DARPA scenario and NDN

Van Jacobson, vanj@cs.ucla.edu

Nov. 20, 2015
FIA-NP PI meeting
Arlington, VA

- At inception, ‘networking’ is created by the people that build network infrastructure.

- This creates a means-ends confusion that marginalizes the edge.

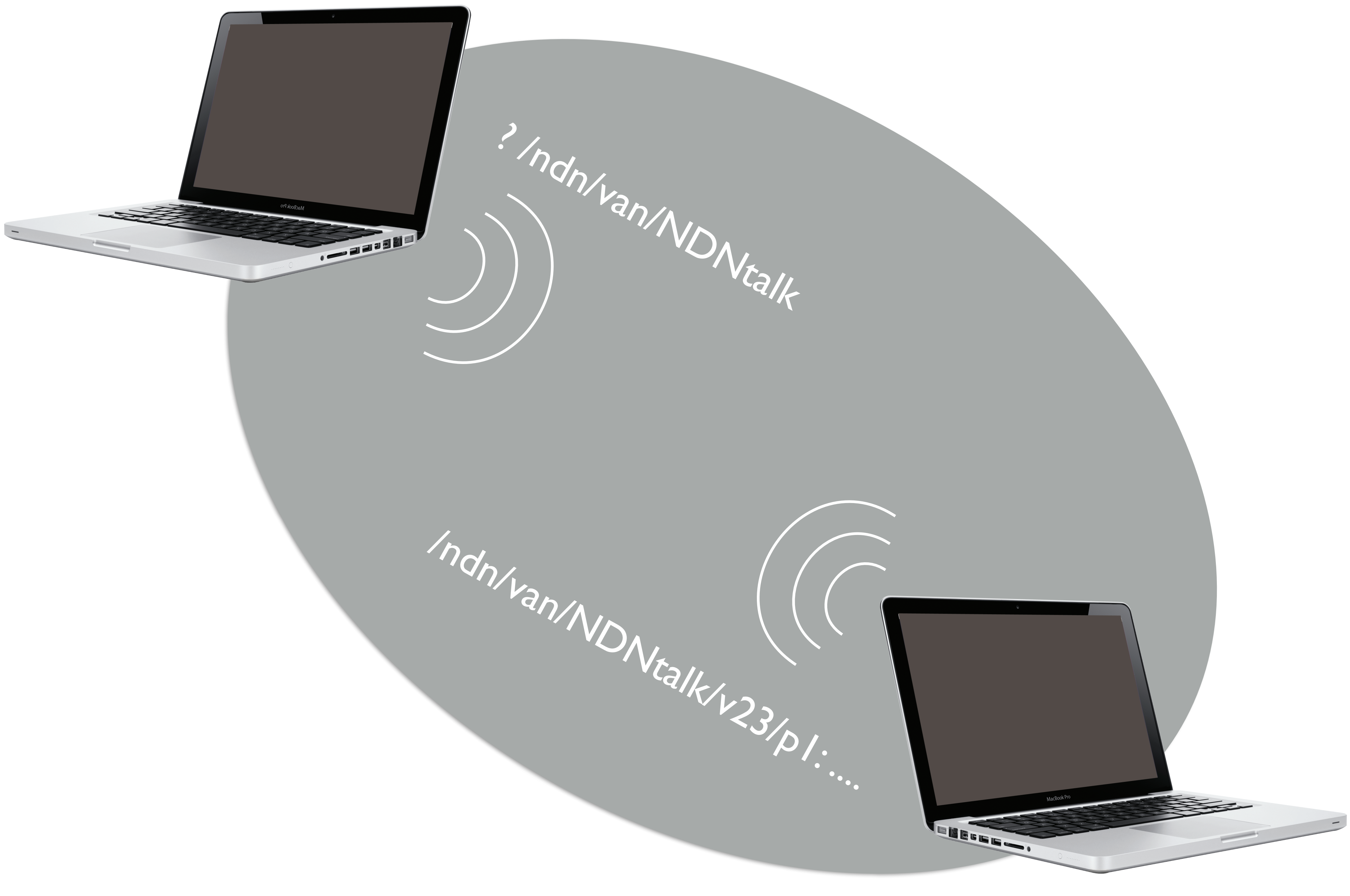
- The genesis of a network is information exchange, not infrastructure.

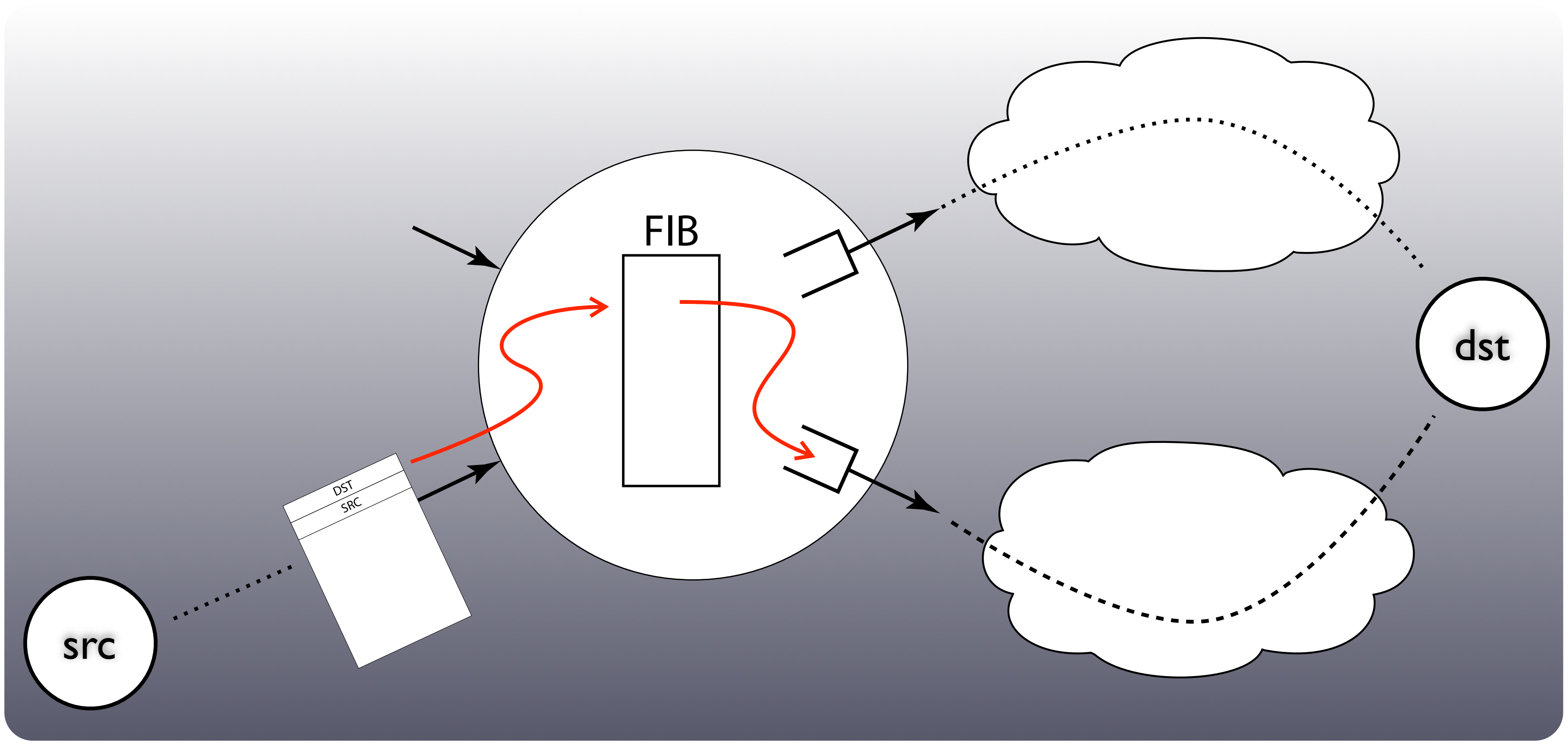
    - *The essence of information exchange is shared convention and context.*

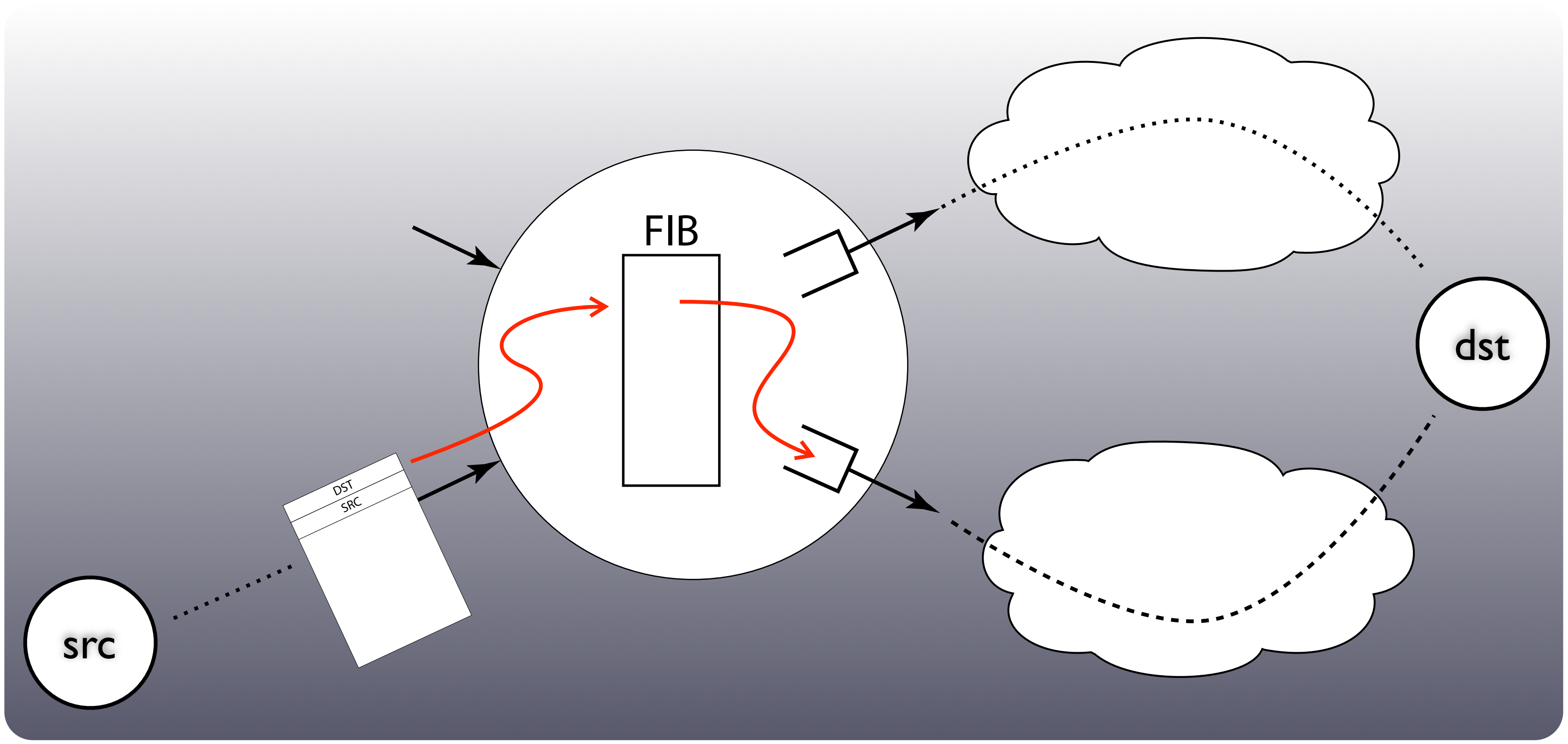
NDN is an information-based networking framework.

It standardizes conventions and patterns that enable devices to achieve their information sharing objectives using any and all available communications technology.

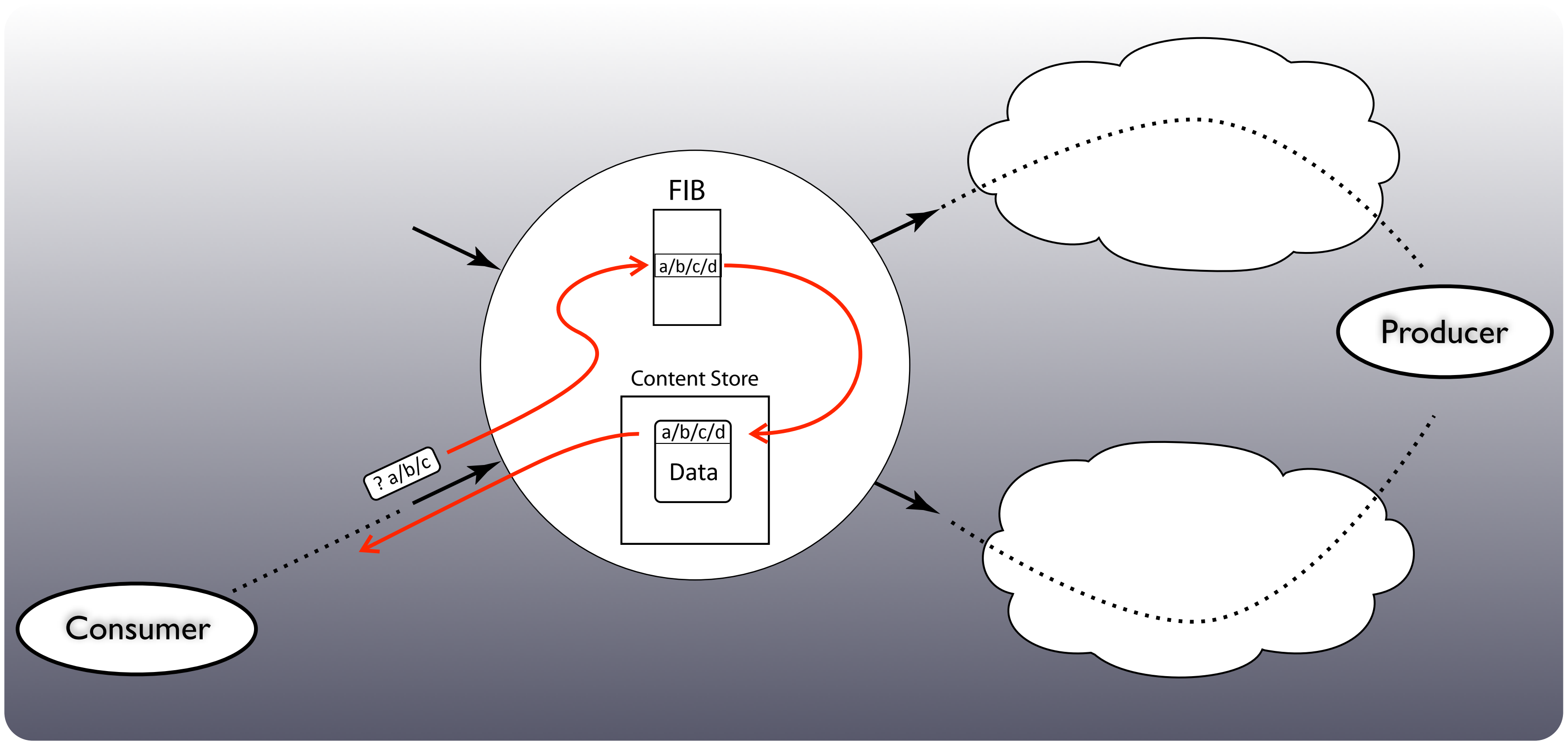
(where “communications technology” is anything that moves bits in time or space)

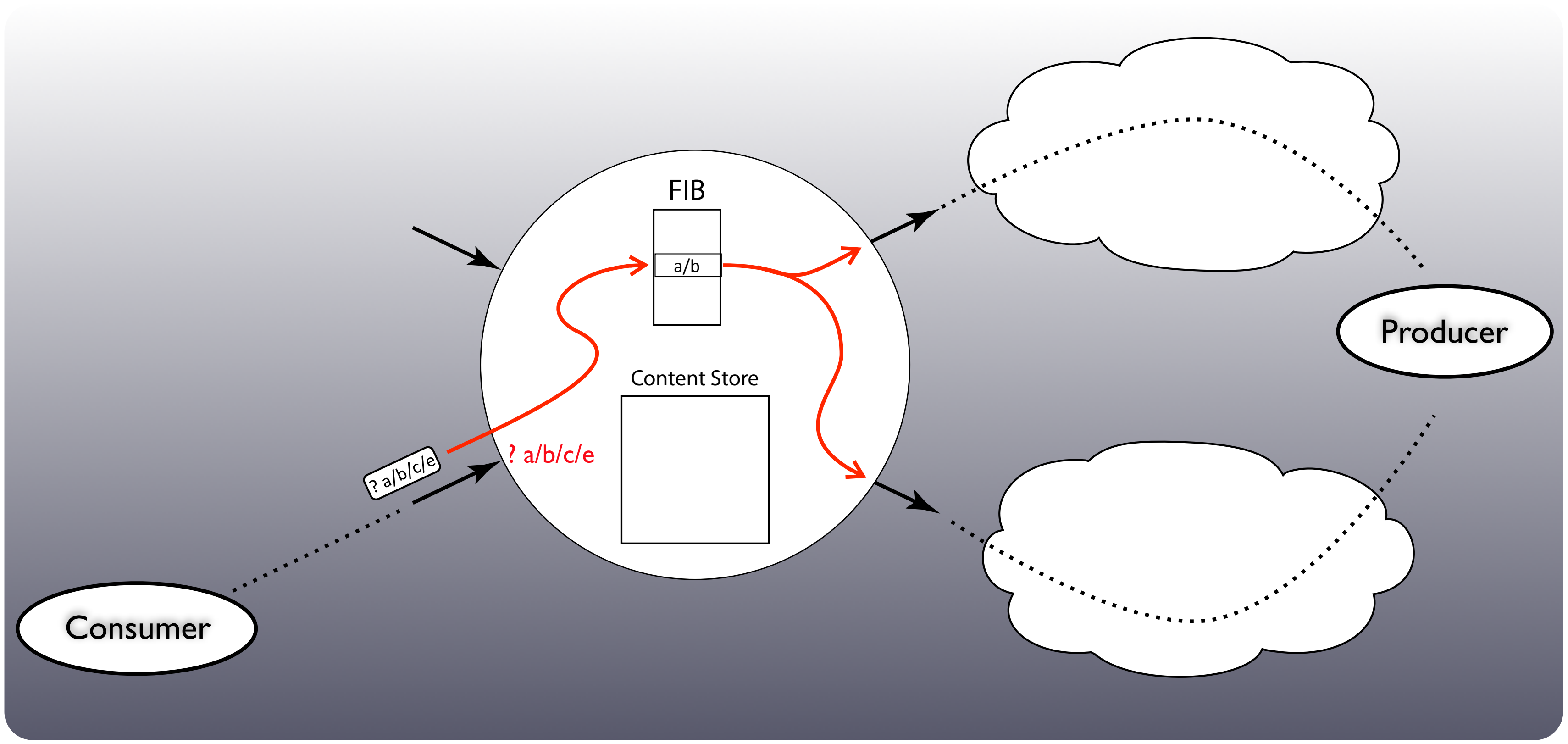

? /ndn/van/NDNtalk
/ndn/van/NDNtalk/v23/p1:.....

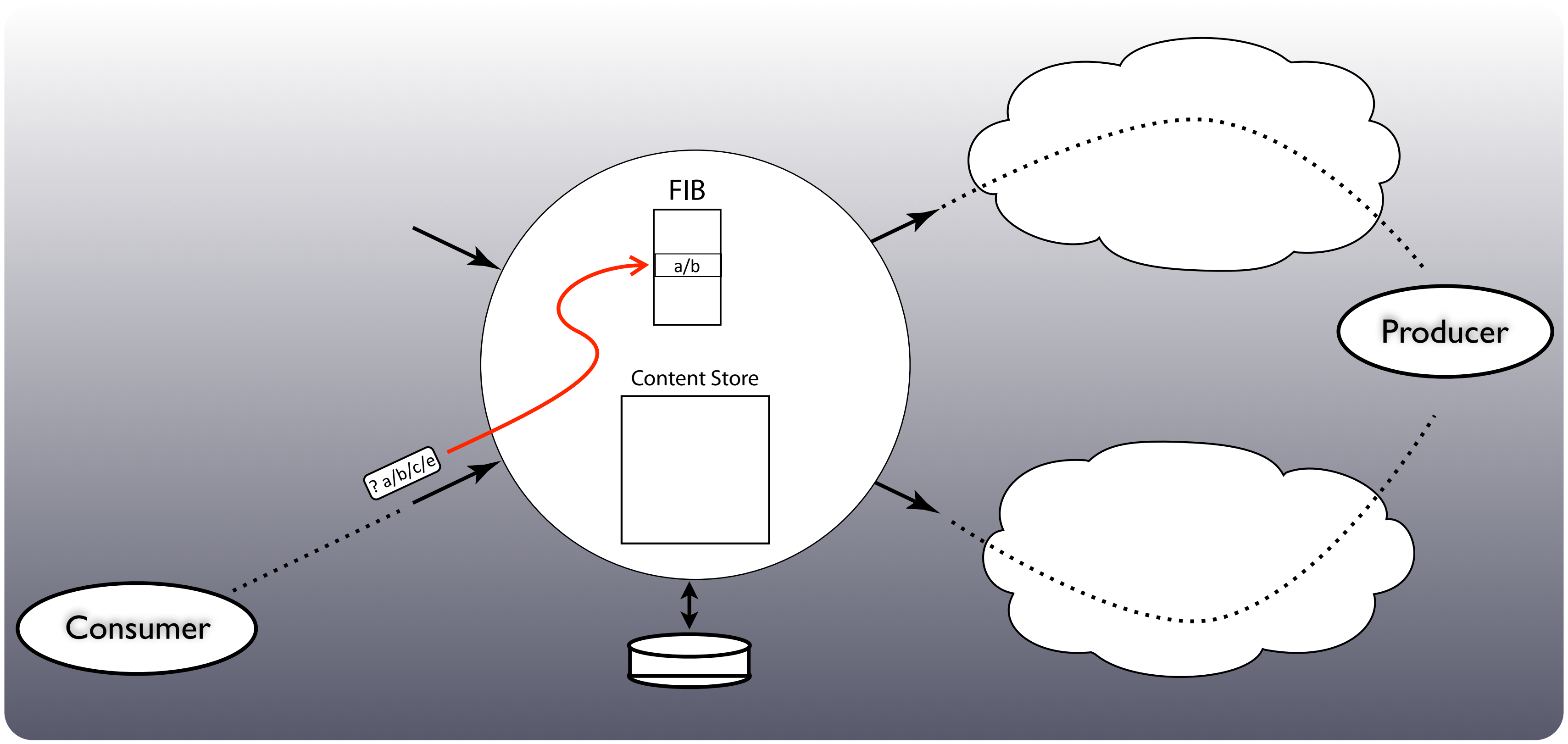

FIB
DST
SRC
src
dst



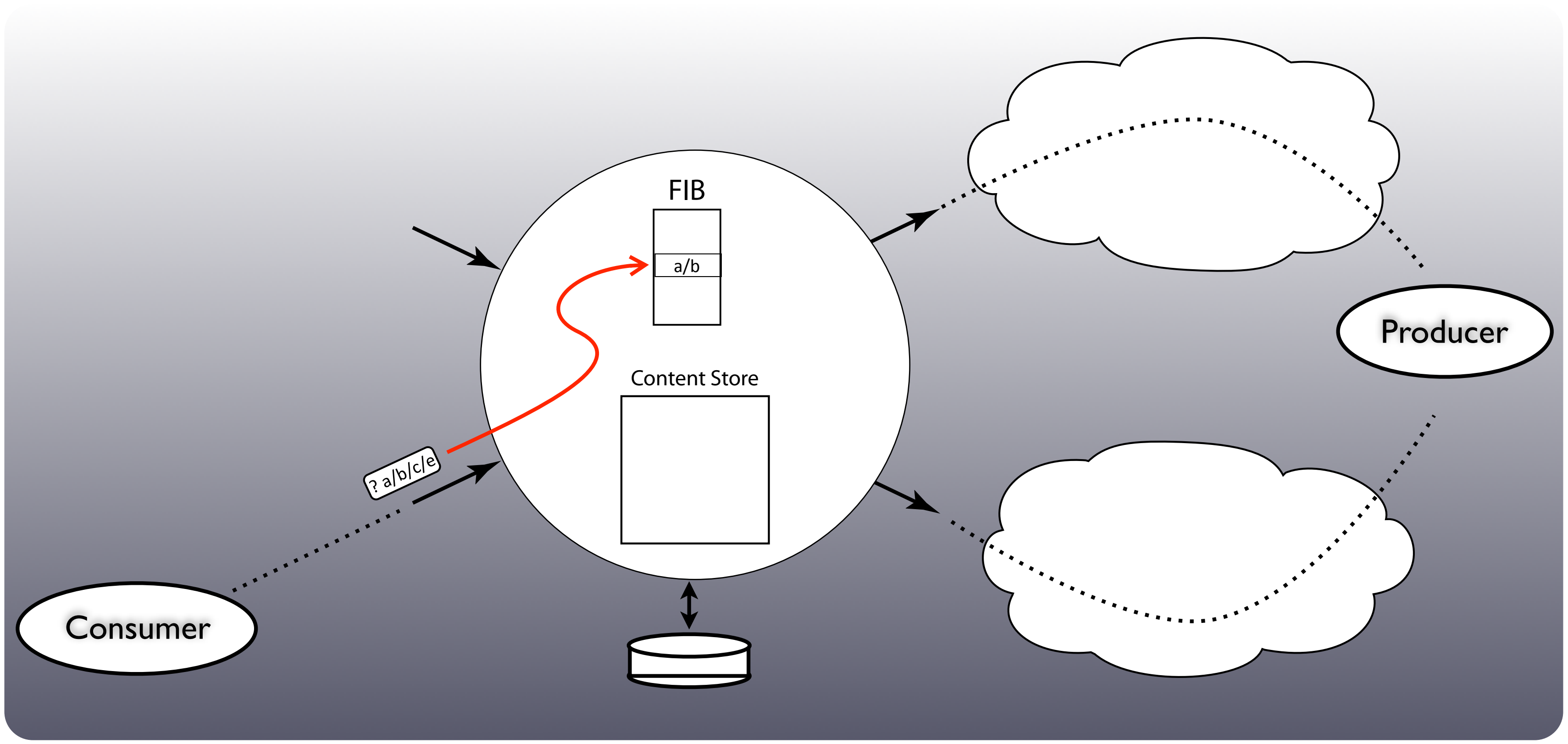
- Intermediate nodes are invisible
- Intermediate nodes can’t choose.
- Intermediate nodes can’t measure success


FIB
a/b/c/d
Content Store
a/b/c/d
Data
? a/b/c
Producer
Consumer


FIB
a/b
Content Store
? a/b/c/e
? a/b/c/e
Consumer
Producer



- Packets say ‘what’ not ‘who’ (no src or dst)
- communication is to local peer(s)
- no lower layer dependencies - works over anything.



- upstream performance is measurable
- memory makes loops impossible
- wireless, wires, storage & cycles look the same

‘Topology’ is an optimization – data goes only where ‘interest’ has been expressed so a time and bandwidth optimal local relay network is automatically constructed for each set of content.

- All the radios in an area behave as a mesh and cooperate to extend range and route around dead spots.

- When multiple technologies are in use (e.g., VHF for normal operations, UWB in-building), adding a bridge will combine them into a seamless, fully connected net.

- Same story for multiple responders (e.g, police and fire) using different technologies or frequencies.

The ability to automatically construct seamless networks from different communications technologies creates a system that evolves gracefully:

- Technology transitions don’t require a ‘flag day’ where everything has to be changed at once.
- Instead, the new technology can be incrementally deployed together with a few ‘bridge boxes’ that combine old and new.

- Content can have multiple names (via links) and names can be context dependent.
- For example, GPS or location limited communications like UWB allow names like “/ThisBuilding” or “/ThisIncident”.
- This would allow, say, /ThisIncident/Police to talk directly with /ThisIncident/Firemen without going through a dispatcher.

- All content is cryptographically signed and automatically authenticated by any receiver.
- The signing framework is multilevel and extensible in a way that allows most communications control policies to be implemented and enforced.
- Content can be automatically encrypted for privacy with automatic key distribution.

Schematized Trust

- Today we (attempt to) secure the <u>process</u> of communication by adding cryptographic wrappers to the packet transport.

- This hasn’t worked well. One serious failing is an intrinsic one-size-fits-all model of trust based on endpoint identity.

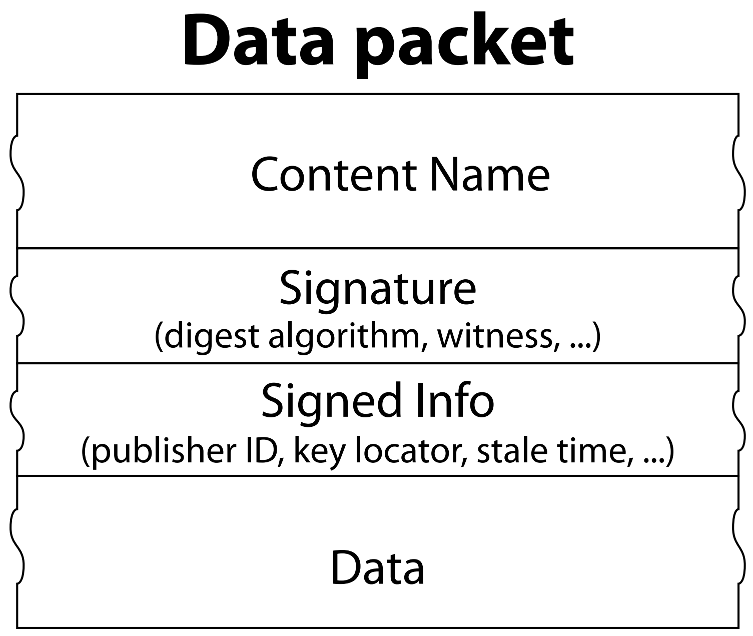
- Information-centric architectures secure content, not just the process of communicating it. They have the potential to support richer and more granular trust.

- To be successful, content-based trust machinery must be easy to understand, configure and use.
- Simple ‘trust schemas’ (patterns / templates / …) that can applied to whole classes of applications would help achieve this.
- As a proof-of-principle, NDN has developed a schematized trust framework and successfully applied it IGP routing and IoT (building control and instrumented environments).

NDN packets

NDN Data packets are structured objects with three parts:

- (opaque) data bytes
- A name for the data
- A signature over the name and data together with the name of the signing key (another NDN packet).


Data packet
Content Name
Signature
(digest algorithm, witness, ...)
Signed Info
(publisher ID, key locator, stale time, ...)
Data

US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/678

Name of a chat packet published into the SFPD Precinct 123 “chat channel” by Sgt. George Adams

$$\begin{tikzcd}
\text{US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/678} \arrow[d, bend right, "\text{signed by}"'] \\
\text{US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/key}
\end{tikzcd}$$

Public key given to the PDA’s chat comm process when Sgt. Adams authenticated himself & unlocked it today. Every packet sent by the process is signed with this key.

signed by

US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/678

US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/key

Public key given to the PDA’s chat comm process when Sgt. Adams authenticated himself & unlocked it today. Every packet sent by the process is signed with this key.

US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/678

US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/key

US/CA/SF/PD/pr123/sgt/george.adams/key

Public key given to the pda when it was configured.

US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/678

US/CA/SF/PD/pr123/**chat/s**gt/george.adams/pkt/key

US/CA/SF/PD/pr123/sgt/george.adams/key

Public key given to the pda when it was configured.

US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/678

US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/key

US/CA/SF/PD/pr123/**sgt/george.adams**/key

US/CA/SF/PD/pr123/**config/empl/975**/key

Public key given to the employee who configured the router.

US/CA/SF/PD/pr123/config/key

Public key authorizing Precinct 123 configuration. (pr123 trust root)

Trust Schema describes how to construct or check the key hierarchy that authorizes a service instance to publish data in some namespace.

(see Schematizing and Automating Trust in Named Data Networking in Proceedings of ACM ICN 2015)

k4 = my.config.root	US/CA/SF/PD/pr123/config/key
k3 = k4 +“empl”+ n	US/CA/SF/PD/pr123/config/empl/975/key
k2 = k3[-3] + rank + name	US/CA/SF/PD/pr123/sgt/george.adams/key
k1 = k2[-2] +“chat”+ k2[2-1] +“pkt”	US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/key
pkt = k1 + n	US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/678

Usage

```
if (validTrustChain(pkt, schema) && signatureValid(pkt))
    process the packet
```

Since schema is just lexical constraints on key names, validation normally only has to check that key name is appropriate for data name.

Only have to validate chain & signature for a key once.

Why so many names?

- Context provided by naming detects and prevents misconfiguration and misbehavior.
- Names provide fine-grain trust that minimizes damage from key exposure.
- Naming strictly limits scope of keys and prevents repurposing.

Why so many levels?

US/CA/F&S/config/key

US/CA/F&S/config/empl/451/key

US/CA/SF/PD/config/key

US/CA/SF/PD/config/empl/51/key

US/CA/SF/PD/pr123/config/key

US/CA/SF/PD/pr123/config/empl/975/key

US/CA/SF/PD/pr123/sgt/george.adams/key

US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/key

US/CA/SF/PD/pr123/chat/sgt/george.adams/pkt/678

Why so many levels?

US/CA/F&S/config/key	US/CA/F&S/config/key
US/CA/F&S/config/empl/3251/key	US/CA/F&S/config/empl/451/key
US/CA/SF/FD/config/key	US/CA/SF/PD/config/key
…	…

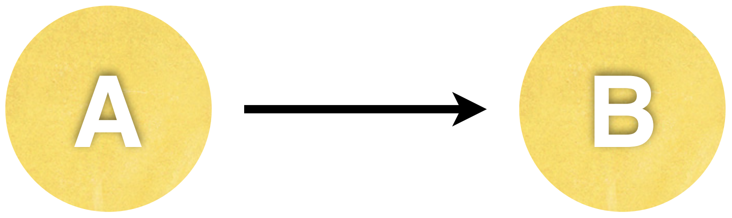
- Same signing keys (with different trust schema) used for cross org authentication / authorization.

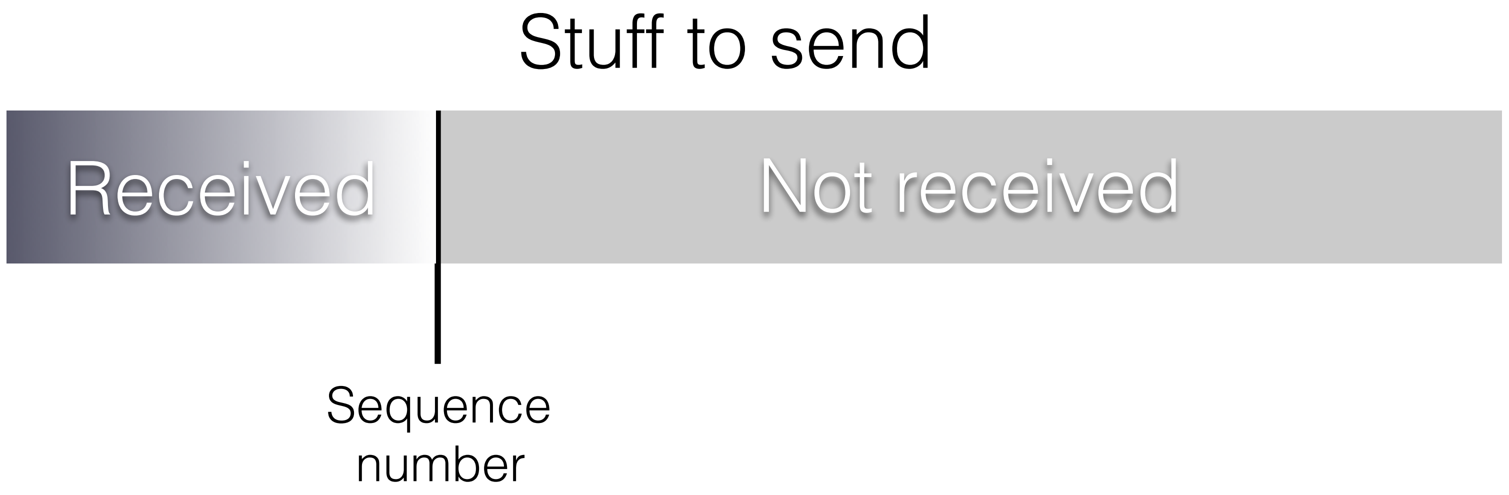
Model Properties

- Complete local autonomy - all keys are locally generated and signed.
- No key distribution problem. Apps get their entire trust chain from pda’s config then announce keys to their peers.
- Once a trust schema has been picked, everything else is simple and automatic.

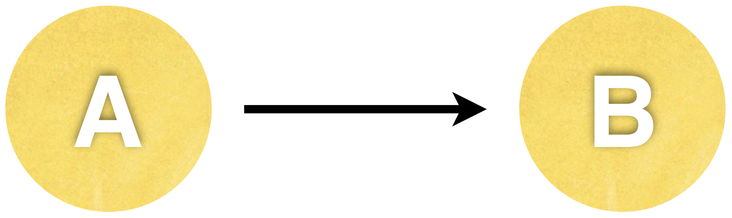
Transport via set-reconciliation (sync)

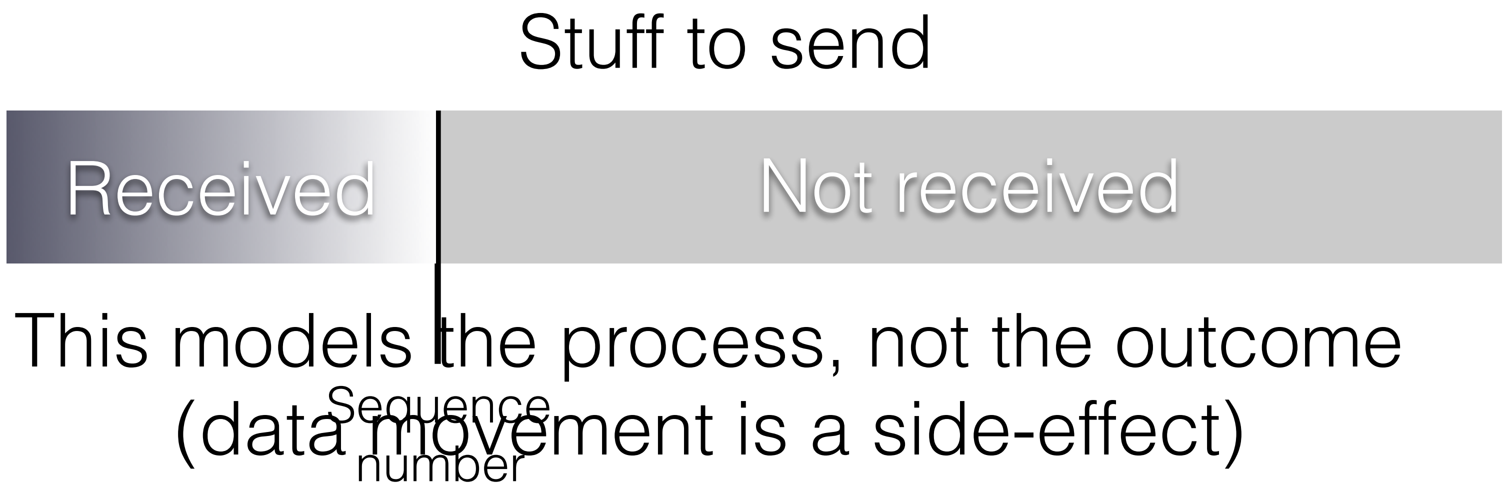
Transport thru the ages





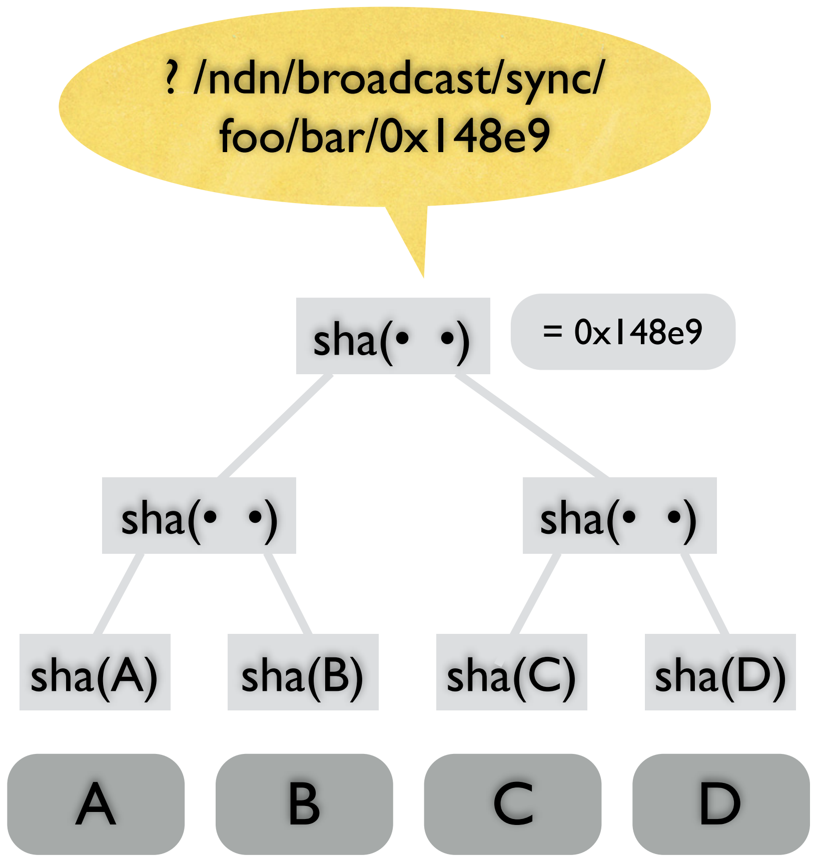
Transport thru the ages





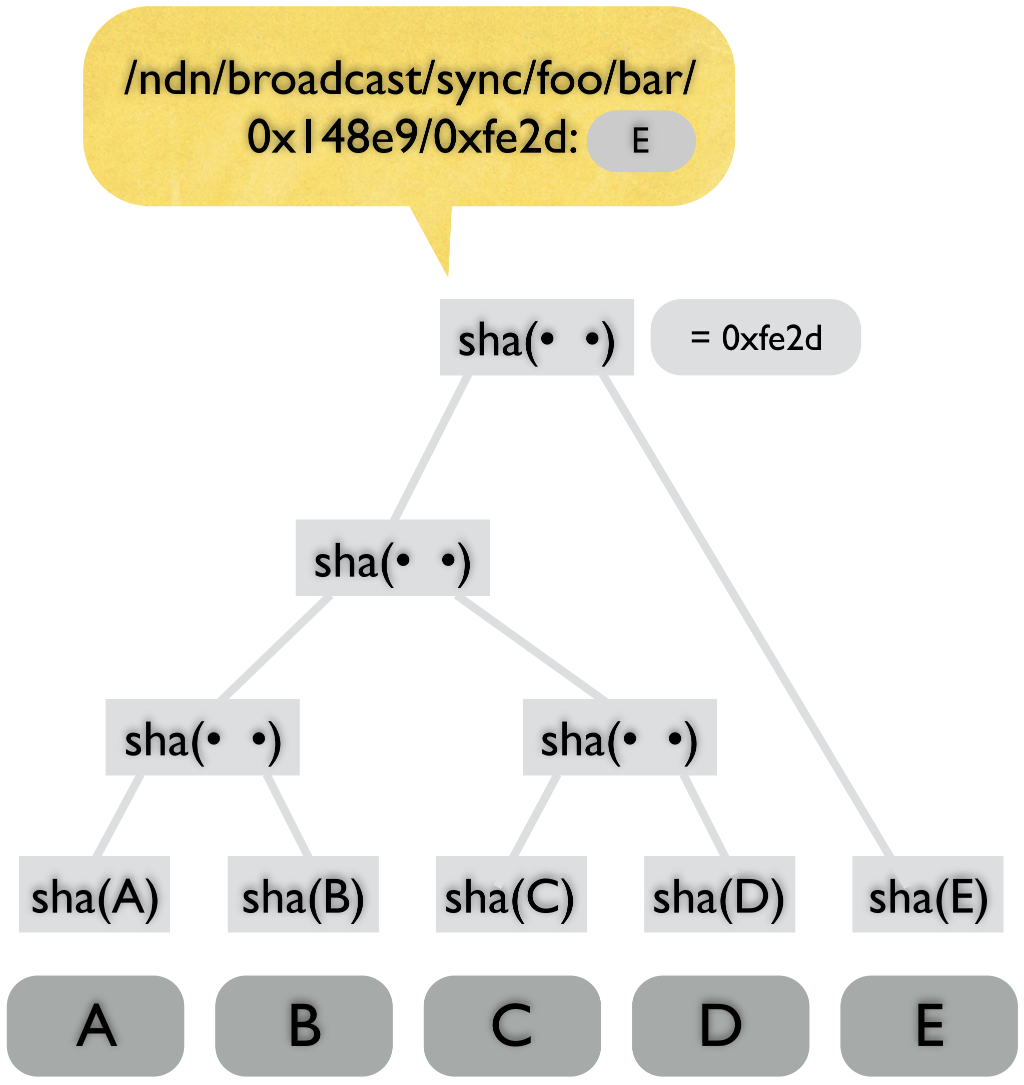
This models the process, not the outcome
(data movement is a side-effect)

A better way

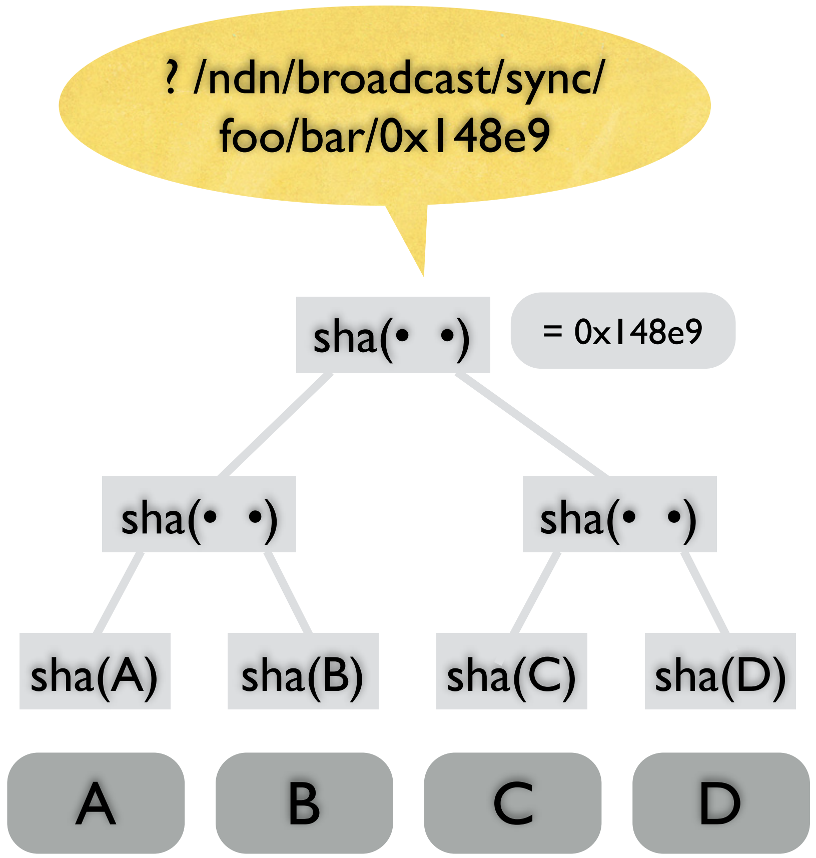


Bob's /foo/bar collection

A better way

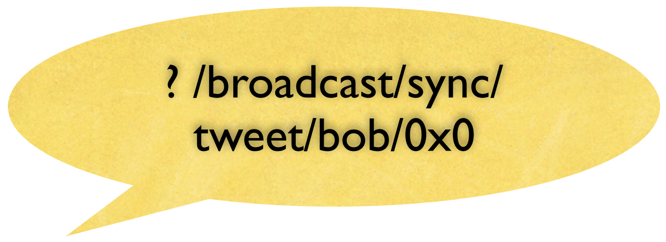


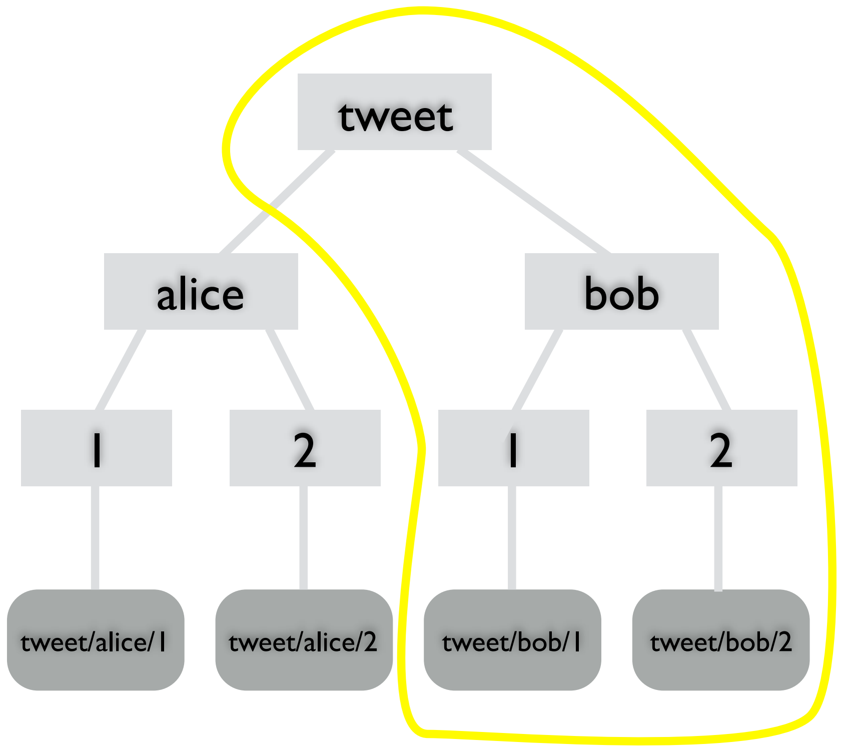
Alice’s /foo/bar collection



Bob’s /foo/bar collection

A better way





Bob's tweet collection