OAK RIDGE National Laboratory

Named Data Networking

Ryan Adamson

National Center for Computational Sciences (NCCS)

Oak Ridge National Laboratory (ORNL)

Advanced Networking (CSC 6730) – Fall 2021

Learning Objectives

After this presentation, you will be able to:

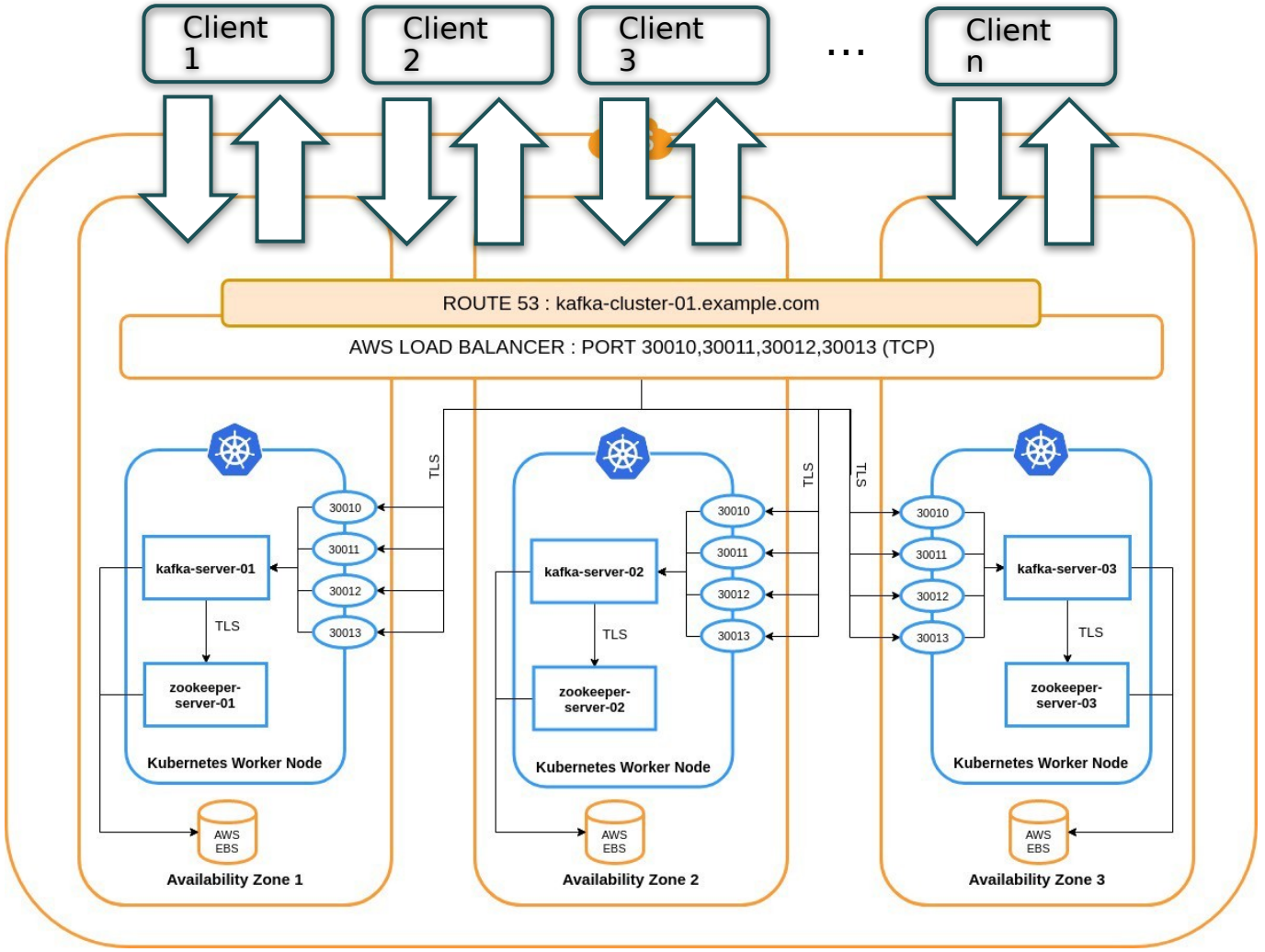
1. Describe key Named Data Networking (NDN) concepts
2. Grok high-level containerization best practices in operations (Docker/Kubernetes)
3. Understand high-level OLCF telemetry and logging needs and best practices (Kafka publish/subscribe model)
4. Analyze the benefits and drawbacks of NDN on cloud-like telemetry deployments (telemetry streaming/failover/clients)

Problem Overview

- Oak Ridge Leadership Computing Facility (OLCF) runs extreme scale systems and has built an analytics and monitoring platform
- This platform consists of Apache Kafka and Elastic applications, deployed on top of a Platform as a Service (PaaS) Kubernetes infrastructure
- In general, the platform is stable, but excess communication due to client connections, bookkeeping, lookups, redundancy, and rebuilding data structures after failure leads to problems

Problem Overview

1. Kafka Application State
2. Container Networking
3. PaaS Load Balancing
4. Client Connection State
5. Massive amounts of real-time data (> 2TB/day)
6. Rebalancing after Failure



Research Question: *"What are the benefits and drawbacks to replacing IP with NDN for streaming analytics and monitoring pipelines?"*

OAK RIDGE National Laboratory

Papers

- L. Zhang, et al., **Named Data Networking**", ACM SIGCOMM CCR, 2014
- Cheng Yi et. al., **A case for stateful forwarding plane**. Comput. Commun. 36, 7 (April, 2013), 779–791
- Peter GUSEV et al., **Real-Time Streaming Data Delivery over Named Data Networking**. In:IEICE Transactions on Communications E99.B (Mai 2016), S. 974–991.DOI:10.1587/transcom.2015AMI0002
- Chengyu Fan, et al., **Managing scientific data with named data networking**. In *Proceedings of the Fifth International Workshop on Network-Aware Data Management* (*NDM '15*). Association for Computing Machinery, New York, NY, USA, Article 1, 1–7. DOI:https://doi.org/10.1145/2832099.2832100

Named Data Networking

- **Research Question:** *"What are the benefits and drawbacks to replacing IP with NDN for streaming analytics and monitoring pipelines?"*

- *"Van Jacobsen: A new way to look at Networking"*

Architectural Differences between IP and NDN

- Instead of point-to-point addressing, NDN addressing is based on data!
  - Every packet has an identifying Name, not IP address
- Motivated by consumption changes in the way systems consume ‘data’ not ‘connections’
- Architecture
  - Narrow waist, much like IP

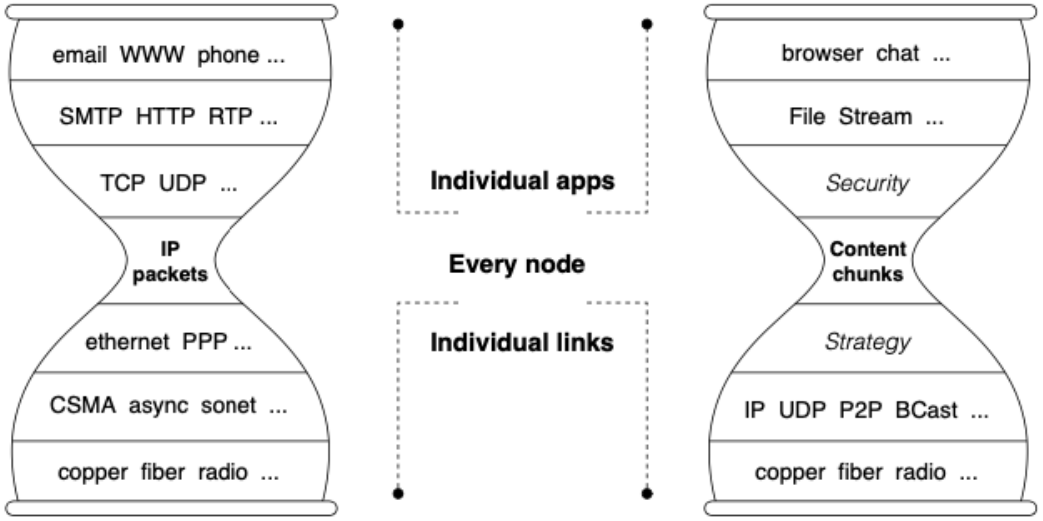


Figure 1: The main building blocks of the NDN architecture are named content chunks, in contrast to the IP architecture’s fundamental unit of communication, which is an end-to-end channel between two end endpoints identified by IP addresses.

Architectural Differences between IP and NDN

- There are two types of packets: Interest and Data
- **Interest packets** are created by a consumer interested in a piece of data
- **Data packets** are returned from the network, along the same path the interest packets took
- Contrast this with IP packets, where each forwarder has simple FIB

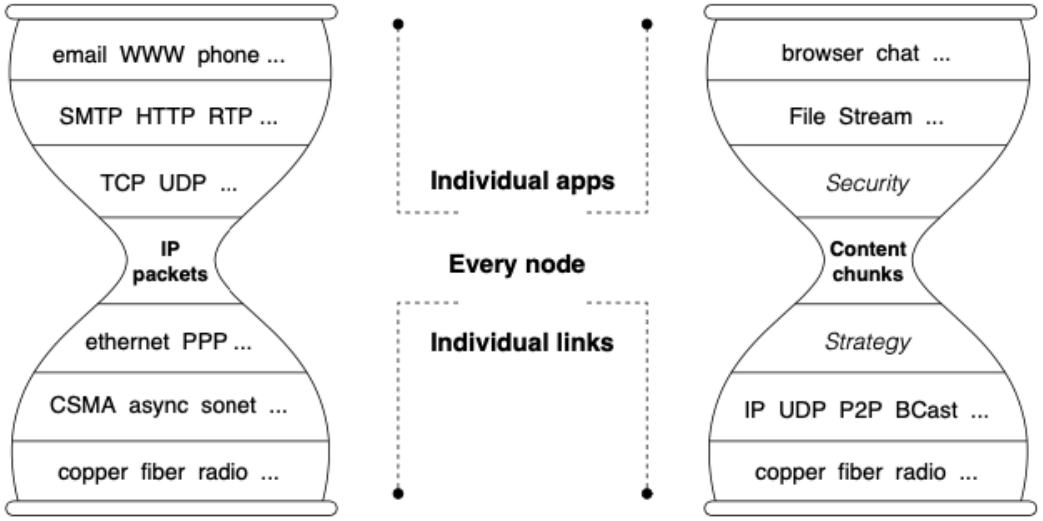


Figure 1: The main building blocks of the NDN architecture are named content chunks, in contrast to the IP architecture's fundamental unit of communication, which is an end-to-end channel between two end endpoints identified by IP addresses.

Packet Forwarding and Return

- NDN nodes maintain **state**
  - Pending Interest Table (PIT)
  - Track unsatisfied forwarded interests to return data packets over same path
  - Forwarding / Routing is similar to IP's Forwarding Information Base (FIB), but allow for much more complex protocols
- NDN nodes keep **cache**
  - Content store can serve future interests that match preventing unnecessary upstream congestion

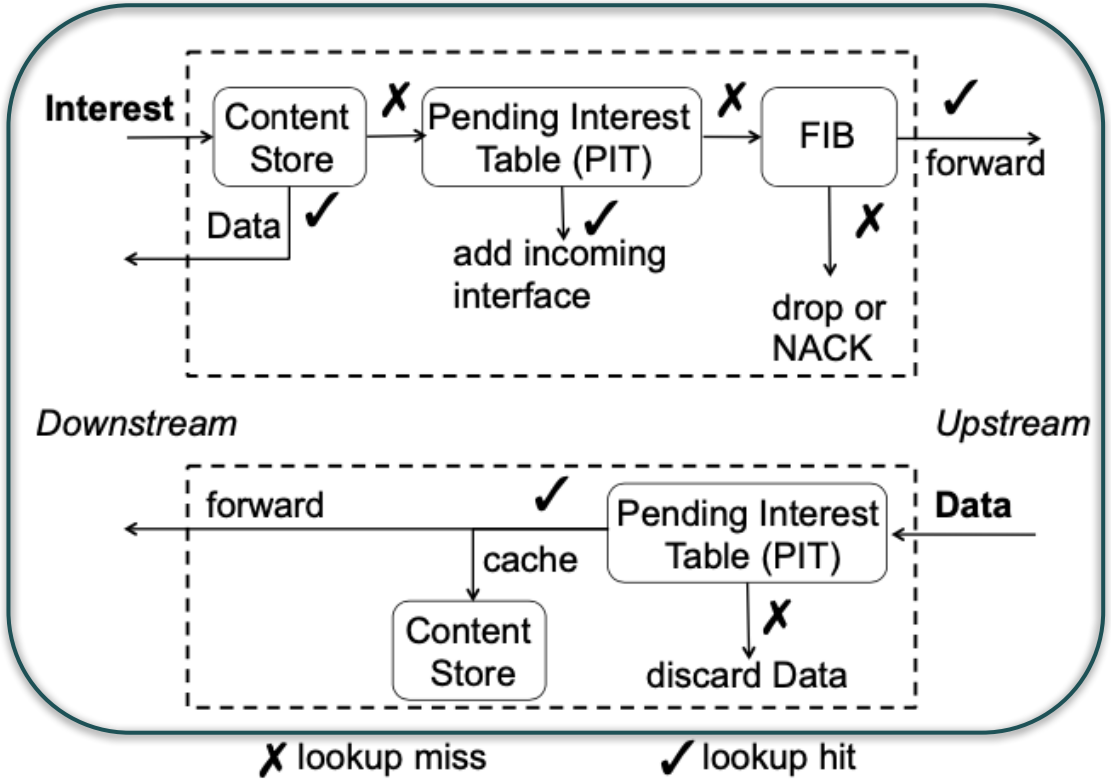


Figure 3: Forwarding Process at an NDN Node.

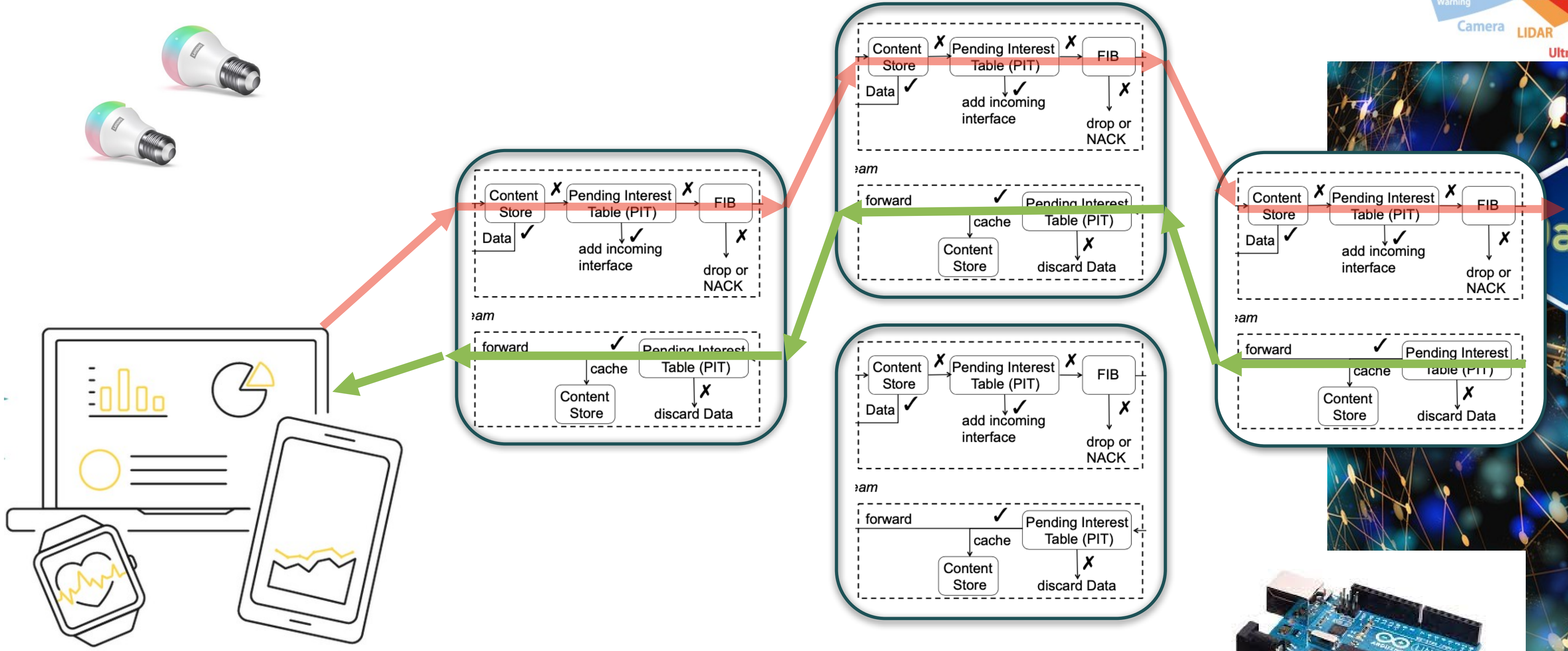
Second-order properties

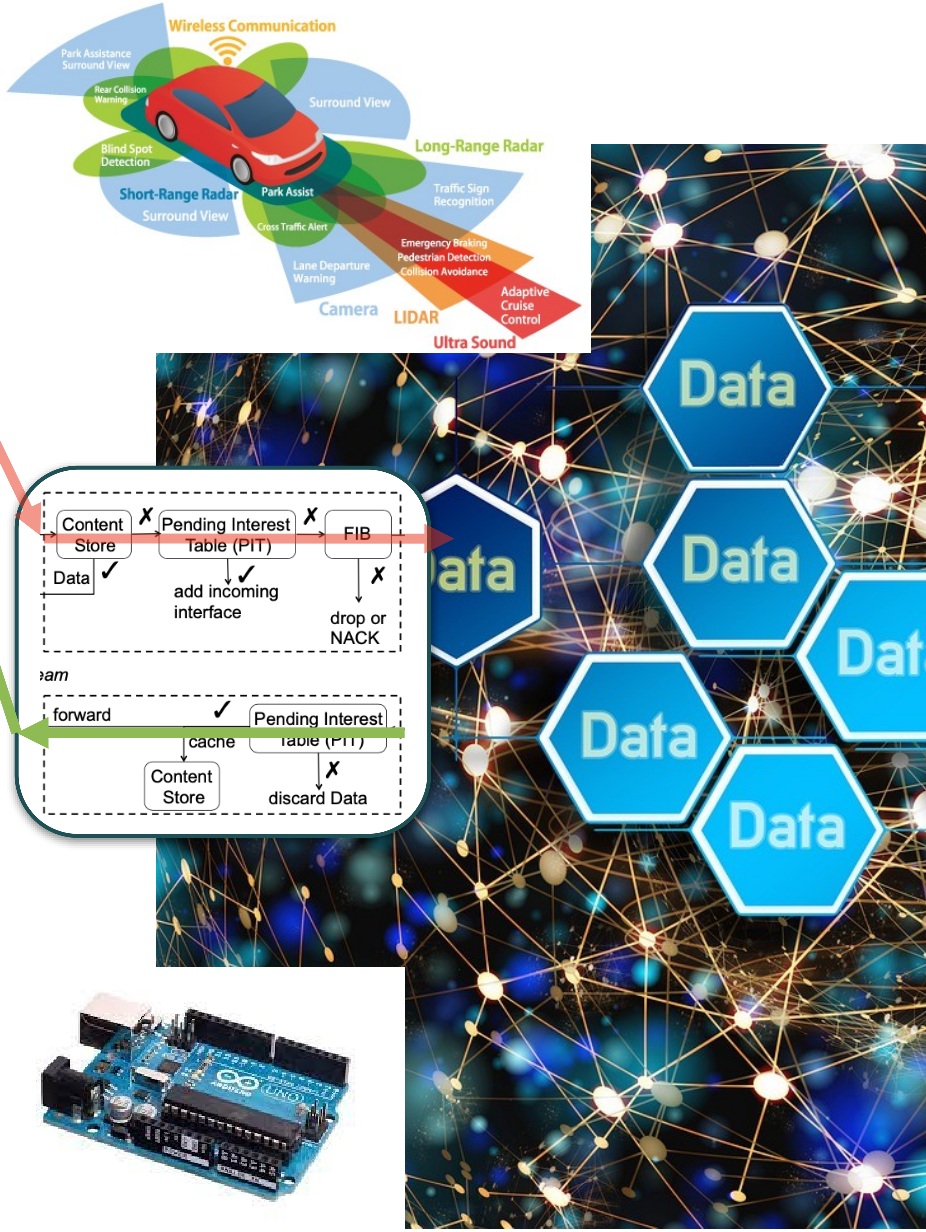
- Routing protocols are ‘similar’ to IP. However:
  - Nonces are added to Interest packets which prevent Interest packet looping
  - Data packets always follow the Interest packet path which prevents Data packet looping
  - NDN Nodes can ‘negatively ack’ or ’NACK’ to the previous node if an Interest will not be fulfilled (due to timeout, congestion, or other issues)
- NDN provides a layer of ‘Security’ by default
  - Signatures of Data packets by producers give some trustworthiness
  - Caching on untrusted nodes is still safe because consumers can still trust these cached Data packets by validating signatures
  - Encryption of data within Data packet is allowed as well

Second-order properties

- Addresses are Names
  - Rather than IP addresses, human-readable names can be given to Data packets, and these names, once published, are ‘immutable’
  - Namespaces can be carved up similar to domain registration (i.e. /amazon.com/tv_series/mr.robot/S1/E10/10m50s/1/3)
- Hierarchical naming can be anything
  - Naming standards are defined by the application that is consuming or producing the data packets
  - There also exist local namespaces such as /localhost for internal use
- Interest and Data packet flags also exist and may vary by application (freshness, local only, other things to help consumers and producers of data)

End to End Data Transfer Example

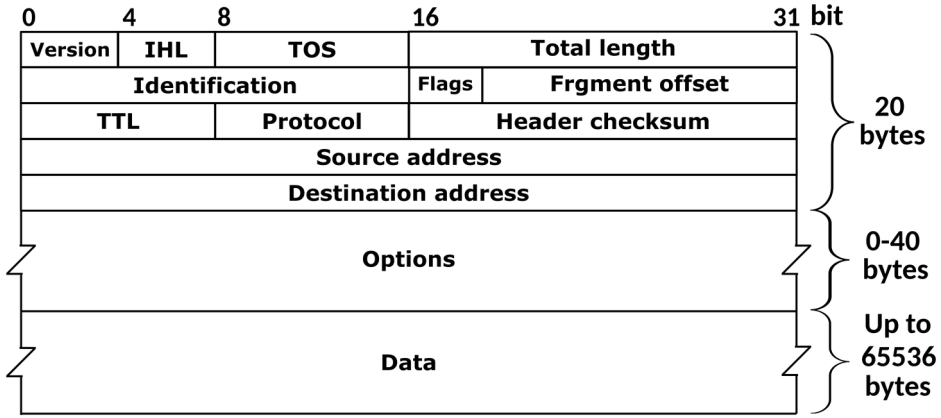




OAK RIDGE National Laboratory

IP and NDN Packet Comparison

- IPv4 Packets have rigorous and specific fields for fast processing in hardware



- NDN Packets utilize TLV, or type-length-value fields for flexible but more compute intensive processing

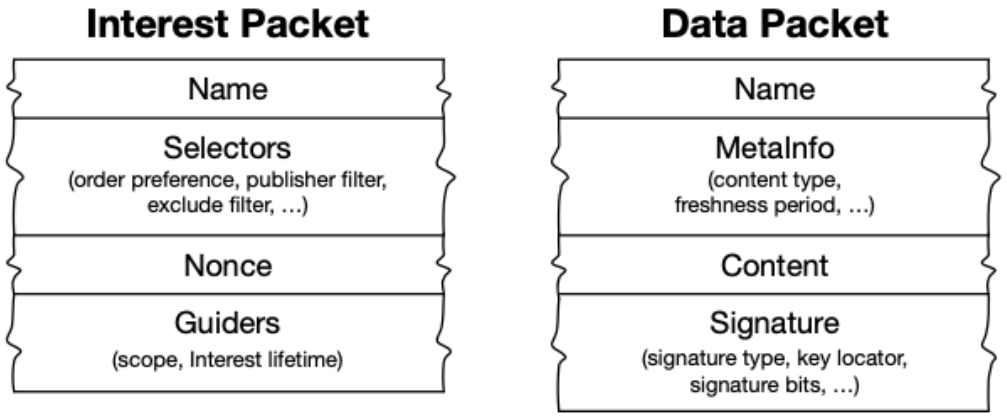


Figure 2: Packets in the NDN Architecture.

Note, this is not the most current version of the NDN Packet Format Specification
Version 0.3 information is here: https://named-data.net/doc/NDN-packet-spec/current/

Named Data Networking (Considerations)

- Names/namespaces
  - Open question as to how applications should create names since flexibility is explicitly granted
  - Each NDN Data packet is immutable, so once a name is used could the producer publish another data packet with same name?
  - NDN name registrars can provide root prefixes, are domain squatting, stealing, etc still an issue?
- Security considerations
  - PKI/Chain of Trust is ‘hard’, but future work could make this easier and enable use of SSO to generate these keys with a short lifetime
  - Use of names may leak sensitive information
  - Encryption / Signing is more computationally expensive, but can be offloaded in hardware in some cases

Named Data Networking (Considerations)

- Data caching
  - Strategic Caching will reduce network congestion
  - Big, local, network cache devices may be useful
- Testbeds and experimental NDN software stacks exist but widespread adoption is still in progress
  - Much like the early days of SDN and the Internet
  - Use cases, consensus, and vendor buy-in are still being developed
- Overlays of NDN on IP are possible though
  - Overlays over TCP/IP or UDP are doable but not performant (TCP max packet size must be < MTU of 1500)
  - Overlays mean that we lose control of routing

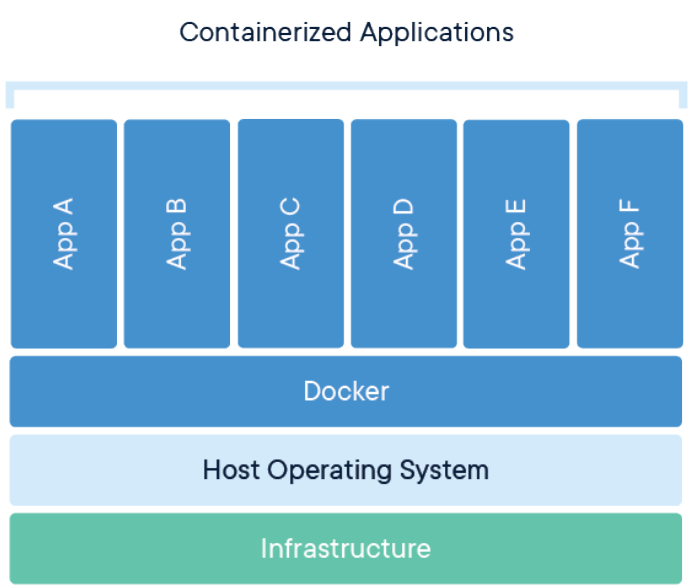
Container Orchestration

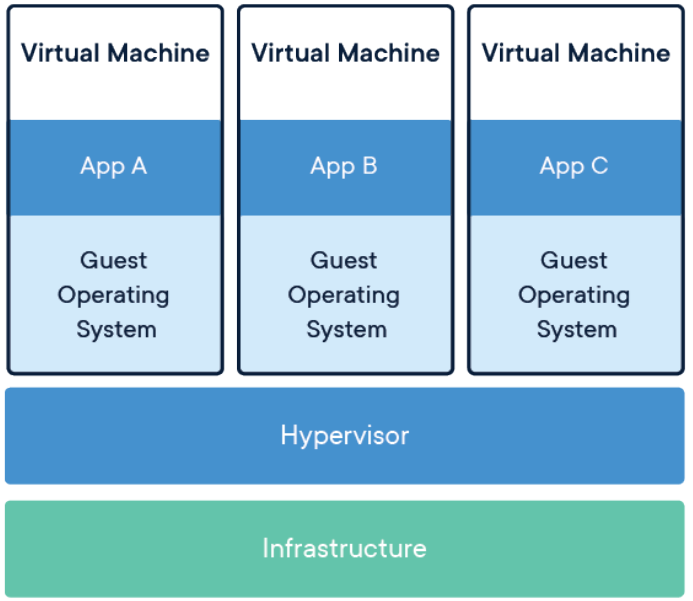
- **Research Question:** *"What are the benefits and drawbacks to replacing IP with NDN for streaming analytics and monitoring pipelines?"*

- *"Google SRE: Avoid touching anything other than code"*

What are Containers? Why should we use them?

- A **container** is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.
- A **virtual machine** is the virtualization/emulation of a computer system. Virtual machines are based on computer architectures and provide functionality of a physical computer.



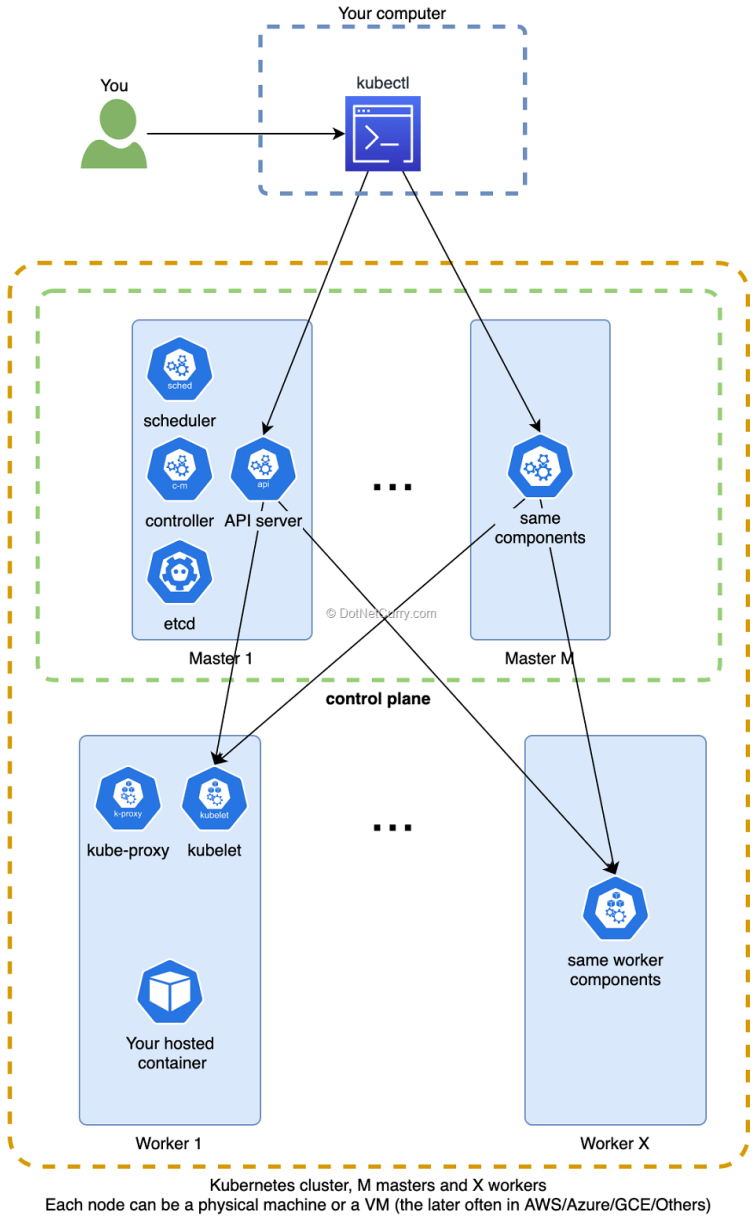


Container Orchestration

- Applications are abstracted and ‘contained’ in containers, so common building blocks of applications are abstracted as well
  - Images and application packaging
  - **Network access**
  - Users
  - Operating System Calls and resource utilization
- Platform as a Service (PaaS) offerings schedule containers as workloads and keep them running/move them to ensure high availability
  - Docker Compose
  - Rancher
  - Kubernetes

Kubernetes as a PaaS

- State is declarative, and is kept by the kuberenetes cluster control plane
  - Applications are farmed out to ‘worker’ nodes
  - Kubelet processes report status
  - Feedback loop in control plane issues commands to implement desired state on worker nodes
- A Pod is the unit of scheduling within Kubernetes
  - Multiple containers, networks, storage attachments, and deployment strategies per pod



Container Orchestration Best Practices

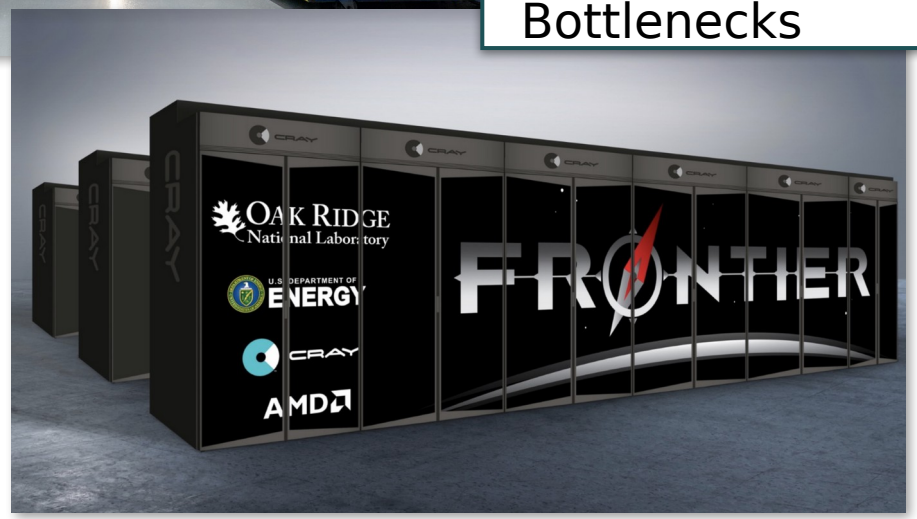
- Treat applications like “Cattle” and not “Pets”
  - Reproducible builds
  - Git-backed configuration for builds and deployment
  - Rapid testing and deployment with CI/CD pipelines
- Applications are deployable without (much?) regard to underlying hardware
  - Memory, CPU, Disk are all abstractions
  - Network addressing is also abstracted
    - “Services” are exposed internal to a Kubernetes cluster, and “routes” are exposed externally. All done via IPTables rule forwarding.

Kafka and Telemetry

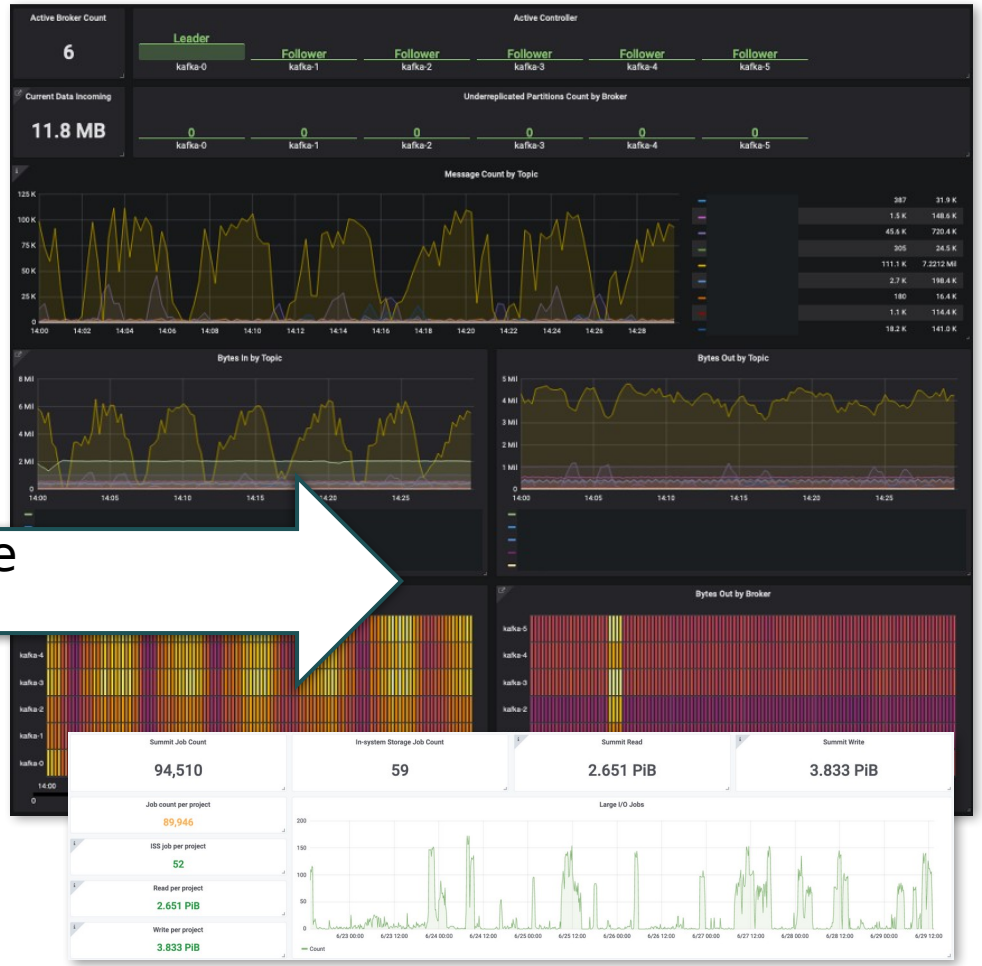
- **Research Question:** *”What are the benefits and drawbacks to replacing IP with NDN for streaming analytics and monitoring pipelines?”*

- *“The data never lies, but it often misleads”*

Hardware and Application Monitoring



Eliminating Performance Bottlenecks

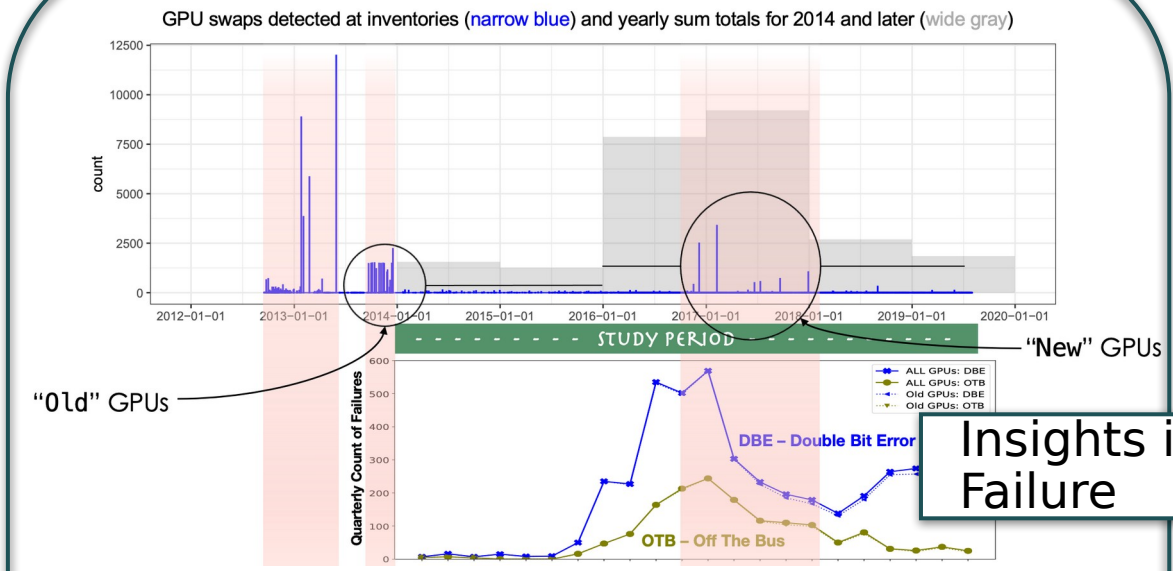


OAK RIDGE National Laboratory

Power, Water, Cooling Infrastructure Analytics

Understanding Infrastructure Efficiency

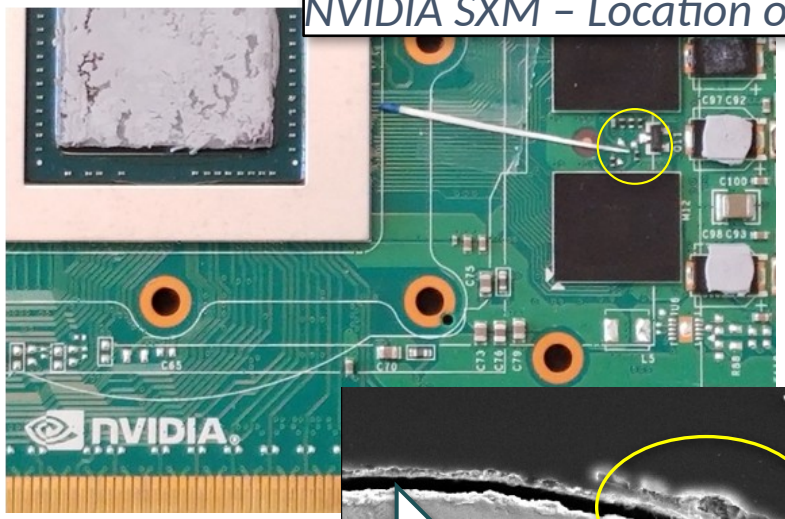
Event Log Monitoring and Failure Analysis

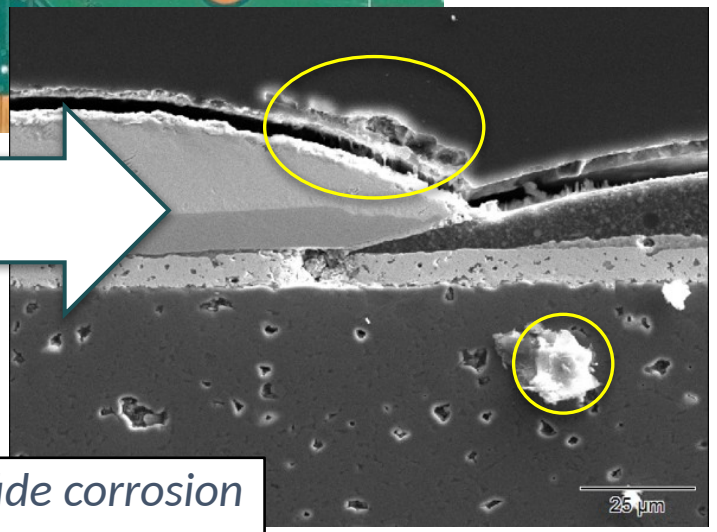


Anti-Sulfur Resistors (ASR) were not appropriately used in Titan GPUs

Insights into Component Failure

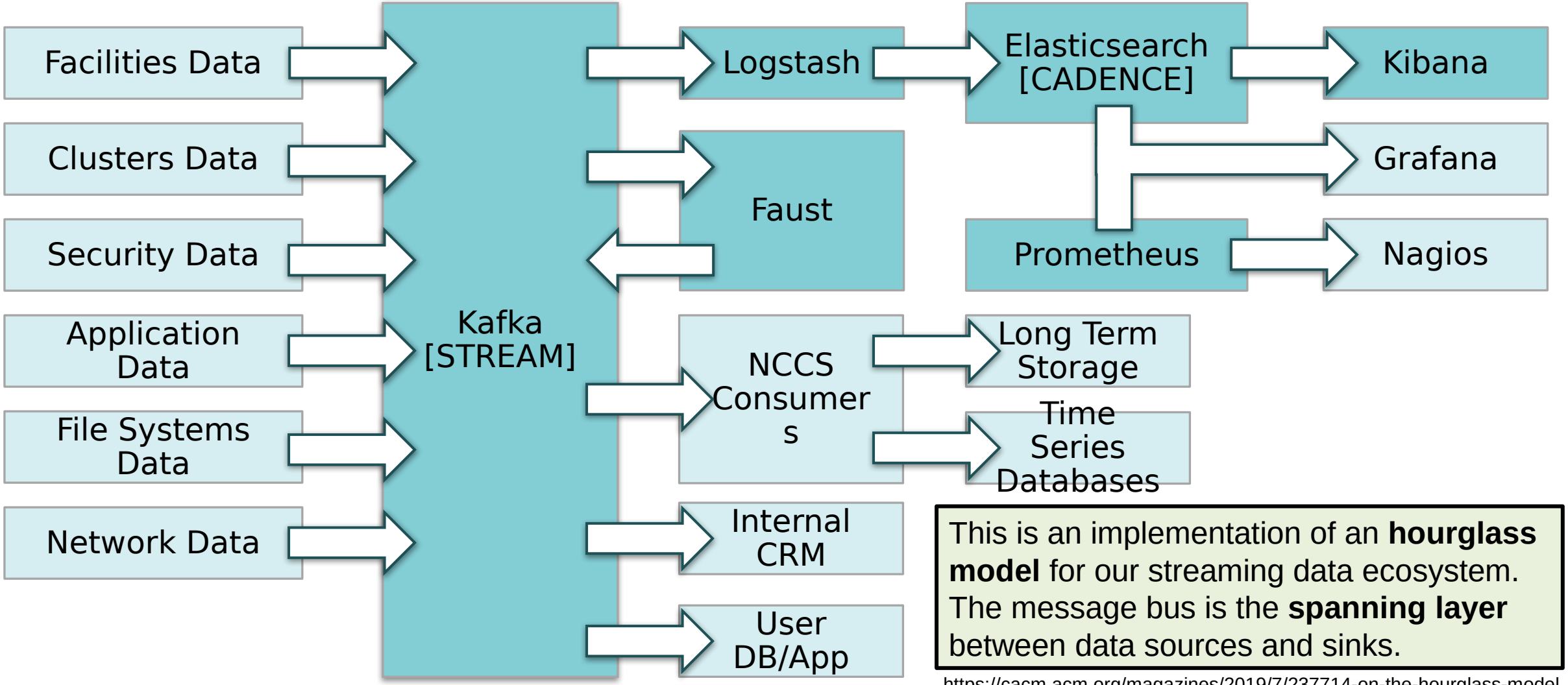
NVIDIA SXM – Location of a non-ASR





Silver-sulfide corrosion "Flowers-of-Sulfur"

Kafka Analytics Platform (Based on IP Networking)

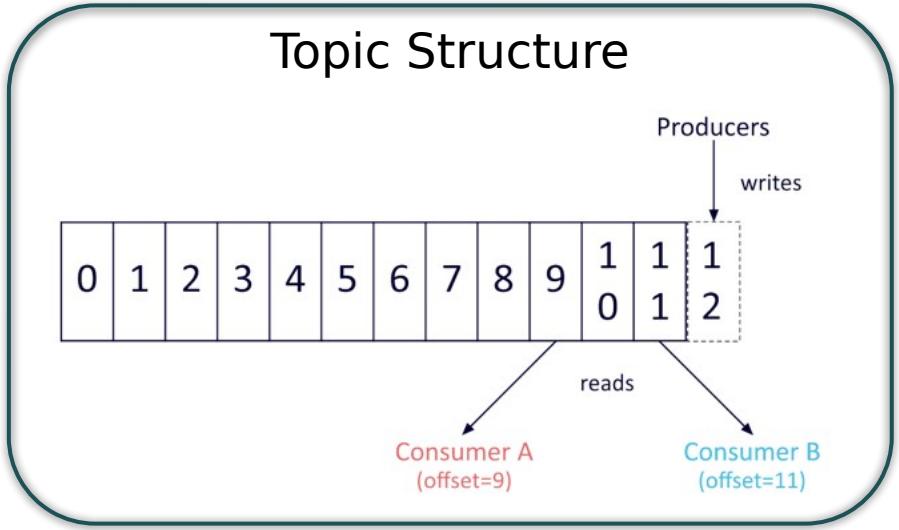


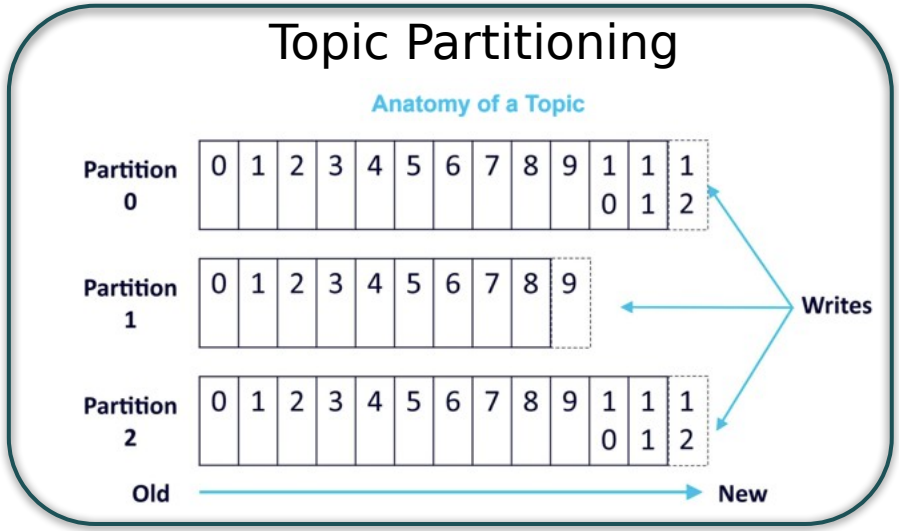
This is an implementation of an **hourglass model** for our streaming data ecosystem. The message bus is the **spanning layer** between data sources and sinks.

https://cacm.acm.org/magazines/2019/7/237714-on-the-hourglass-model

Kafka: Publish / Subscribe (PubSub) Message Bus

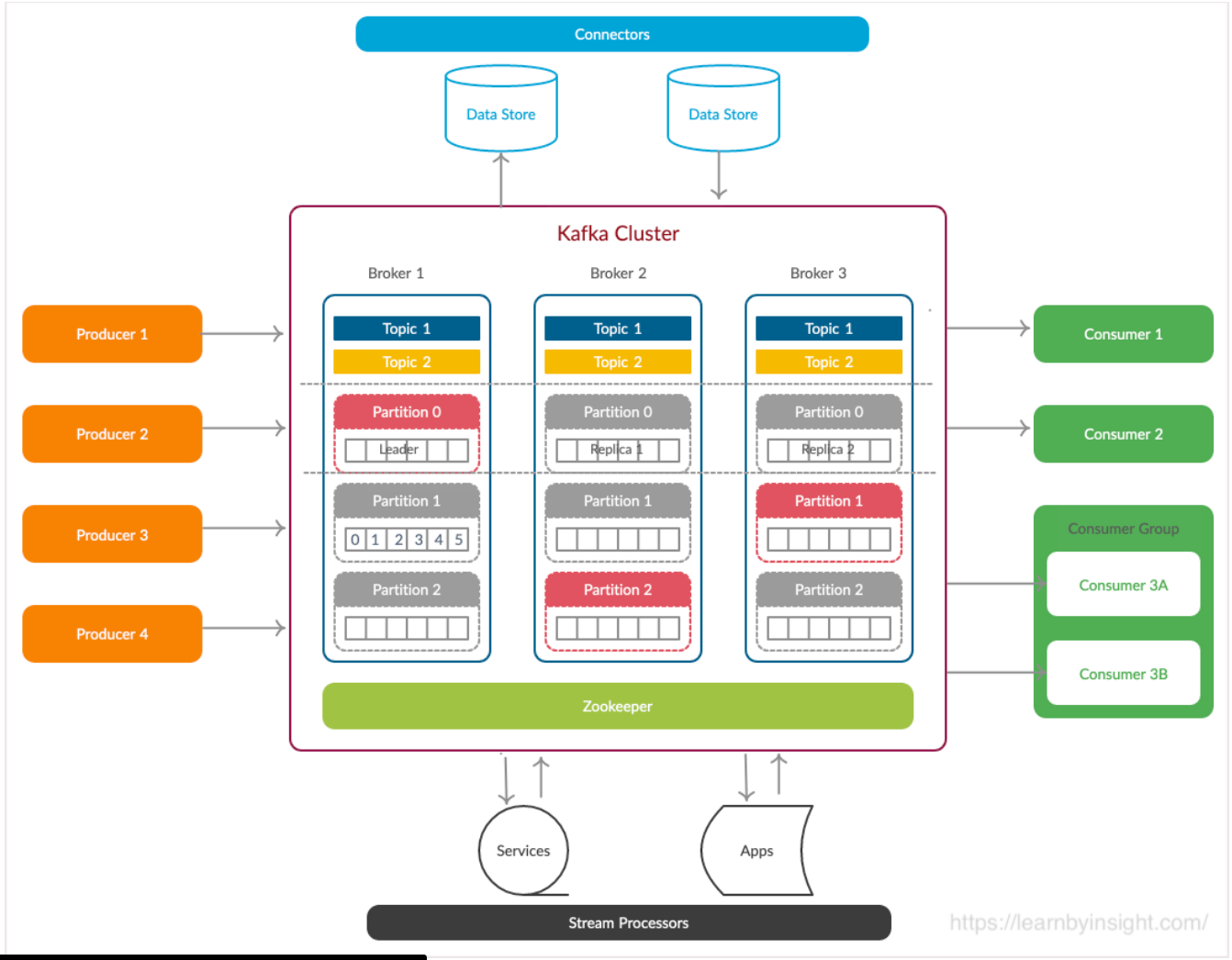
- A message bus serves as a ‘spanning layer’
  - Subscribers ask message bus for all messages related to a ‘topic’
  - Publishers publish messages to a ‘topic’ without needing subscriber details
- Kafka is highly-performant
  - Highly-scalable (Brokers)
  - Distributed (Partitions)
  - Redundant (Replication Factor)
  - **Topics are a form of cache**





Client (Producer or Subscriber) Overview

1. Client connects to any Kafka Broker
2. Client declares desired topic to read/write
3. Broker responds with list of brokers/partitions related to the topic
4. Client connects to those brokers
5. Producer begins writing, Subscriber seeks to desired offset



How many TCP connections are made and IP packets sent?

Key properties of log and telemetry data

- There is an inherent ordering to the data, based on timestamp
- Telemetry streams are well defined, generally have specific schemas, and have concrete topic names
- Low latency is desired for real-time streaming
- Data is ‘continuous’ and gaps are bad
- Log data can be very bursty around for system failure events
- Data written is the ‘truth’, and will never need to be updated

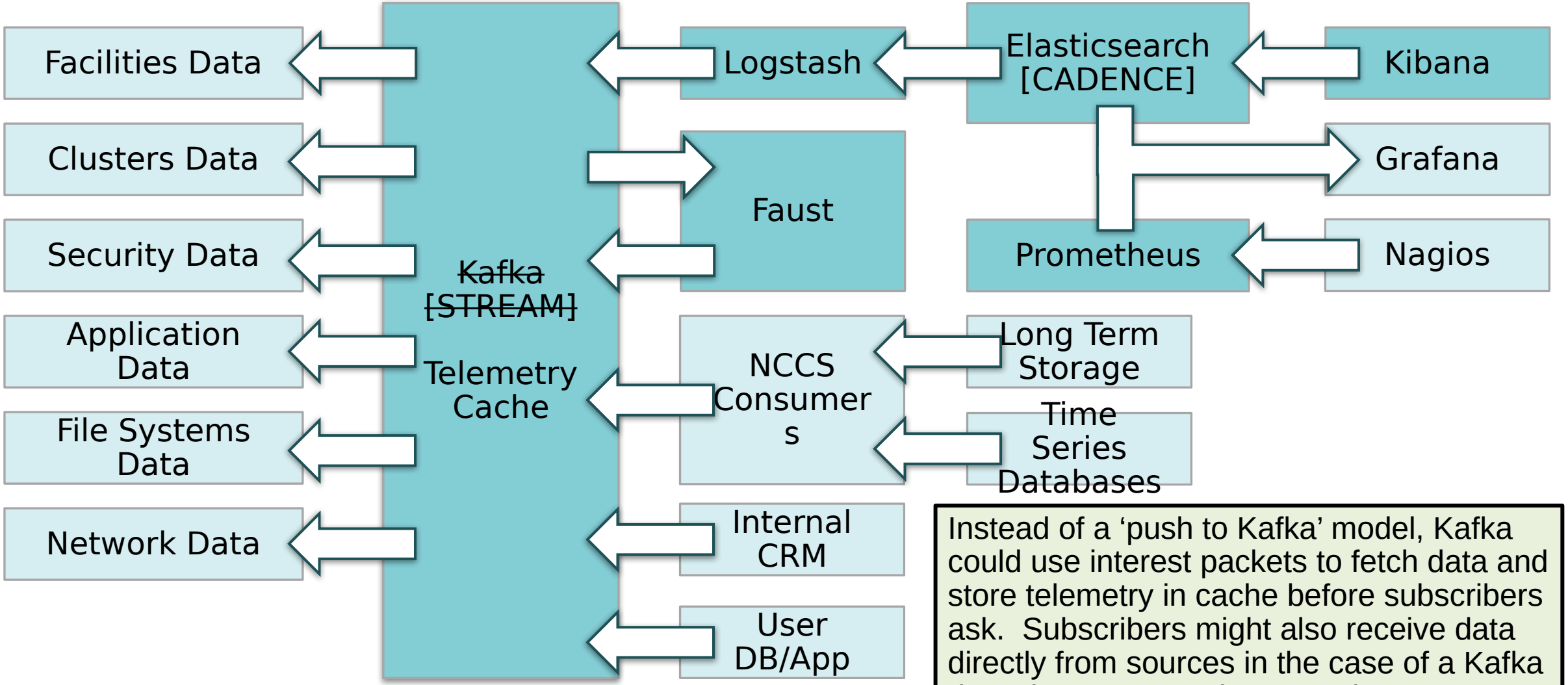
Research Question

- **Research Question:** *"What are the benefits and drawbacks to replacing IP with NDN for streaming analytics and monitoring pipelines?"*

Analyze the benefits of NDN on cloud-like telemetry deployments

1. Do we even need message busses if we have NDN?
   - In particular, subscribers can send interest packets for a particular namespace and subscribe to telemetry streams *without* a message bus
2. Yes, we do, for topic cataloging purposes
3. Yes, we do, for ‘Strategic Caching’ capabilities
4. Has this been done before?
   - Managing Scientific Data with NDN (Catalog and presentation)
   - Real Time Streaming Data Deliver over NDN (Caching, real time)

Kafka Analytics Platform (Using NDN)



Instead of a 'push to Kafka' model, Kafka could use interest packets to fetch data and store telemetry in cache before subscribers ask. Subscribers might also receive data directly from sources in the case of a Kafka downtime or network congestion.

Major Changes Required to replace IP with NDN

Producer Changes

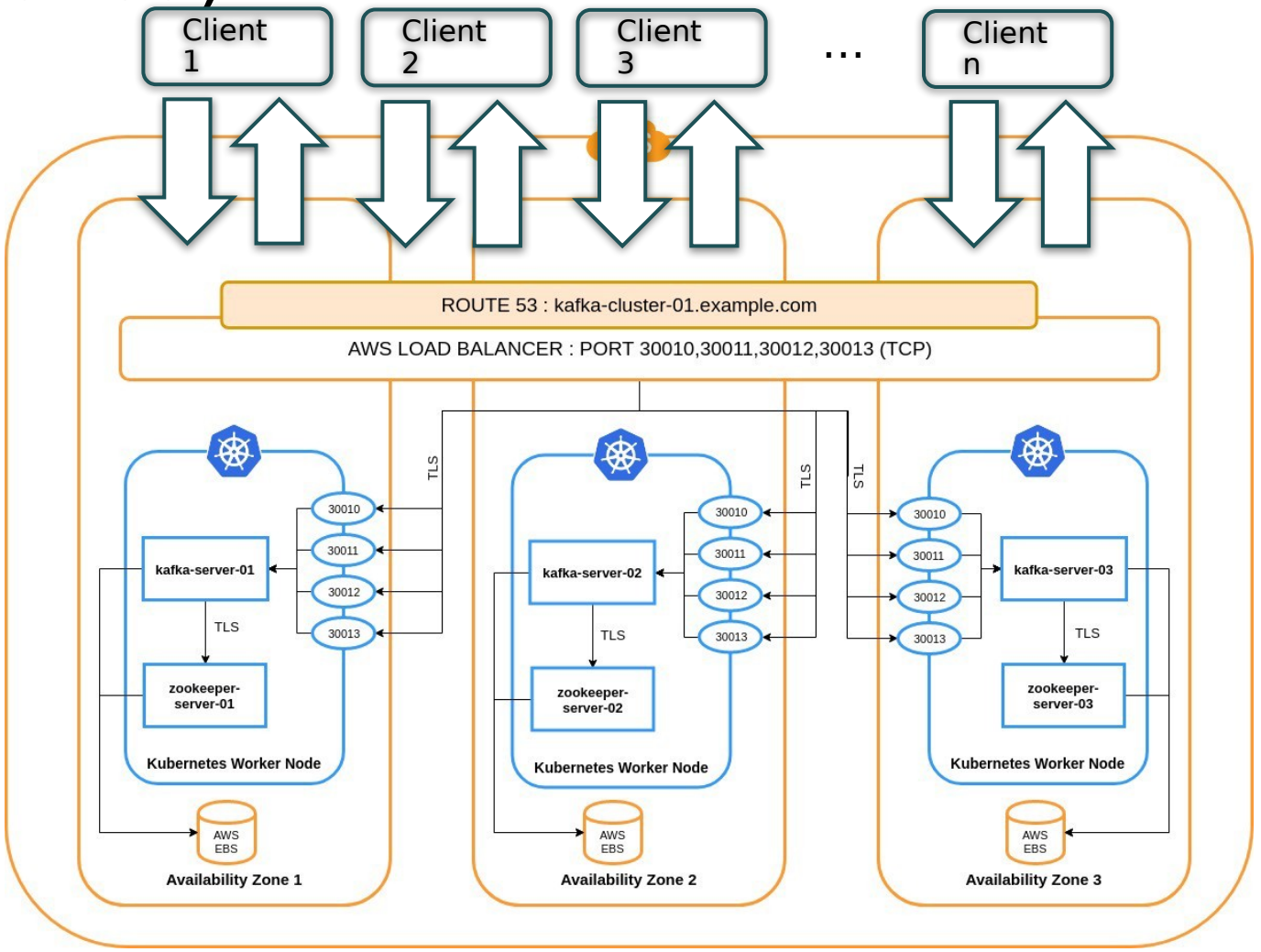
- Producers no longer need to push data to Kafka
- Kafka is configured with ‘topics’ it will ingest prior to any consumer interest in the topic
- Producers need Kafka to pull data before system telemetry buffers fill up or consumers may see telemetry gaps
- Producers need to sign telemetry data packets
- Producers need a coordinated way/namespace scheme for producing to Kafka

Consumer Changes

- Subscriber requests could be of the form /kafka/topic/msg_offset/1
- Kafka only needs to ensure <replication factor> number of Data packets exist in the NDN cache within the Kafka system
- After a Kafka broker failure, recovery and duplication of Data packets can be performed from the still-alive brokers with a copy, OR from the consumers that have the Data packets still cached
- Recovery can be performed ‘backwards’ from Consumer -> Cache which is not possible with IP (it would need to be supported at the application layer)

Problem Overview (Revisited)

1. Kafka Application State
2. Container Networking
3. PaaS Load Balancing
4. Client Connection State
5. Massive amounts of real-time data (> 2TB/day)
6. Rebalancing after Failure



Research Question: *"What are the benefits and drawbacks to replacing IP with NDN for streaming analytics and monitoring pipelines?"*

Final Thoughts

Benefits of replacing IP with NDN

- Standing up a third party (or remote) Kafka bus would be trivial and wouldn't require configuration for the main bus
- Broker recovery could be much faster, especially if Data packets could also be recovered from subscribers in the event of data loss
- Reduction in TCP handshakes and where-is-the-data style lookups may reduce latency
- Most subscribers subscribing to the 'latest' data won't even hit the Kafka cache, because pending interests will be satisfied simultaneously

Drawbacks to replacing IP with NDN

- Support for NDN is still not pervasive
- All (most?) of the producers, subscribers, and supporting applications and infrastructure would need to support NDN to gain benefit
- Caching and network design strategies need future thought, especially if recovery of lost Kafka brokers relies on a producer and subscriber driven rebuilding of cache.
- Security model based on non-zero trust (i.e. username and password SASL authentication) needs to be engineered

Research Question: *"What are the benefits and drawbacks to replacing IP with NDN for streaming analytics and monitoring pipelines?"*

Thanks!

Ryan Adamson: adamsonrm@ornl.gov

Acknowledgements:

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Depart- ment of Energy under Contract No. DE-AC05-00OR22725.