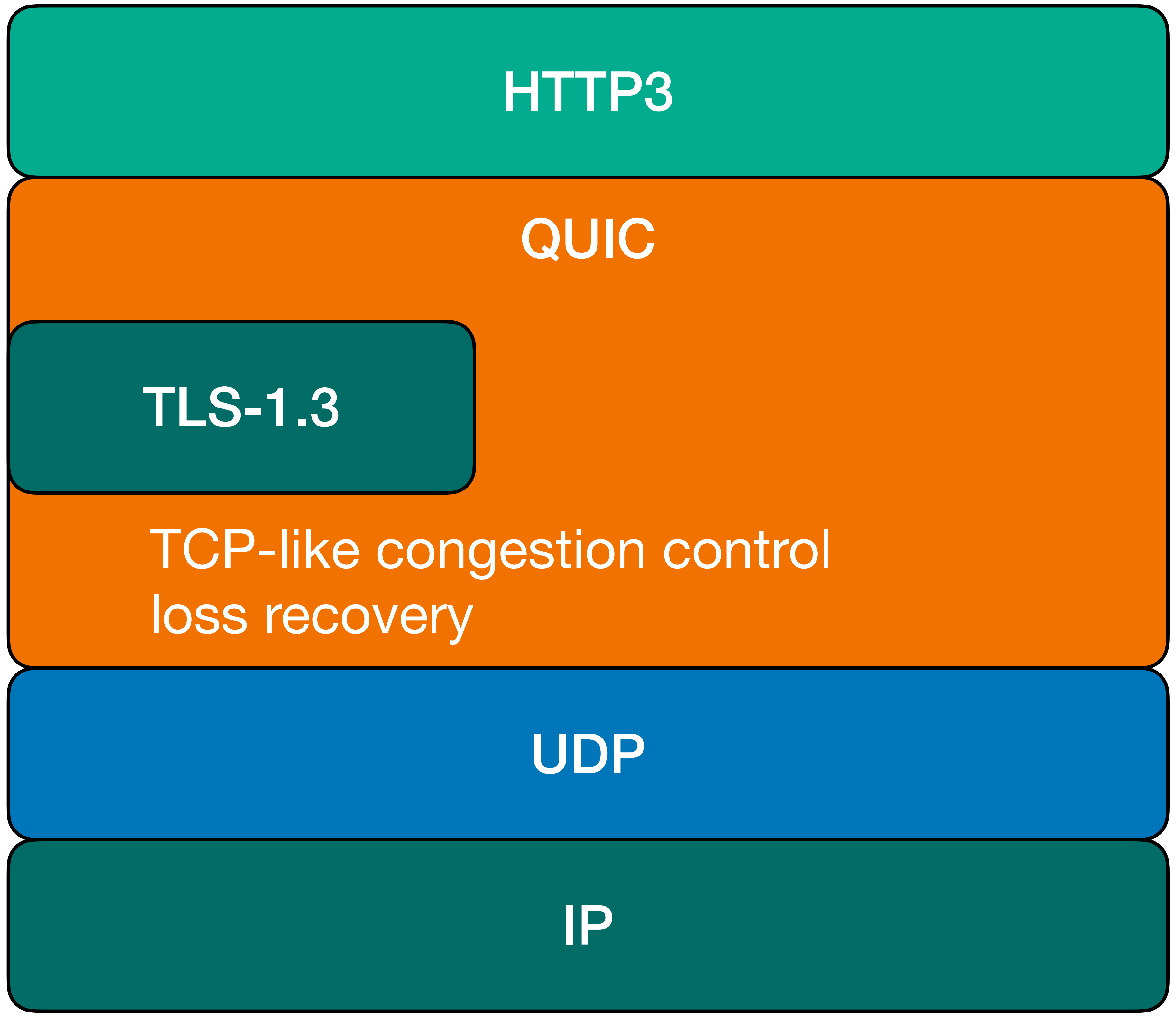
Response
; representation of server state

MODIFY /resource/A
; including all request & content parameters

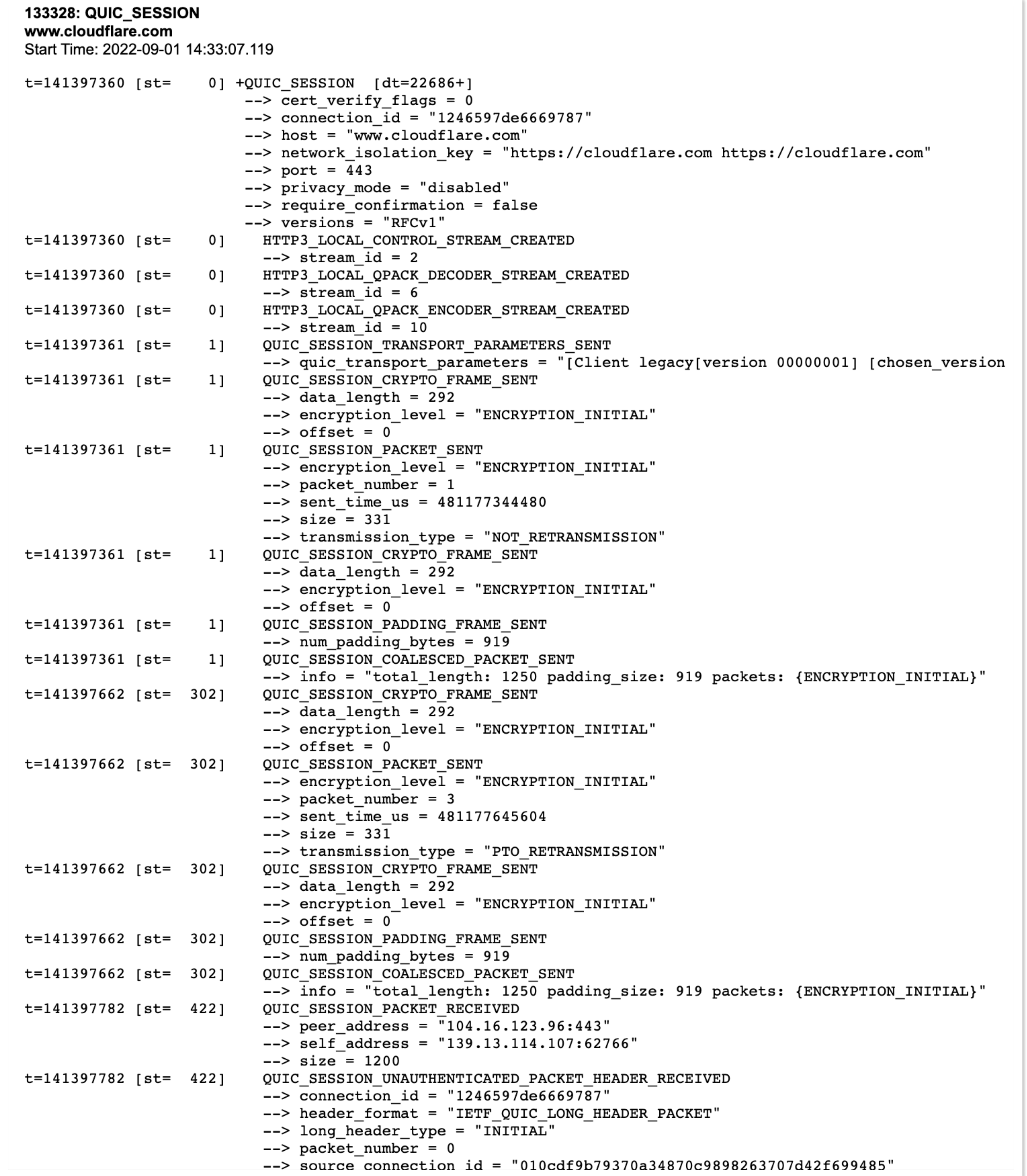
Response
; representation of server state

RESTful Reality

HTTP3



- **Connections, security contexts, channels**
- **Request parameters, cookies**



133328: QUIC_SESSION
www.cloudflare.com
Start Time: 2022-09-01 14:33:07.119

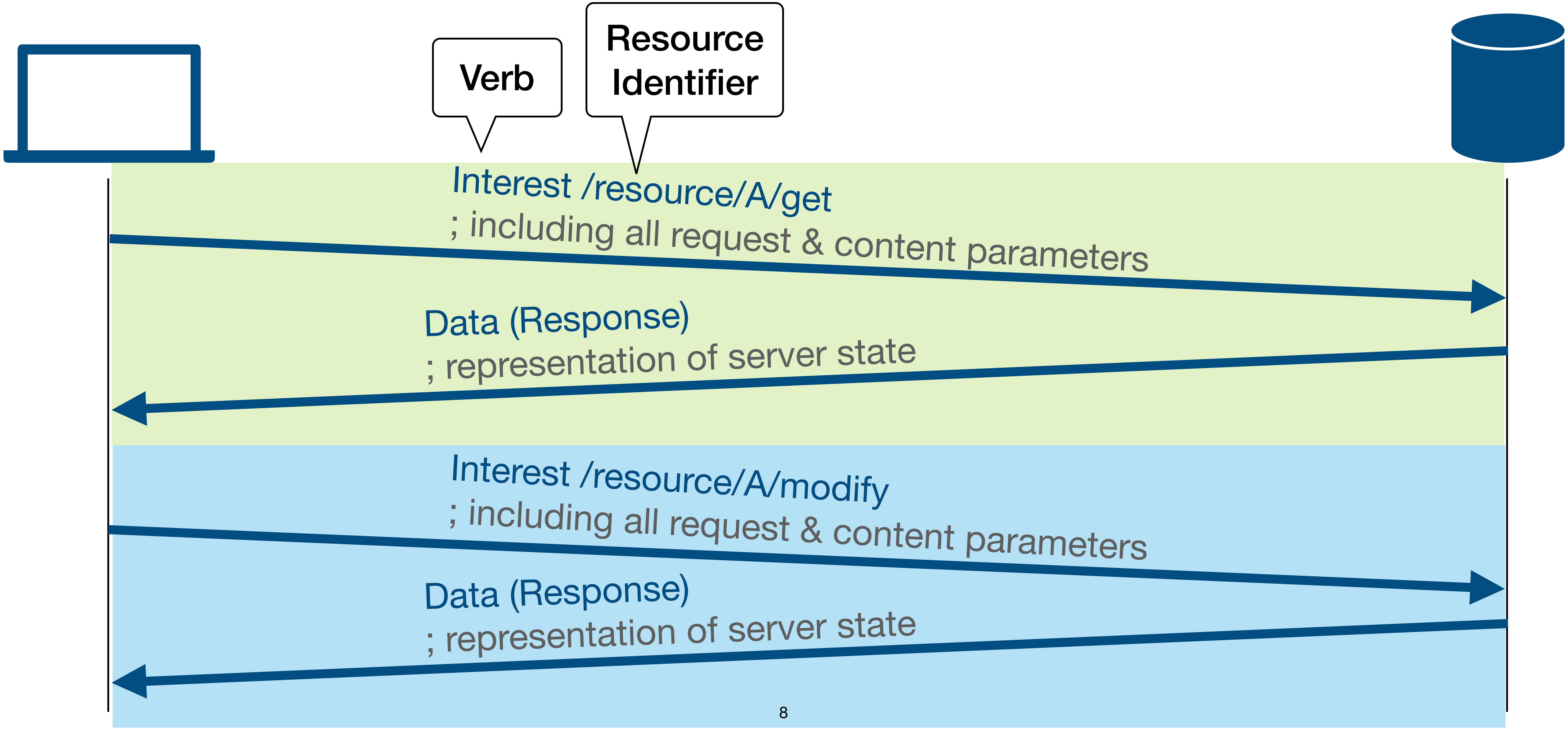
```
t=141397360 [st=    0] +QUIC_SESSION  [dt=22686+]
                        --> cert_verify_flags = 0
                        --> connection_id = "1246597de6669787"
                        --> host = "www.cloudflare.com"
                        --> network_isolation_key = "https://cloudflare.com https://cloudflare.com"
                        --> port = 443
                        --> privacy_mode = "disabled"
                        --> require_confirmation = false
                        --> versions = "RFCv1"
t=141397360 [st=    0]    HTTP3_LOCAL_CONTROL_STREAM_CREATED
                          --> stream_id = 2
t=141397360 [st=    0]    HTTP3_LOCAL_QPACK_DECODER_STREAM_CREATED
                          --> stream_id = 6
t=141397360 [st=    0]    HTTP3_LOCAL_QPACK_ENCODER_STREAM_CREATED
                          --> stream_id = 10
t=141397361 [st=    1]    QUIC_SESSION_TRANSPORT_PARAMETERS_SENT
                          --> quic_transport_parameters = "[Client legacy[version 00000001] [chosen_version
t=141397361 [st=    1]    QUIC_SESSION_CRYPTO_FRAME_SENT
                          --> data_length = 292
                          --> encryption_level = "ENCRYPTION_INITIAL"
                          --> offset = 0
t=141397361 [st=    1]    QUIC_SESSION_PACKET_SENT
                          --> encryption_level = "ENCRYPTION_INITIAL"
                          --> packet_number = 1
                          --> sent_time_us = 481177344480
                          --> size = 331
                          --> transmission_type = "NOT_RETRANSMISSION"
t=141397361 [st=    1]    QUIC_SESSION_CRYPTO_FRAME_SENT
                          --> data_length = 292
                          --> encryption_level = "ENCRYPTION_INITIAL"
                          --> offset = 0
t=141397361 [st=    1]    QUIC_SESSION_PADDING_FRAME_SENT
                          --> num_padding_bytes = 919
t=141397361 [st=    1]    QUIC_SESSION_COALESCED_PACKET_SENT
                          --> info = "total_length: 1250 padding_size: 919 packets: {ENCRYPTION_INITIAL}"
t=141397662 [st=  302]    QUIC_SESSION_CRYPTO_FRAME_SENT
                          --> data_length = 292
                          --> encryption_level = "ENCRYPTION_INITIAL"
                          --> offset = 0
t=141397662 [st=  302]    QUIC_SESSION_PACKET_SENT
                          --> encryption_level = "ENCRYPTION_INITIAL"
                          --> packet_number = 3
                          --> sent_time_us = 481177645604
                          --> size = 331
                          --> transmission_type = "PTO_RETRANSMISSION"
t=141397662 [st=  302]    QUIC_SESSION_CRYPTO_FRAME_SENT
                          --> data_length = 292
                          --> encryption_level = "ENCRYPTION_INITIAL"
                          --> offset = 0
t=141397662 [st=  302]    QUIC_SESSION_PADDING_FRAME_SENT
                          --> num_padding_bytes = 919
t=141397662 [st=  302]    QUIC_SESSION_COALESCED_PACKET_SENT
                          --> info = "total_length: 1250 padding_size: 919 packets: {ENCRYPTION_INITIAL}"
t=141397782 [st=  422]    QUIC_SESSION_PACKET_RECEIVED
                          --> peer_address = "104.16.123.96:443"
                          --> self_address = "139.13.114.107:62766"
                          --> size = 1200
t=141397782 [st=  422]    QUIC_SESSION_UNAUTHENTICATED_PACKET_HEADER_RECEIVED
                          --> connection_id = "1246597de6669787"
                          --> header_format = "IETF_QUIC_LONG_HEADER_PACKET"
                          --> long_header_type = "INITIAL"
                          --> packet_number = 0
                          --> source_connection_id = "010cdf9b79370a34870c9898263707d42f699485"
```

Information-Centric REST?

- **ICN-idiomatic RESTful communications as a building block for applications**
  - Clients and servers in a sessions
  - Common understanding of state evolution
  - Suitable for a broad range of applications
  - At least HTTP/TLS's security and privacy features
- **Can we do this better than state of the art (HTTP3/QUIC/TLS-1.3)?**
  - Simpler protocol machinery
  - Less overhead on the wire
  - Leveraging typical ICN benefits

Naïve ICN Approach

Interests as Vehicles for Requests



Verb
Resource Identifier
Interest /resource/A/get
; including all request & content parameters
Data (Response)
; representation of server state
Interest /resource/A/modify
; including all request & content parameters
Data (Response)
; representation of server state

Naïve ICN Approach

Interests as Vehicles for Requests

- **Flow balance**
  - Request parameters can require a lot of bytes (often more than the state representation in the response)
  - Interests are intended to regulate Data packets
- **Computational overload attacks on server**
- **Application layer processing time vs. network layer timeouts**
- **Secure sessions and name confidentiality**

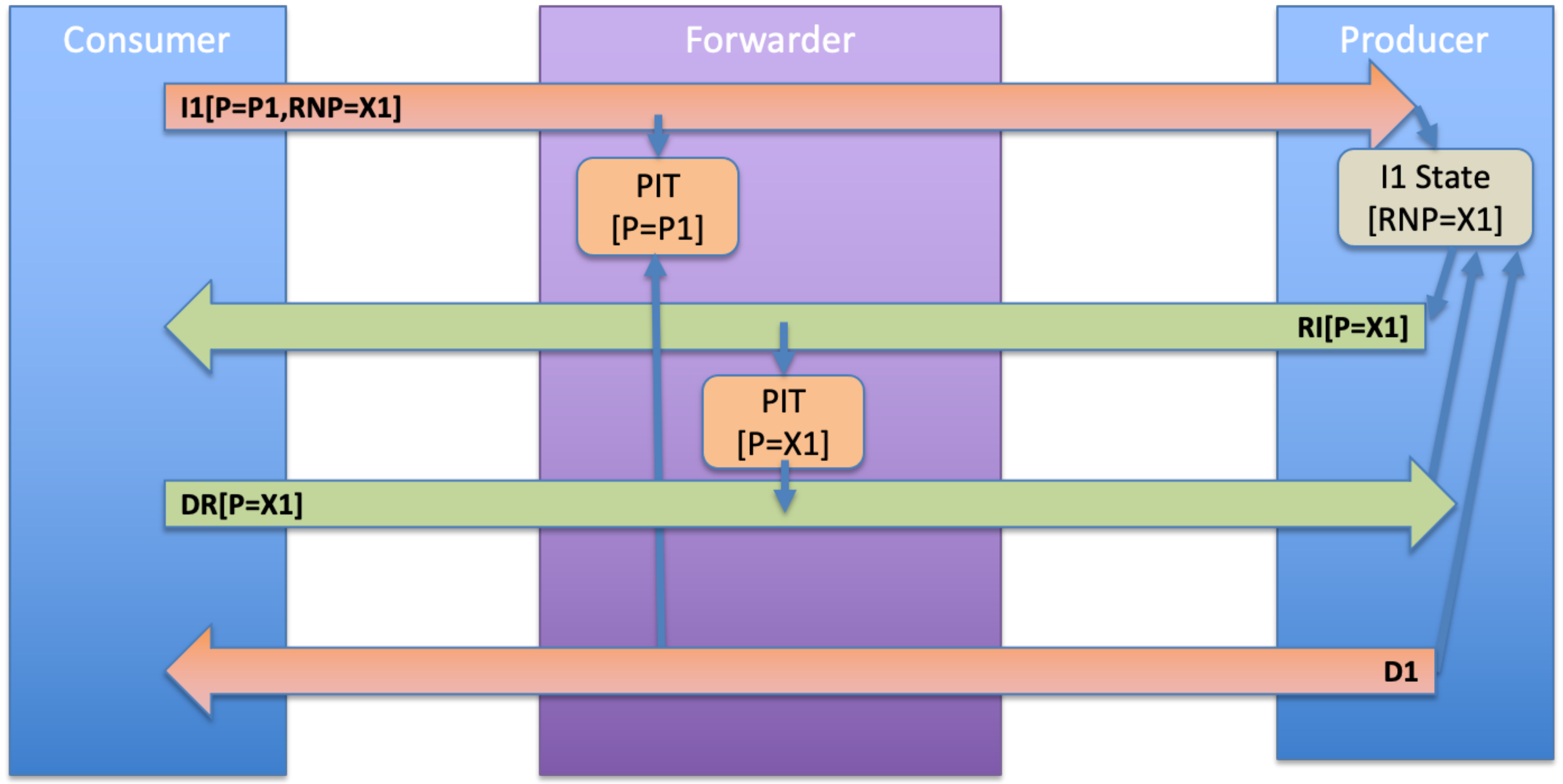
Naïve ICN Approach

Interests as Vehicles for Requests

- **Flow balance**
    - Request parameters can require a lot of bytes (often more than the state representation in the response)
    - Interests are intended to regulate Data packets
- **Computational overload attacks on server**
- **Application layer processing time vs. network layer timeouts**
- **Secure sessions and name confidentiality**

Reflexive Forwarding and RICE

draft-oran-icnrg-reflexive-forwarding



RESTful ICN Design

Data-oriented REST Sessions

- **Enable client/server communication**
    - With a series of request/response interactions in a session context
- **Employ Reflexive Forwarding for RPC communication**
    - Allow for robust ICN-idiomatic client/server communication with client parameter passing
    - For both key exchange and actual RESTFul communication
- **Enable secure RESTful communication using standard ICN mechanisms**
    - Content Object encryption and signatures
    - Without forcing all interactions into TLS-like tunnels

RESTful ICN Design 2

Efficiency

- **Supporting a series of requests (in a session)**
  - Avoid setting up context state for every request and the corresponding protocol interactions
- **Establish and maintain shared "session" state**
  - Using identifiers of keys and associated security context negotiated by setup phase
  - Reflexive Forwarding Parameter passing machinery for clients to refer to previously created application state
  - Emulating HTTP cookies

RESTful ICN Design 3

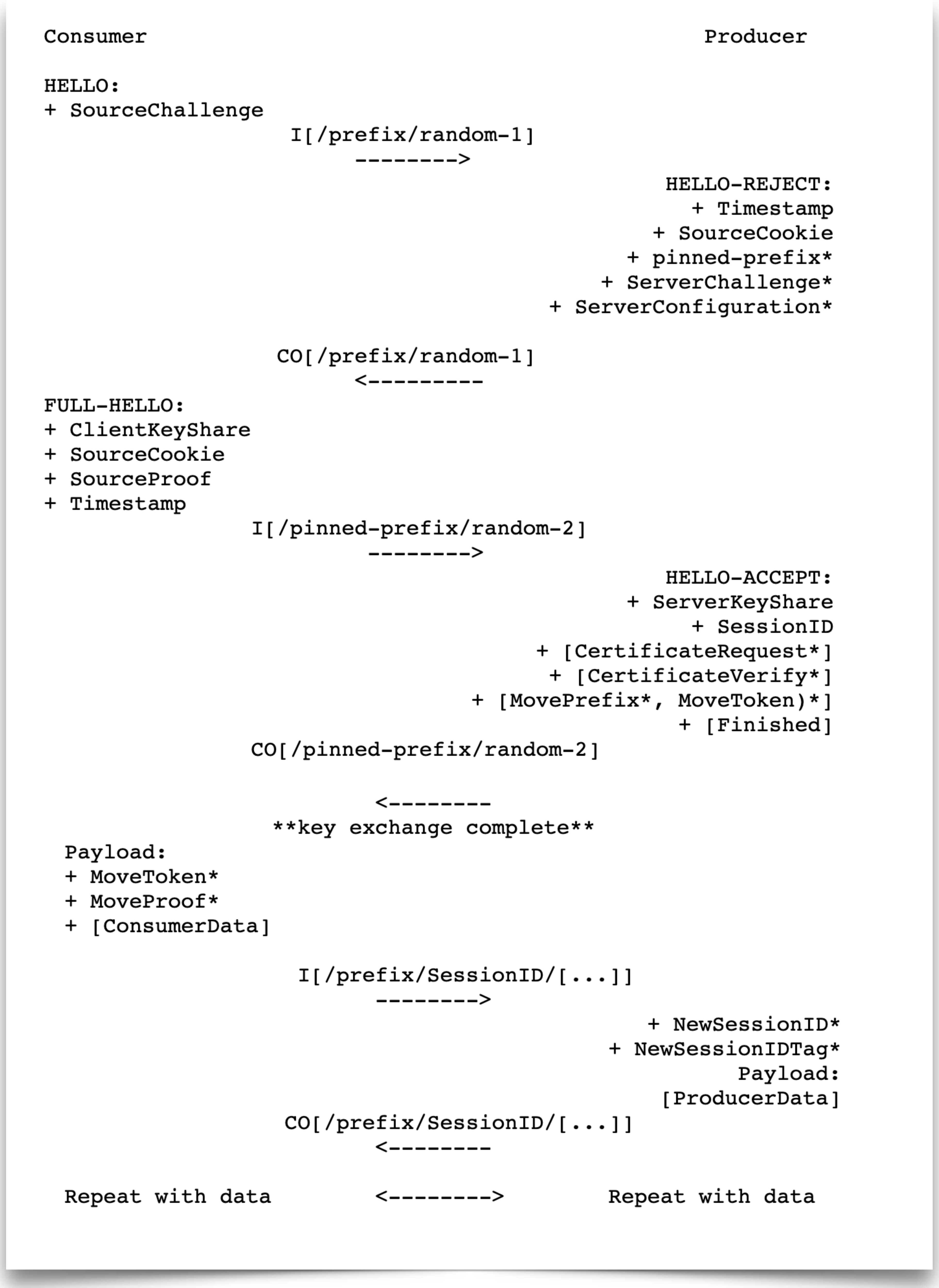
State Management

- **Secure referent state held on a particular server (through key-ids) and a referent to application state through parameters secured through those keys**
- **Basis for enabling key features of today's session based RESTful protocols**
  - Application state caching on clients to allow server agility
  - Securing application state exchanged through pair-wise session keys with particular server
  - Rapid setup of these keys using TLS 1.3-compliant key exchange protocol
  - Efficient state evolution (minimizing round-trips and state representation overhead)
  - RESTful semantics for multiple interactions with the application through the same server
- **Caveat**
  - Have to make sure that client talks to the same server over multiple requests
  - Or that there is some server-side state synchronization machinery

CCNx Key Exchange

Mosko, Ersin, Wood: draft-wood-icnrg-ccnxkeyexchange

- **TLS-1.3-like key exchange protocol between two peers**
  - For establishing a shared, forward-secure key for secure and confidential communication
- **Wraps "inner" ICN communication (Interest/Data) into "outer", TLS-style secured Interest/Data exchanges**
  - Orthogonal to reliability and congestion control
- **Designed for client/server scenarios**
  - Protection against computational overload attacks
  - Can use different infrastructure for security and service functions

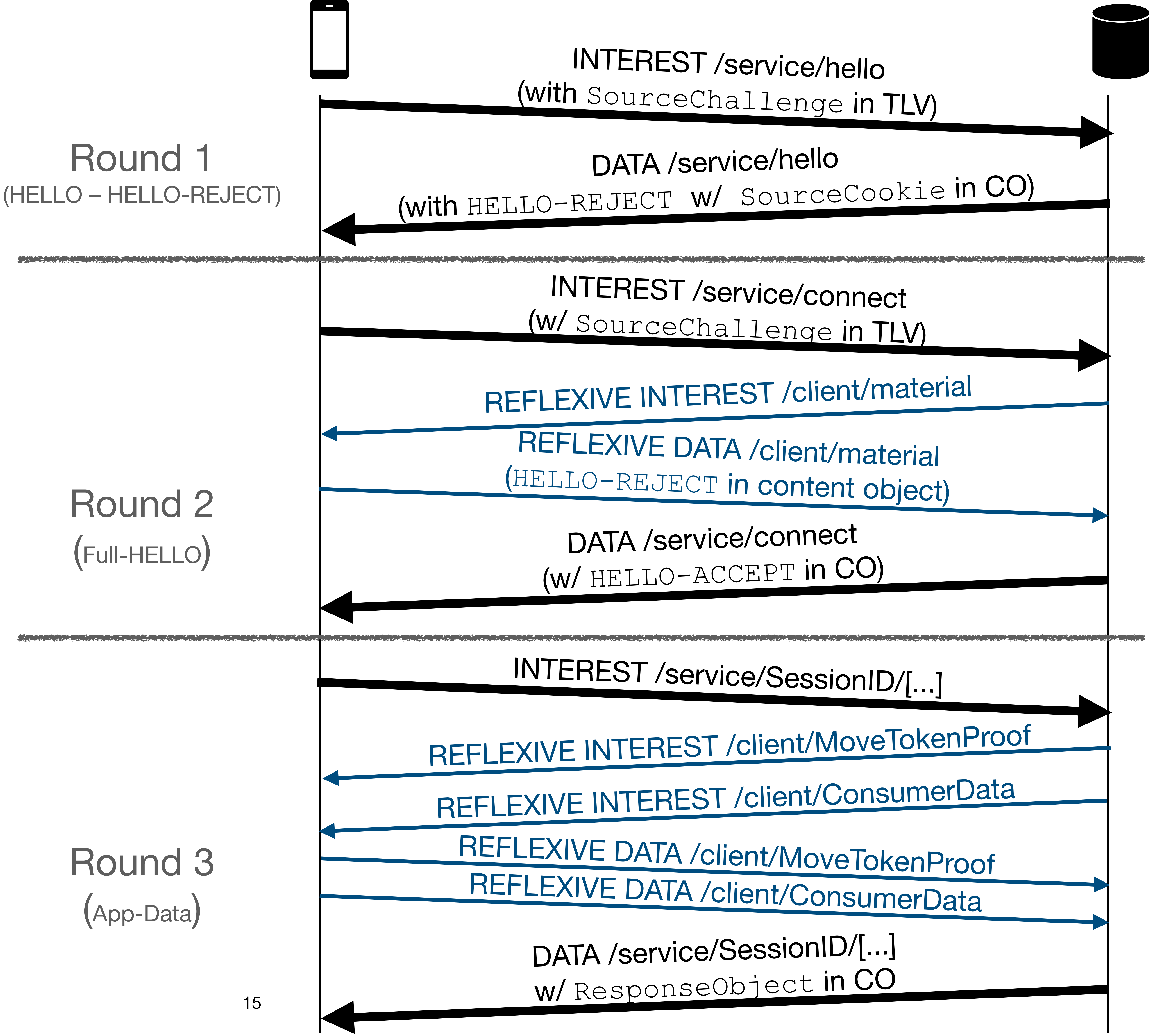


```
Consumer                                           Producer

HELLO:
+ SourceChallenge
                   I[/prefix/random-1]
                        -------->
                                                HELLO-REJECT:
                                                  + Timestamp
                                               + SourceCookie
                                             + pinned-prefix*
                                           + ServerChallenge*
                                       + ServerConfiguration*

                  CO[/prefix/random-1]
                        <---------
FULL-HELLO:
+ ClientKeyShare
+ SourceCookie
+ SourceProof
+ Timestamp
                I[/pinned-prefix/random-2]
                         -------->
                                                HELLO-ACCEPT:
                                             + ServerKeyShare
                                                  + SessionID
                                      + [CertificateRequest*]
                                       + [CertificateVerify*]
                                 + [MovePrefix*, MoveToken)*]
                                                 + [Finished]
                CO[/pinned-prefix/random-2]

                          <--------
                  **key exchange complete**
  Payload:
  + MoveToken*
  + MoveProof*
  + [ConsumerData]

                    I[/prefix/SessionID/[...]]
                          -------->
                                               + NewSessionID*
                                            + NewSessionIDTag*
                                                      Payload:
                                               [ProducerData]
                   CO[/prefix/SessionID/[...]]
                          <--------

  Repeat with data        <--------->        Repeat with data
```

RESTful ICN

Session Setup

- **Integrating CCNx-style kex exchanges in Reflexive Forwarding framework**
  - Same semantics
  - Less data in unsolicited Interests
  - A few more roundtrips
- **Coupling session state and keying**
  - Key revocation => session termination



Round 1
(HELLO – HELLO-REJECT)

INTEREST /service/hello
(with `SourceChallenge` in TLV)

DATA /service/hello
(with `HELLO-REJECT` w/ `SourceCookie` in CO)

Round 2
(Full-HELLO)

INTEREST /service/connect
(w/ `SourceChallenge` in TLV)

REFLEXIVE INTEREST /client/material

REFLEXIVE DATA /client/material
(`HELLO-REJECT` in content object)

DATA /service/connect
(w/ `HELLO-ACCEPT` in CO)

Round 3
(App-Data)

INTEREST /service/SessionID/[...]

REFLEXIVE INTEREST /client/MoveTokenProof

REFLEXIVE INTEREST /client/ConsumerData

REFLEXIVE DATA /client/MoveTokenProof

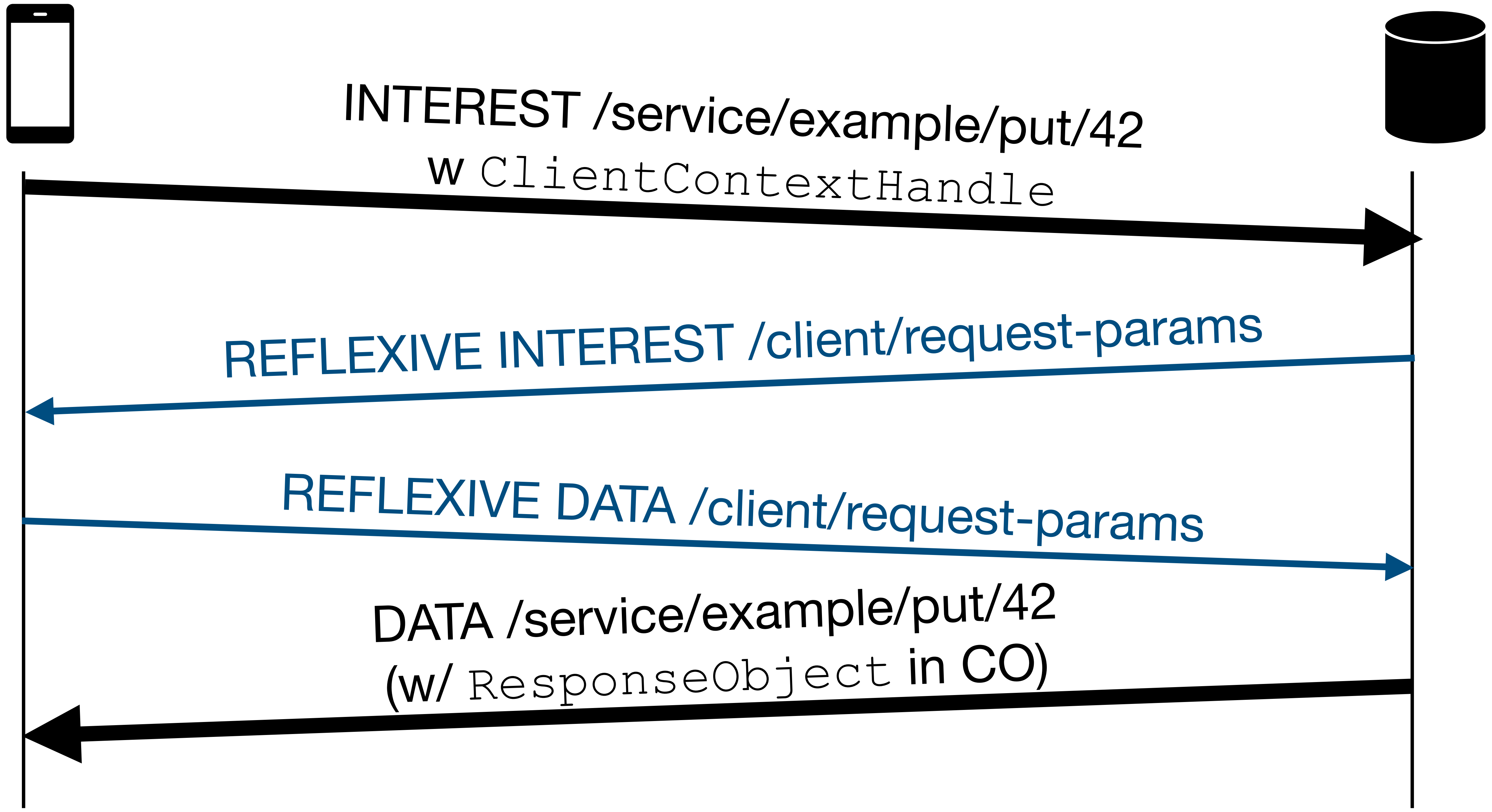
REFLEXIVE DATA /client/ConsumerData

DATA /service/SessionID/[...]
w/ `ResponseObject` in CO

RESTful ICN

Requests and Responses

- **Reflexive Forwarding**
  - ClientContextHandle in initial Interest
  - Contains necessary SessionID and key-id for the security context
  - Plus encrypted name for application state representation
- **Reponses**
  - Request results
  - Encrypted name for new session state representation
- **Not using tunnel-like encryption**
  - Encrypting content objects with symmetric key


INTEREST /service/example/put/42
w ClientContextHandle
REFLEXIVE INTEREST /client/request-params
REFLEXIVE DATA /client/request-params
DATA /service/example/put/42
(w/ ResponseObject in CO)

Conclusions

- **Time to think about web over ICN: basic Interest/Data not enough**
- **Key idea here: Integrating key exchange with reflexive forwarding**
  - Provide required context handles in initial Initial interest
  - Use negotiated keys for symmetric content object encryption
- **Approximate capabilities of current state of the art (HTTP3/QUIC or TCP)**
  - Overcoming complexities of 3 layer approach with isolated implementations and protocol machinery
  - Potentially easier to implement
  - Still enjoying the usual ICN greatness
- **Future work**
  - Name privacy
  - Build it