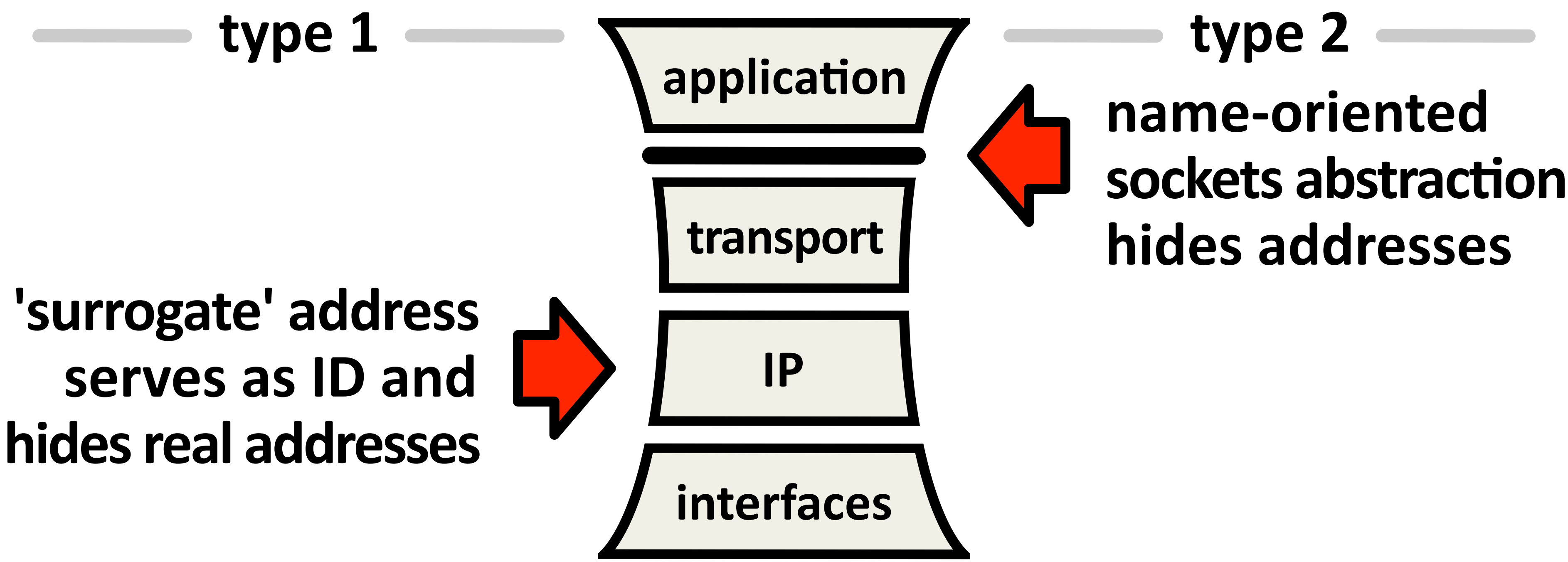
Improving Routing Scalability with Name-Oriented Sockets

Christian Vogt **Ericsson Research**

IRTF Routing research group meeting. November 2009

2 Types of Identifier-Locator Separation



- main difference is application transparency
- growing deployment for type 2, not for type 1

Transparency Implies Deployment Hurdle

transparency to application → **extra namespace** → **limited benefits for applications**

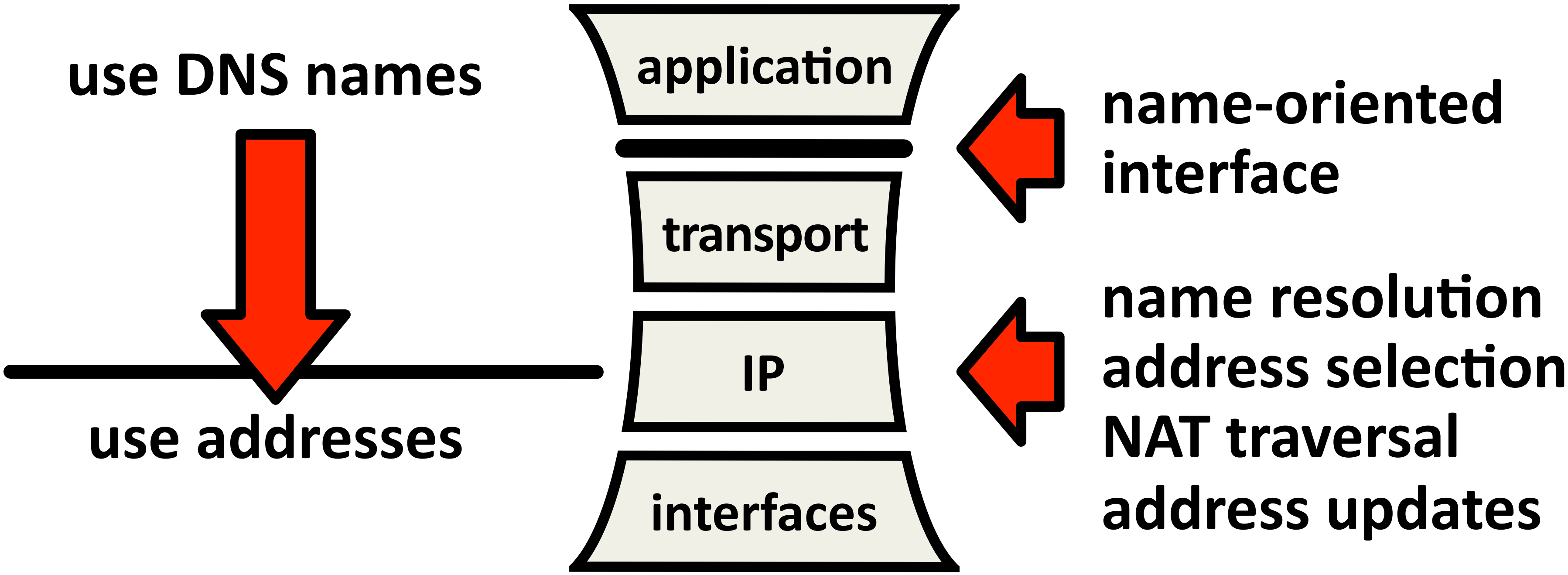
extra resolution
extra security
extra infrastructure

more complex implementation
more administration
more dependencies

→ **deployment hurdle**

application transparency does <u>not</u> aid deployment as commonly believed

Name-Oriented Sockets



- applications use DNS names bilaterally
- IP address management at IP layer
- standard IP packets

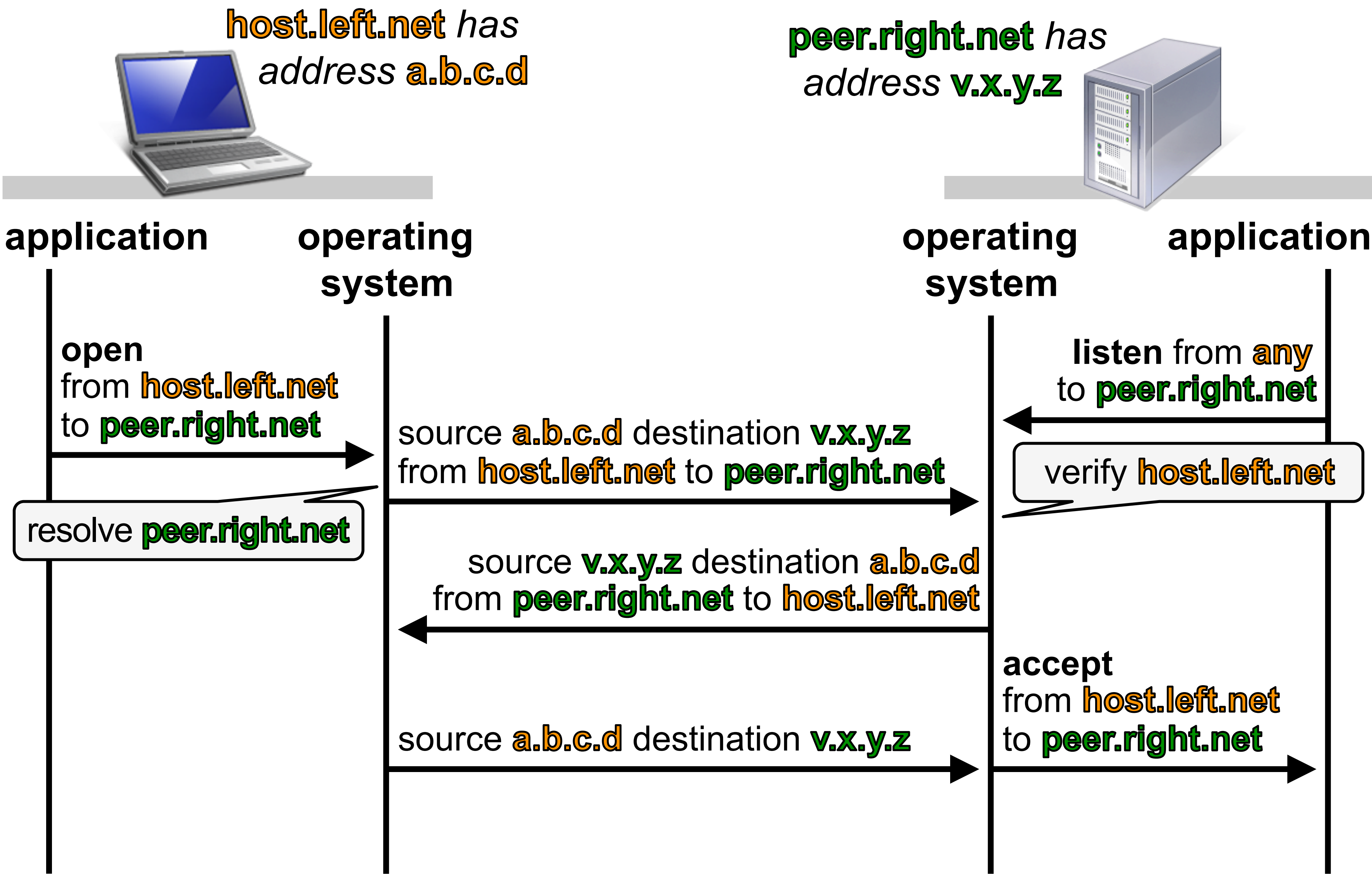
New Interface For Applications

- Listen method — prepare for incoming session
  service handle = Listen (source name, destination name, local port, transport)
- Open method — initiate outgoing session
  session handle = Open (source name, destination name, remote port, transport)
- Accept method — receive incoming session
  (source name, destination name, session handle) = Accept (handle)
- Write method — send data
  Write (session handle, data)
- Read method — receive data
  data = Read (session handle)
- Close method — close session
  Close (session handle)

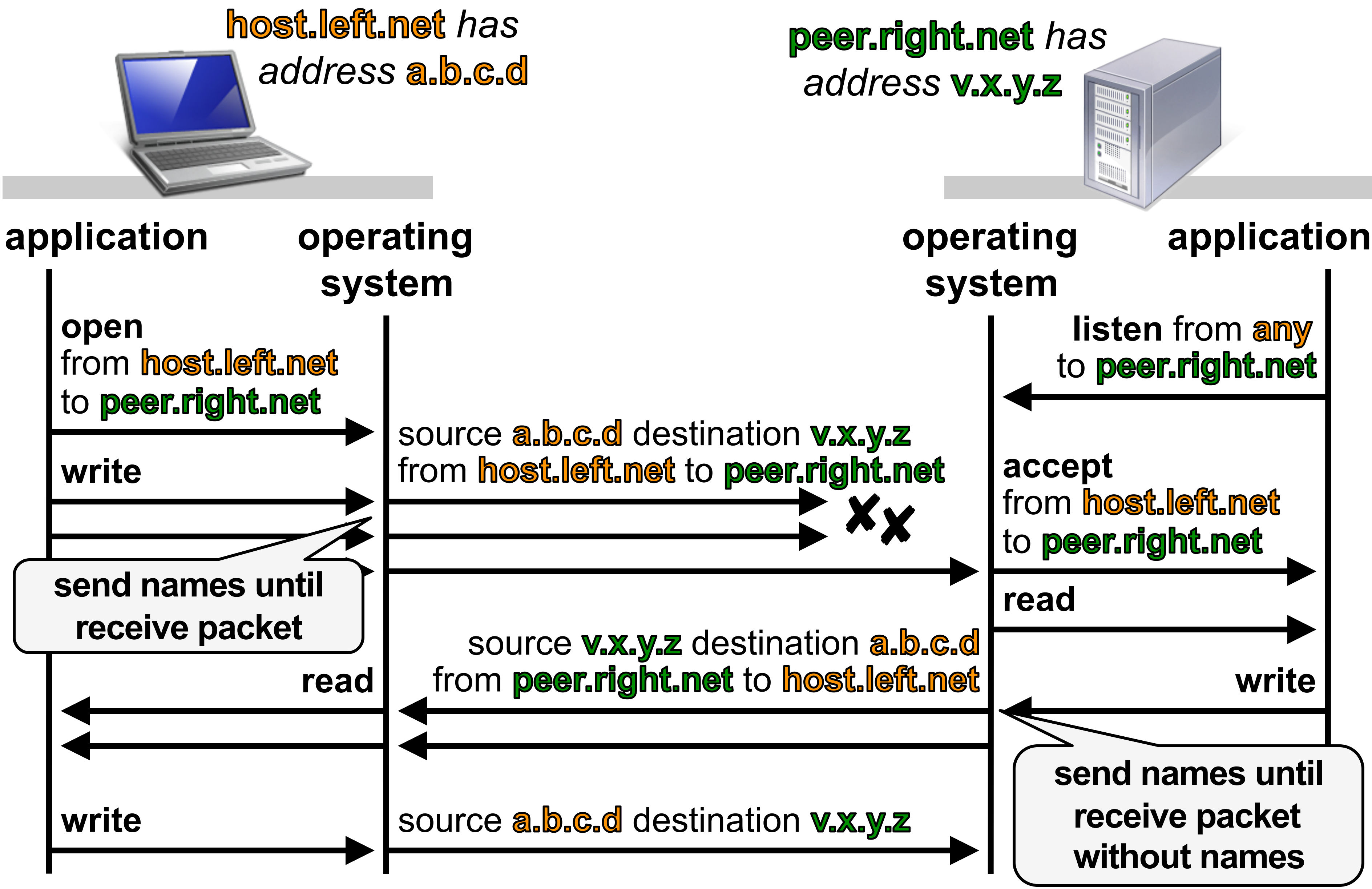
Other Components

- initial name exchange
- address updates
- backwards compatibility
- hosts without registered DNS name
- security

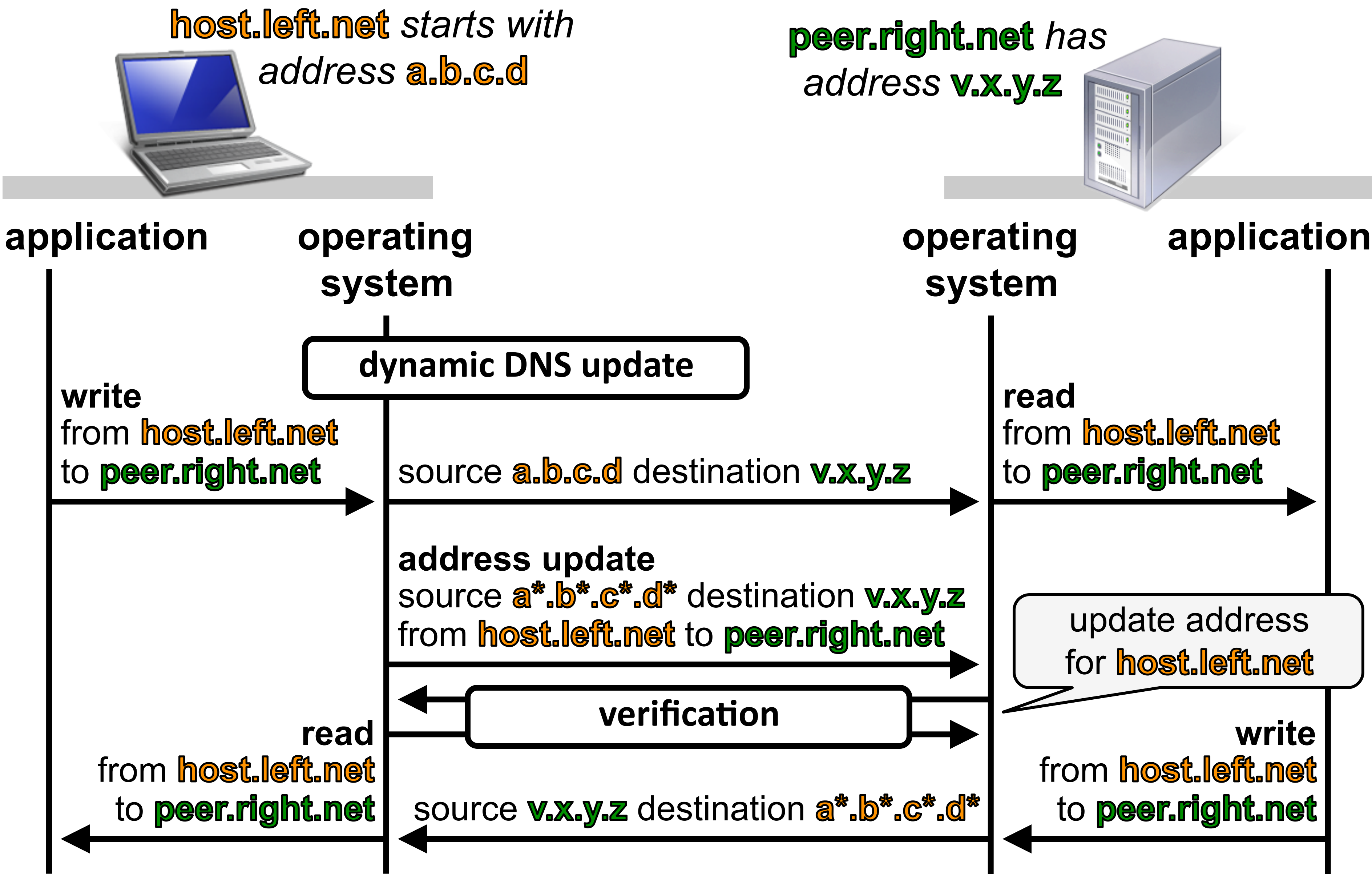
Initial Name Exchange



Initial Name Exchange



Address Updates

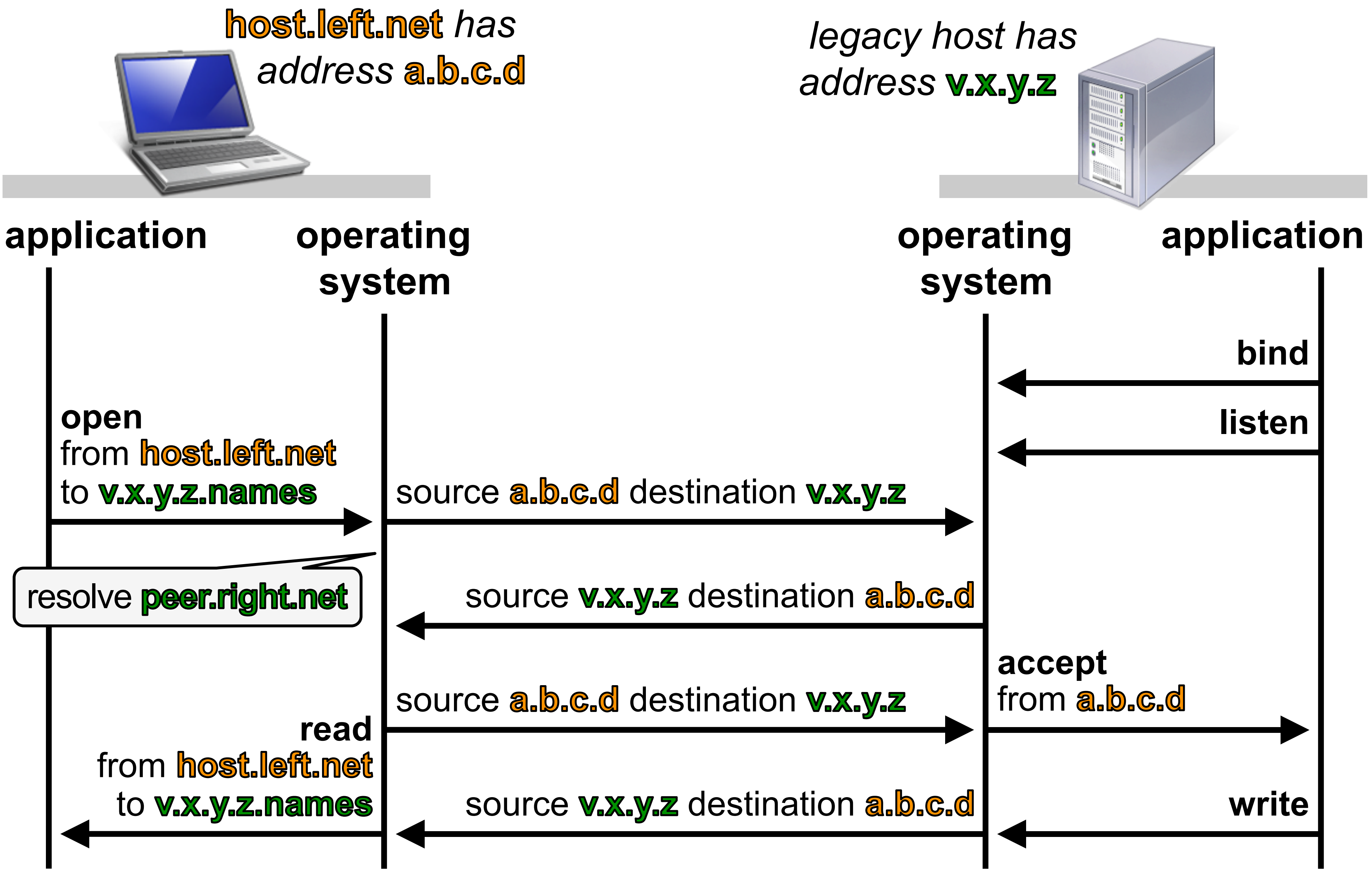


Backwards Compatibility

two types

1. legacy local application
   - old interface alongside new interface
   - remote peer sees legacy host
2. legacy remote peer
   - unilaterally name-oriented
   - address-derived name for peer

Backwards Compatibility



Security

- initial name exchange: DNS lookup by peer
  - same security as for initiating host
  - security strength depends on DNSSEC
- address update: return routability end to end
  - retains security of non-mobile Internet
- dynamic DNS update: crypto authentication
  - provisioned by hosting provider

New Dependencies On the DNS

- name-oriented stack will increase DNS load
  - more lookups
  - more dynamic updates
- scalability and convergence perhaps problematic
  - load increase never tested
  - low time-to-live values often not supported
- analysis results so far promising
  - load increase affects only lowest-level servers
  - missing time-to-live support fixable

Conclusion

- name-oriented sockets improve routing scalability
  - enable multi-homing and mobility
  - simplify renumbering
- good deployment prerequisites
  - backwards compatibility
  - advantages for application developers
  - incentives to change operating systems
  - no new infrastructure or administrative procedures
  - no dependency between stakeholders
- early prototype at Ericsson