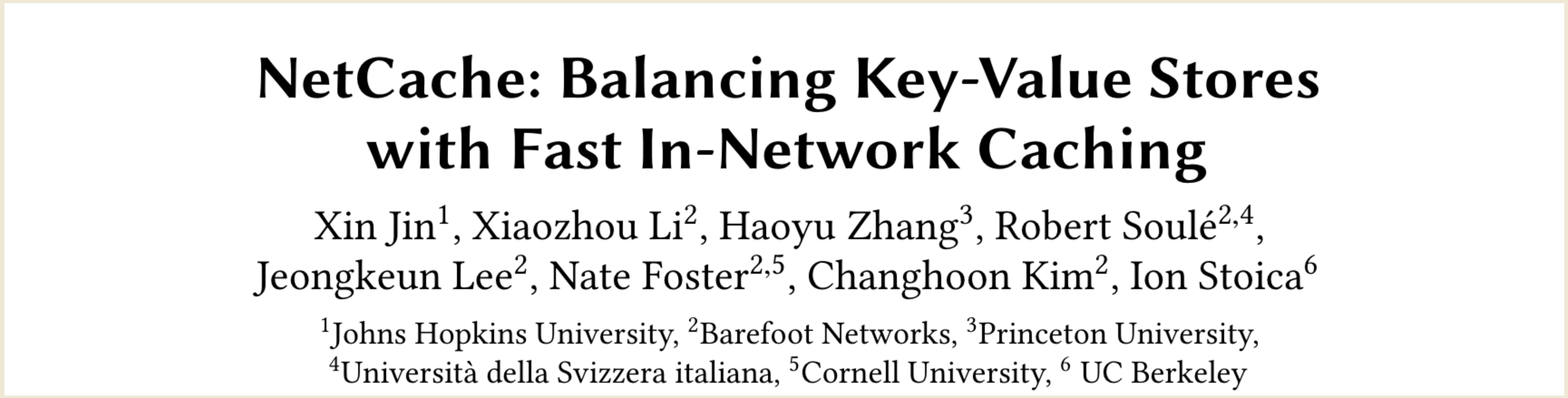
Bertha: Tunneling Through the Network API

Akshay Narayan Aurojit Panda Mohammad Alizadeh

Hari Balakrishnan Arvind Krishnamurthy Scott Shenker

MIT, NYU, UW, UC Berkeley, ICSI

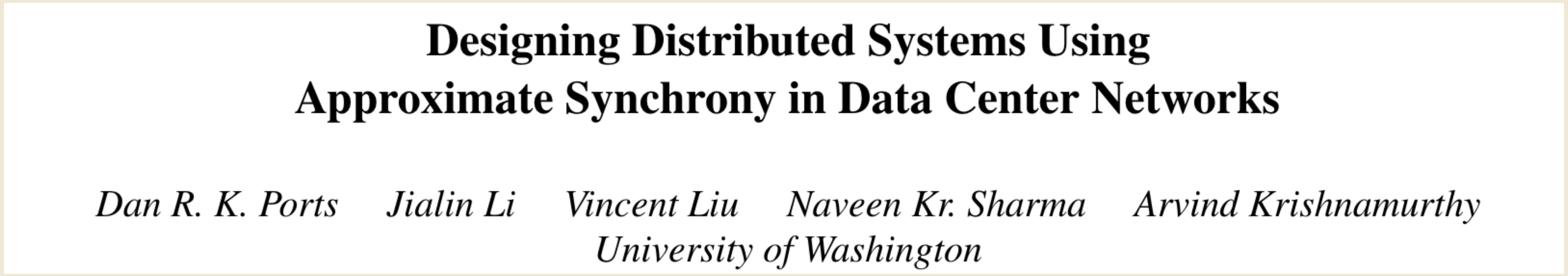
Recent Developments



NetCache: Balancing Key-Value Stores with Fast In-Network Caching

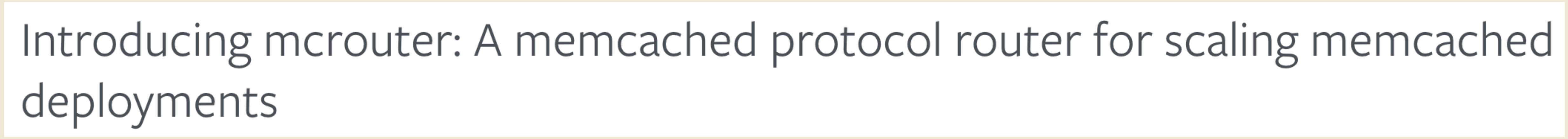
Xin Jin[1], Xiaozhou Li[2], Haoyu Zhang[3], Robert Soulé[2,4], Jeongkeun Lee[2], Nate Foster[2,5], Changhoon Kim[2], Ion Stoica[6]

[1]Johns Hopkins University, [2]Barefoot Networks, [3]Princeton University, [4]Università della Svizzera italiana, [5]Cornell University, [6] UC Berkeley



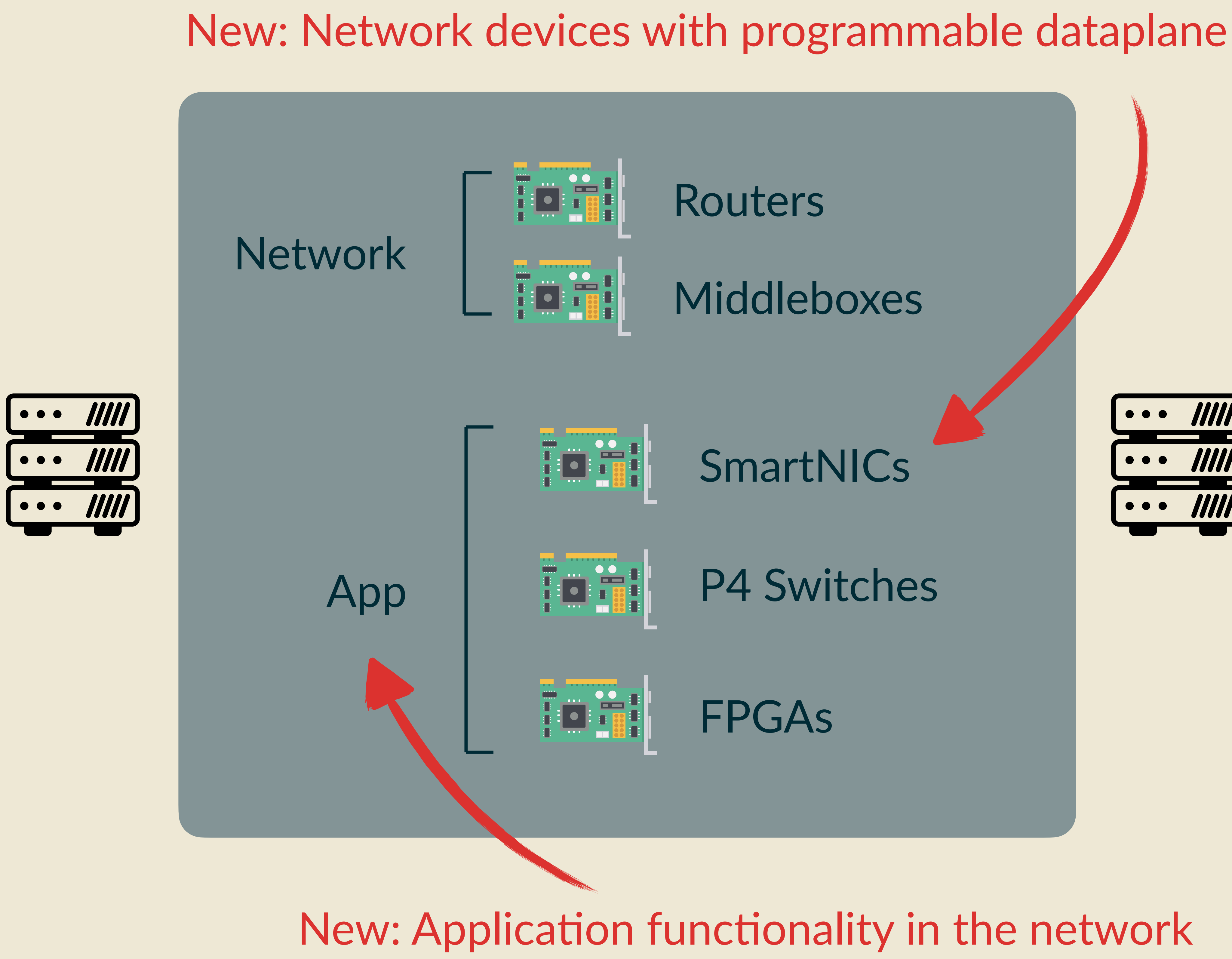
Designing Distributed Systems Using Approximate Synchrony in Data Center Networks

Dan R. K. Ports *Jialin Li* *Vincent Liu* *Naveen Kr. Sharma* *Arvind Krishnamurthy*
University of Washington

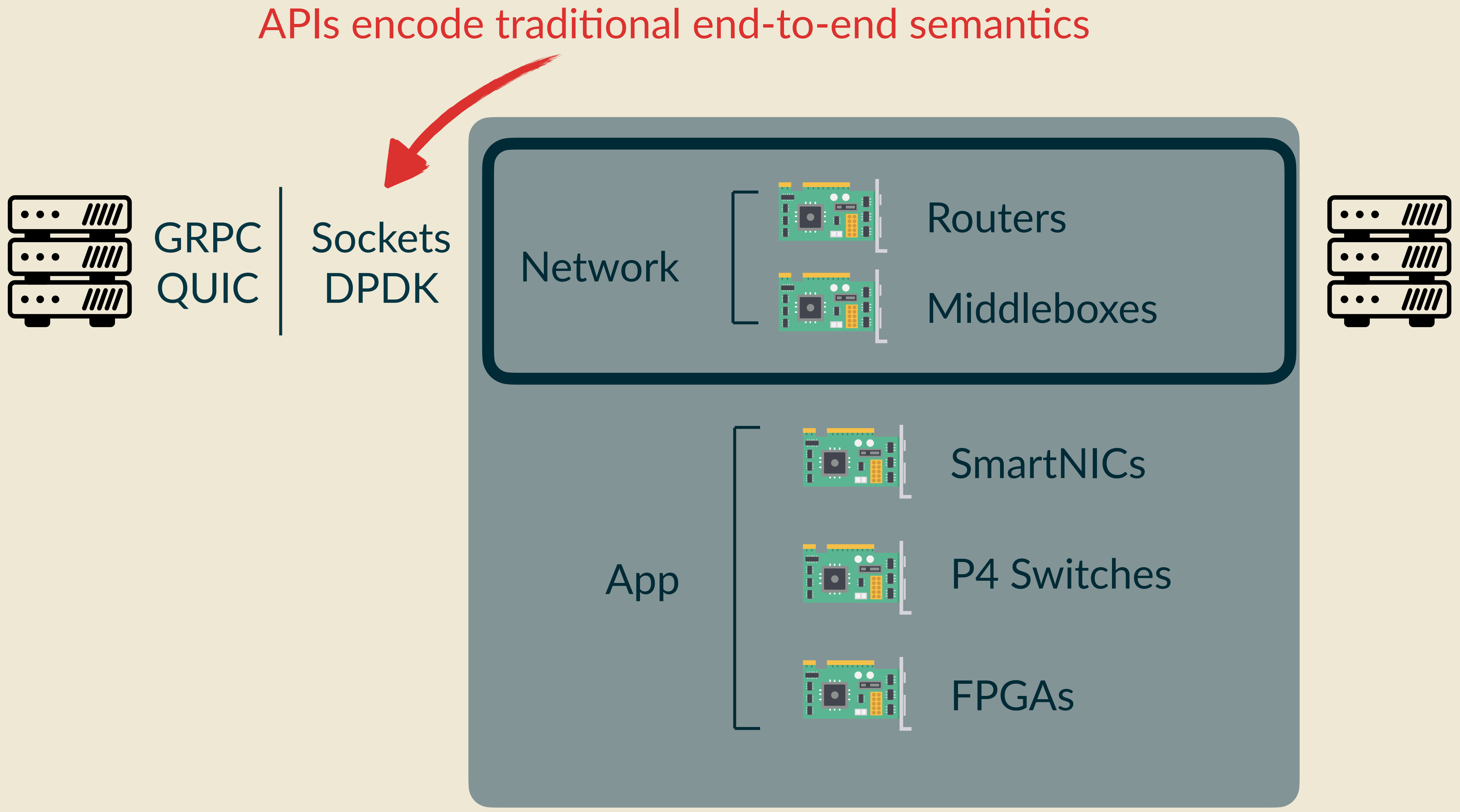


Introducing mcrouter: A memcached protocol router for scaling memcached deployments

New Network Infrastructure

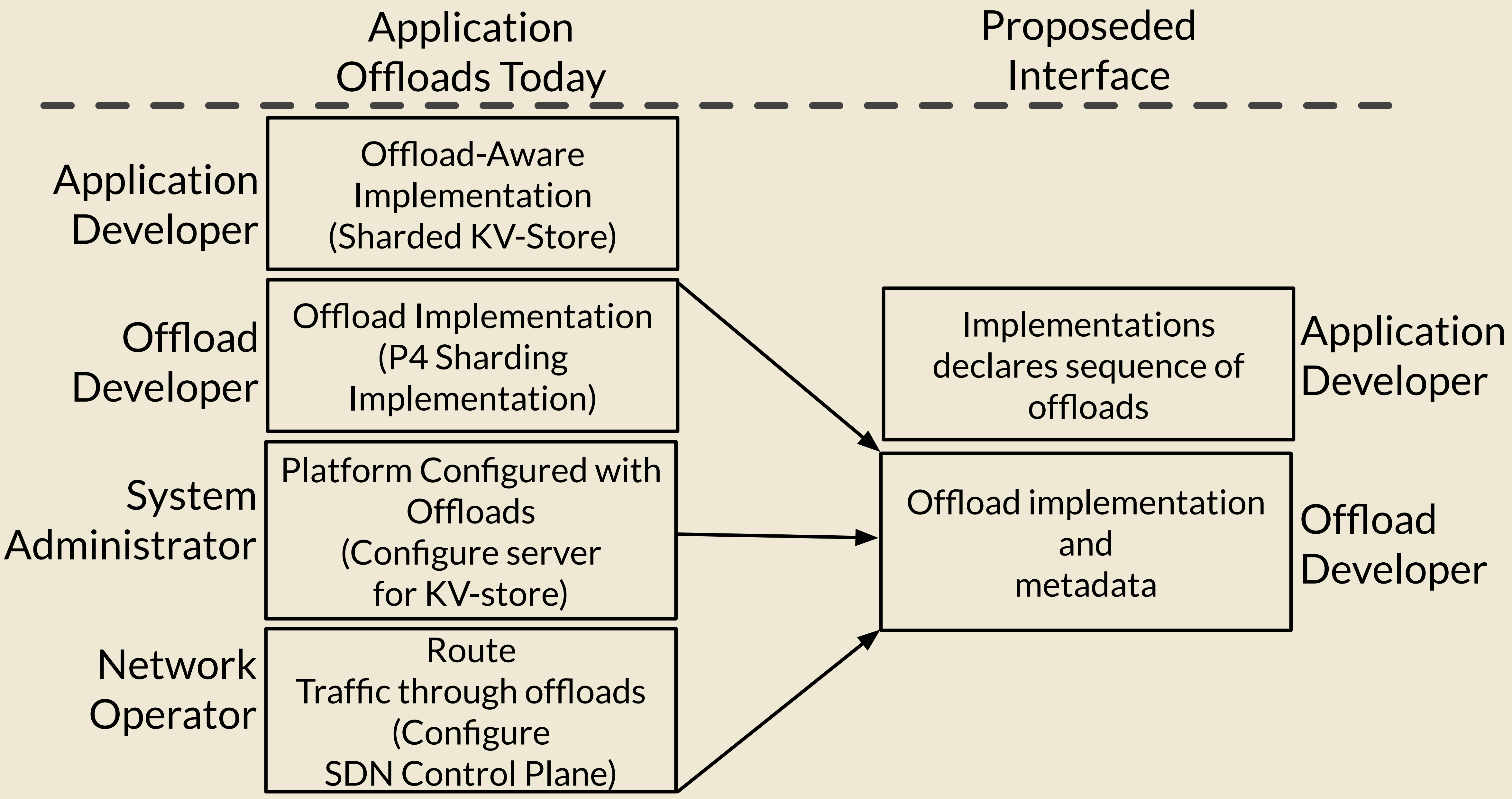

New: Network devices with programmable dataplane
Network
Routers
Middleboxes
App
SmartNICs
P4 Switches
FPGAs
New: Application functionality in the network

Architectural API

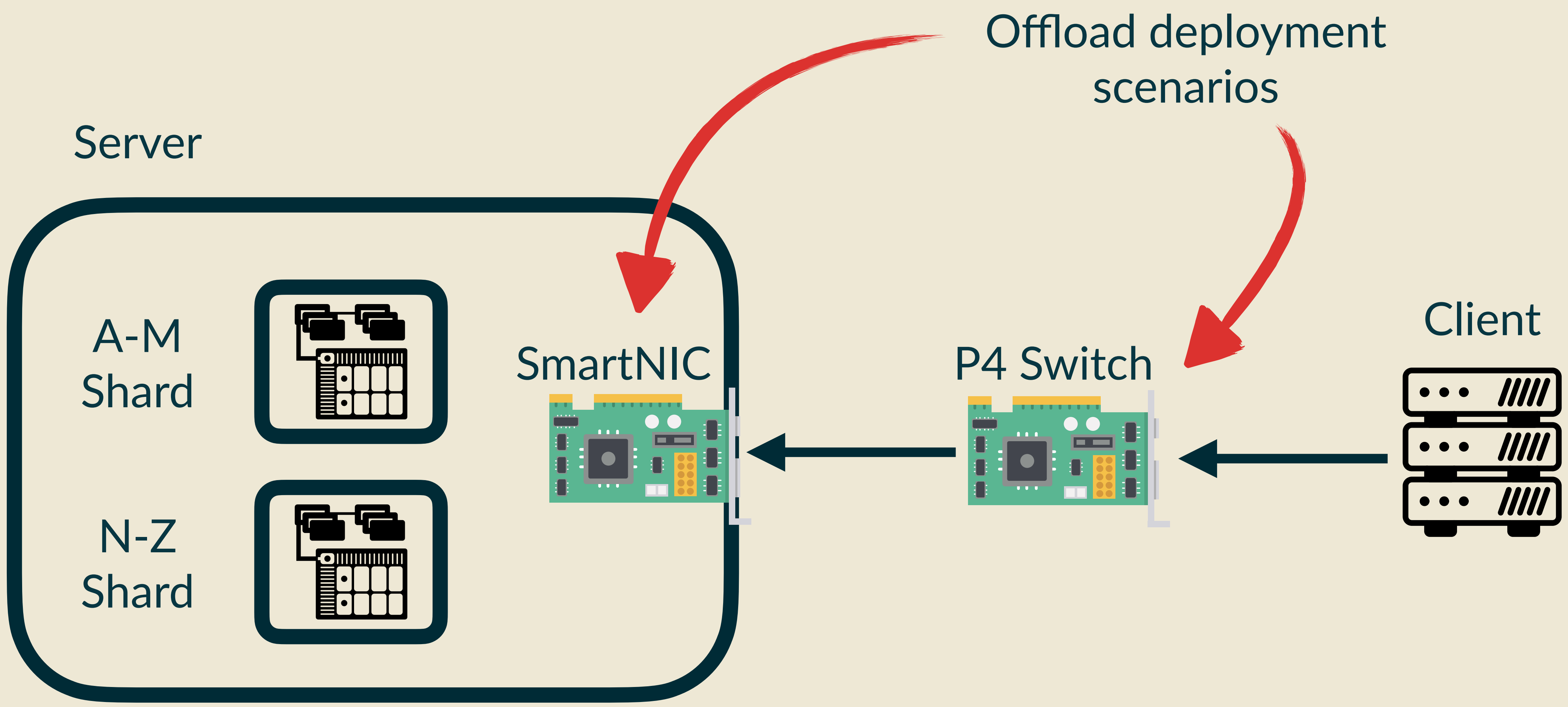


What would an API that encoded application offload semantics look like?

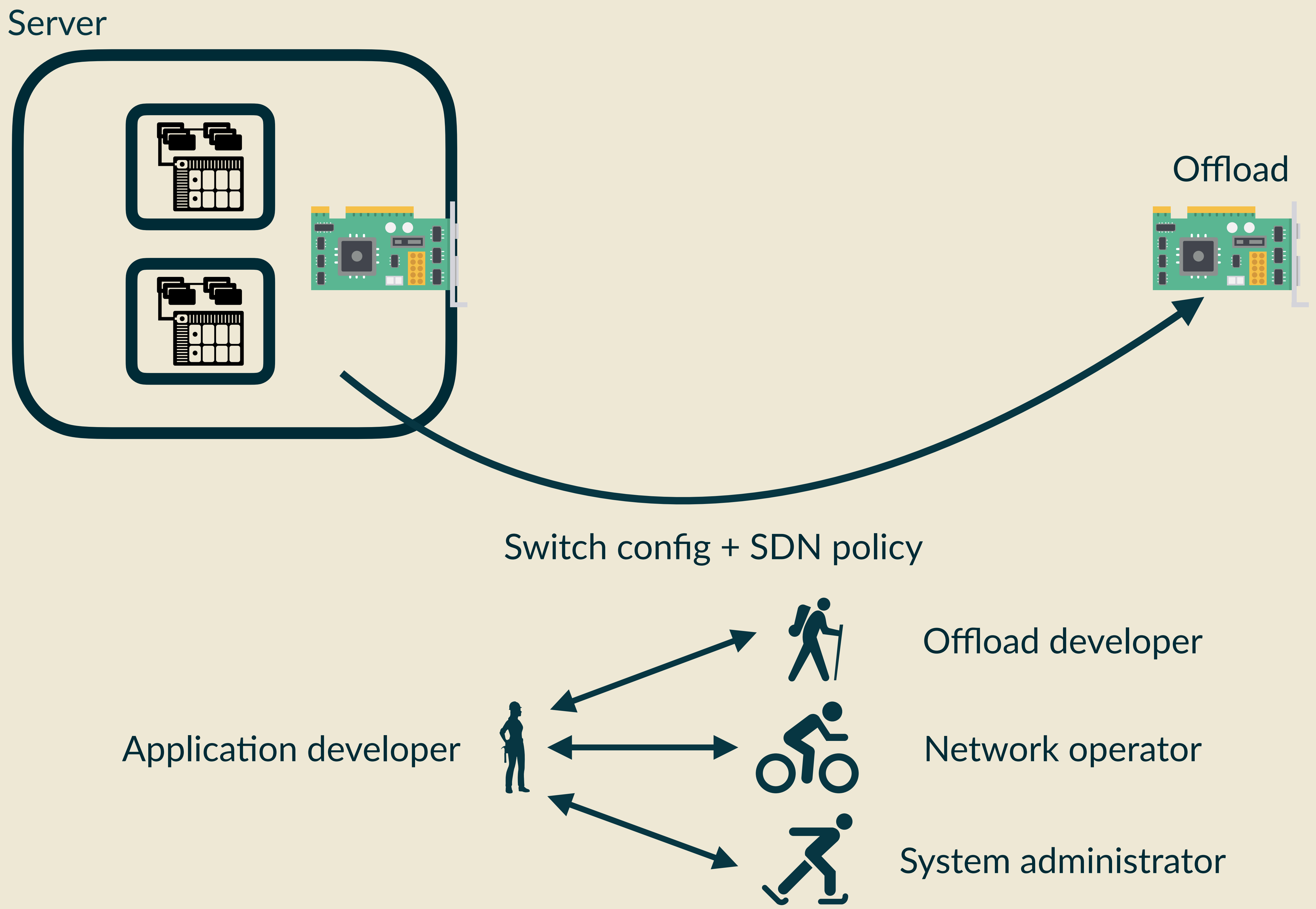
Want: An Interface


Application Offloads Today
Proposeded Interface
Application Developer
Offload-Aware Implementation (Sharded KV-Store)
Offload Developer
Offload Implementation (P4 Sharding Implementation)
System Administrator
Platform Configured with Offloads (Configure server for KV-store)
Network Operator
Route Traffic through offloads (Configure SDN Control Plane)
Implementations declares sequence of offloads
Application Developer
Offload implementation and metadata
Offload Developer

Offload Deployment



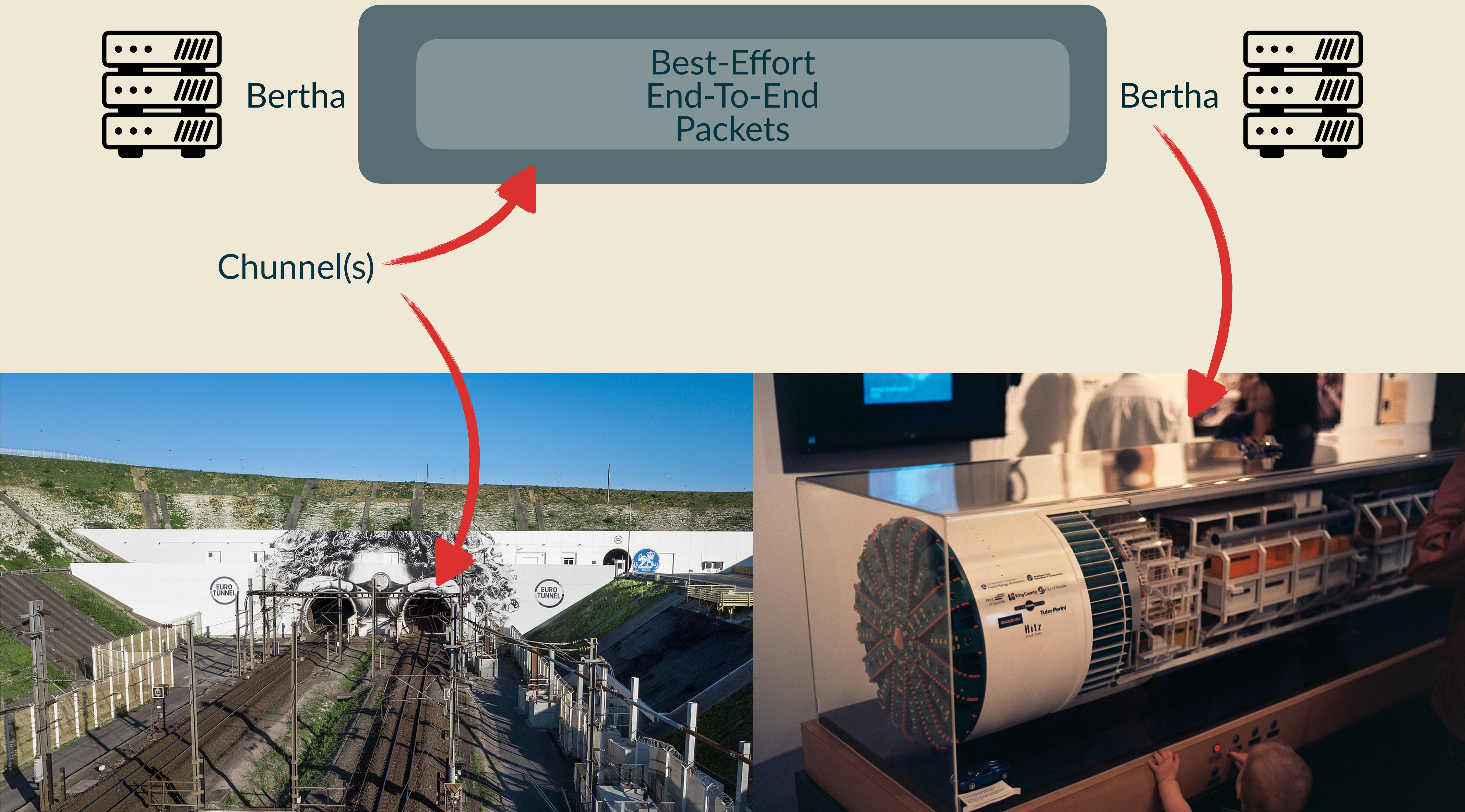
Out-of-Band Coordination



What would an API that encoded application offload semantics look like?

Our Answer: Chunnels + Bertha

Chunnels

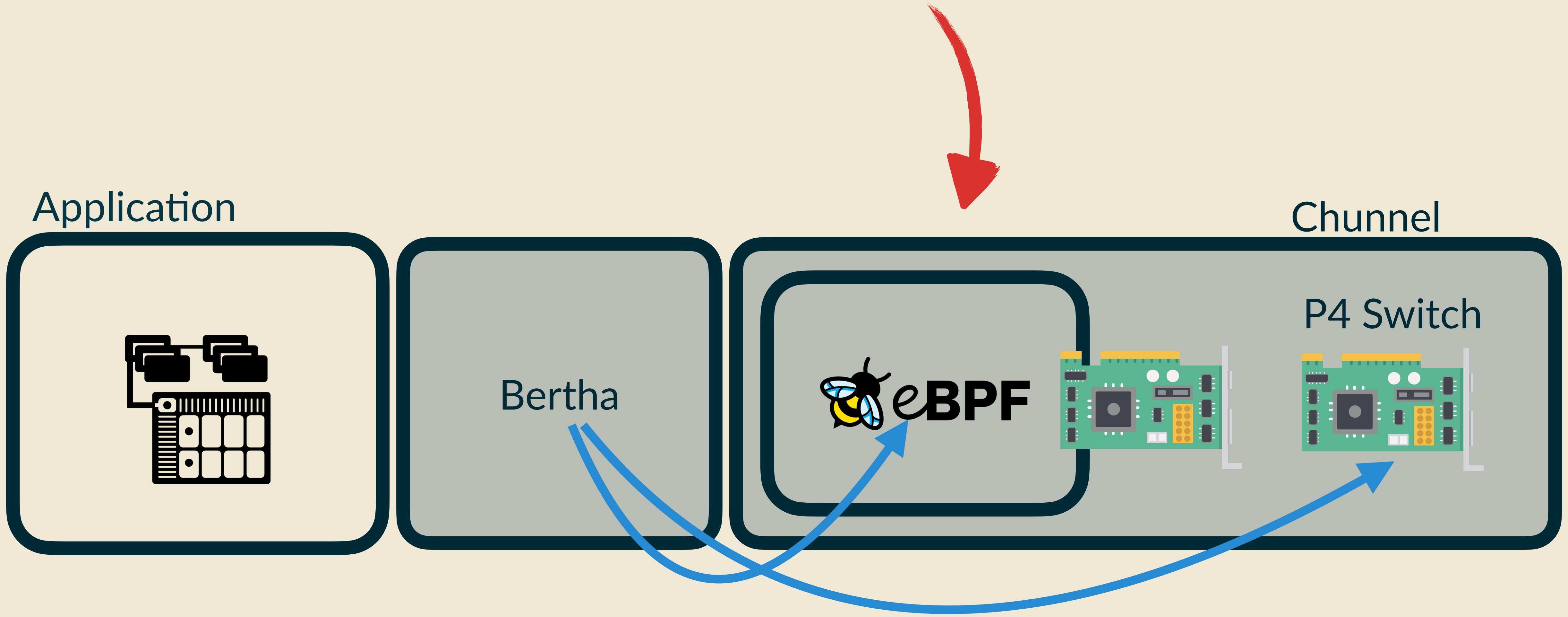


https://upload.wikimedia.org/wikipedia/commons/c/cb/Fresque_25_ans_Eurotunnel.jpg

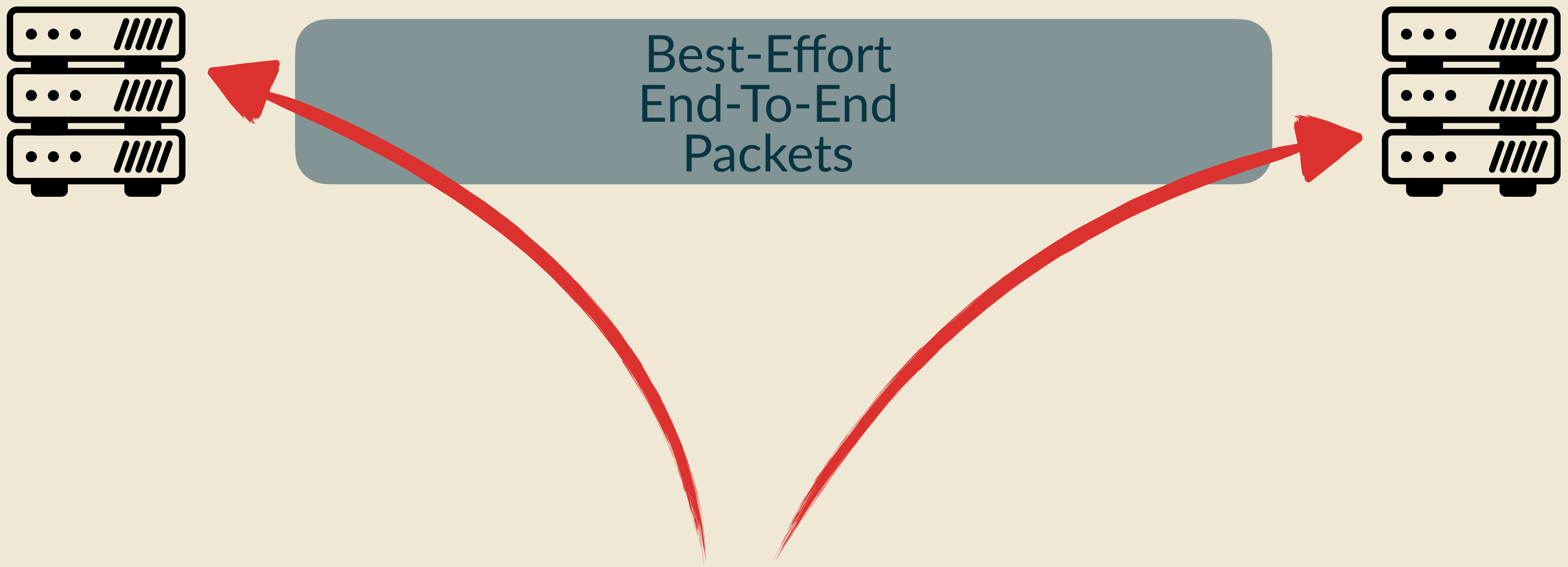
https://en.wikipedia.org/wiki/File:Front_of_model_of_tunnel_boring_machine_at_Milepost_31.JPG

Chunnels

From the application's perspective,
in-machine and in-network offloads are the same



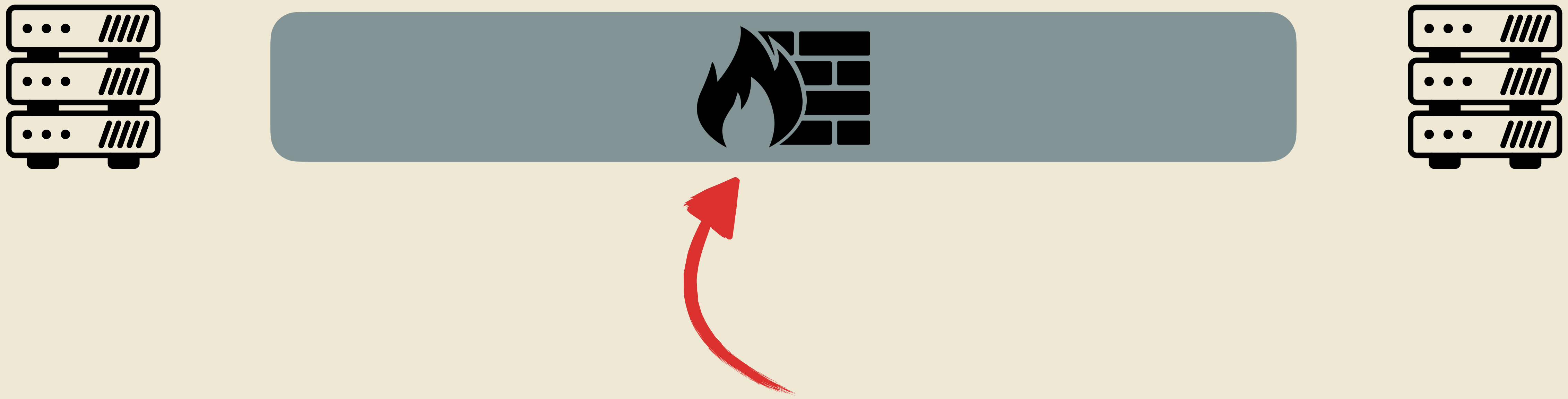
Chunnel Properties



Fallback: Functionality implementable by end-host application software

Chunnel Properties

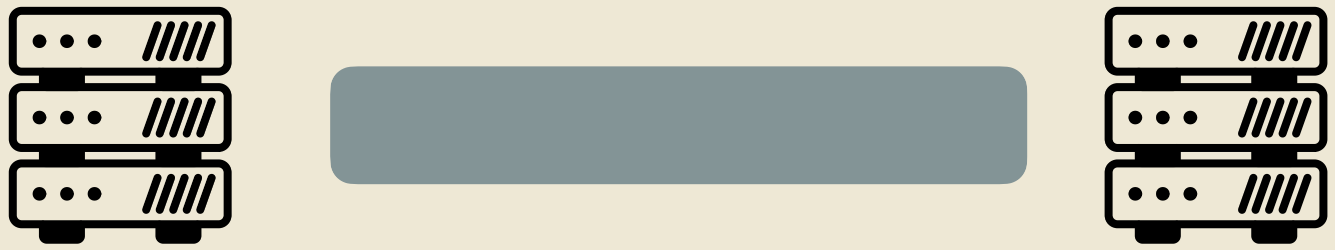
Fallback: Portability



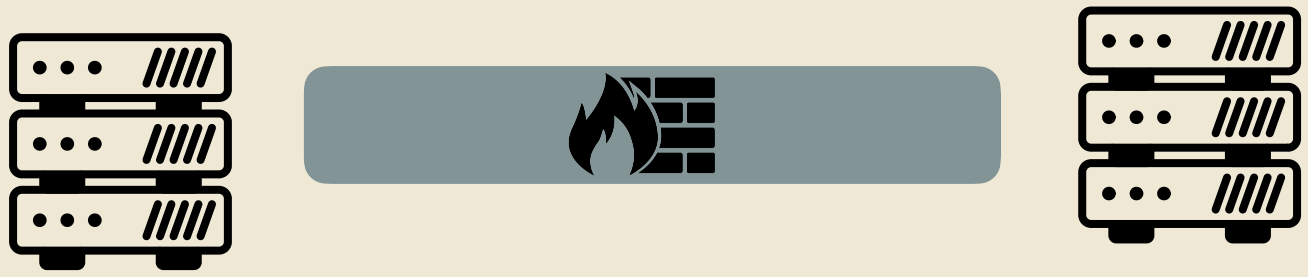
Application Relevance

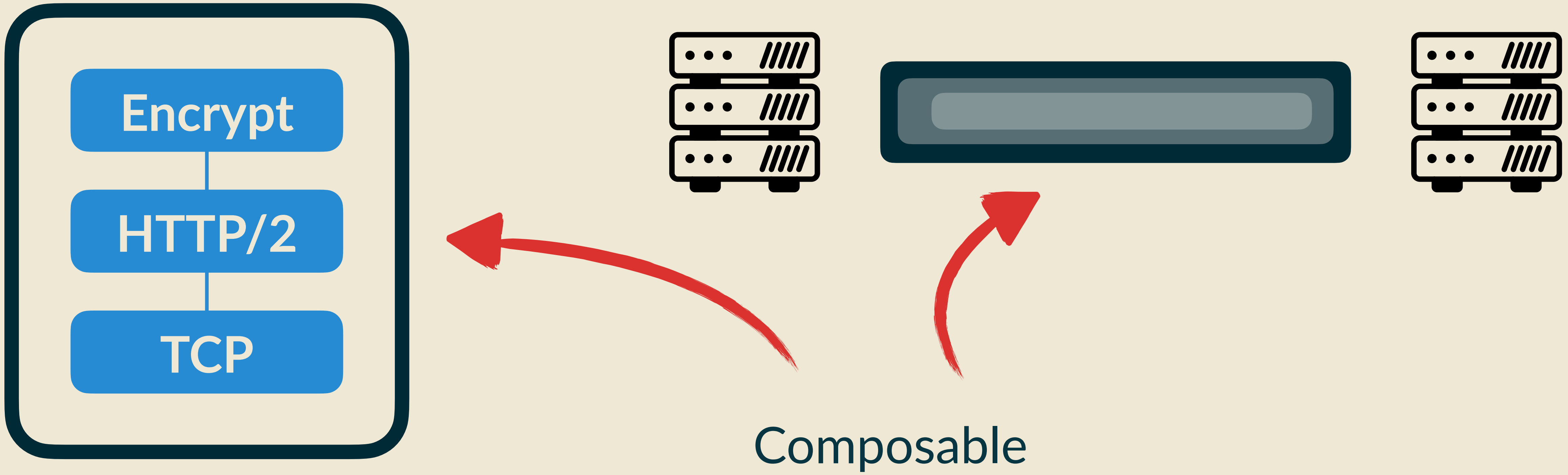
Chunnel Properties

Fallback: Portability



Application relevance: Safety





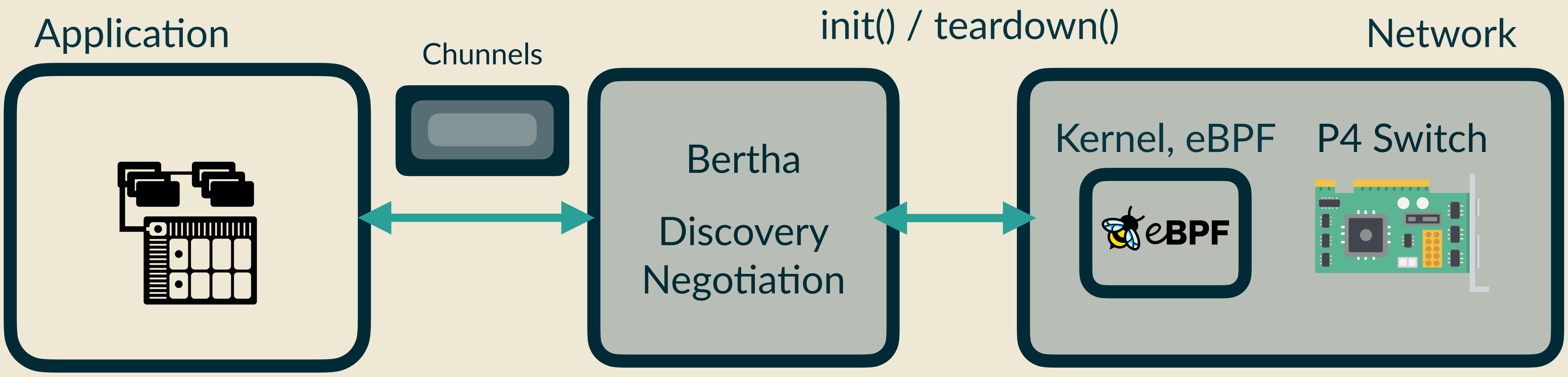
Composable

Bertha Connection

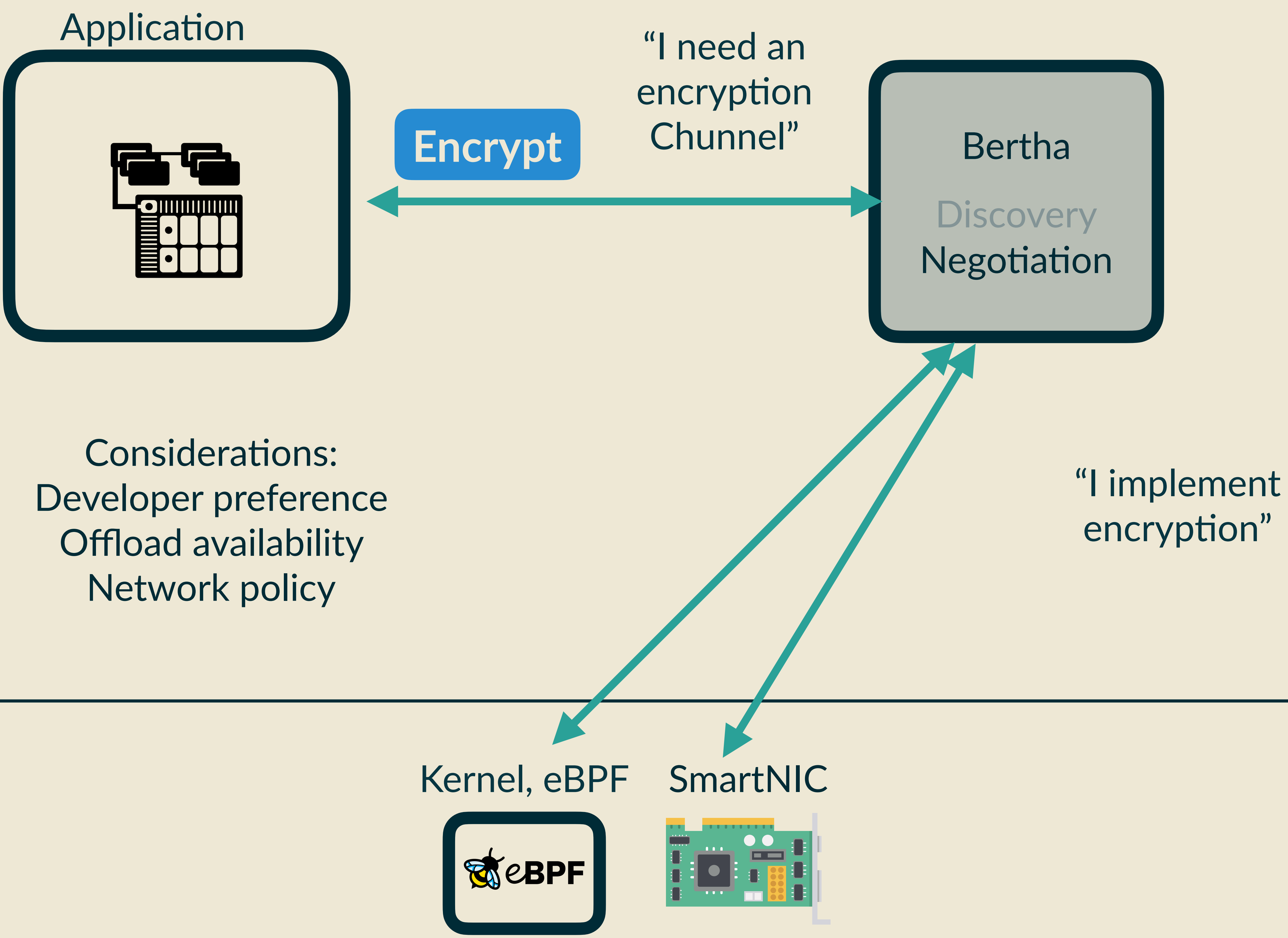
Bertha Roles

Discovery: Figure out which Chunnel implementations are available (e.g., eBPF, Kernel)

Negotiation: Decide which implementation to use.



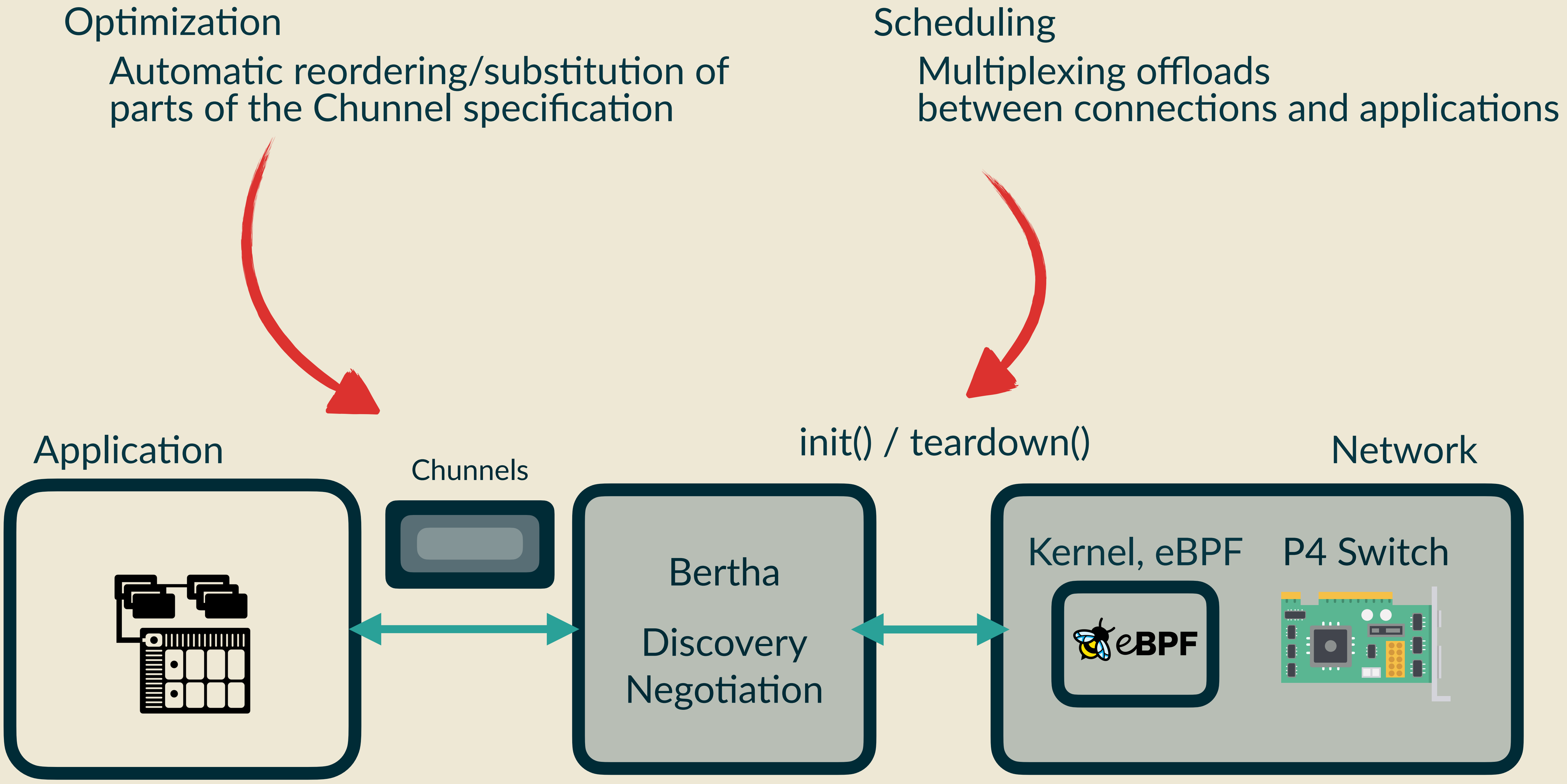
Negotiation



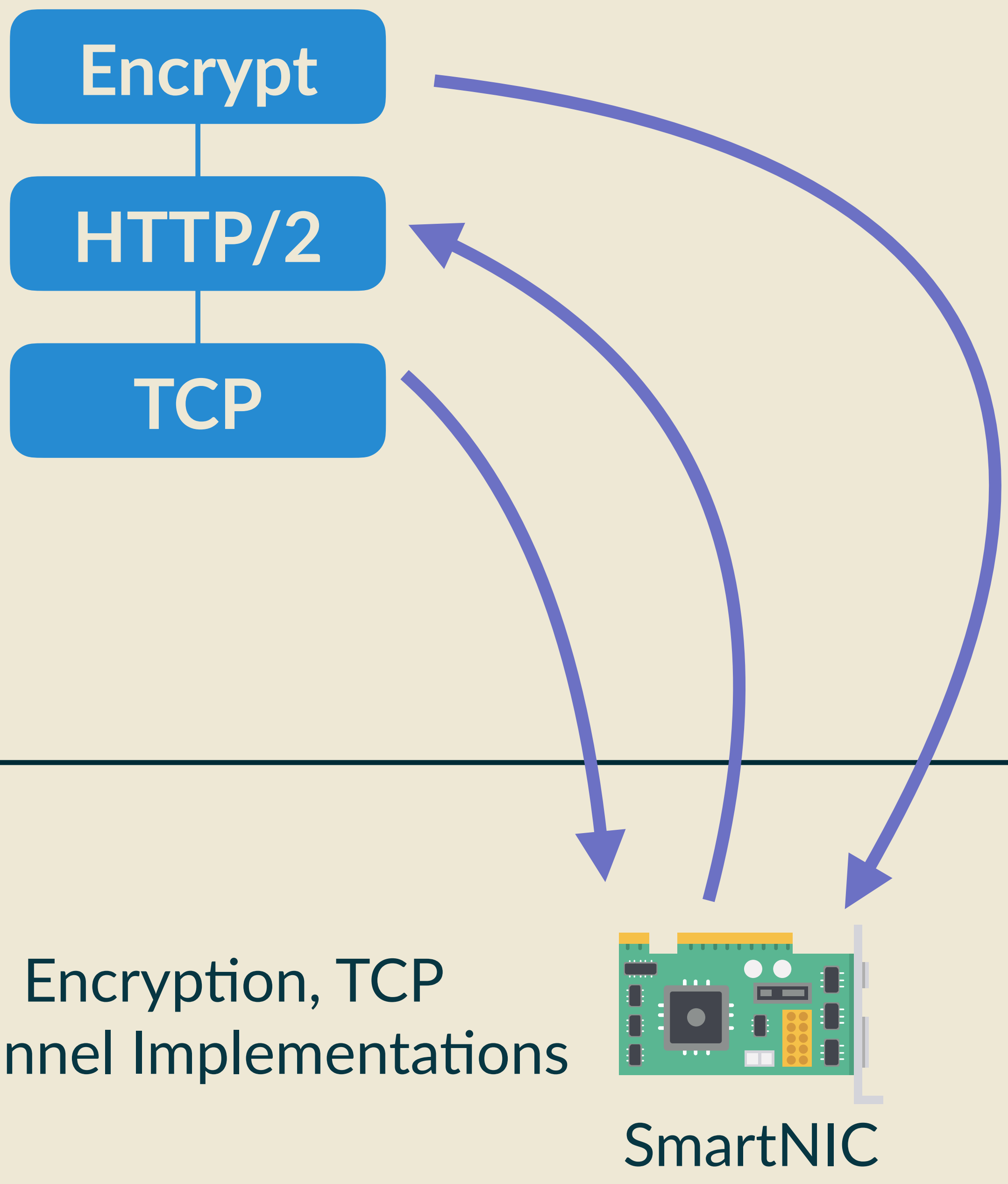
Considerations:
Developer preference
Offload availability
Network policy

Cool Implications

Research Directions


Optimization
Automatic reordering/substitution of parts of the Chunnel specification
Scheduling
Multiplexing offloads between connections and applications
Application
Chunnels
init() / teardown()
Network
Bertha
Discovery
Negotiation
Kernel, eBPF
eBPF
P4 Switch

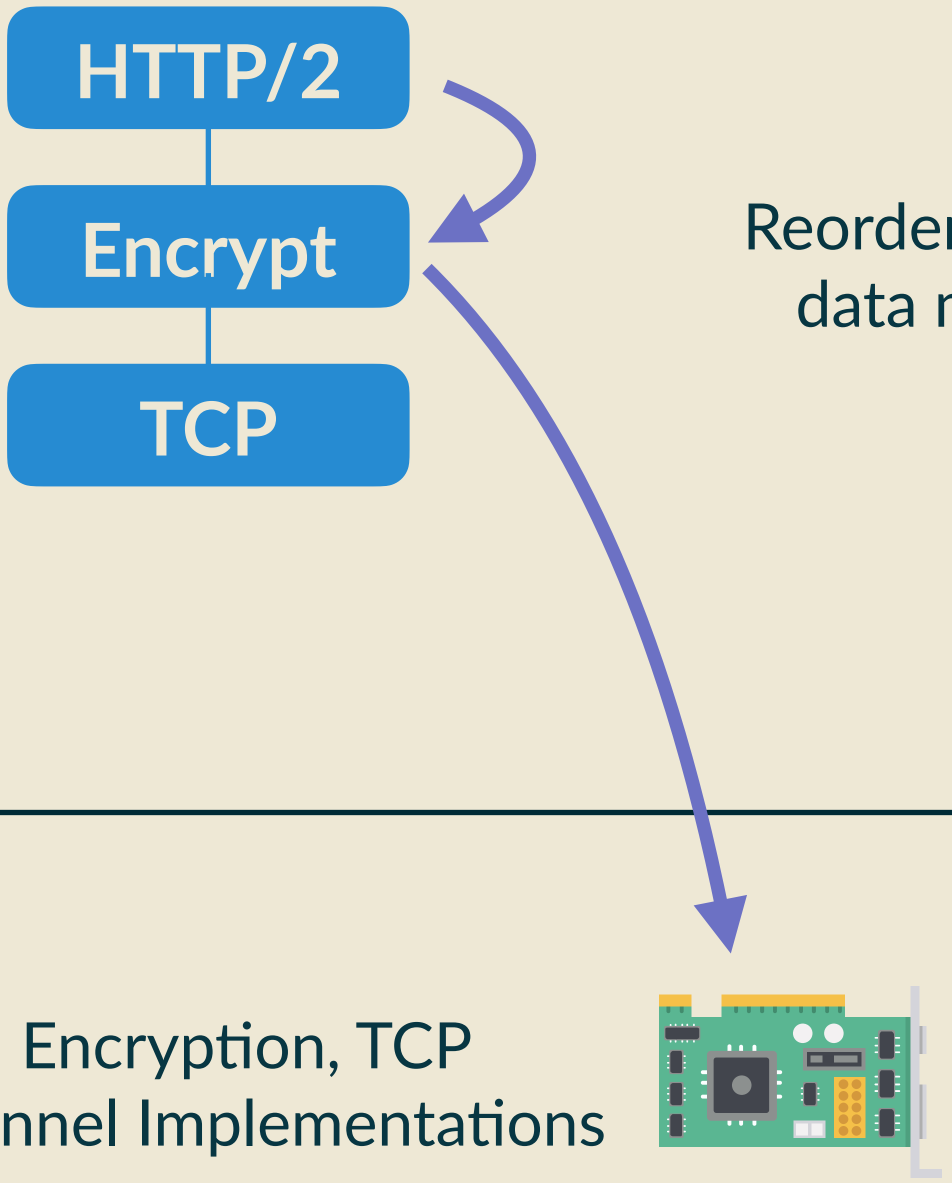
Chunnel Optimization



To use these offloads,
must to traverse PCIe 3x

Encryption, TCP
Chunnel Implementations

Chunnel Optimization



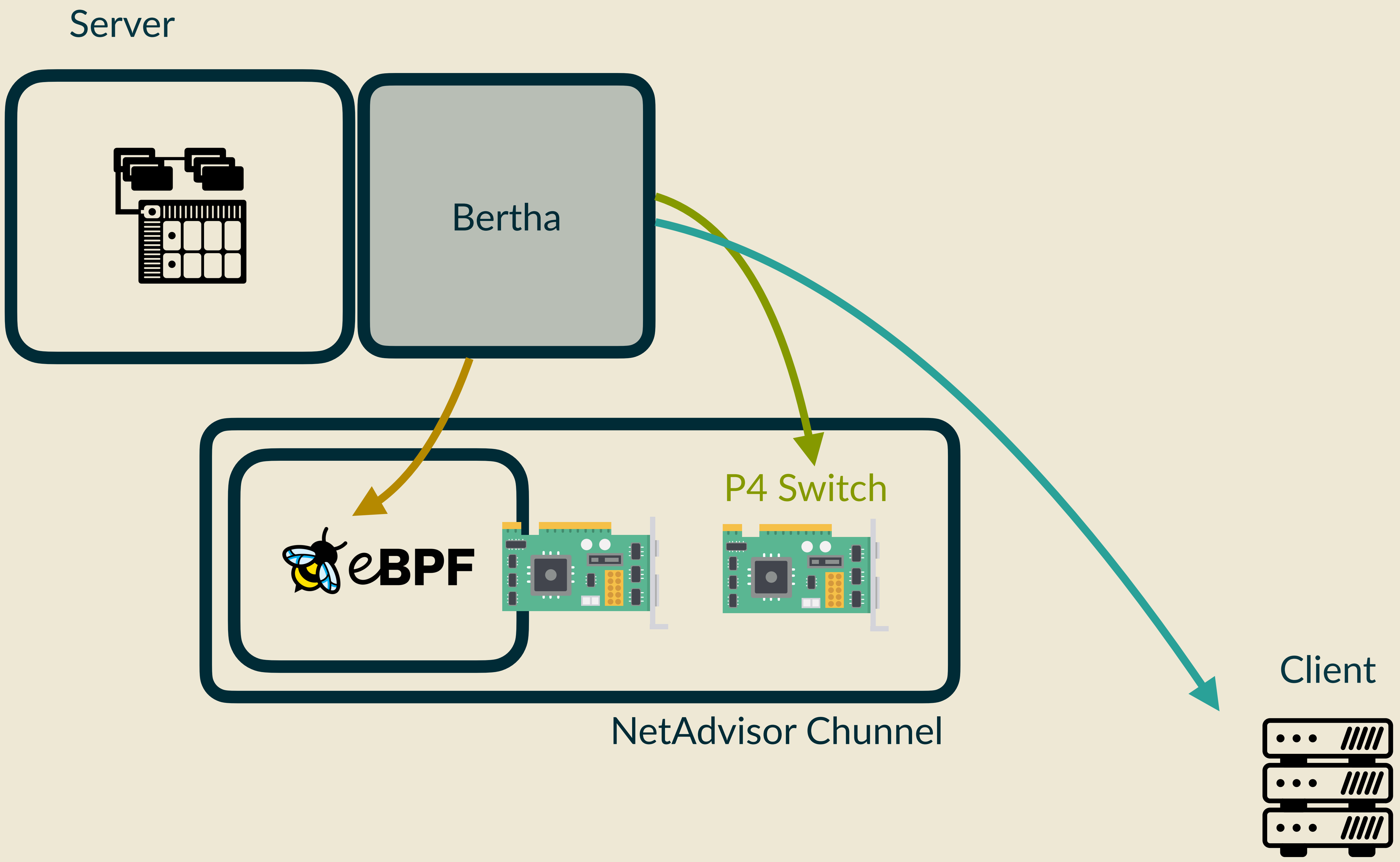
Reordering reduces data movement

Similar to optimizations in Weld, TensorFlow, ONNX

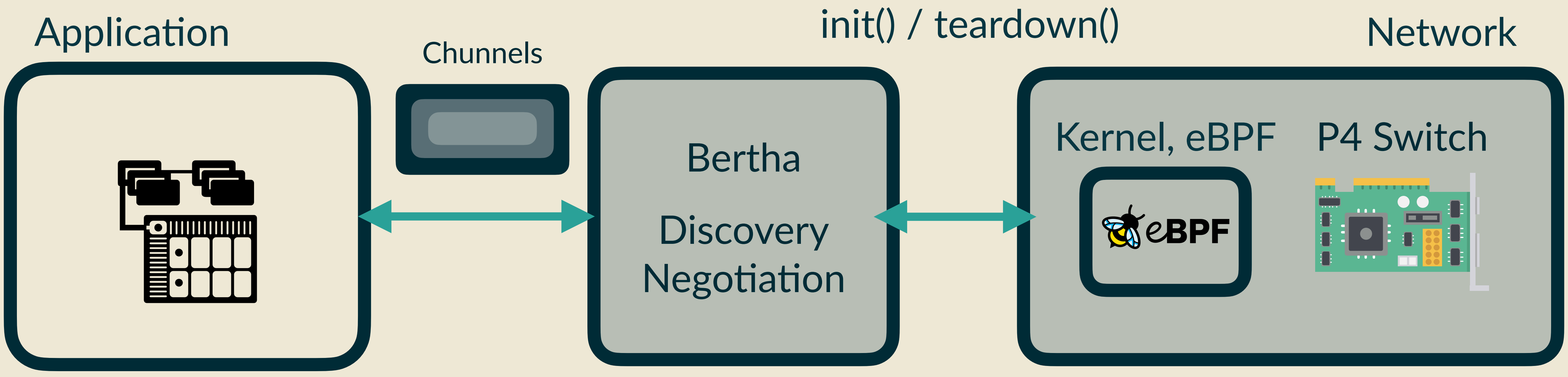
Encryption, TCP
Chunnel Implementations

SmartNIC

Client Push



End



Contact: akshayn@mit.edu