Secure Replicas and Nomad Sessions with CCNxKE

Christopher A. Wood
UCI and PARC

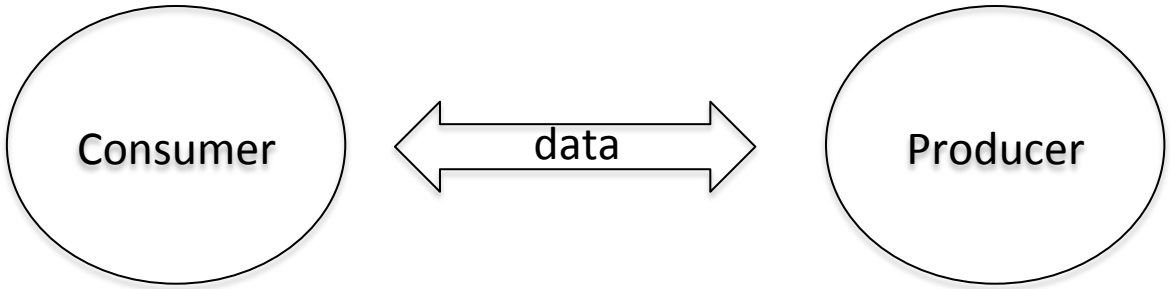
ICNRG Interim Meeting – IETF 96 – Berlin

July 17, 2016

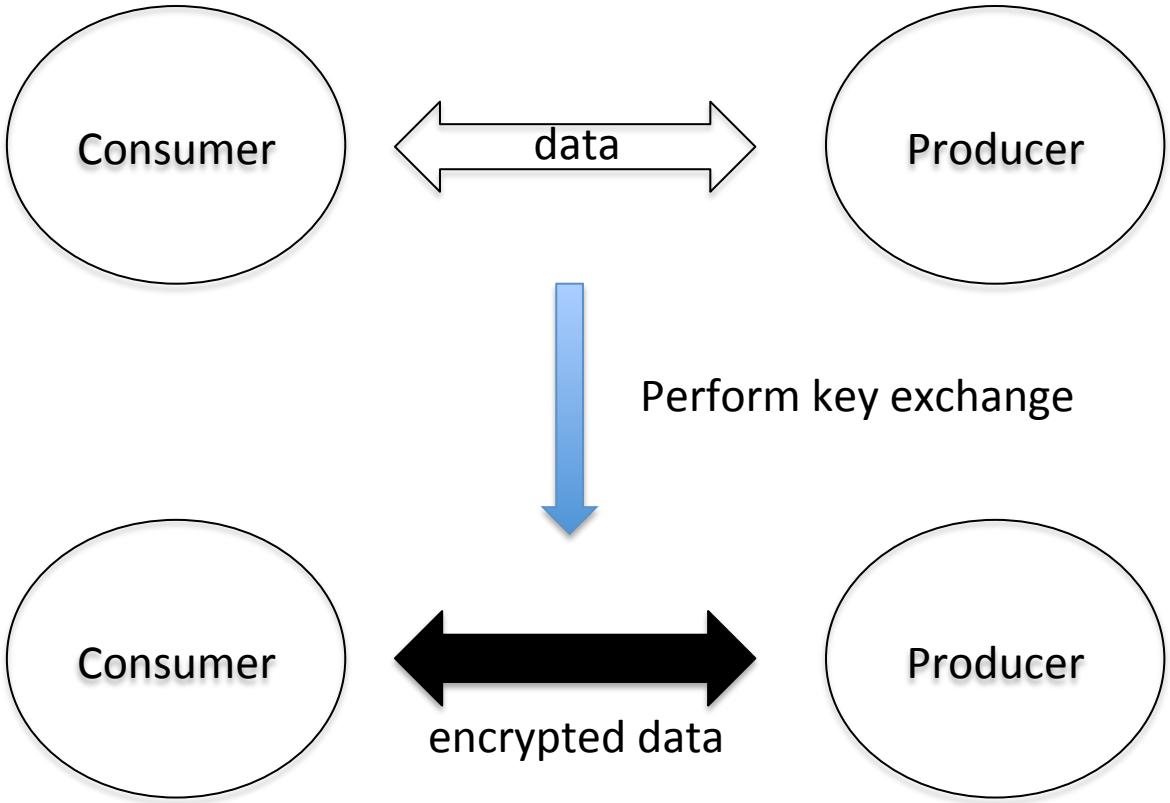
Session-Based Communication in CCN

- Problem:
  - A client and server (replica) want to establish a secure session in which all messages will be encrypted
- One approach:
  - Use CCNx-KE – a TLS-like key exchange protocol tailored for CCN
  - Clients authenticate the server (and vice versa) and the parties establish a shared forward-secure session key
  - The session key is used to encrypt all subsequent traffic carrying application data

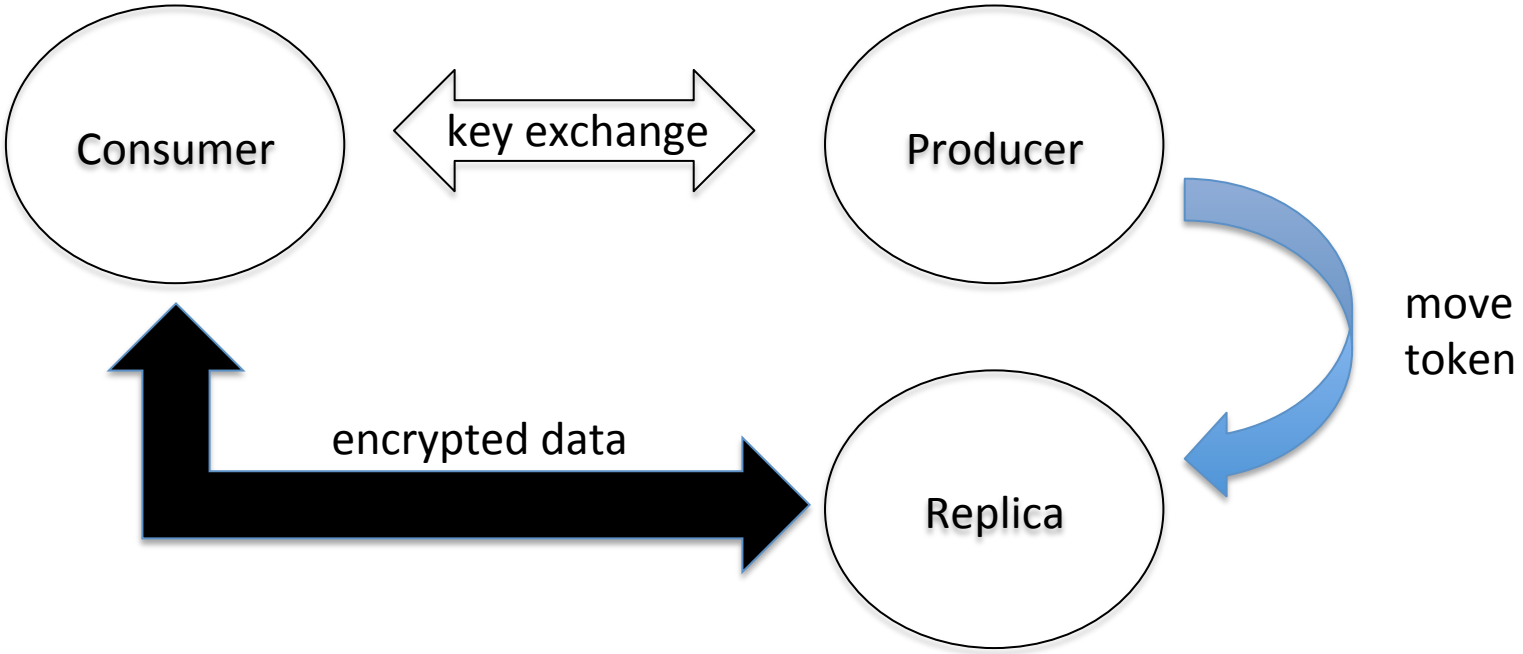
Standard CCN Session Communication



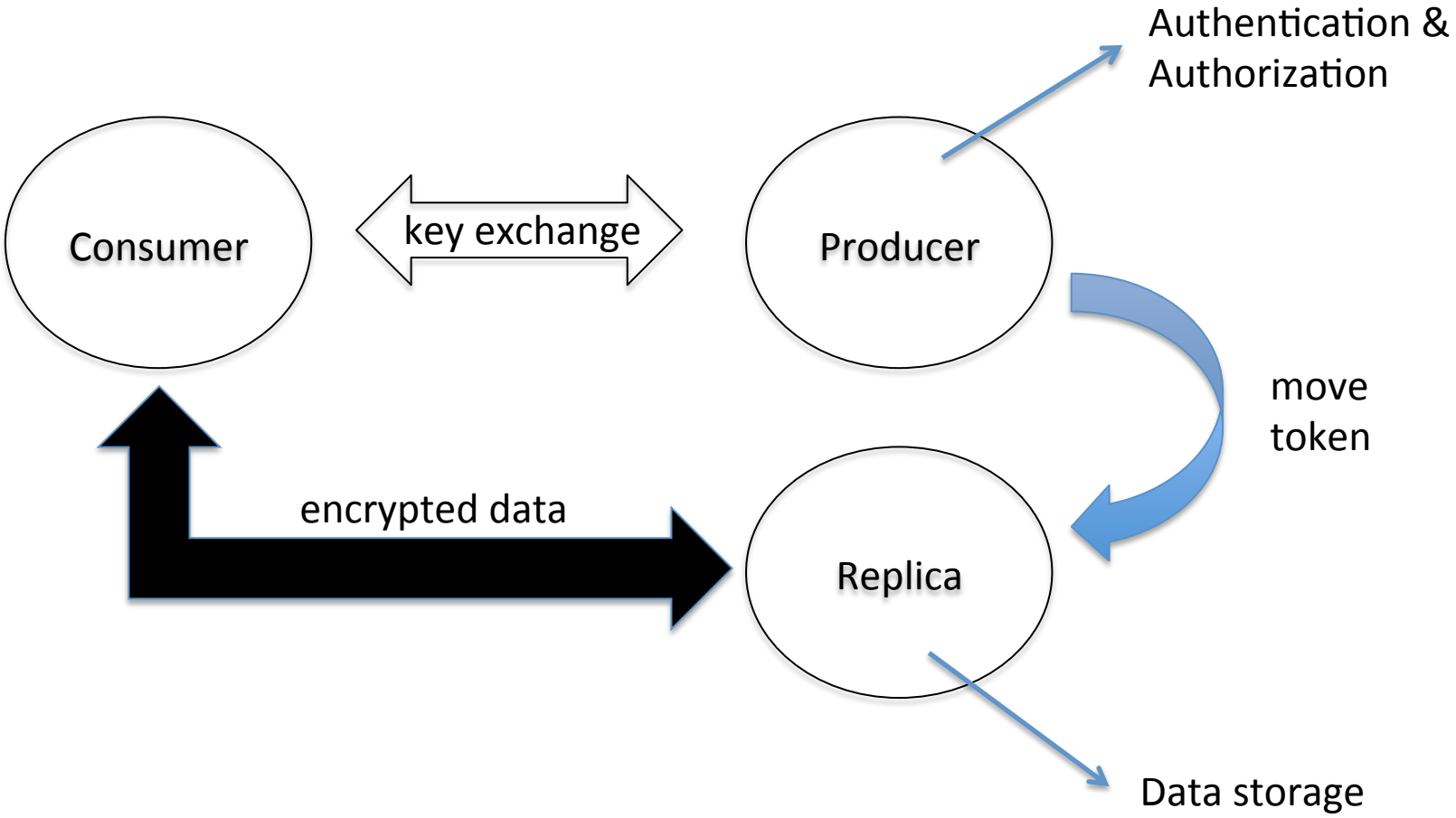
Standard CCN Session Communication



Session Relocation



Session Relocation



Problems to Address

1. What is the trust relationship between the producer and the replica?
   - Same or different owner?
2. How is the session transferred from the producer or the replica?
   - Are tokens stateful or stateless?

Trust Model #1

- The producer and replica are owned by the same entity
  - They can share a key that's frequently rotated

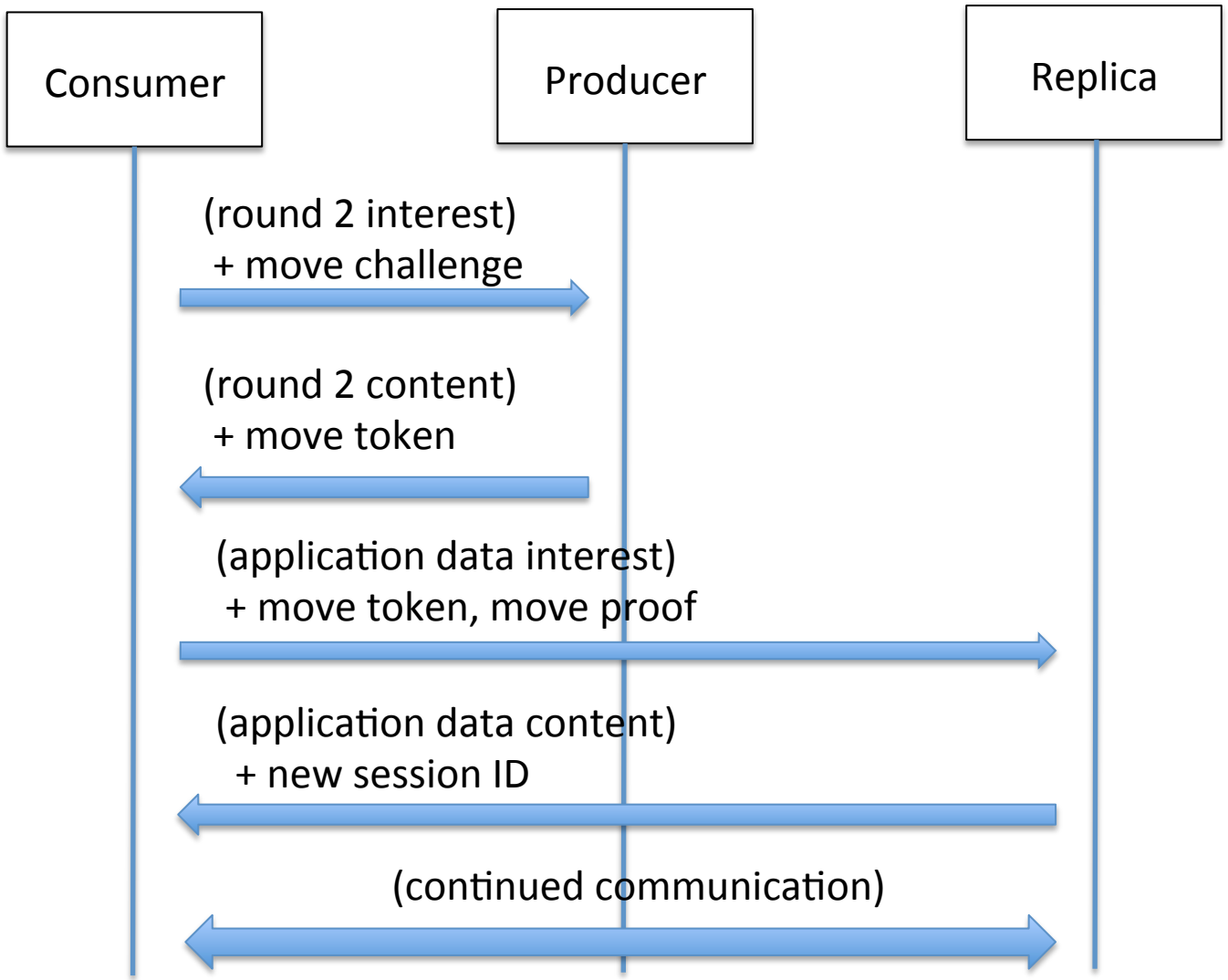
Trust Model #2

- The producer and replica have some relationship
  - The producer pays for replica services
  - A MNO distributes users to the best replica
  - The authentication server passes the user to a load balancer (via a move token)
- The producer and replicate create a session and re-key on a regular basis

Relocation Approach

- Session relocation requires the traffic secret to be recovered at the replica
- Trust model #1 (easy):
  - Tokens are stateful
  - Consumer tokens prove nothing
- Trust model #2 (hard):
  - Consumer tokens must prove that they came from the producer

Move Token Usage



Move Token Construction

- Move challenge

  $Y = H(X)$, for some $X \leftarrow \{0,1\}^{128}$

- Move token

  $T = k_{ID} \,||\, Enc_k(Y \,||\, \text{traffic\_secret})$

- Move proof

  X

Move Token Construction

- Move challenge

  $Y = H(X)$, for some $X \leftarrow \{0,1\}^{128}$

- Move token

  $T = k_{ID} \,||\, Enc_k(Y \,||\, traffic\_secret)$

- Move proof

  X

Replica check:

1. If k_{ID} not valid, drop
2. $Y \,||\, traffic\_secret = Dec_k(T)$
3. If H(X) != Y, drop

Properties

- k_{ID} is a key that's routinely refreshed between the producer and replica (e.g., on a daily basis).
- Replica work is minimized:
  - no public-key crypto
  - single symmetric decryption and hash computation
- Two round trips before data can be retrieved
  - 1) Authenticate with the producer (2)
  - 2) Start a new session with the replica and get the first chunk of data (0)

Summing Up

- CCNx-KE is used to separate authentication and authorization from the retrieval of actual application data.
- Producers can upload encrypted data to a replica that only authorized consumers can decrypt.
- The replica session is used as a form of “transport encryption.”

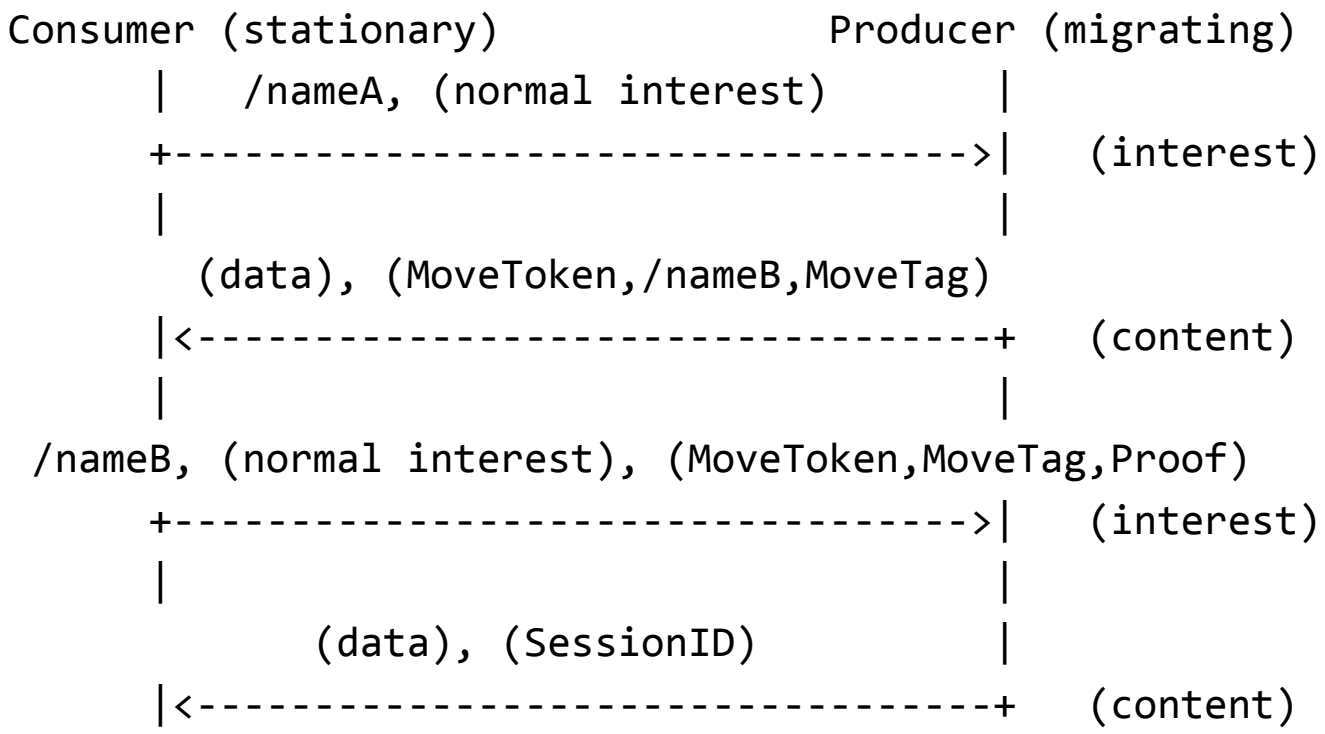
Session Identifiers and Secrets

- Traffic secrets are bound to a session identifier
- Session identifiers are bound to a **name prefix**
- CCNxKE handshakes can establish bidirectional session identifiers
  - Consumer to producer
  - Producer to consumer

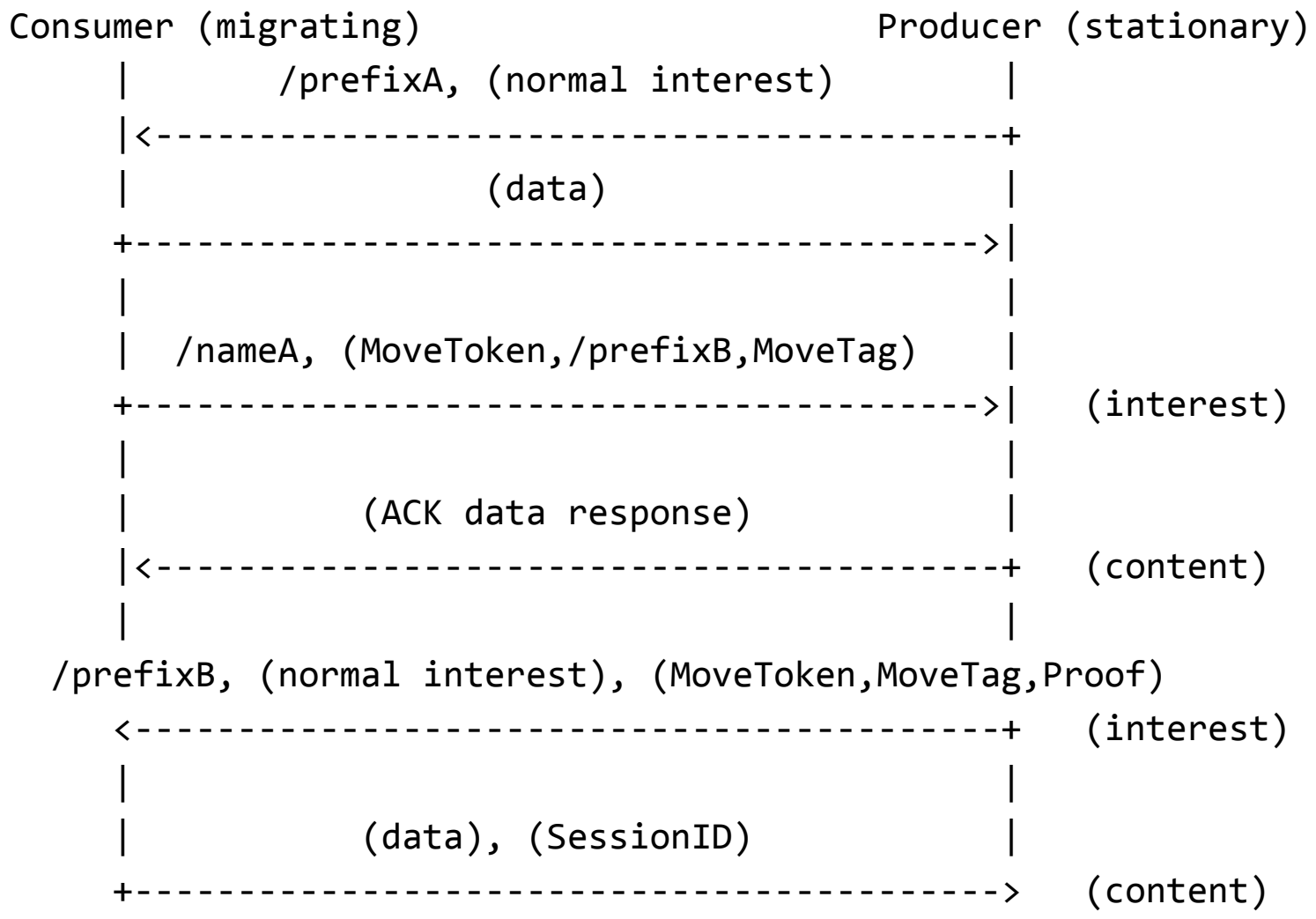
Nomad Sessions

- If names are location-agnostic, consumers and producers can move freely without re-establishing sessions
  - Contrast to TCP-based TLS sessions
- If either end-host moves we want to minimize or prevent re-keying
  - How? Generalize move tokens

Nomad Example #1
(soft handoff)



Nomad Example #2
(soft handoff)



Don’t Reinvent the Wheel

- RFC 5169: Handover Key Management and Re-Authentication Problem Statement
- RFC 6696: EAP Extensions for the EAP Re-authentication Protocol (ERP)
- RFC 6697: Handover Keying (HOKEY) Architecture Design
- Mobile DTLS (draft-barrett-mobile-dtls-00)