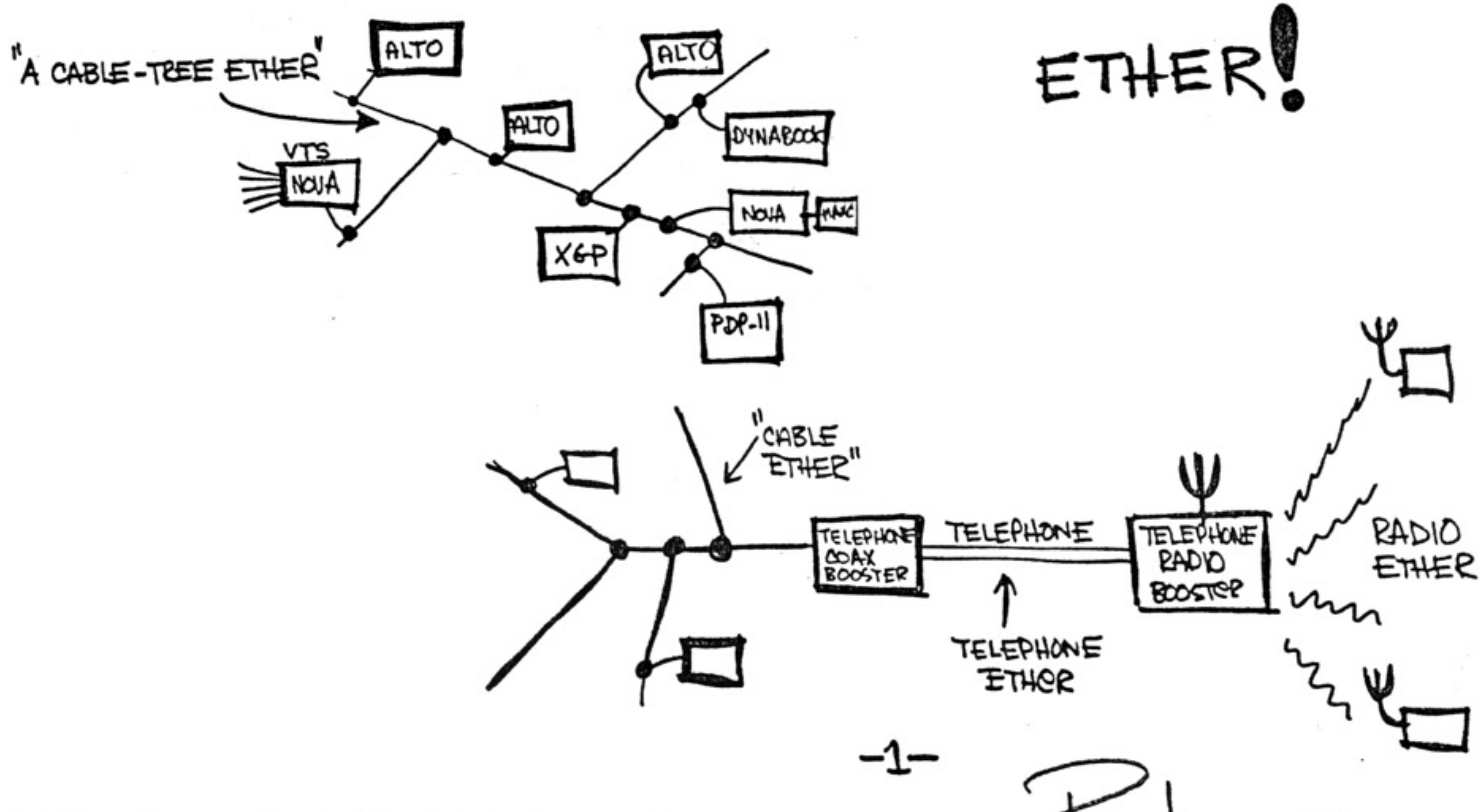
CCNx Conference 2013

September 5th & 6th
PARC, Palo Alto, California

Happy
40th
Birthday

Welcome to PARC



Bob Metcalfe's 1972 sketch of his original "ethernet" vision
Image provided courtesy of Palo Alto Research Center Inc., a Xerox Company

Overview

CCN Update

CCN Tenets

CCN Demo

CCN Progress

CCN Direction

CCN Update

CCNx Conference 2013

	CCNx 2011	CCNx 2012	CCNx 2013
Attendees	130	135	160
Accepted Talks	12	29	28
Posters/demos	31	19	16

CCN Ecosystem

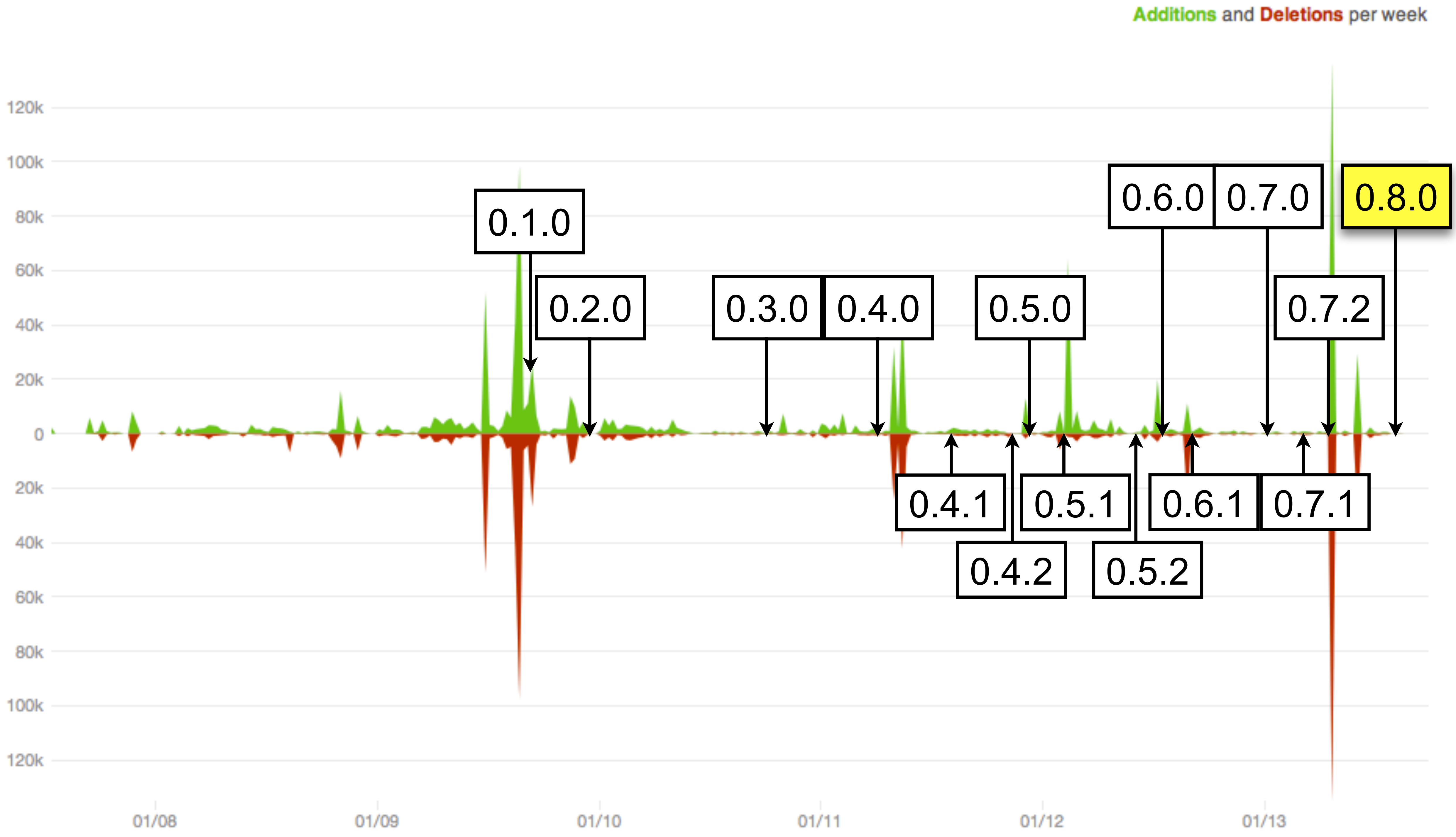
Three **activities sponsored by PARC**:

- CCN open source reference implementation
- CCN developer community - CCNx.org, CCNx Conference
- Emerging Networks Consortium - www.parc.com/enc

A **large worldwide research & development community** including academic and industrial research laboratories, automotive, telecommunications, aerospace, media, manufacturing and semiconductor companies

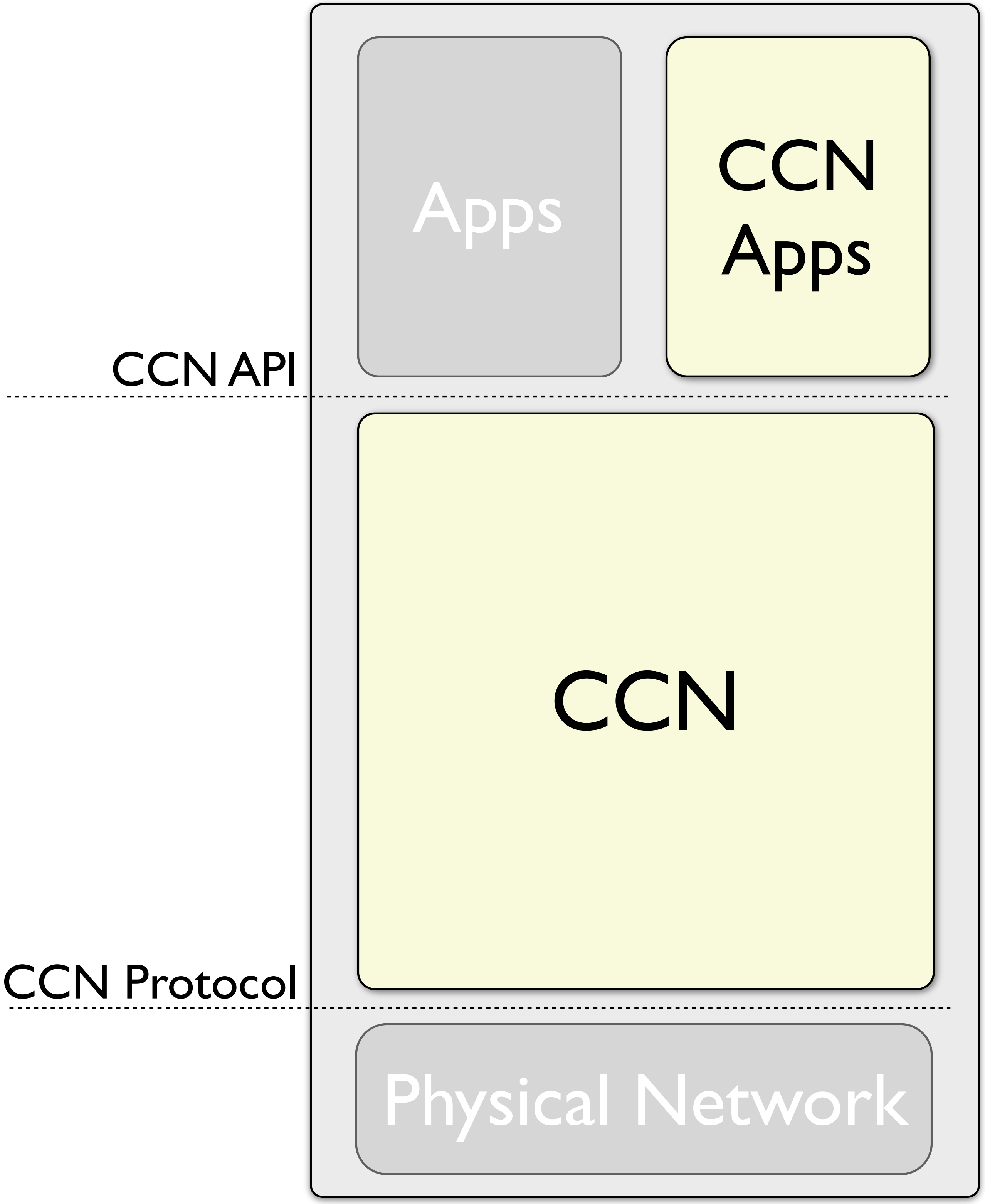
Projects underway at Alcatel, Cisco, Huawei, Tellabs, Ericsson, Intel, Nokia, Hitachi Data Systems, Fujitsu, Samsung, BT, Orange, FT, AT&T, IBM, Toyota, Xerox and others....

CCN Code Base



CCN Community Progress

Project Vitality	Adoption	Code Stats	Releases	Sub-projects
23 Releases 650+ Citations 134 Github Watchers 31 Forks 55 CCNx related projects 830+ Issues closed.	15+ major research universities engaging in research in the area of Content Centric Networking	100+ issues closed	0.7.0 0.7.1 0.7.2 0.8.0	CCN-Python CCN Core CCN Transport CCN Routing FLAN Forwarder
127 Tweets 130 Followers	Commercial projects based on CCNx are moving from the lab to PoC, reaching the market	145,500 lines of code removed, 138,951 lines added	0.8.1 soon	
160 CCNxCon2013 attendees	Internal work with embedded devices and hardware	~301 commits per release	0.9.0 and 1.0 are on the public roadmap: http://redmine.ccnx.org/projects/ccn/roadmap	



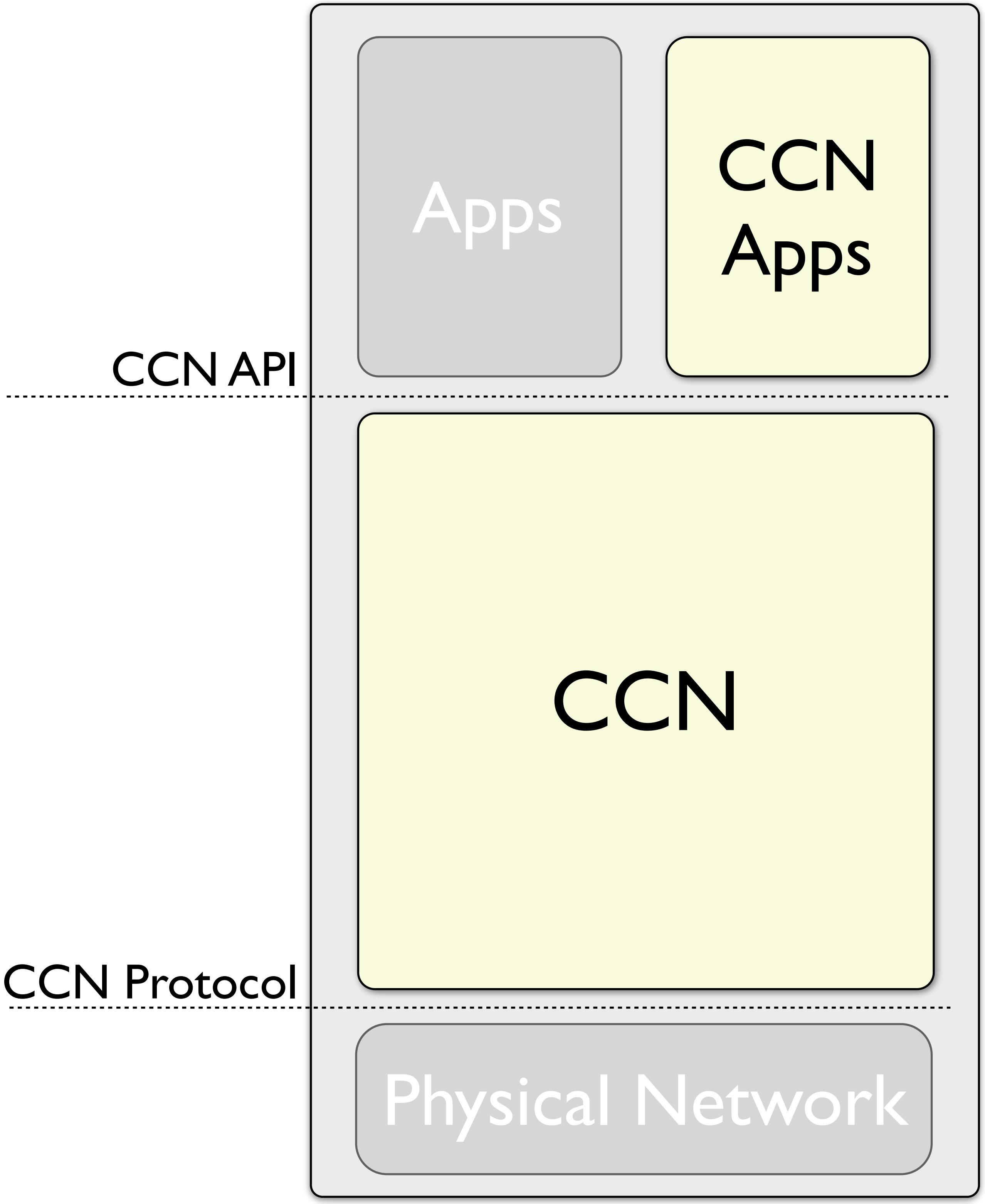
Research prototype

↓

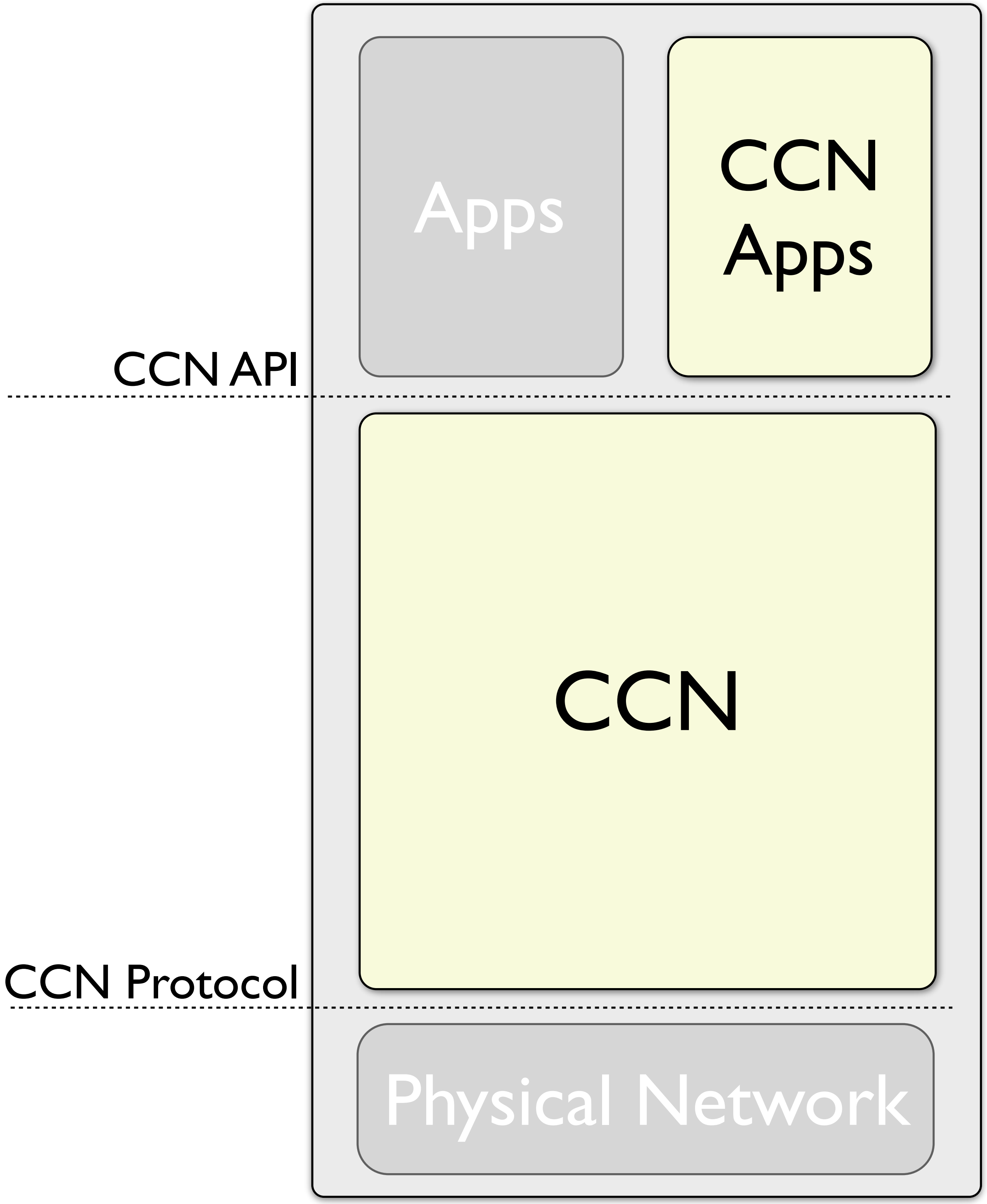
Production prototype

Roadmap 1

Improve Networking



Core protocol and encoding
Fast forwarding
Routing
Auto-configuration
Advanced flow control
Improved performance



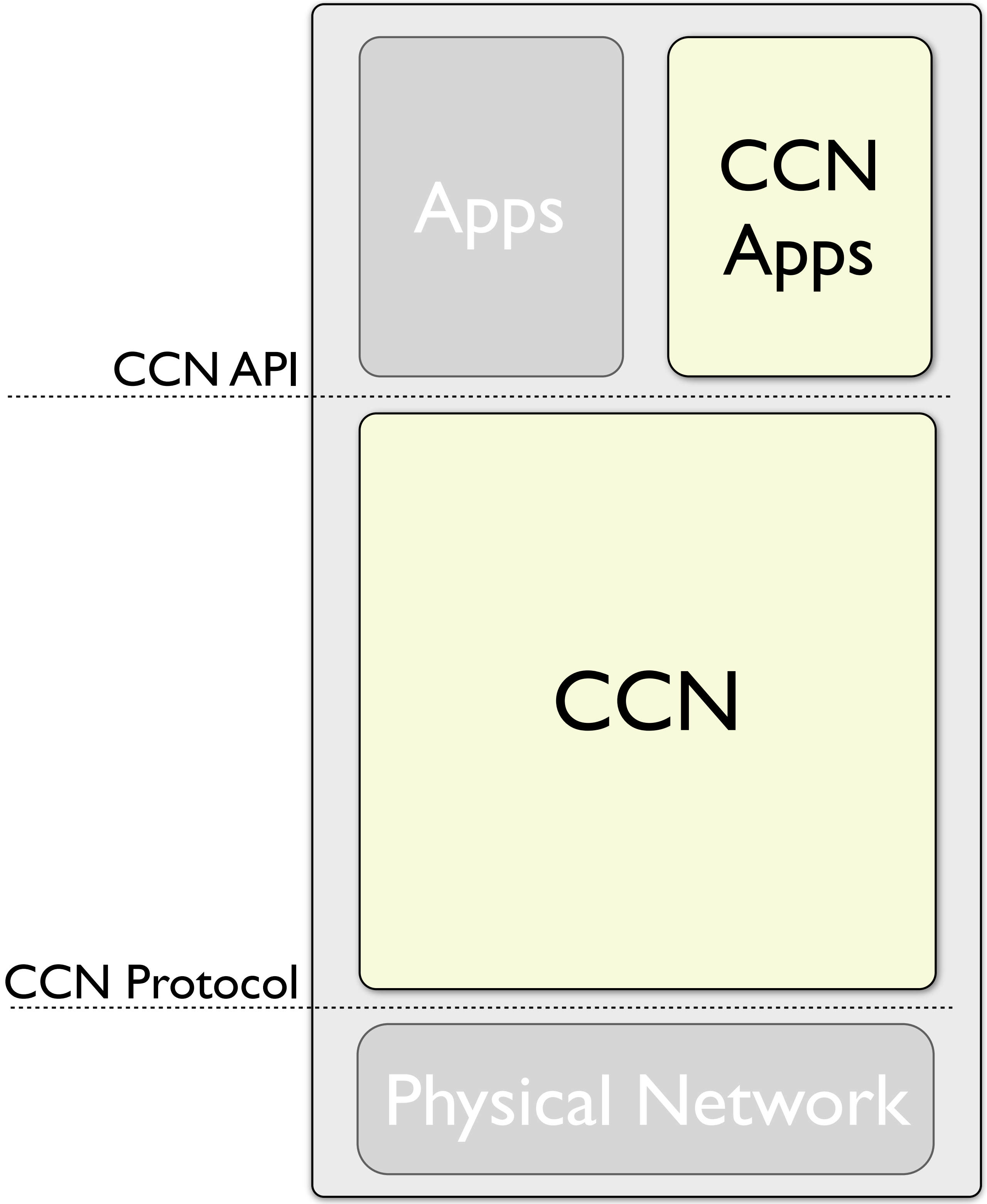
Roadmap II

Improve Services

Advanced Repo
Advanced Sync
Content organization
Trust Model
Efficient security
High level protocol suite

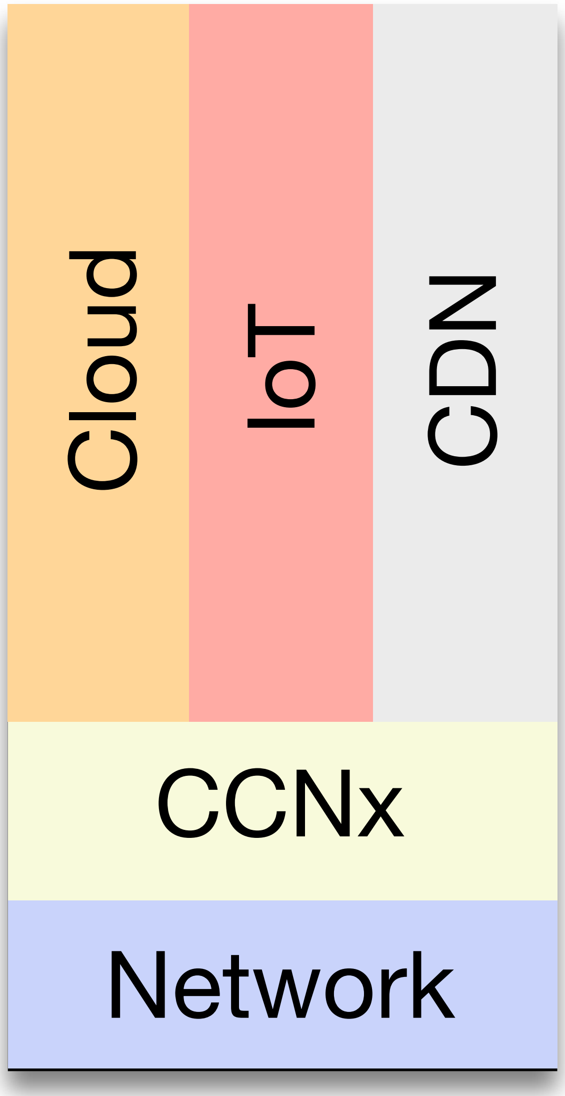
Roadmap III

Improve Adoption



Clean library and API
Language bindings
Testing framework
Development tools
Documentation
Examples
Tutorials

CCN Applications Focus



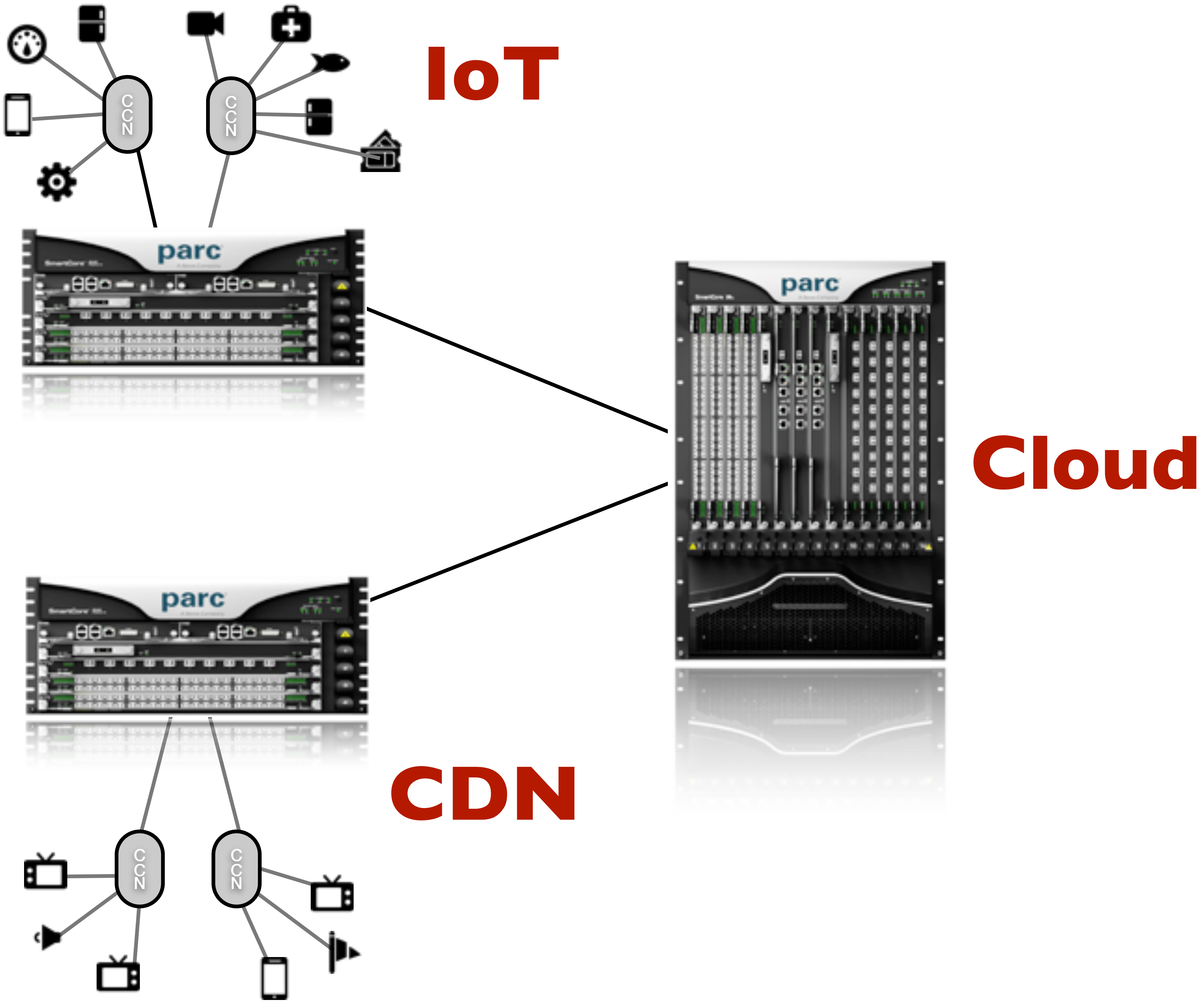
Cloud

IoT

CDN

CCNx

Network



CCN Ecosystem

Exciting collaborations:

Visiting scientists from BT, KDDI & INRIA

Customer and partner engagements - ENC, Samsung, Cisco & very interesting startups

ICNRG, conferences & workshops

Exciting research projects:

NDN, alternative approaches, industrial projects, PoCs

An invitation:

PARC welcomes visiting scientists & many forms of collaboration

Multiple solutions, intellectual property, papers & discussion

Many hard problems let to investigate & solve!

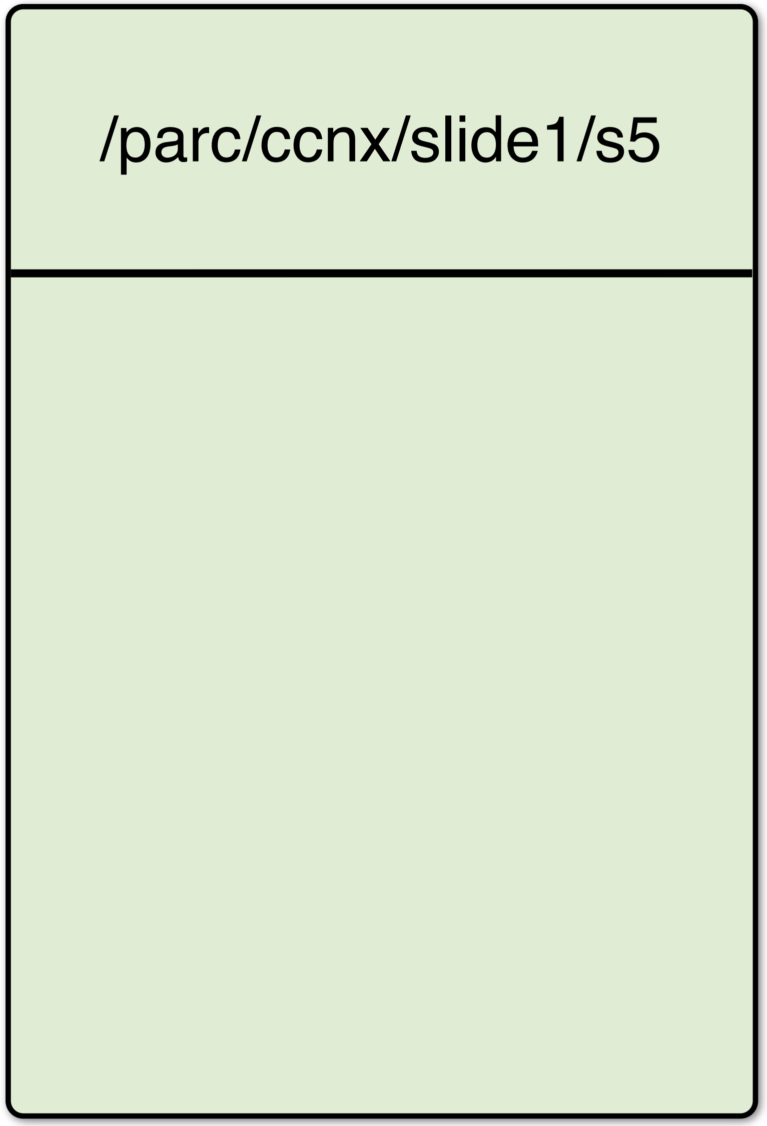
CCN Tenets

CCN - What is it?

Content Centric Networking (CCN) is a communications architecture based on dissemination rather than conversation

Communicate via Named Data

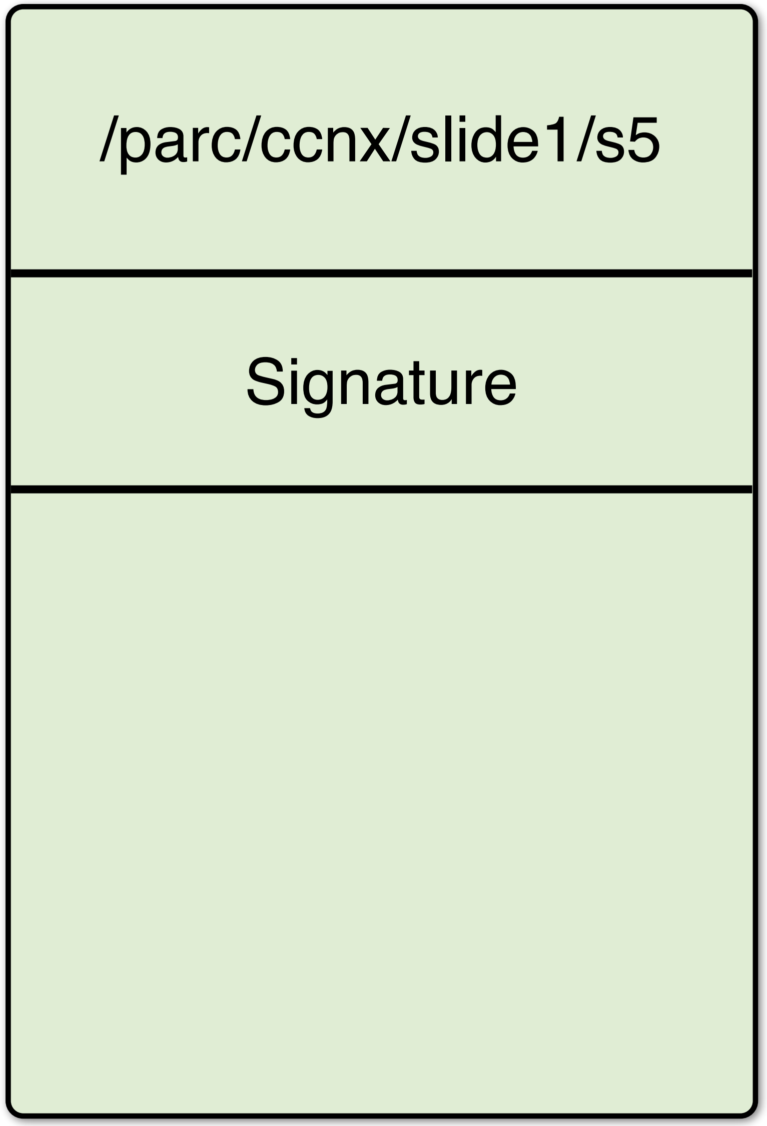
Content Object



/parc/ccnx/slide1/s5

Secure
Named Data

Content Object



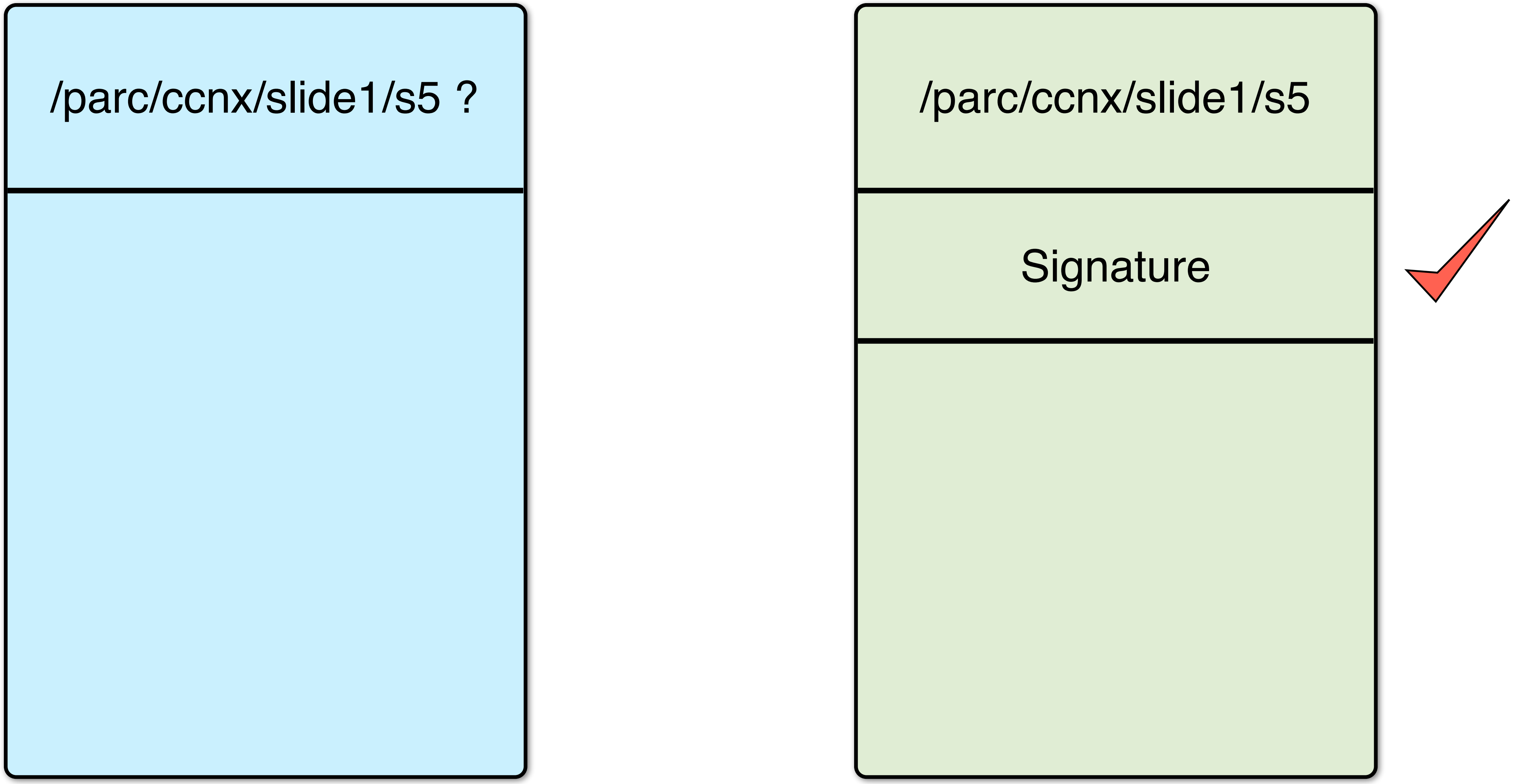
/parc/ccnx/slide1/s5

Signature

Request by name

Interest

Content Object

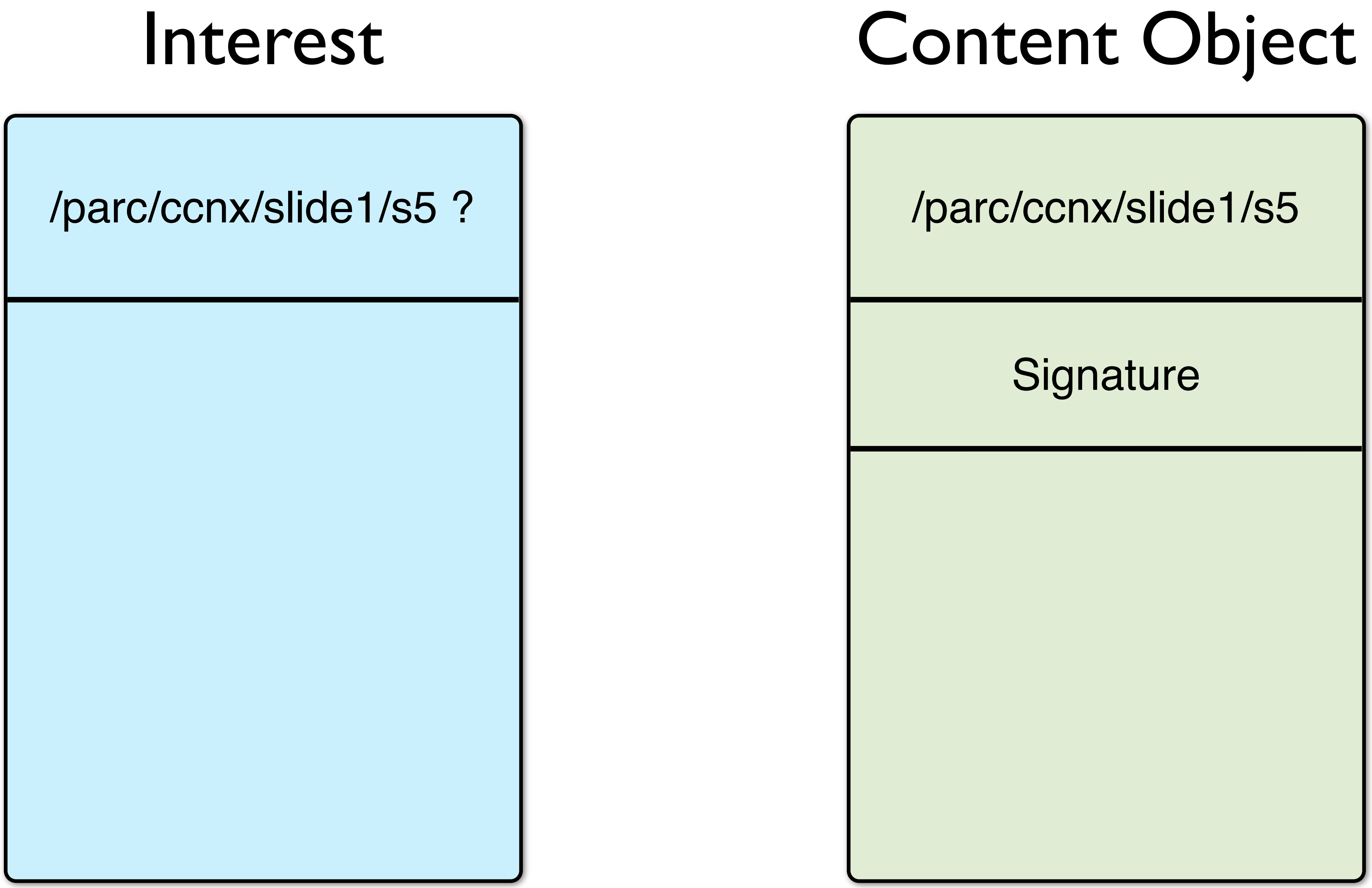


/parc/ccnx/slide1/s5 ?

/parc/ccnx/slide1/s5

Signature

✓



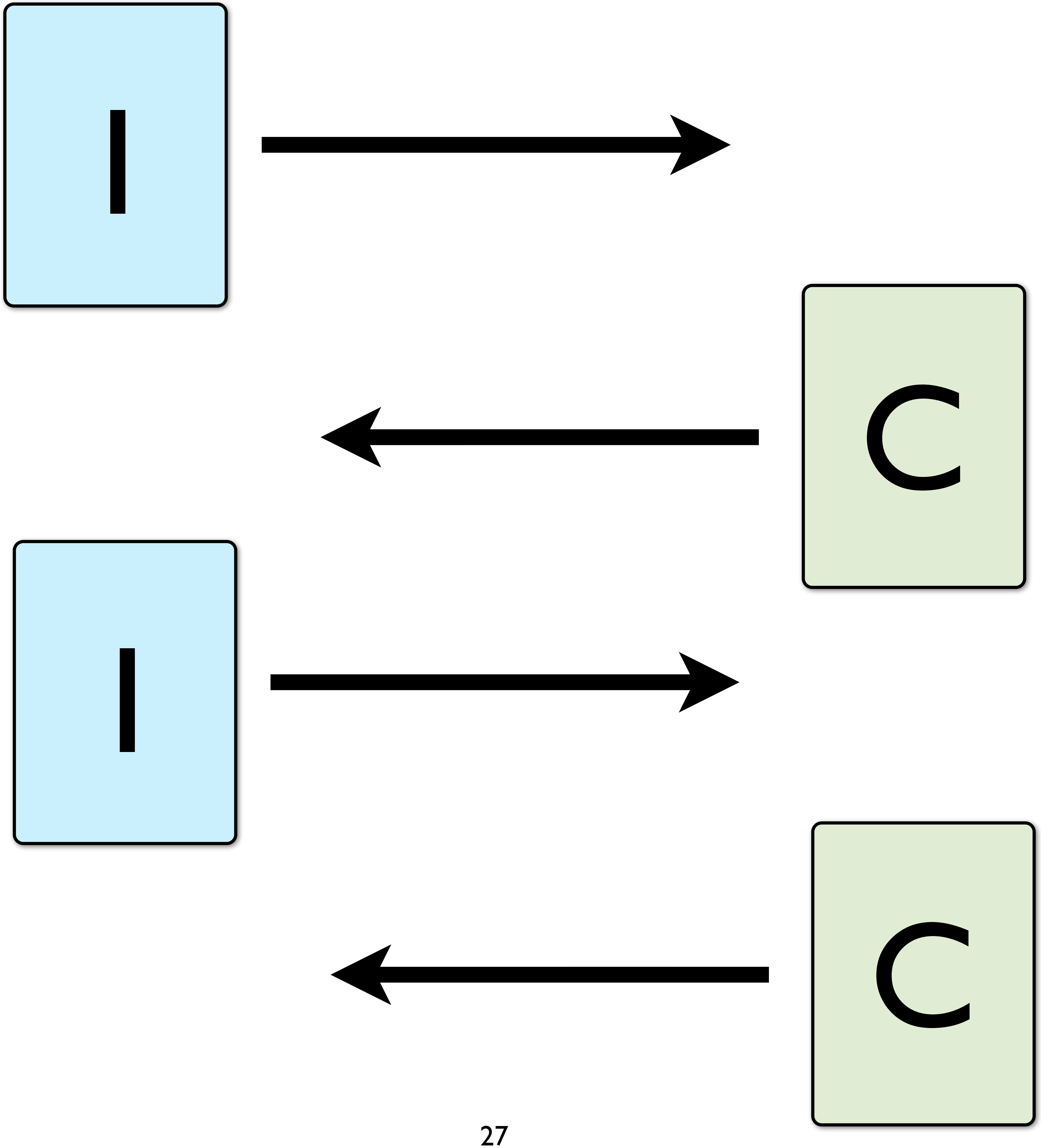
Interest

/parc/ccnx/slide1/s5 ?

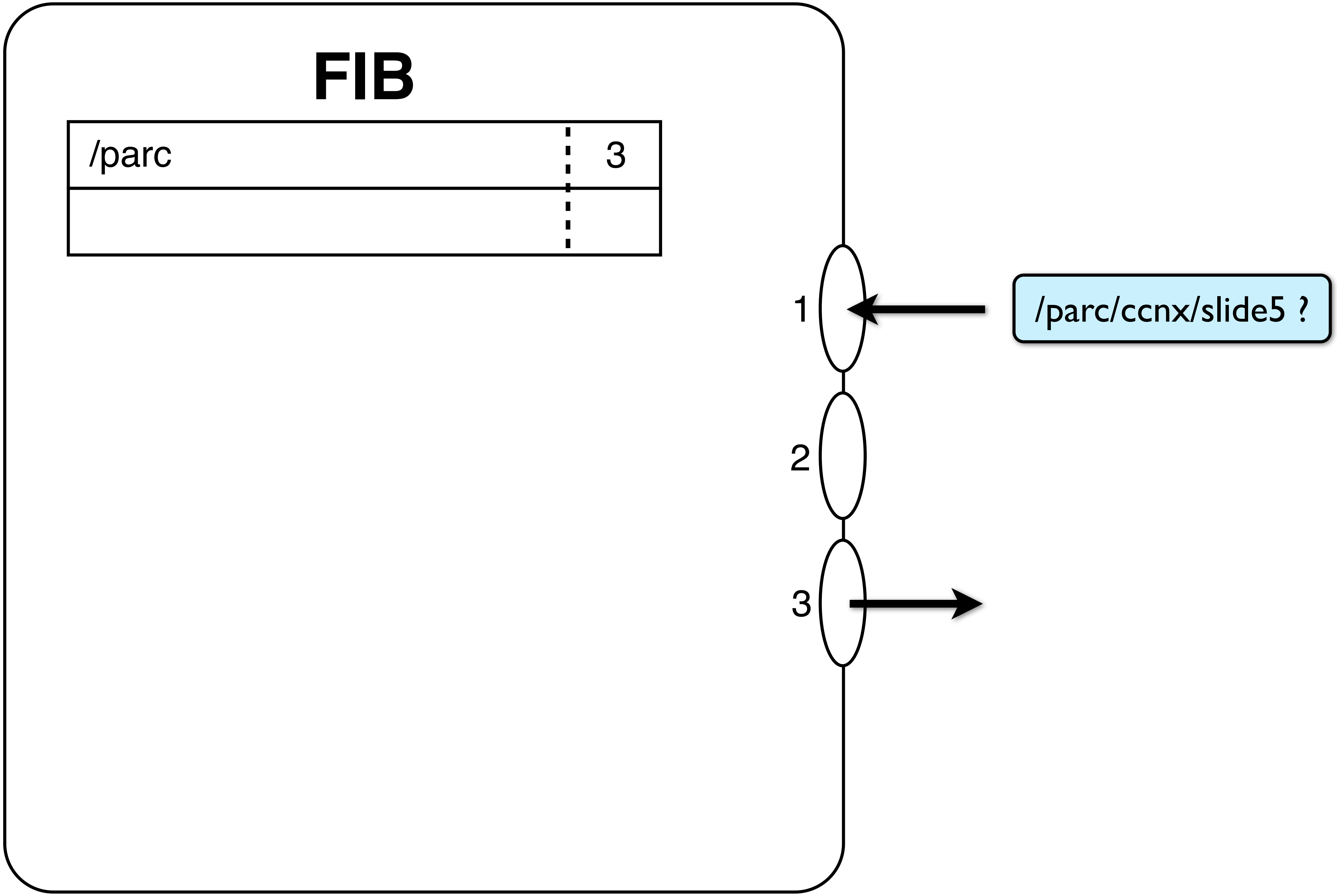
Content Object

/parc/ccnx/slide1/s5

Signature


I
C
I
C

Route by name



FIB

/parc	3

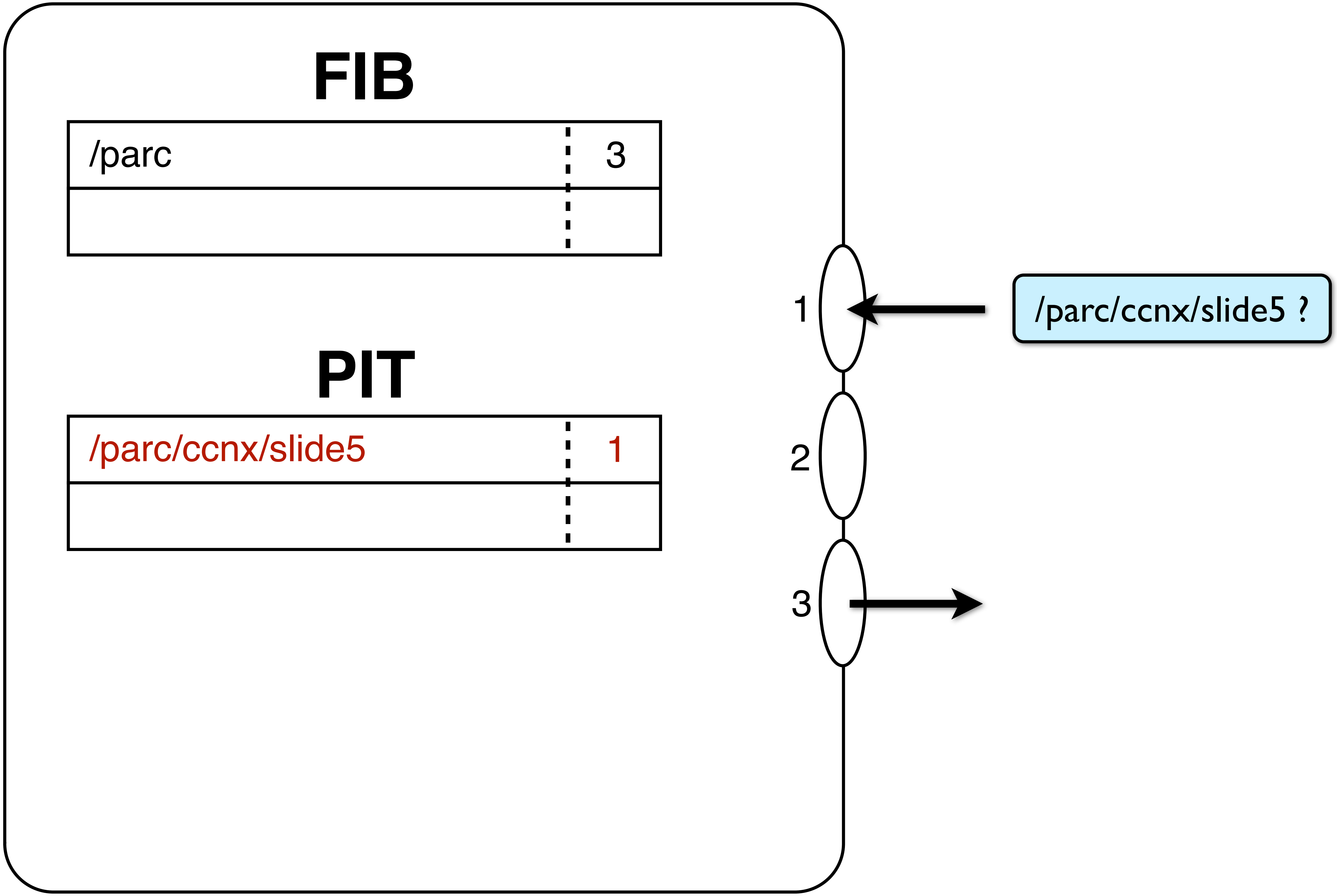
1

2

3

/parc/ccnx/slide5 ?

Keep state



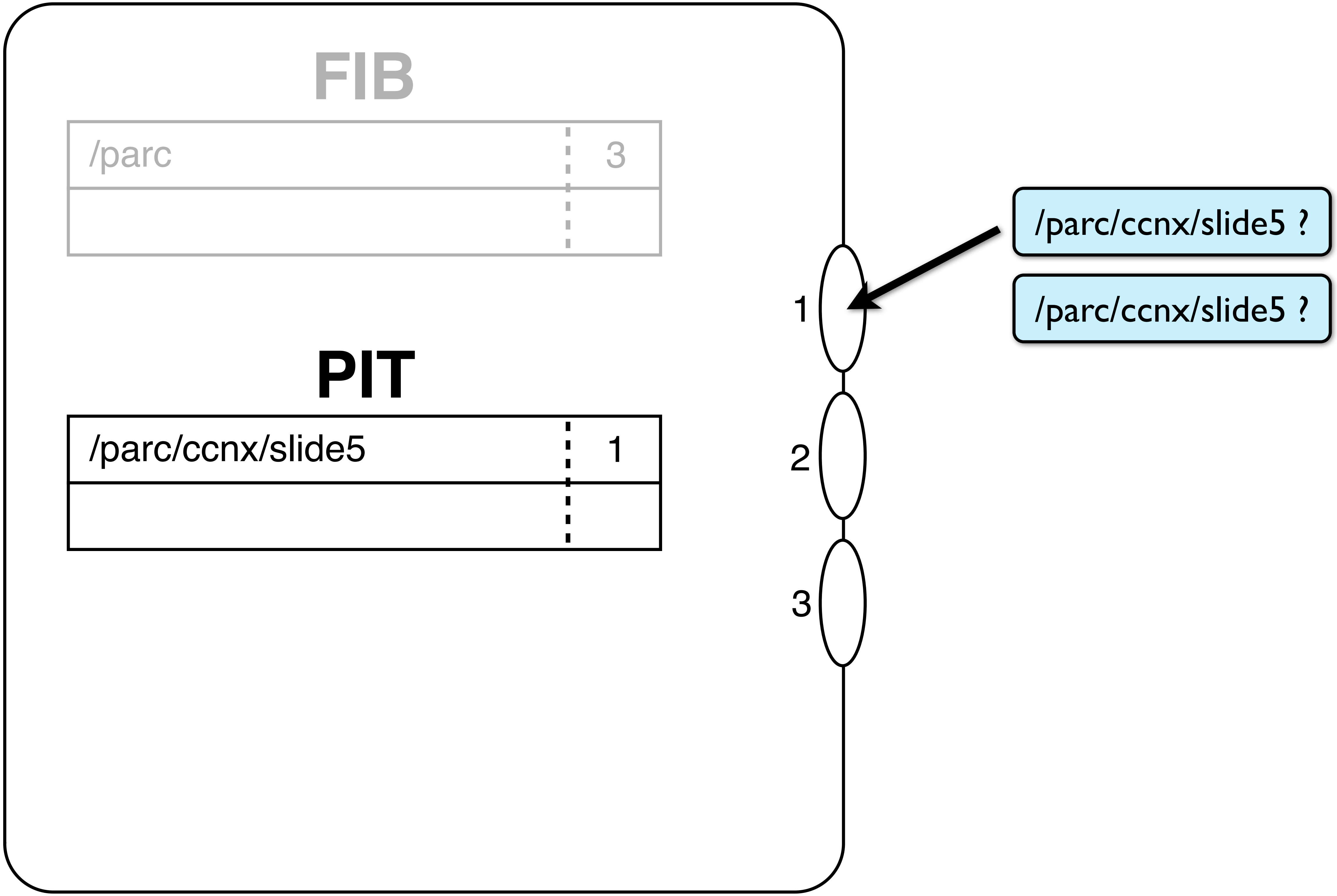
FIB

/parc	3

PIT

/parc/ccnx/slide5	1

/parc/ccnx/slide5 ?

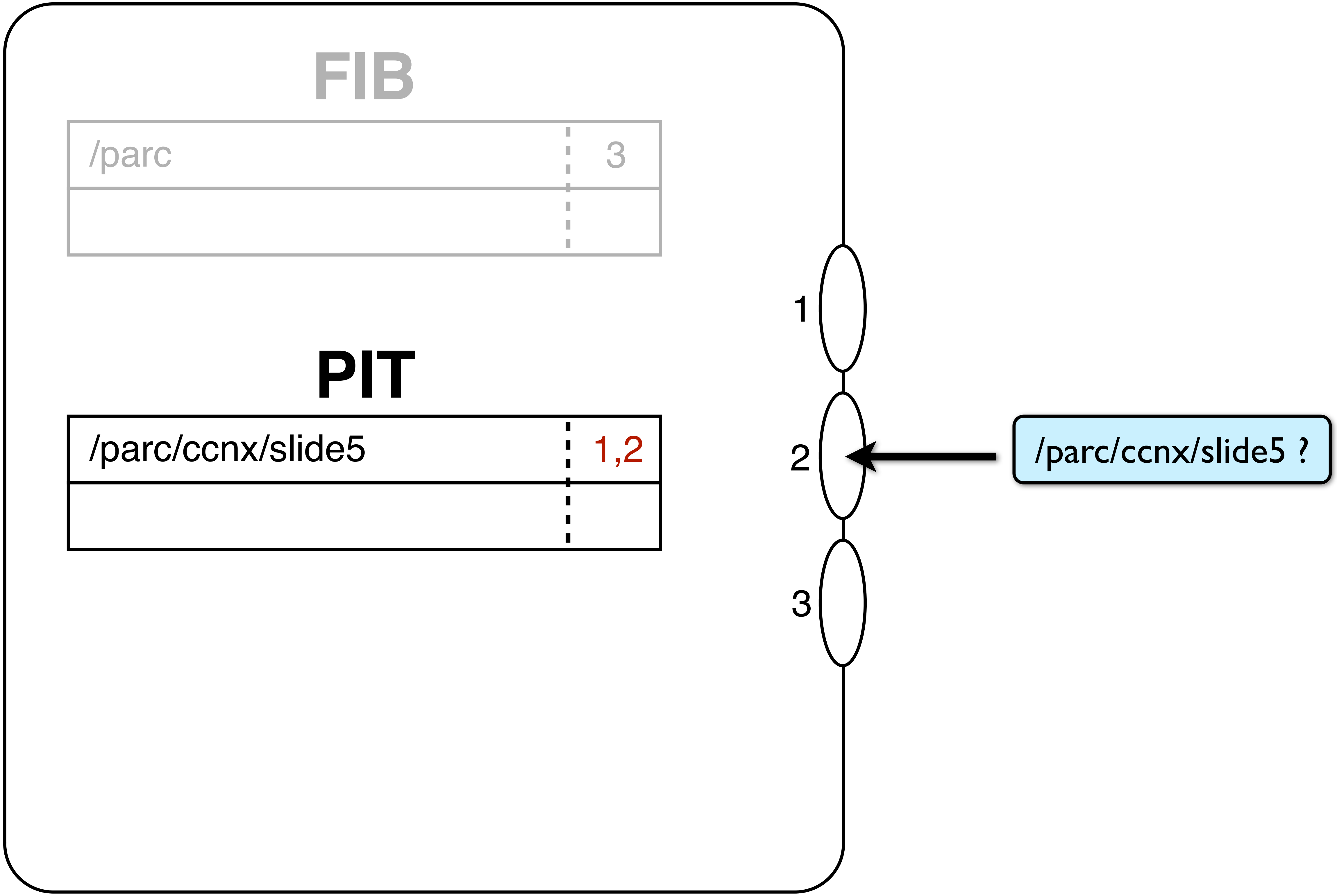


FIB

/parc	3

PIT

/parc/ccnx/slide5	1



FIB

/parc	3

PIT

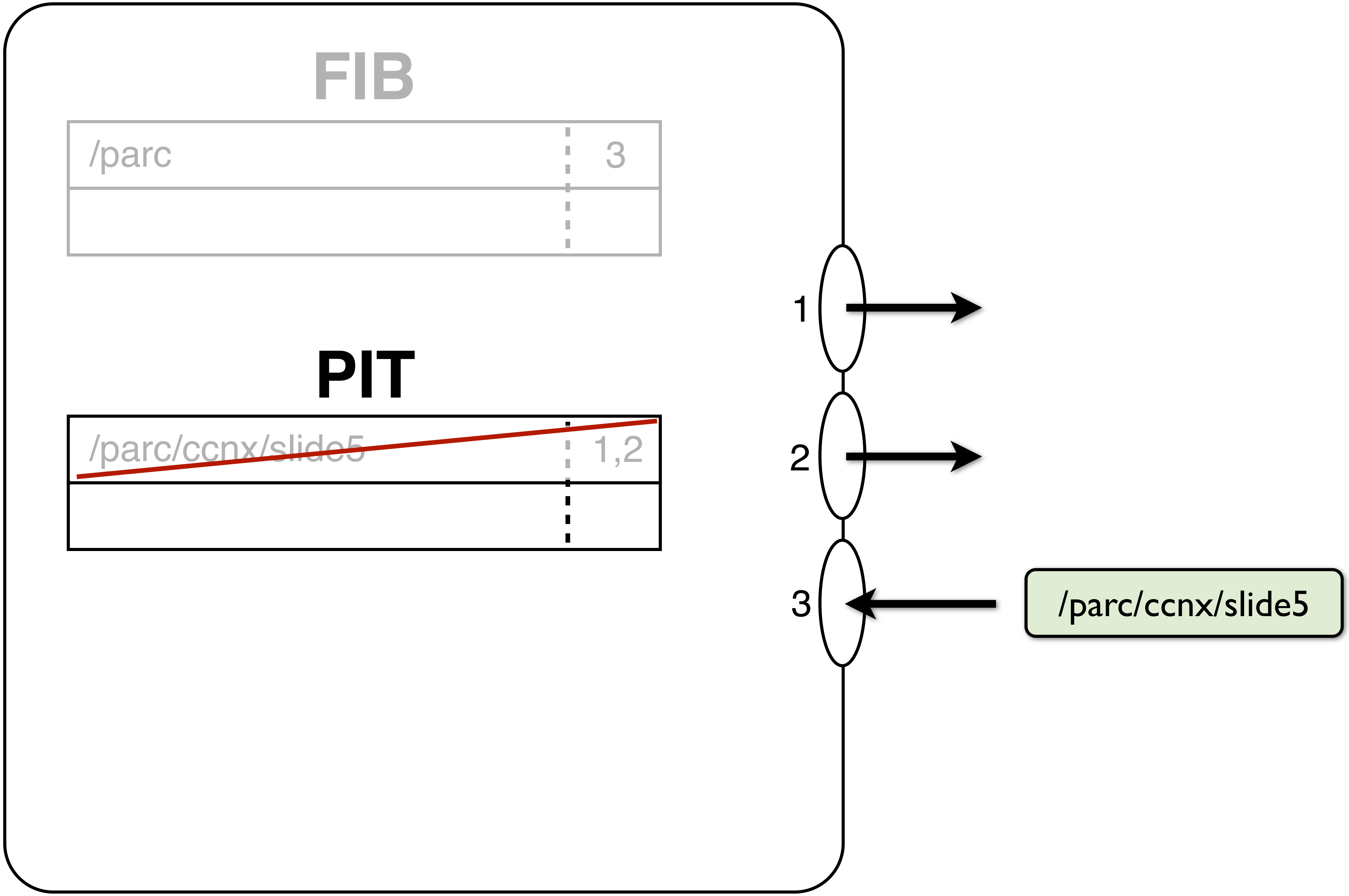
/parc/ccnx/slide5	1,2

1

2

3

/parc/ccnx/slide5 ?



FIB

/parc	3

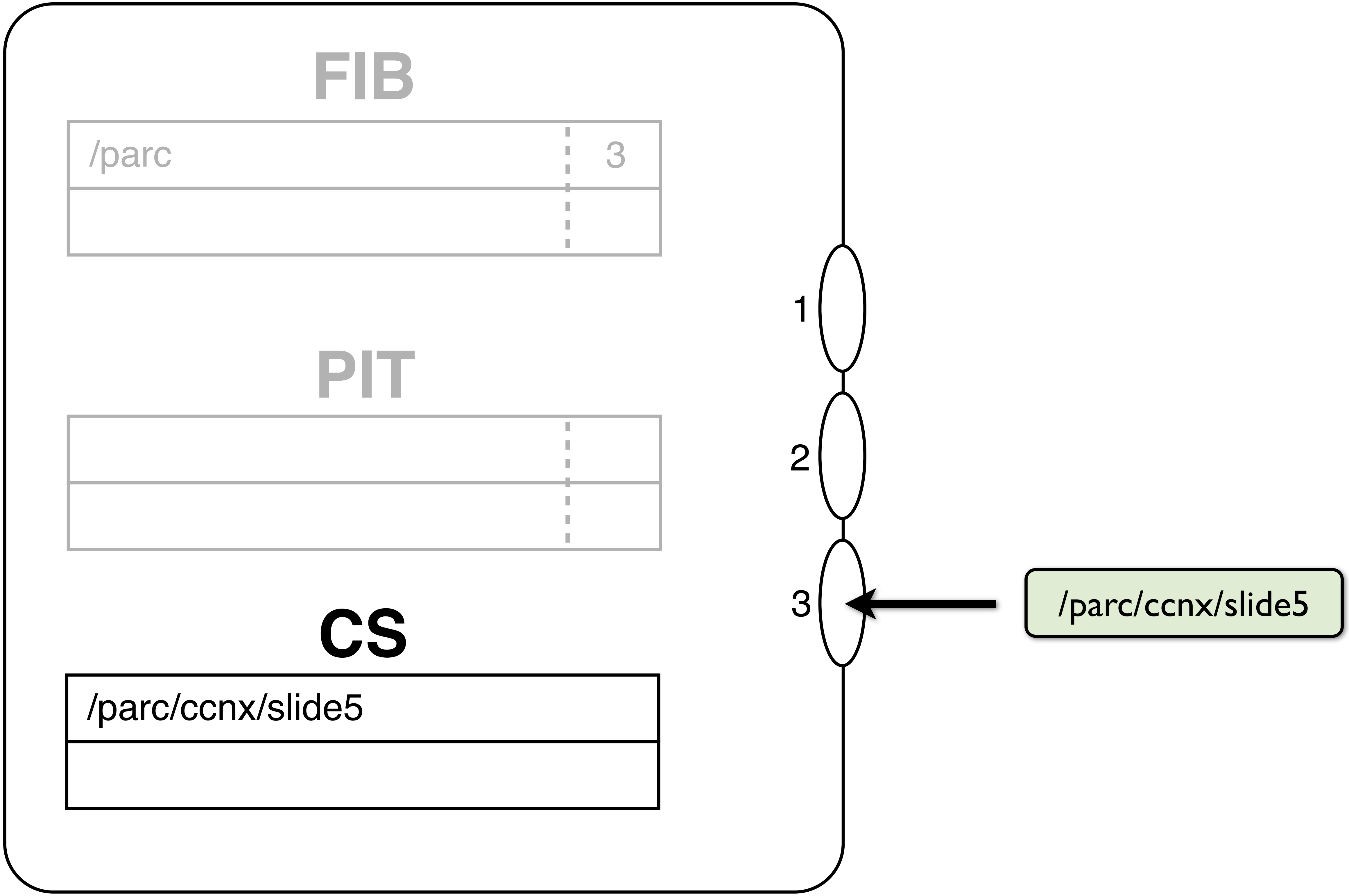
PIT

~~/parc/ccnx/slide5~~	~~1,2~~

optionally

Keep data

(more state)



FIB

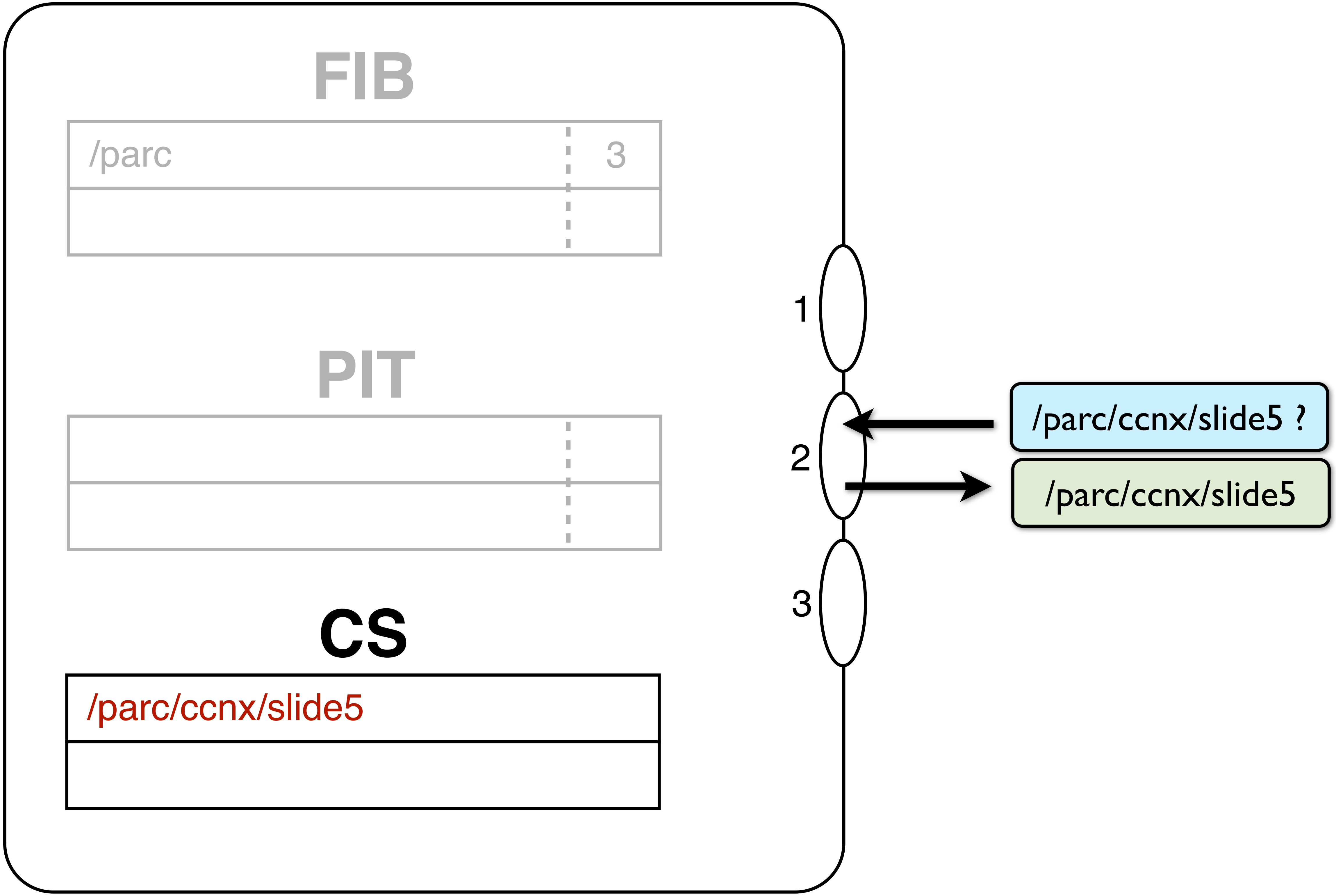
/parc	3

PIT

CS

/parc/ccnx/slide5

/parc/ccnx/slide5



FIB

/parc	3

PIT

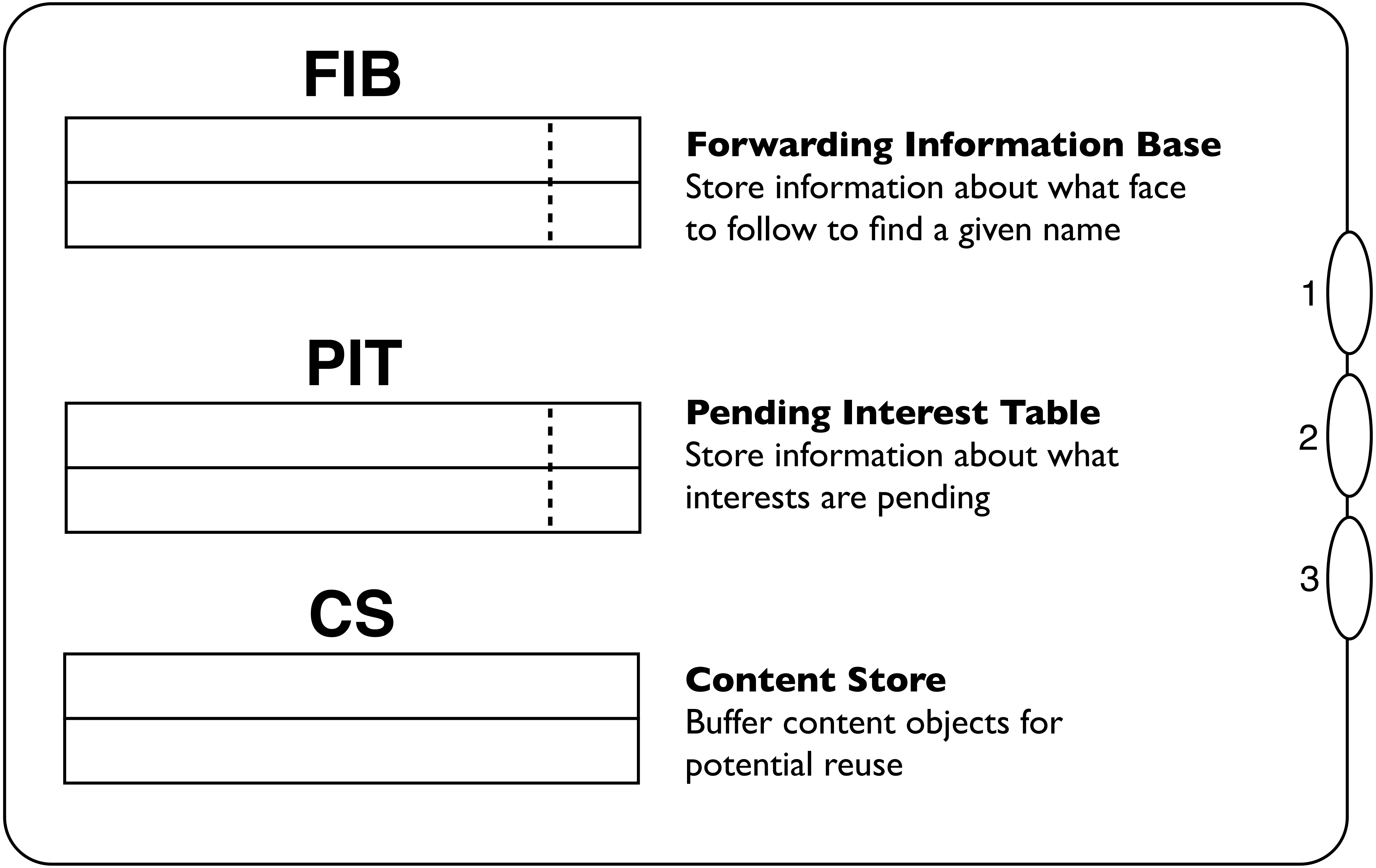
CS

/parc/ccnx/slide5

/parc/ccnx/slide5 ?

/parc/ccnx/slide5

Unify Architecture



Forwarding Information Base
Store information about what face to follow to find a given name

Pending Interest Table
Store information about what interests are pending

Content Store
Buffer content objects for potential reuse

The CCN Tenets

Name content objects

Secure content objects

Retrieve content by name

Keep state in the network

Unify the architecture

CCN - It’s Hot

ICN is HOT (Information-Centric Networks)

Universities - UCLA, UCI, UCSC, UCSD, Stanford, MIT, UMass, etc.
Companies - Cisco, Huawei, Alcatel-Lucent, Ericsson, NEC, IBM, Intel, Orange, BT, AT&T
Conferences - IETF/ACM/IEEE - SIGCOMM, INFOCOM, ICNP, ANCS, Mobicom, etc.

CCN is ICN

CCN is the baseline for all research
85% of papers and demos at SIGCOMM ICN 2013 are CCN based

You are CCN

You are leading the next wave of networking
Together we will change digital communication

CCN - Why It’s Hot

CCN Networks are Manageable

offer simple configuration

reduce deployment time

are easy to maintain

CCN Networks are Secure

don’t depend on link security

secure every object

protect privacy

CCN Networks are Resilient

are more resistant to attacks
require less infrastructure
support multiple traffic models
provide dynamic rerouting

CCN Networks are Smart

adapt to network conditions
provide better flow control
use resources more efficiently

CCN Networks are Flexible

support mobility
provide programmable packets
adapt as network changes

A CCN Network is

Manageable

because it names every content object

Secure

because it secures every content object

Resilient

because it retrieves content by name

Smart

because it uses state in the network

Flexible

because it's a unified system and architecture

CCN On Small Devices

Internet of Things

IoT is

Embedded devices (often limited resources)

that

Interact with the world through sensors and effectors (often not a keyboard and screen)

Communicate with other devices and infrastructure

Raspberry Pi

700MHz ARM

512 MB RAM

1.5 Watts

USB x2

Ethernet

HDMI

$35

An example thing

Platform for CCN

Installed on Raspberry Pi:

CCN & source (including Java)

CCN Wireshark

CCN VLC Media Player

Enter the Raffle to win one!

BeagleBone Black

1GHz ARM

512 MB RAM

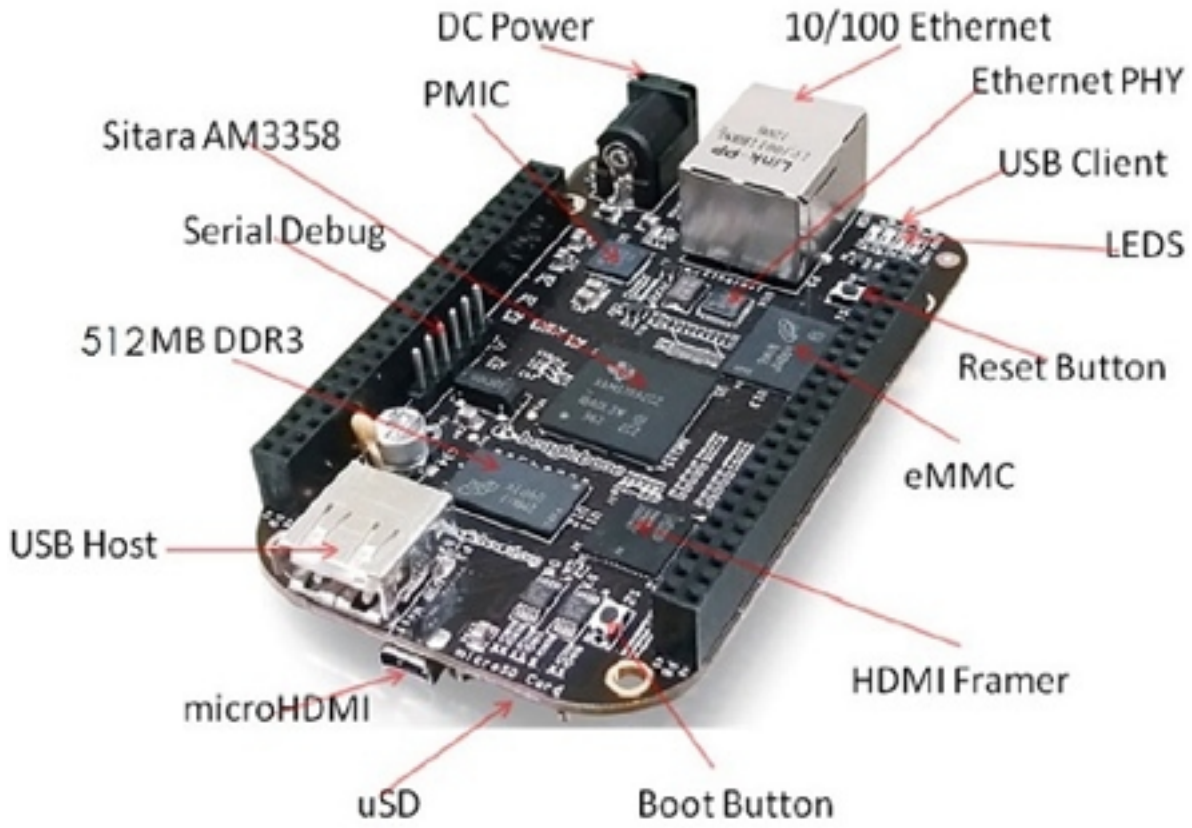
~1 Watt

Ethernet/USB

HDMI

66 GPIO/ADC

$45



Fits in a mint tin

Demo

CCN Progress

CCNx Releases

Approximately quarterly pace

Oct 3 2012 ccnx-0.6.2

Dec 9 2012 ccnx-0.7.0

Feb 4 2013 ccnx-0.7.1

May 19 2013 ccnx-0.7.2

Aug 13 2013 ccnx-0.8.0

CCNx **0.6.2**

Improvements to sync library support in both C and Java

Automatic key generation

Better tools for autoconfiguration

Stability improvements on Android

CCNx 0.7.0

Routing agents learn about the adjacency of the network that they find themselves participating in.

Interest timeout/retransmit managed better.

CCNx Android Service uses WebView to load ccndstatus

VLC plugin uses separate preferences for version timeout and media

Content Objects now have a way to carry experimental extension fields.

CCNx Android Services have a toggle to enable/disable sync on startup

Local ccnd / local prefix auto-configuration

CCNx **0.7.1**

Generalize signing/verification code to support MACs

Restructure sync slice code for java library

Improve and modernize java library timer mechanisms

Improve in-line documentation of java sync code

Start a repository automatically in ccndstart

Add guest prefix support

Java BloomFilter is deprecated

CCNx **0.7.2**

Junit tests and System Tests are separated from main code.

An easier to read encoding for multiple escape characters in ccnx URIs

ccnd prints Excludes in an Interest to improve debug-ability

Wireshark plugin updated for 1.8.6

CCNx **0.8.0**

Optional use of symmetric key HMACs in place of public/private key signatures.

ccnd content store uses a flatname representation for indexing, in the same way that ccnr does.

The more readable escaping convention in URIs is now the default.

An example Ubuntu rc startup script is now included.

Notable Features

Auto-discovery - broadcast and DNS-based

Guest Prefix Support

Adjacency Prefixes

Content Object Extensibility

Symmetric Key MAC Alternative

Revised Content Store Implementation

CCN Direction

What’s Next?

Enable **Adoption**

Research Prototype → Production Prototype

Enable **Experimentation**

From wire-formats to application models
Routing, security, trust, performance

Enable **Stability and Interoperability**

Object protocols, application protocols

Enable **Productivity**

Clean API, language bindings, IDE integration, documentation

Models

Programming Model

What are the programmatic entities?

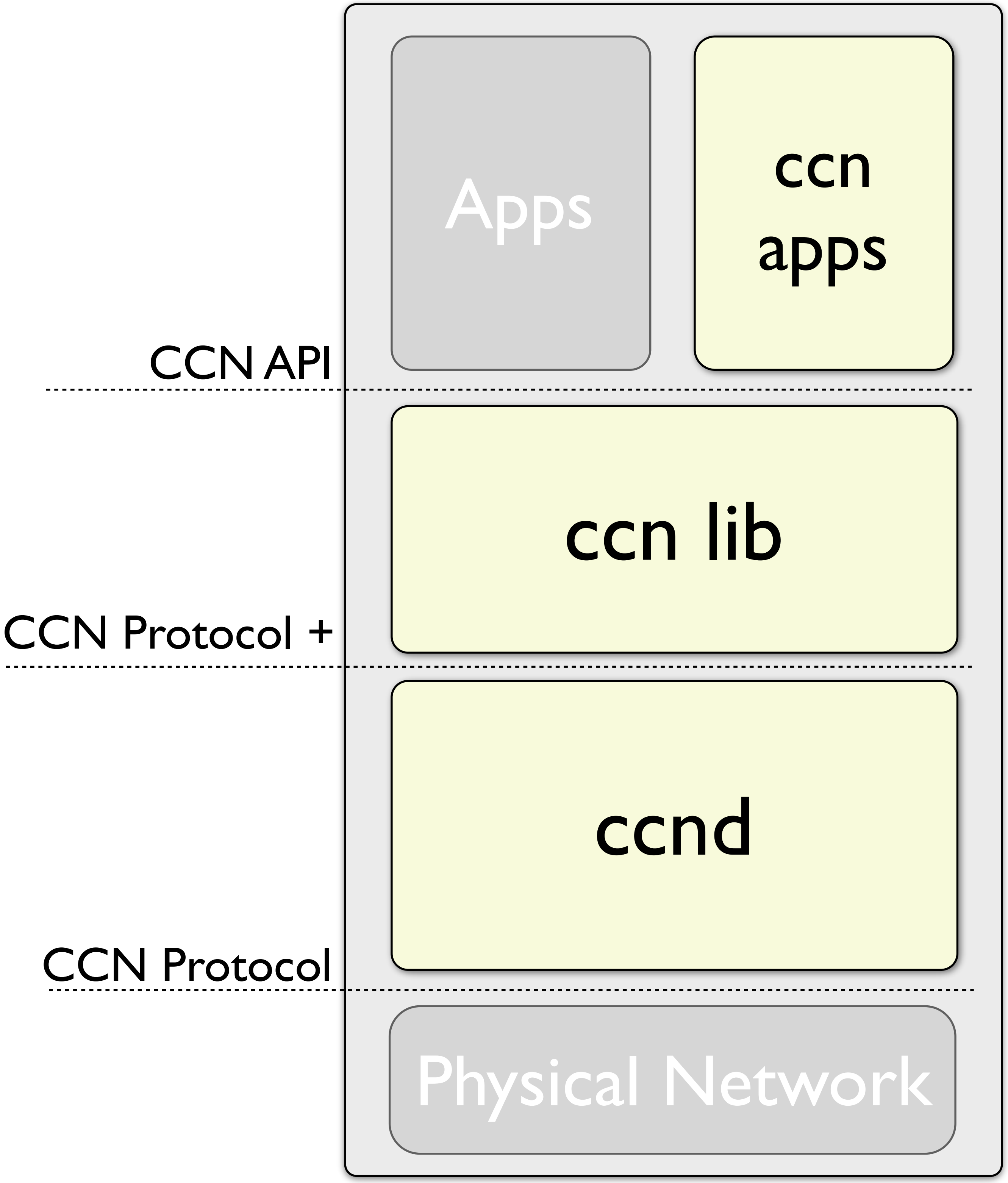
How do they map to functional, object-object oriented or imperative paradigms?

Application Model

How does an application use CCN?

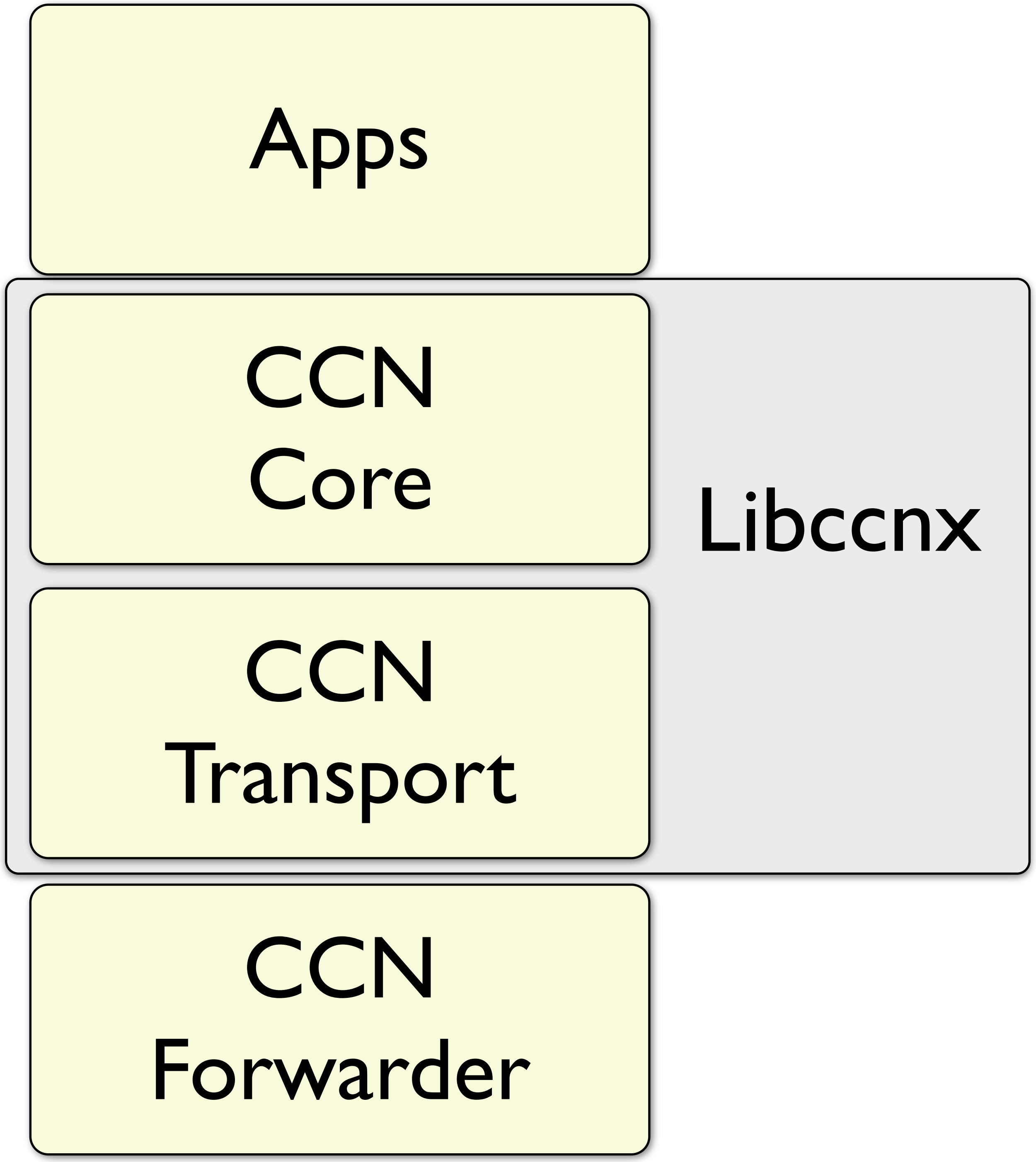
Does CCN change application design?

Today

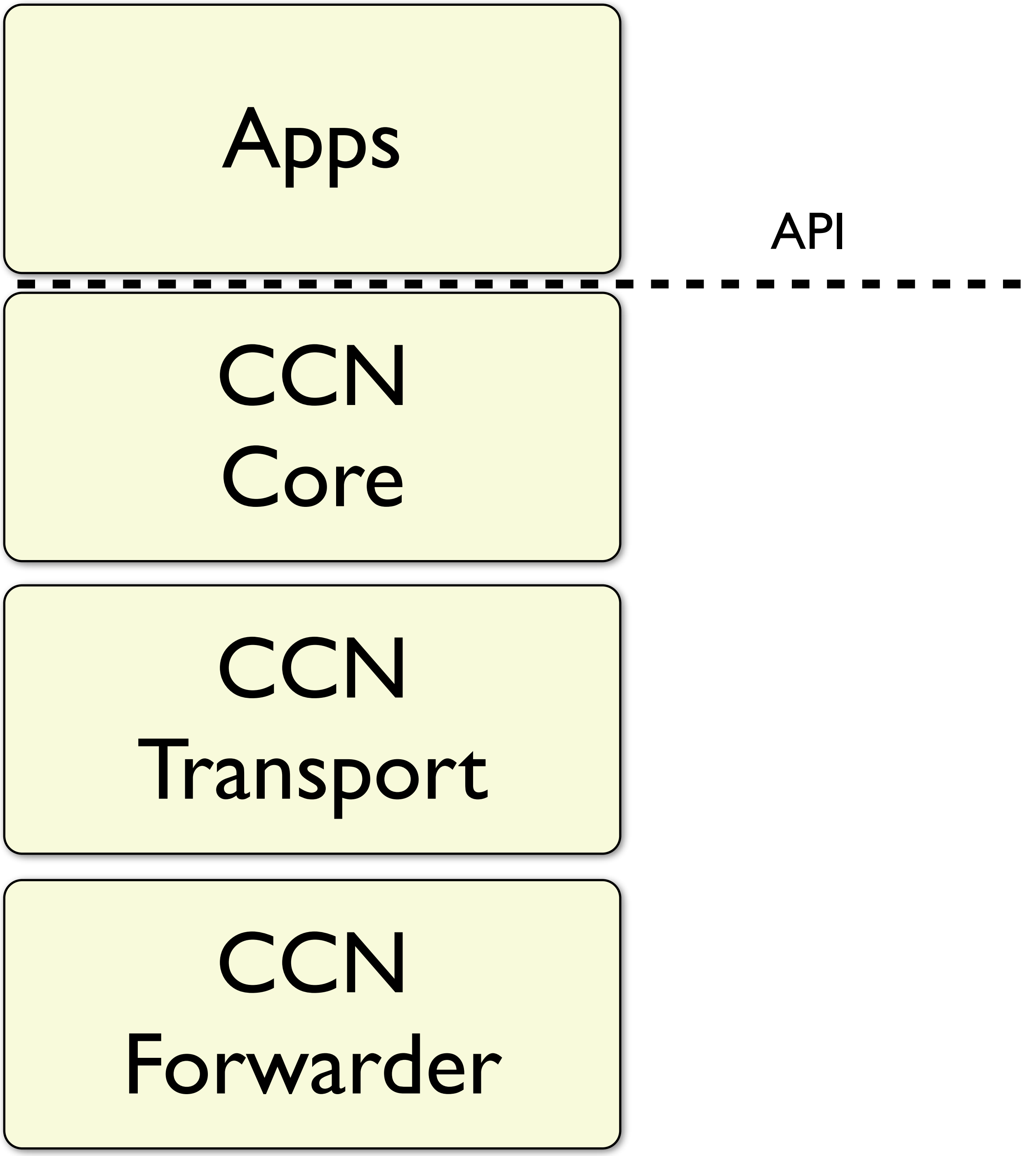

Apps
ccn apps
CCN API
ccn lib
CCN Protocol +
ccnd
CCN Protocol
Physical Network

CCN Library Architecture

CCN API
CCN Core
CCN Transport



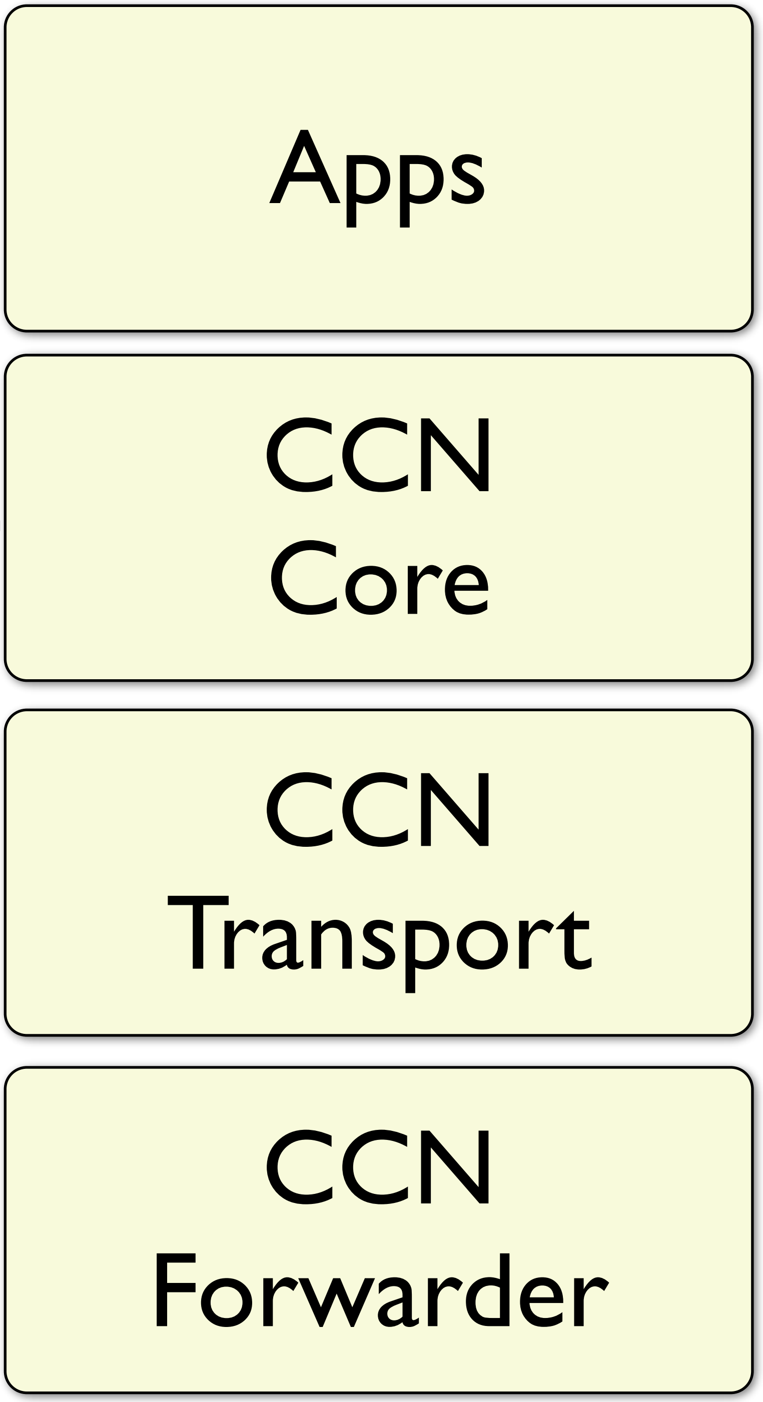
Libccnx API



Language Bindings
Separates Concerns
Enables Experimentation

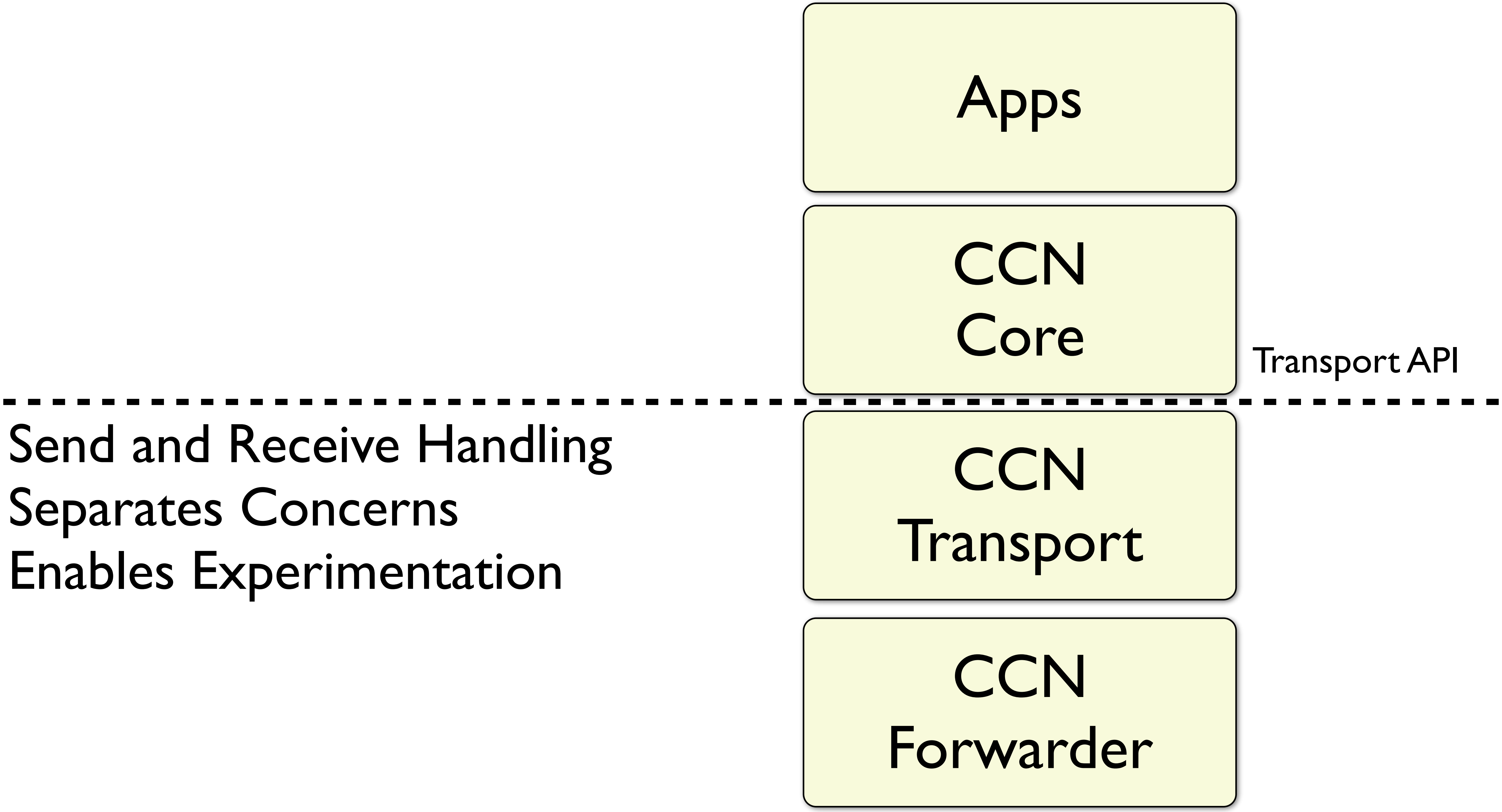
Stable
Curated
Familiar
Readable
Memorable
Hard to Misuse

Libccnx Core



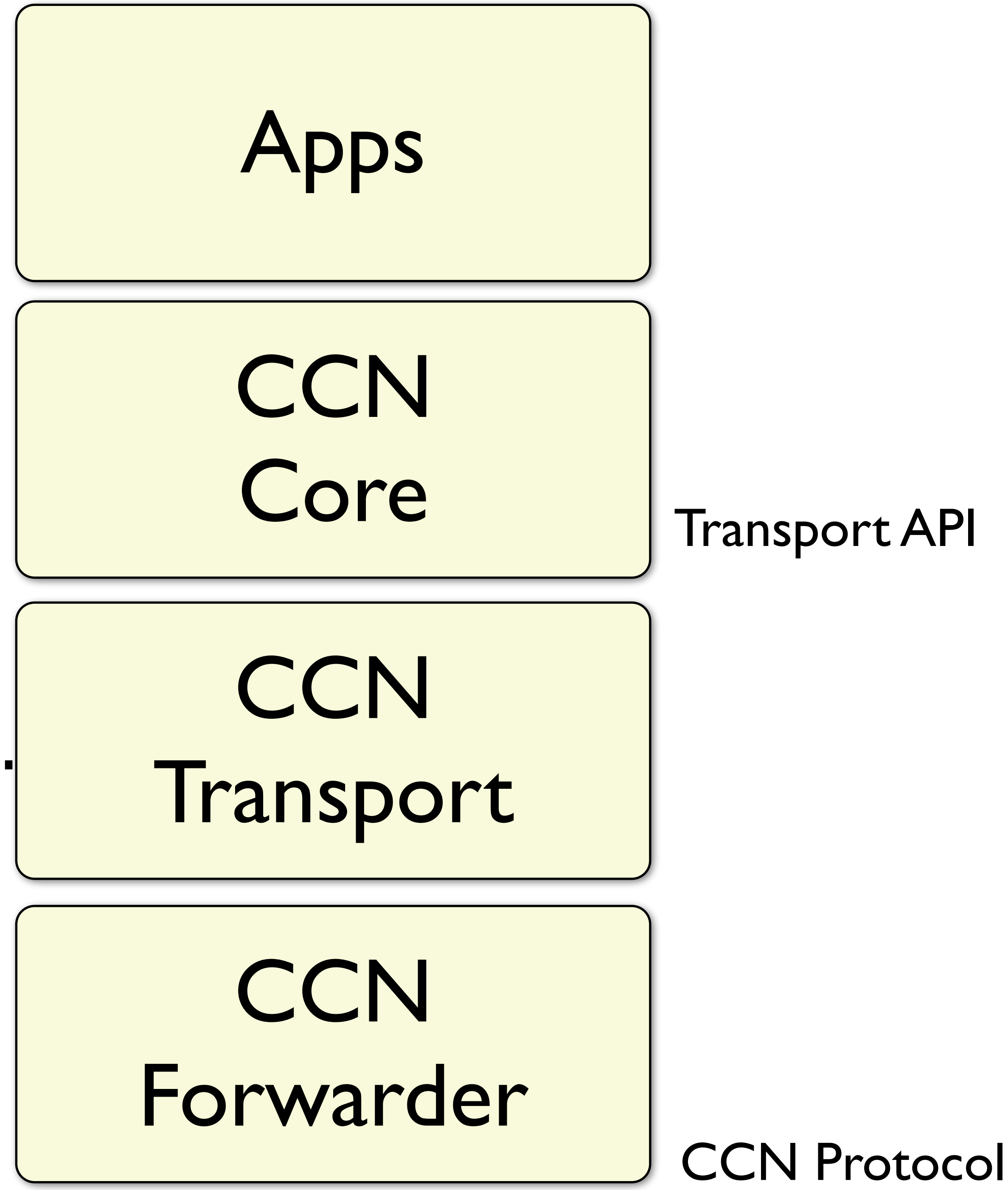
Message Semantics
Object Protocols
Substitutable

Libccnx Transport API



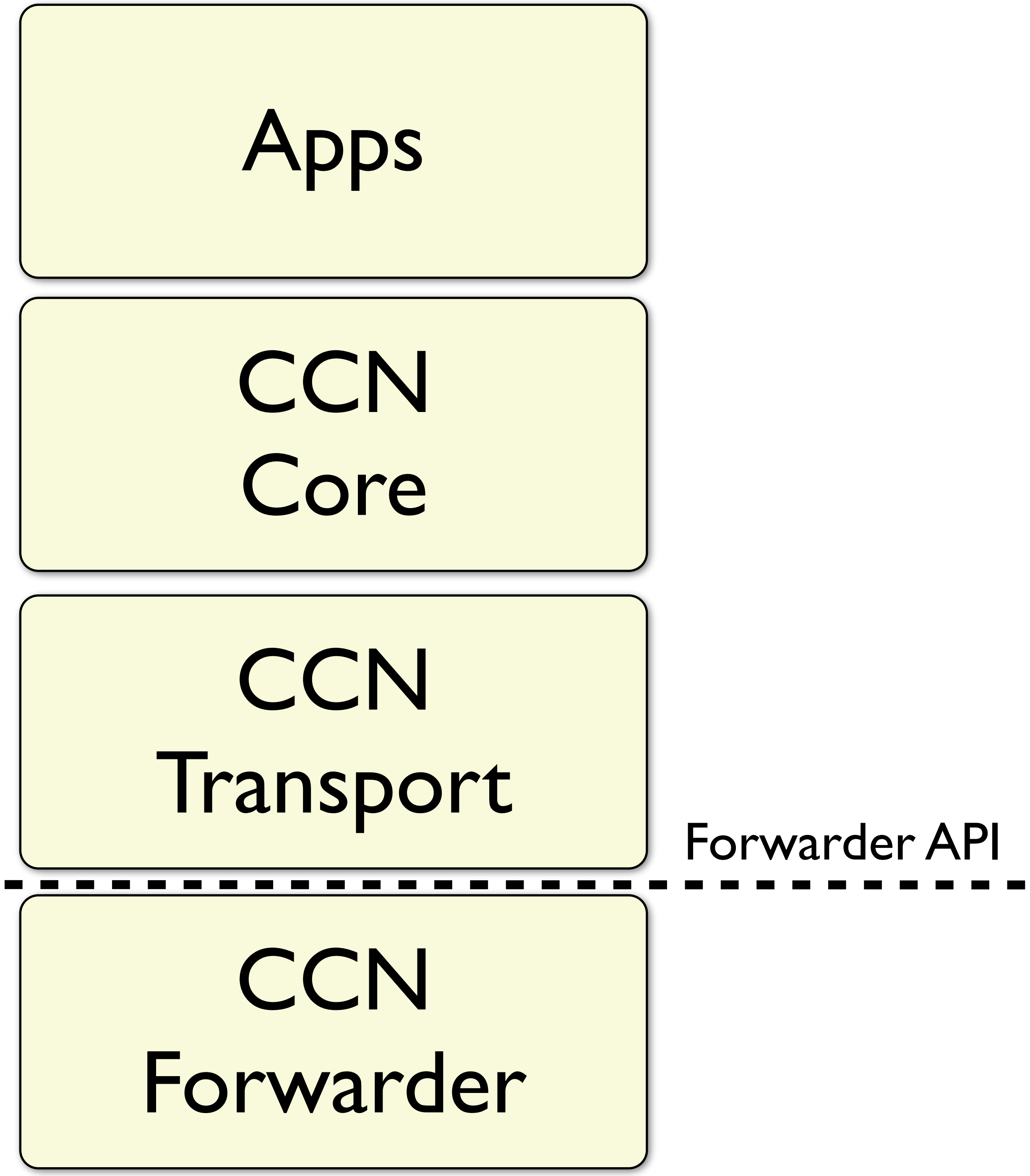
Send and Receive Handling
Separates Concerns
Enables Experimentation

Libccnx Transport



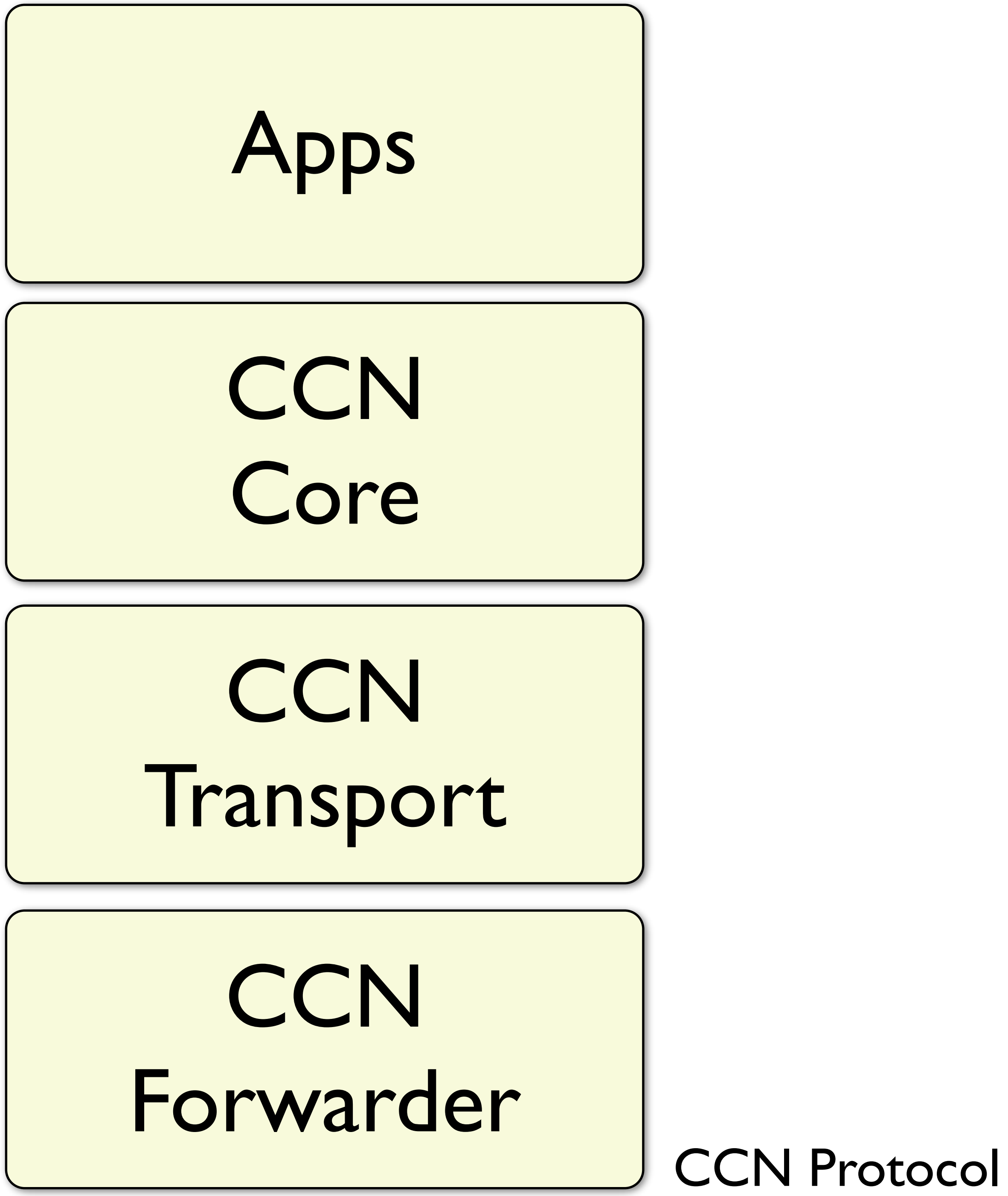
Message Handling
Flow Control
Wire Formats
Substitutable

Libccnx Forwarder API



Send and Receive
Forwarder Control
Forwarder Inspection

CCN Protocol



Well defined
Interoperable
Thin waist

Example (Datagram)

```
int fd = ccnx_socket(PF_CCNX, SOCK_DGRAM, PROTO_TLV);

CCNxName name = CCNxName_Create("ccnx:/parc.com/object");

CCNxInterest *interest = ccnxInterest_Create(name);

struct msghdr *message = ccnxInterest_Encode(interest);
ccnxInterest_Send(socket, message, 0);
ccnxContentObject_Recv(socket, message, MSG_WAITALL);
write(1,
    message.msg_iov[CCN_DTAG_Content].iov_base,
    message.msg_iov[CCN_DTAG_Content].iov_len);
ccnx_close(fd);
```

Example (Stream)

```
int fd = ccnx_socket(AF_CCNX, SOCK_STREAM, PROTO_TLV);

CCNxName name = CCNxName_Create("ccnx:/parc.com/stream");

struct ccnaddr address = {
  .name = ccnxName_Encoded(name)
};

ccnx_connect(fd, &address, sizeof(address));

char buffer[8192];
while (1) {
  int nread = ccnx_read(fd, buffer, sizeof(buffer));
  write(1, buffer, nread);
}

ccnx_close(fd);
```

Open Topics

These seem equivalent, should they be?

```
ccnx_socket() == socket(2) ?

ccnx_connect() == connect(2) ?

ccnx_read() == read(2) ?

ccnx_write() == write(2) ?

ccnx_select() == select(2) ?

ccnxInterest_Send() == sendmsg(2) ?

ccnxContentObject_Recv() == recvmsg(2) ?

ccnx_close() == close(2) ?
```

Thank you!

One more thing.....

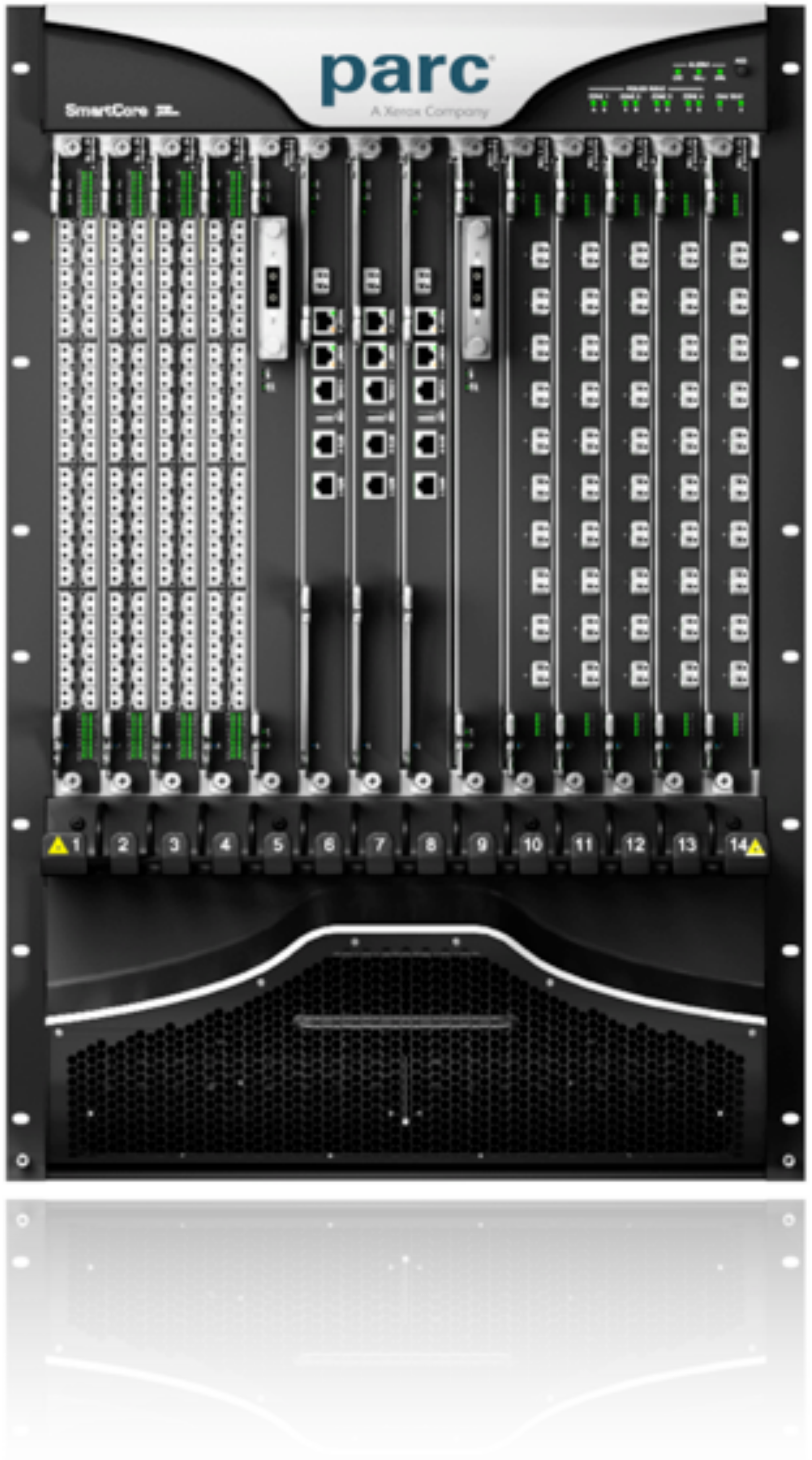
Project 42

Penn

12 Terabit non-blocking fabric

14 slot chassis

1 Terabit per Slot
40x1GbE
10x10GbE
20x1GbE + 5x10GbE
100GbE



Teller

4.4 Terabit non-blocking fabric

6 slots chassis

Hardware Architecture

Distributed Architecture

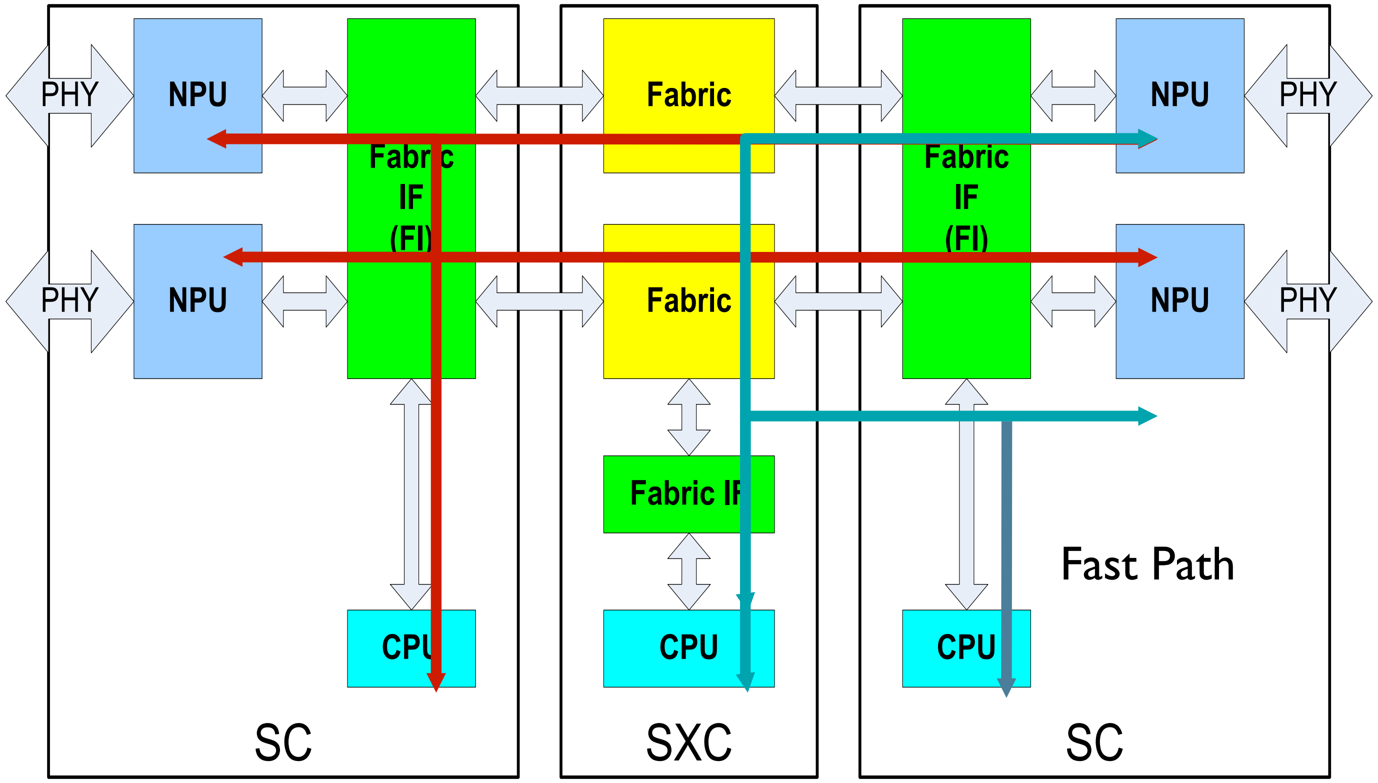
Separate Control Network for Inter-card IPC

Redundant design for high availability

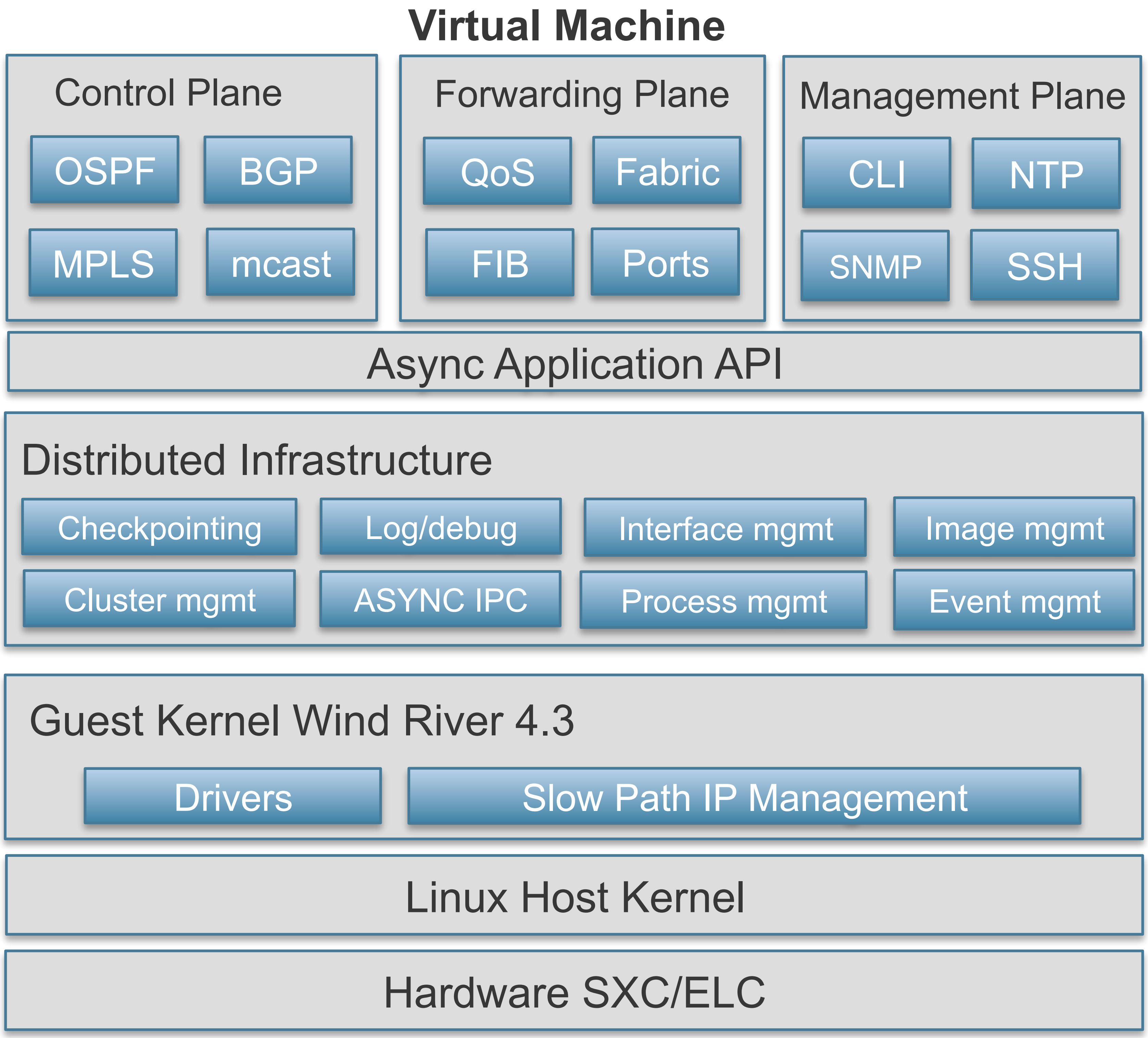
Large memories on line cards and storage interfaces

32 core processor for packet inspection and data collection

Wire speed programmability in fast path



Software Architecture



Open Source Linux Kernel and KVM Hypervisor

P42 Software runs on a VM enables In Service Software upgrade, multi-tenant and network slicing

VM technology isolation allows running other services in a VM without impacting Routing

Functional modules are processes and processes are restartable

Fault monitoring and recovery mechanism for high availability

Asynchronous to avoid thread switching overhead.

Let's go change the world!

Thank you!