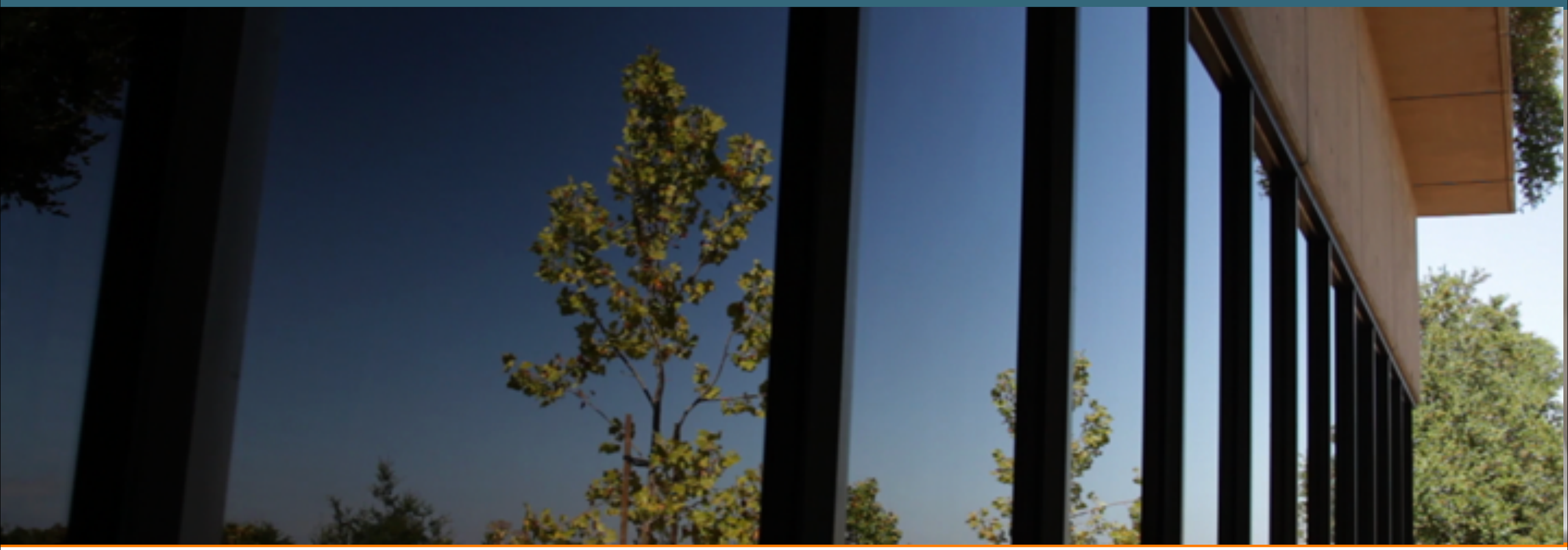
parc
A Xerox Company

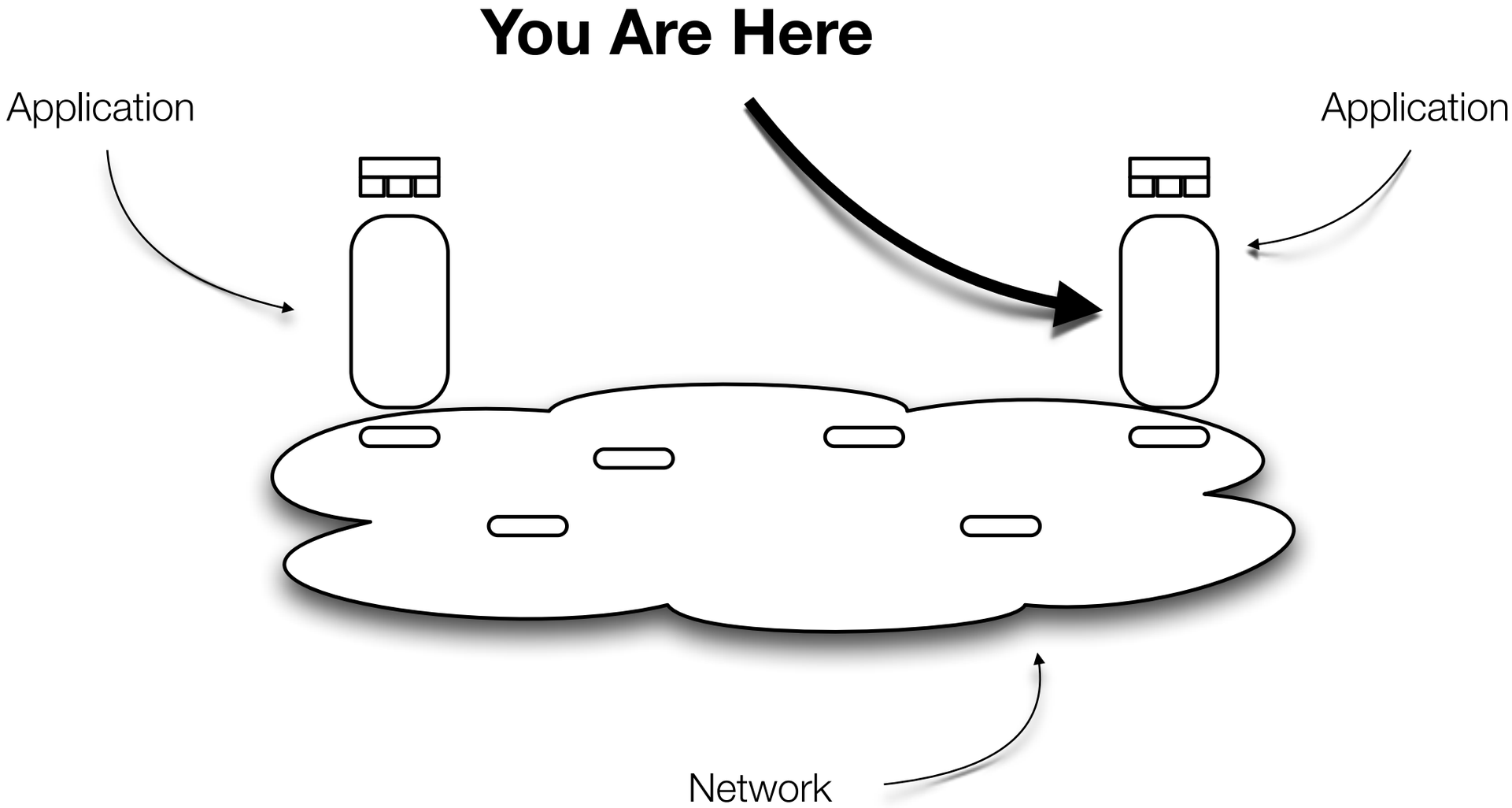
CCNx 1.0 Network Software Stack

Architectural Overview



glenn.scott@parc.com

Overview



Requirements

Provide Structure and a Vocabulary

- Design simple parts connected by clean interfaces.
- Implement modularity and separate concerns.

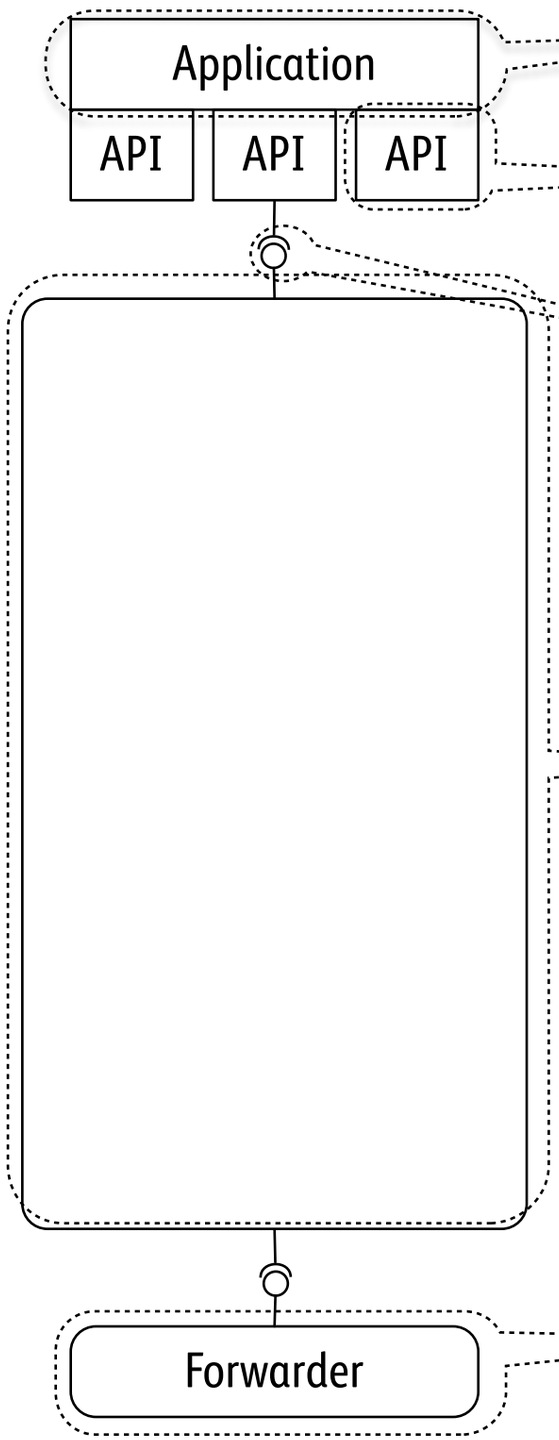
Promote Stability and Enable Extensibility

- Design for composition and substitution of components.
- Implement the core functionality plus examples and proofs-of-concept.
- Design for evolution.

Enable Productivity

- Design for inspection and debugging.
- Implement for usability and “learnability.”

Vocabulary



A **program** that uses one or more CCN APIs to communicate.

Application Programming Interfaces implement a particular kind of interaction with the CCN Transport.

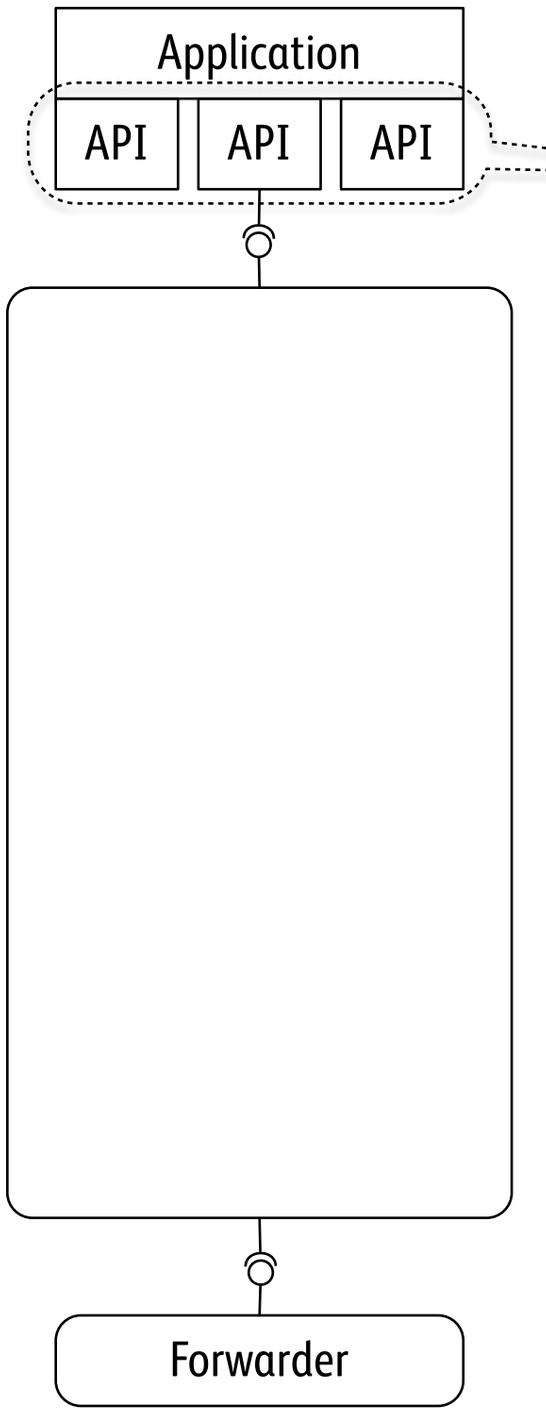
The **Transport Programming Interface** providing a uniform interface to a **Transport Stack.**

The **Transport Stack** conveying data between the application and the network.

The CCN **Forwarder** directing traffic on the network.

glenn.scott@parc.com

Application Programming Interface



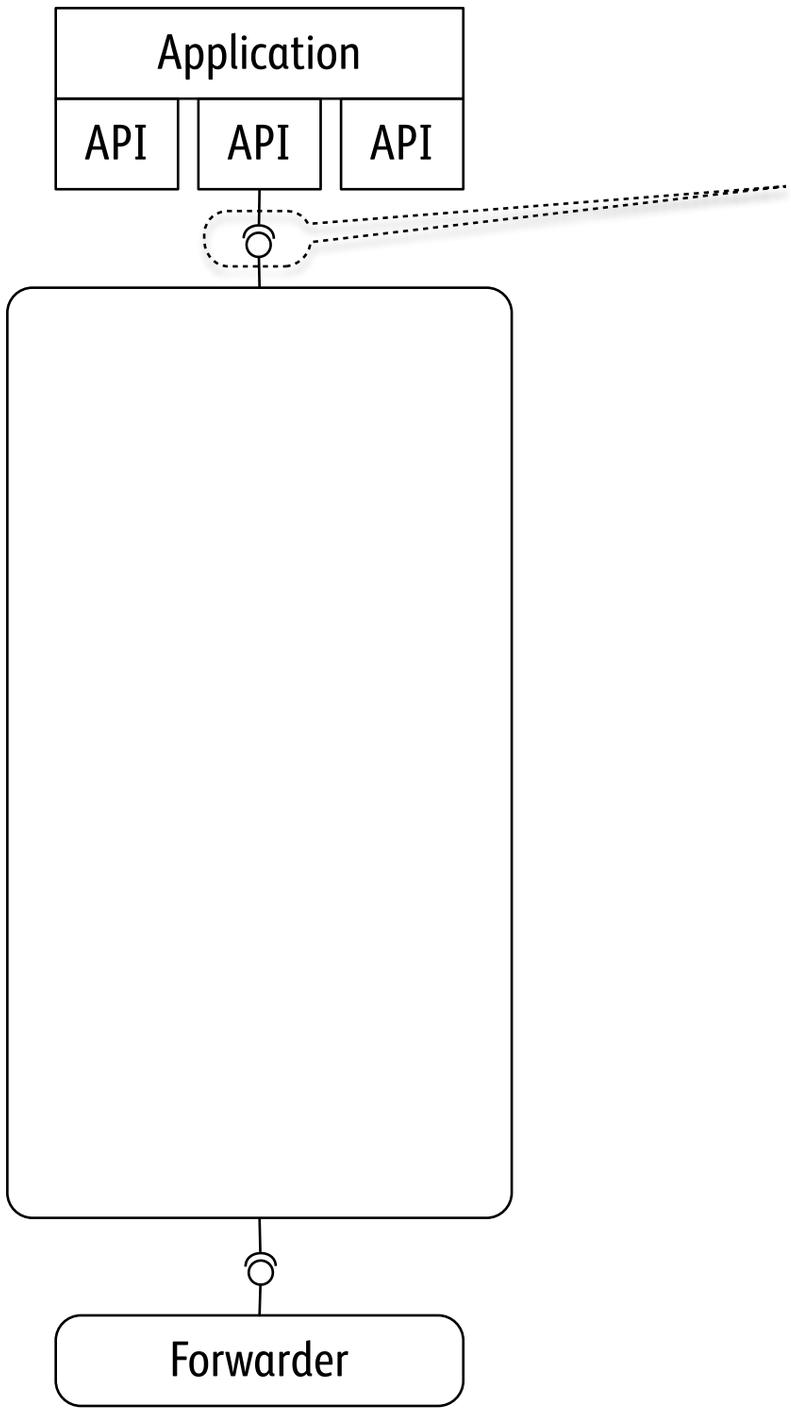
Many kinds of APIs are possible.

- Berkeley Sockets
- Legacy CCN/NDN
- Repository
- Streaming
- Message Passing
- Callbacks
- You decide what you want.

Applications may use different APIs at the same time.

glenn.scott@parc.com

Transport Programming Interface



Passes messages to and from the API

`Transport_Open`
: Opens and configure a Transport Stack

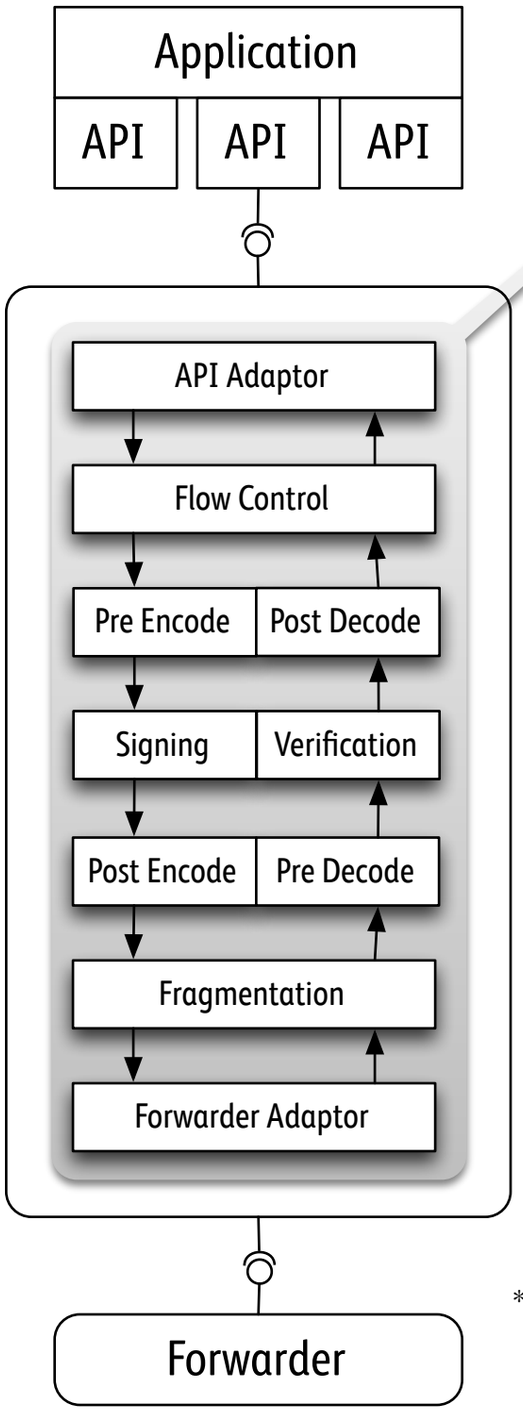
`Transport_Close`
: Closes a Transport Stack

`Transport_Send`
: Send a message to a Transport Stack

`Transport_Receive`
: Receive a message from a Transport Stack

`Transport_Notify`
: Signal that an event occurred.

Transport Stack



*Not actual size.

Assembled and initialized at create-time

Implements CCN object protocols

- Segmented objects
- Versioned objects

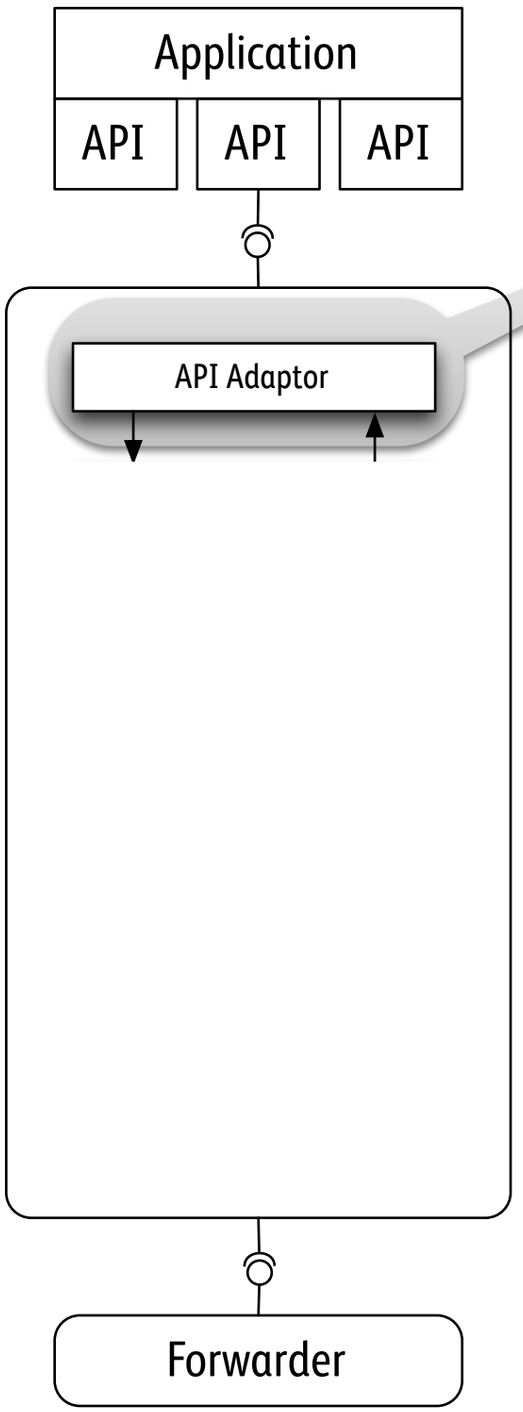
Integrates external services:

- External Caches
- Identity, Key and Certificate services
- Trust-model implementations

Control messages coordinate operations.

Data messages contain Interests and Content Objects.

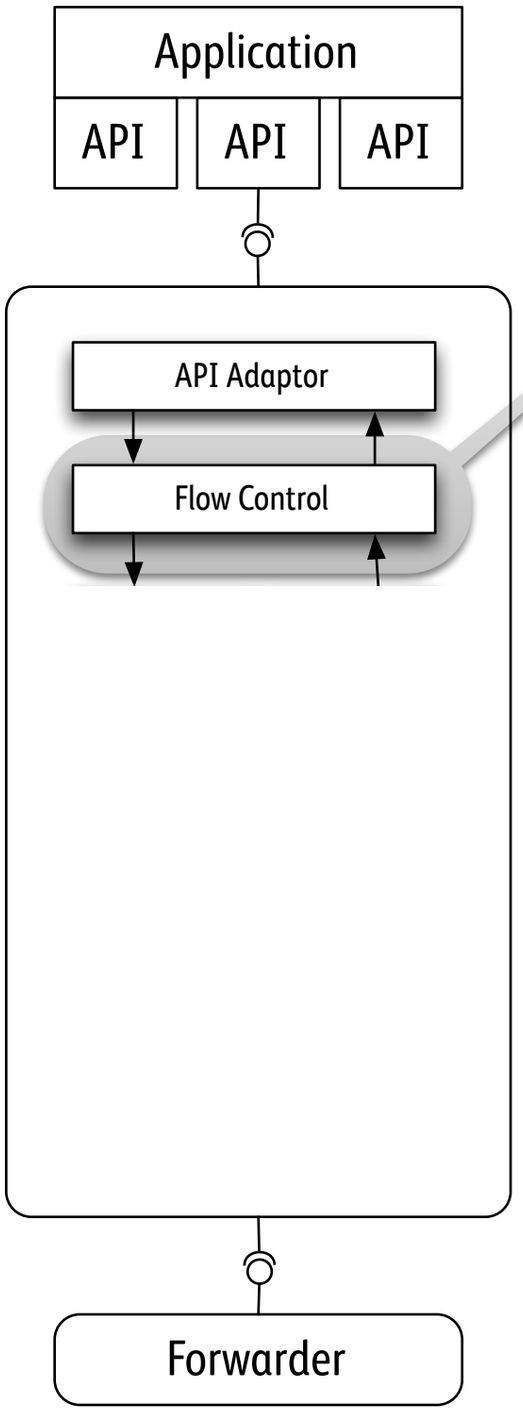
API Adaptor



Required Component

Communicates with a specific Transport Stack instance.

Flow Control



Optional Component

Traffic Shaping/Management

Handles Interests and Content Objects

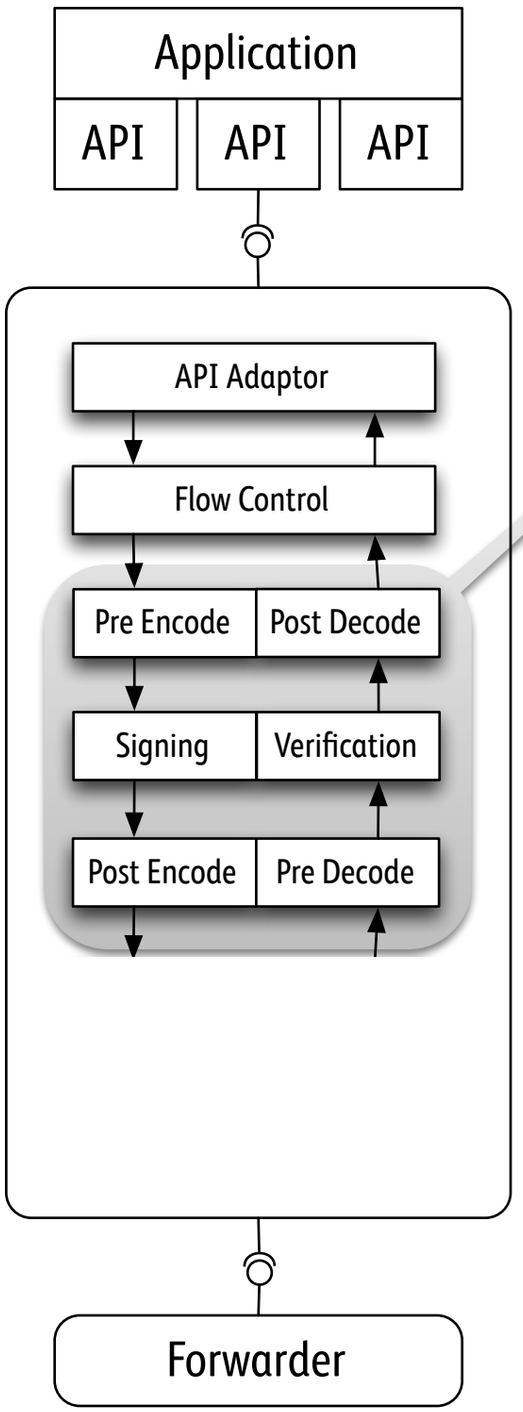
Interest retransmissions

Pipelining Interests

Content Object ordering

Link resolution

Verification and Encoding



Optional Component

Validate outbound Content Objects.

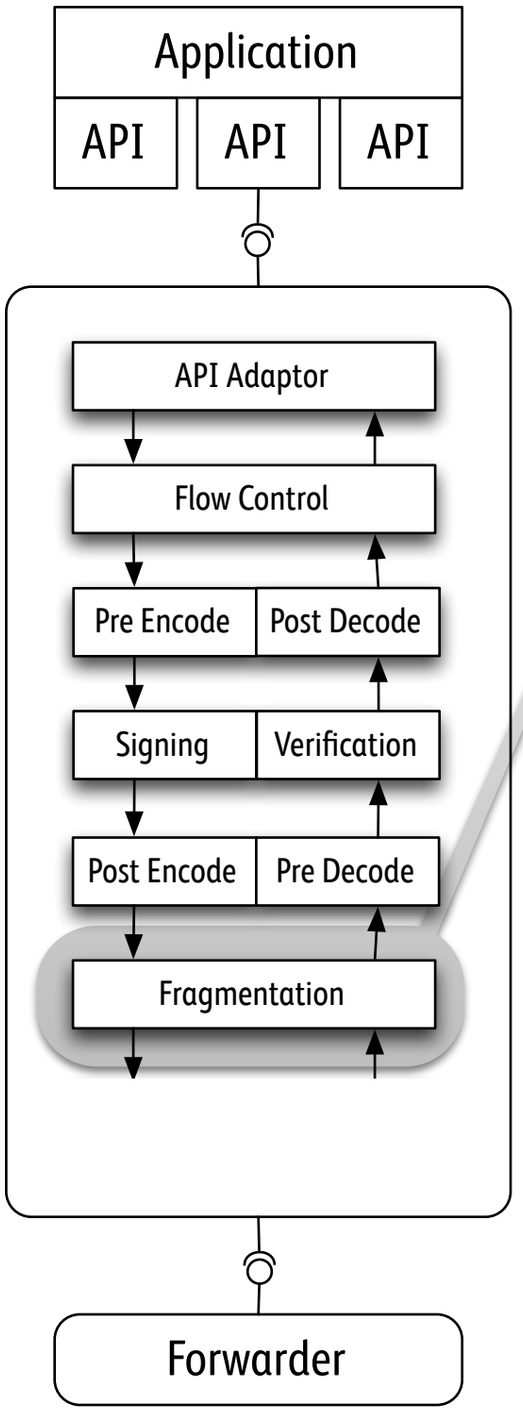
Encode and sign outbound encoded Content Objects.

Verify inbound Content Objects.

Interface to external key stores and key management systems.

glenn.scott@parc.com

Fragmentation



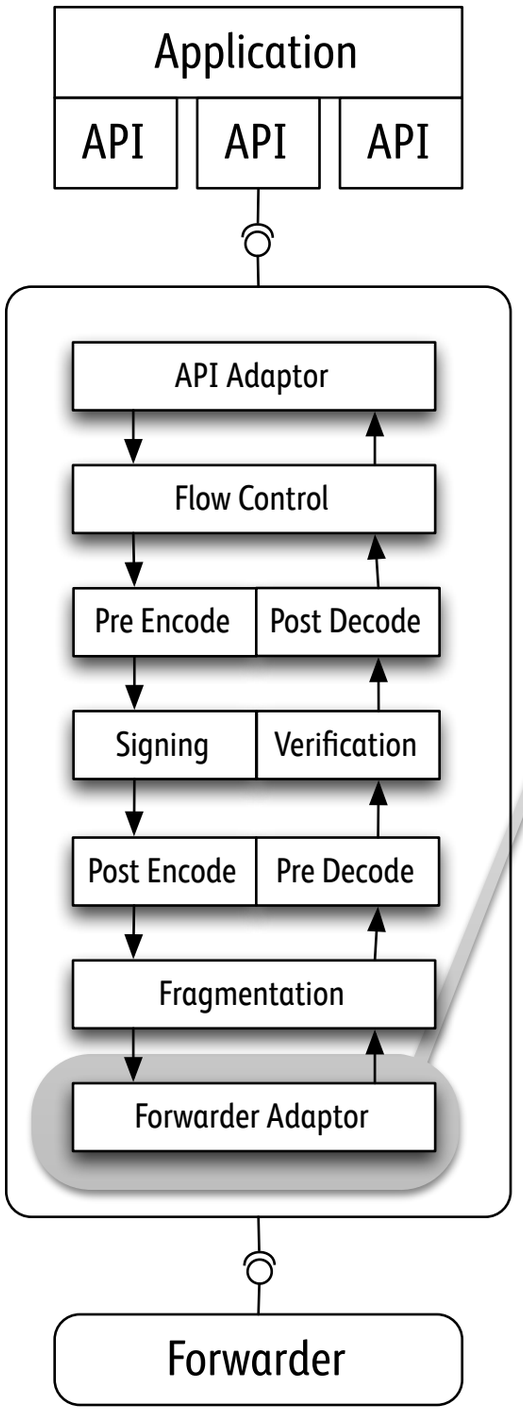
Optional Component

Outbound packet fragmentation.

Inbound packet reassembly.

glenn.scott@parc.com

Forwarder Adaptor



Required Component

Communicates with a Forwarder.

glenn.scott@parc.com

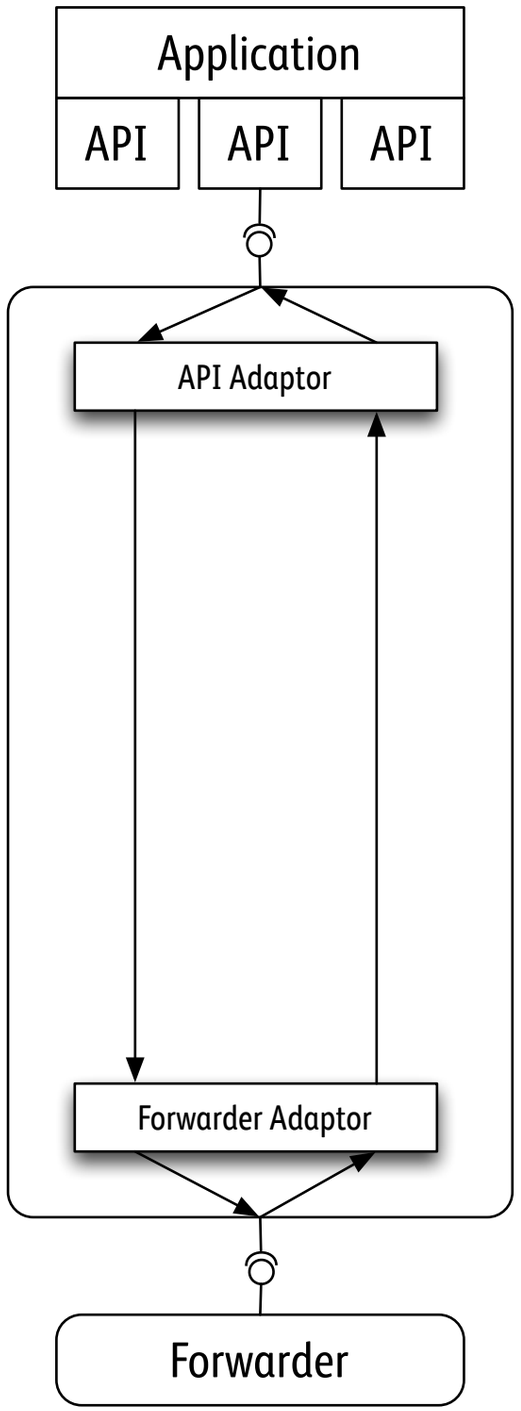
Examples

Simple Stack

Legacy CCN Stack

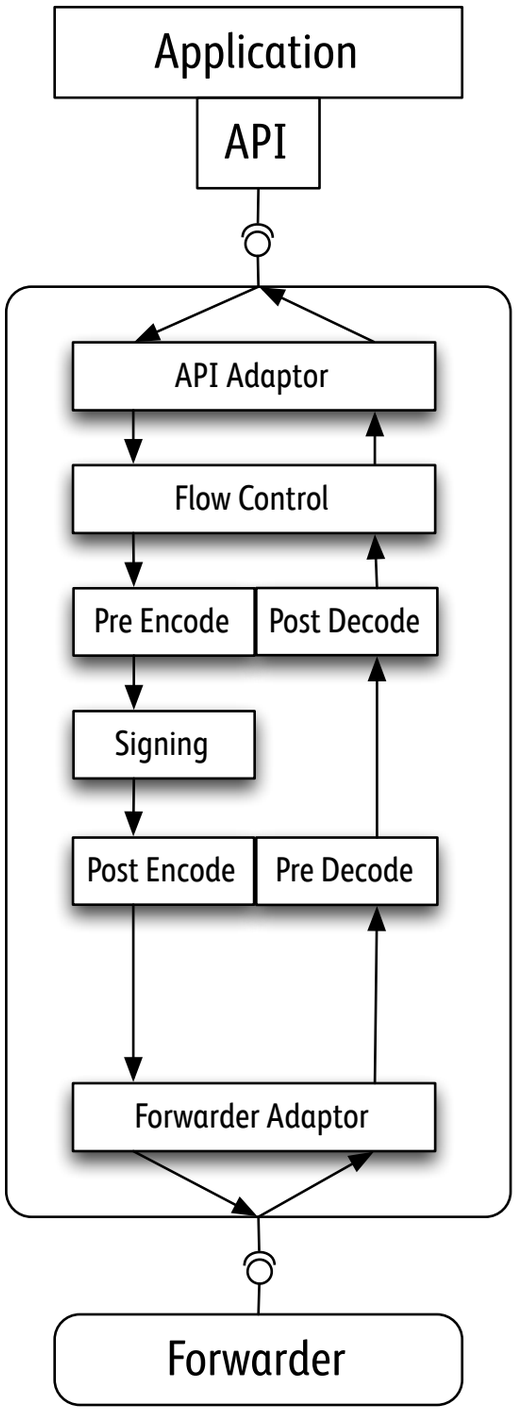
glenn.scott@parc.com

Simple Stack



The API is responsible for everything.

Legacy CCNx Stack



API callback with get(), put(), listen(), and expressInterest().

Flow control is Interest re-expression.

Applications check signatures.

No fragmentation.

glenn.scott@parc.com

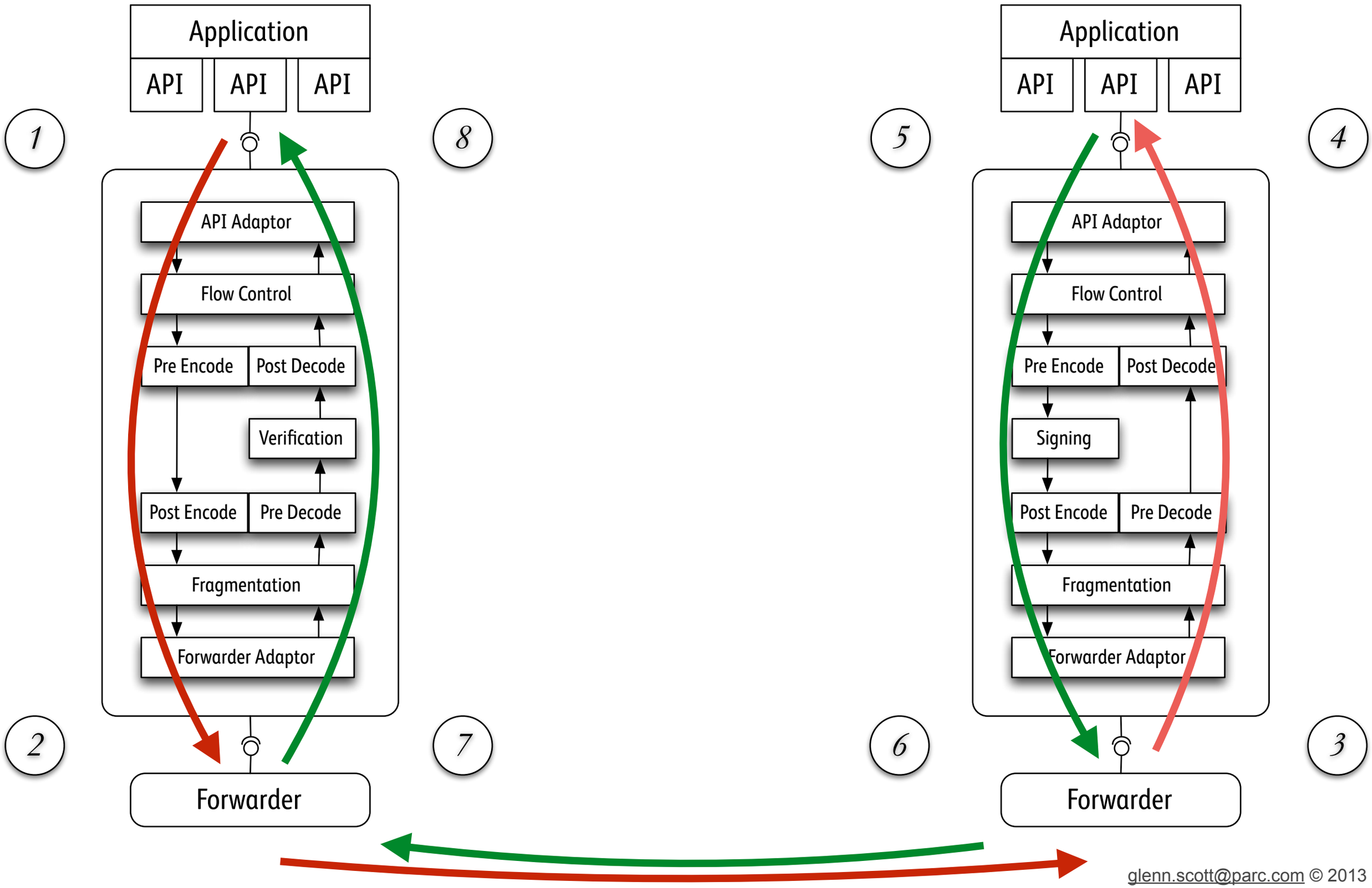
http://www.parc.com/ccn

http://www.ccnx.org

Fin

glenn.scott@parc.com

Example Path



Software Modules

Algorithm Library - General purpose facilities (buffers, lists, maps, etc)

Security Library - Cryptographic and other security facilities

CCN Core - CCN constructs as C objects

Transport *“Ready To Assemble”* A Transport Stack implementation

Legacy CCN API - The legacy API found in CCNx < 1.0

CCN Socket API - A socket interface to CCN >= 1.0

Runtime Development Tools - Developer aids

Unit Testing Tools - TDD, developer and release aids