

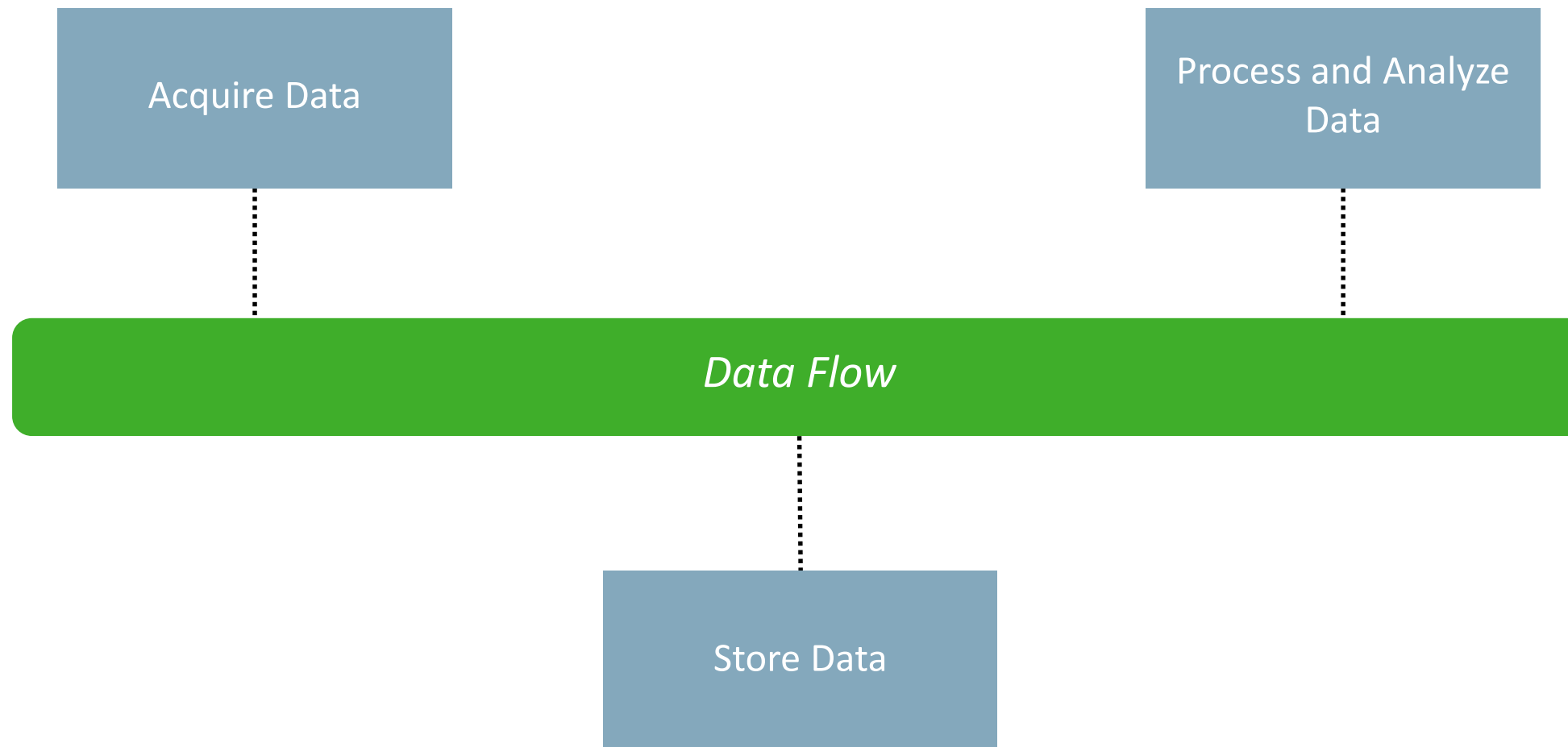
# Apache NiFi Deep Dive

**Bryan Bende – Member of Technical Staff**

NJ Hadoop Meetup – May 10<sup>th</sup> 2016

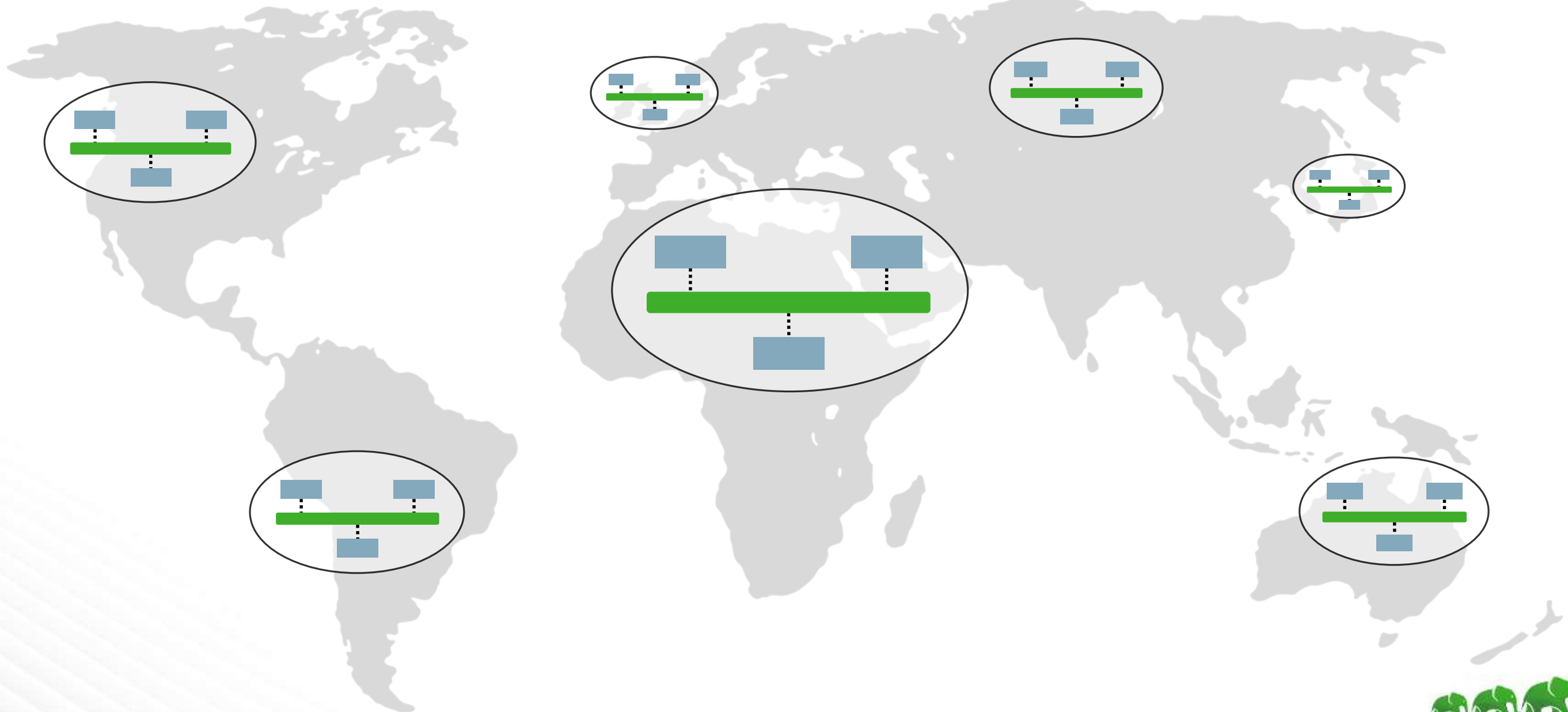


# Simplistic View of Enterprise Data Flow



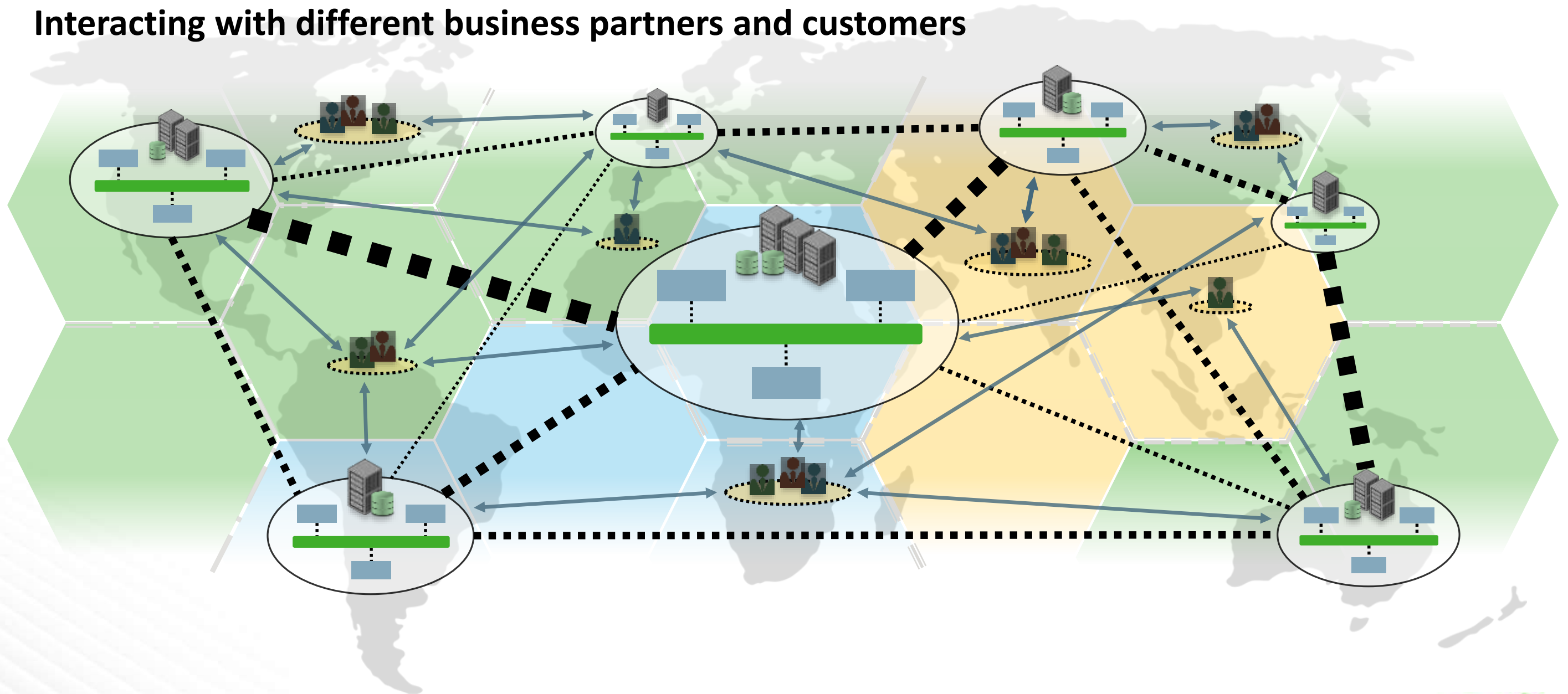
# Realistic View of Enterprise Data Flow

Different organizations/business units across different geographic locations...



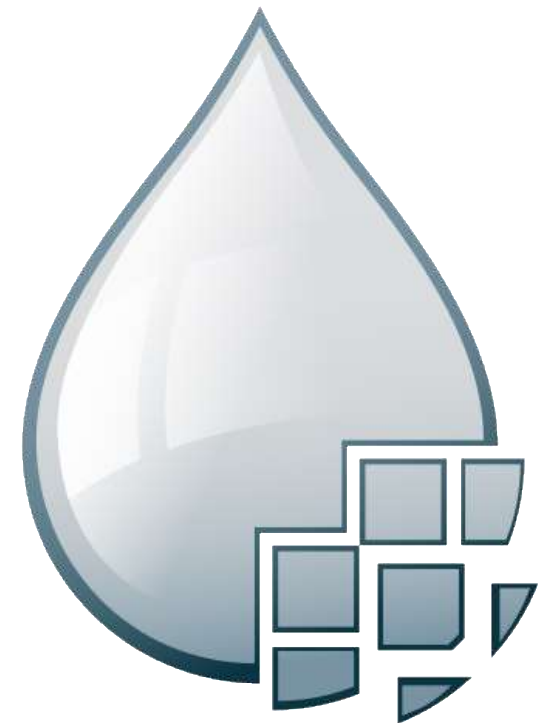
# Realistic View of Enterprise Data Flow

Interacting with different business partners and customers



# Apache NiFi

- Created to address the challenges of global enterprise dataflow
- Key features:
  - Visual Command and Control
  - Data Lineage (Provenance)
  - Data Prioritization
  - Data Buffering/Back-Pressure
  - Control Latency vs. Throughput
  - Secure Control Plane / Data Plane
  - Scale Out Clustering
  - Extensibility



# Apache NiFi

## ***What is Apache NiFi used for?***

- Reliable and secure transfer of data between systems
- Delivery of data from sources to analytic platforms
- Enrichment and preparation of data:
  - Conversion between formats
  - Extraction/Parsing
  - Routing decisions

## ***What is Apache NiFi NOT used for?***

- Distributed Computation
- Complex Event Processing
- Joins / Complex Rolling Window Operations



# Apache NiFi Deep Dive

# Terminology

## FlowFile

- Unit of data moving through the system
- Content + Attributes (key/value pairs)

## Processor

- Performs the work, can access FlowFiles

## Connection

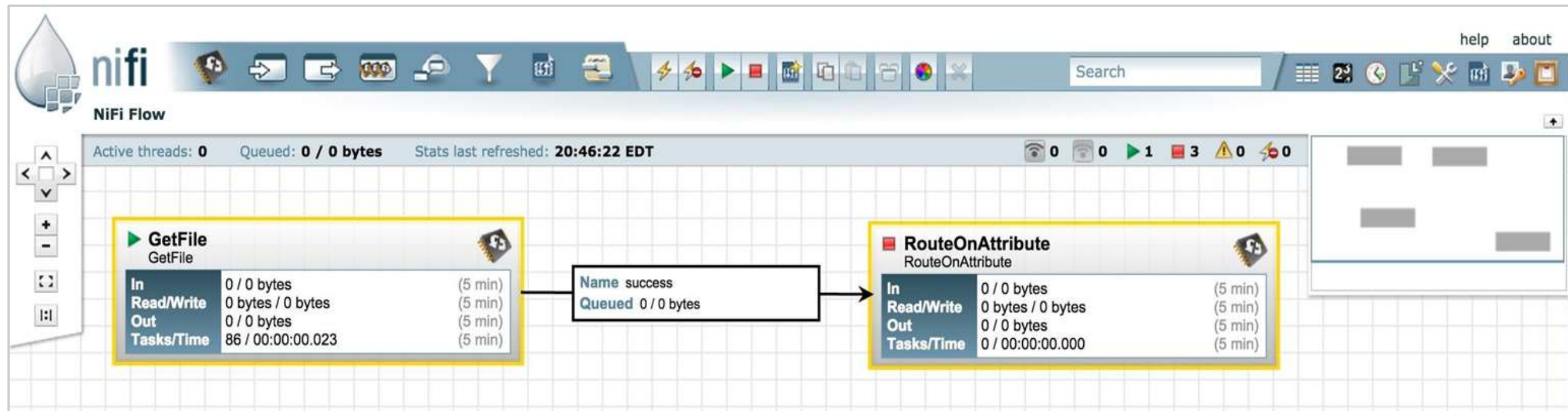
- Links between processors
- Queues that can be dynamically prioritized

## Process Group

- Set of processors and their connections
- Receive data via input ports, send data via output ports



# Visual Command & Control



- Drag and drop processors to build a flow
- Start, stop, and configure components in real time
- View errors and corresponding error messages
- View statistics and health of data flow
- Create templates of common processor & connections

# Provenance/Lineage

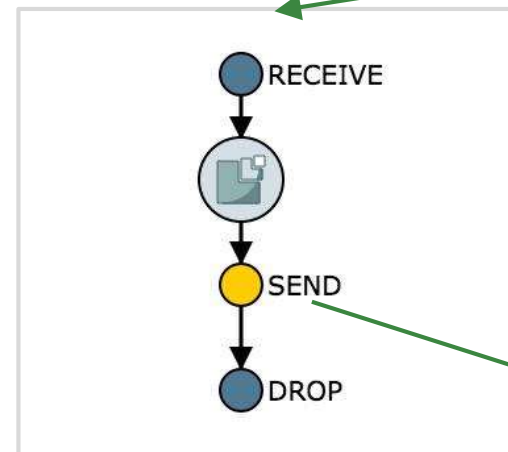
**NiFi Flow Data Provenance**  
Oldest event available: 07/29/2015 14:08:06 EDT

Filter  by component name ▼  
Displaying 1,000 of 1,000

Last updated: 21:12:00 EDT Showing the most recent 1,000 of 62,293 events, please refine the search. Search

	Date/Time ▼	Type	FlowFile Uuid	Size	Component Name	Component Type	
ⓘ	07/29/2015 16:21:34.368 EDT	DROP	3b9f20bc-031e-4af8-ad8a-fedce...	158 bytes	PutSolrContentStream	PutSolrContentStream	🔗 ➔
ⓘ	07/29/2015 16:21:34.367 EDT	SEND	3b9f20bc-031e-4af8-ad8a-fedce...	158 bytes	PutSolrContentStream	PutSolrContentStream	🔗 ➔
ⓘ	07/29/2015 16:21:34.366 EDT	DROP	6f5036bc-1768-476d-9b6d-1f83...	2.15 KB	PutSolrContentStream	PutSolrContentStream	🔗 ➔

- Tracks data at each point as it flows through the system
- Records, indexes, and makes events available for display
- Handles fan-in/fan-out, i.e. merging and splitting data
- View attributes and content at given points in time

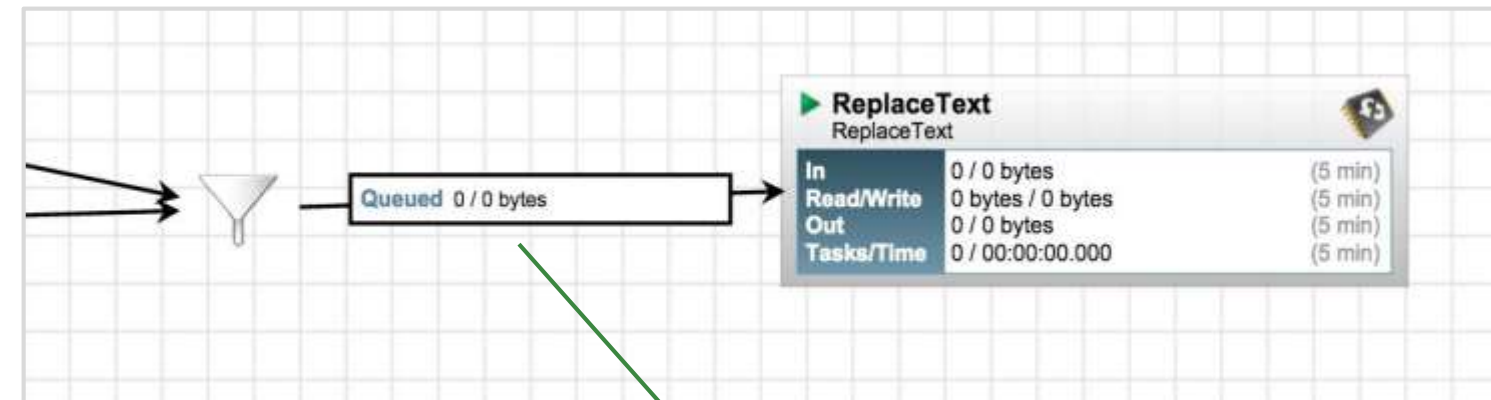


**Provenance Event**

Details	Attributes	Content
Time	07/29/2015 16:21:34.367 EDT	
Event Duration	00:00:00.001	
Lineage Duration	00:00:00.117	
Type	SEND	
FlowFile Uuid	3b9f20bc-031e-4af8-ad8a-fedcef4e0099	
File Size	158 bytes	
Component Id	fa7b551f-c405-4fde-b004-0b0d69c03472	
Component Name	PutSolrContentStream	
Component Type	PutSolrContentStream	
Transit Uri	solr://http://localhost:8984/solr/chronicle	
Details	No value set	

# Prioritization

- Configure a prioritizer per connection
- Determine what is important for your data – time based, arrival order, importance of a data set
- Funnel many connections down to a single connection to prioritize across data sets
- Develop your own prioritizer if needed



**Available prioritizers ?**

- FirstInFirstOutPrioritizer
- NewestFlowFileFirstPrioritizer
- OldestFlowFileFirstPrioritizer

**Selected prioritizers ?**

- PriorityAttributePrioritizer

# Back-Pressure

- Configure back-pressure per connection
- Based on number of FlowFiles or total size of FlowFiles
- Upstream processor no longer scheduled to run until below threshold

**FlowFile expiration** ?

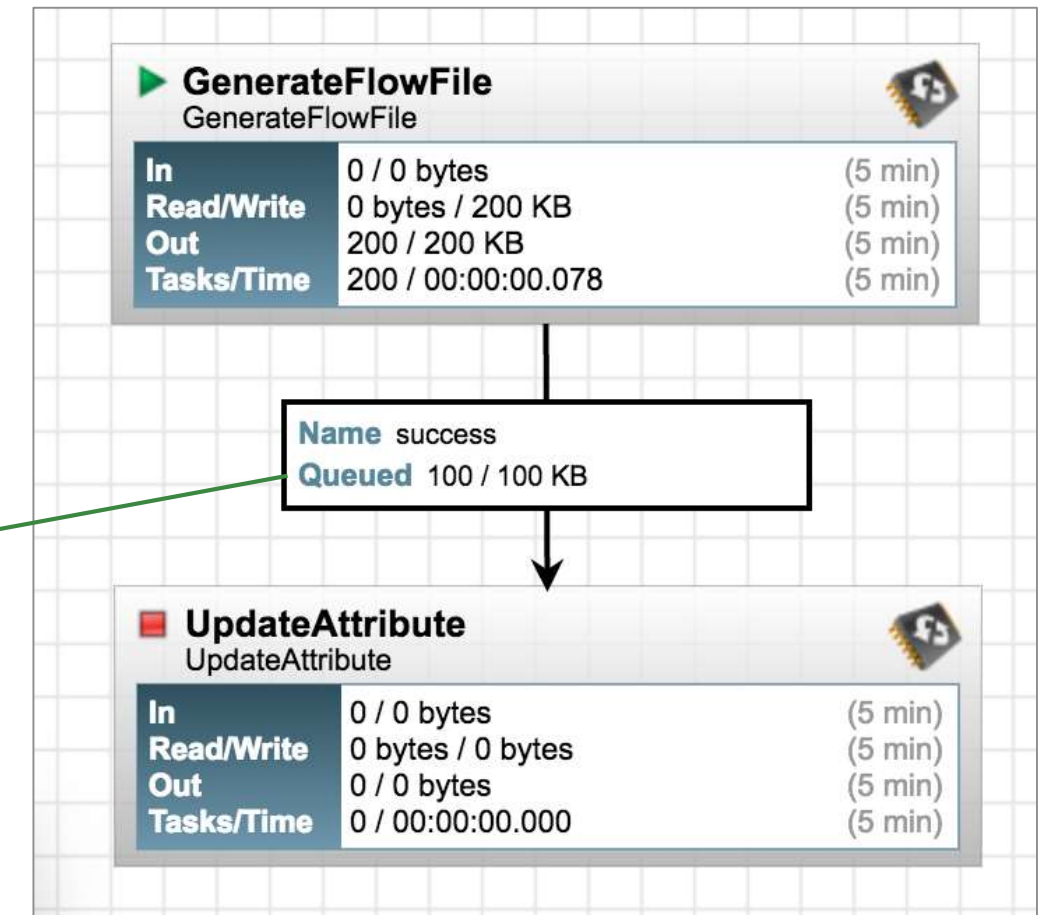
0 sec

**Back pressure object threshold** ?

100

**Back pressure data size threshold** ?

0 MB





# Latency vs. Throughput

- Choose between lower latency, or higher throughput on each processor
- Higher throughput allows framework to batch together all operations for the selected amount of time for improved performance
- Processor developer determines whether to support this by using `@SupportsBatching` annotation

### Configure Processor

Settings

Scheduling

Properties

Comments

Scheduling strategy ?

Timer driven

Concurrent tasks ?

1

Run schedule ?

0 sec

Run duration ?

0ms25ms50ms100ms250ms500ms1s2s

Lower latencyHigher throughput

# Security

## Control Plane

- Pluggable authentication
  - 2-Way SSL, LDAP, Kerberos
- Pluggable authorization
  - File-based authority provider out of the box
  - Multiple roles to defines access controls
- Audit trail of all user actions

## Data Plane

- Optional 2-Way SSL between cluster nodes
- Optional 2-Way SSL on Site-To-Site connections (NiFi-to-NiFi)
- Encryption/Decryption of data through processors
- Provenance for audit trail of data



# Extensibility

Built from the ground up with extensions in mind

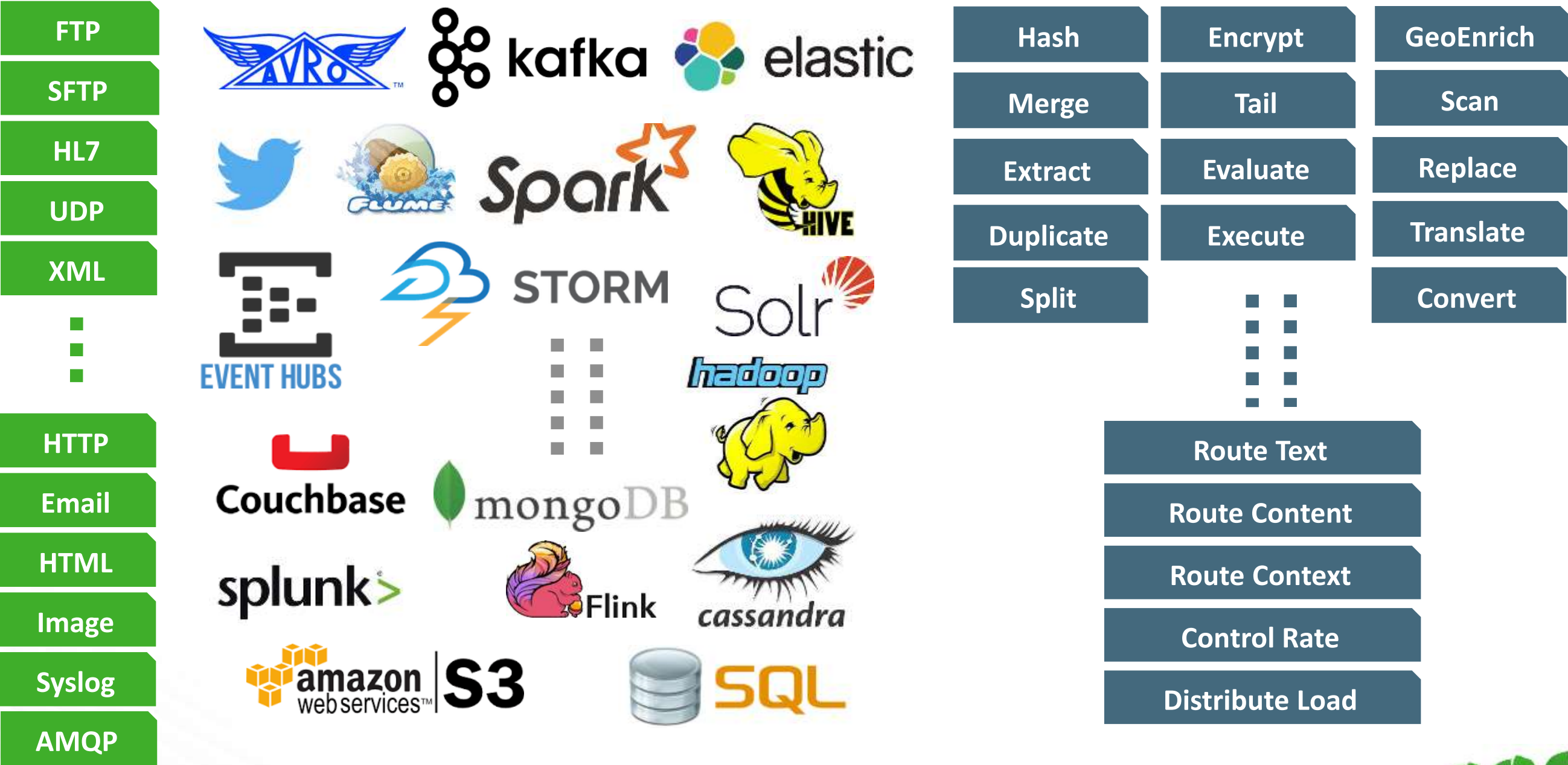
Service-loader pattern for...

- Processors
- Controller Services
- Reporting Tasks
- Prioritizers

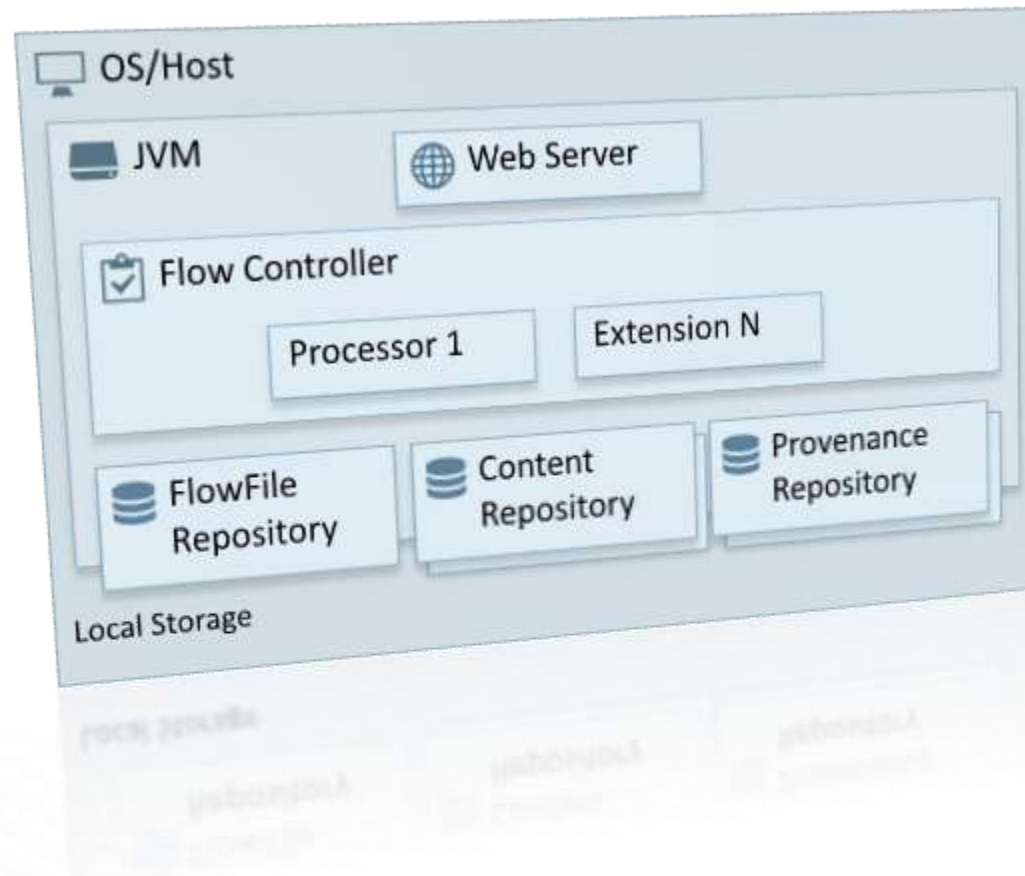
Extensions packaged as NiFi Archives (NARs)

- Deploy NiFi lib directory and restart
- Provides ClassLoader isolation
- Same model as standard components

# Rapid Ecosystem Adoption: 130+ Processors

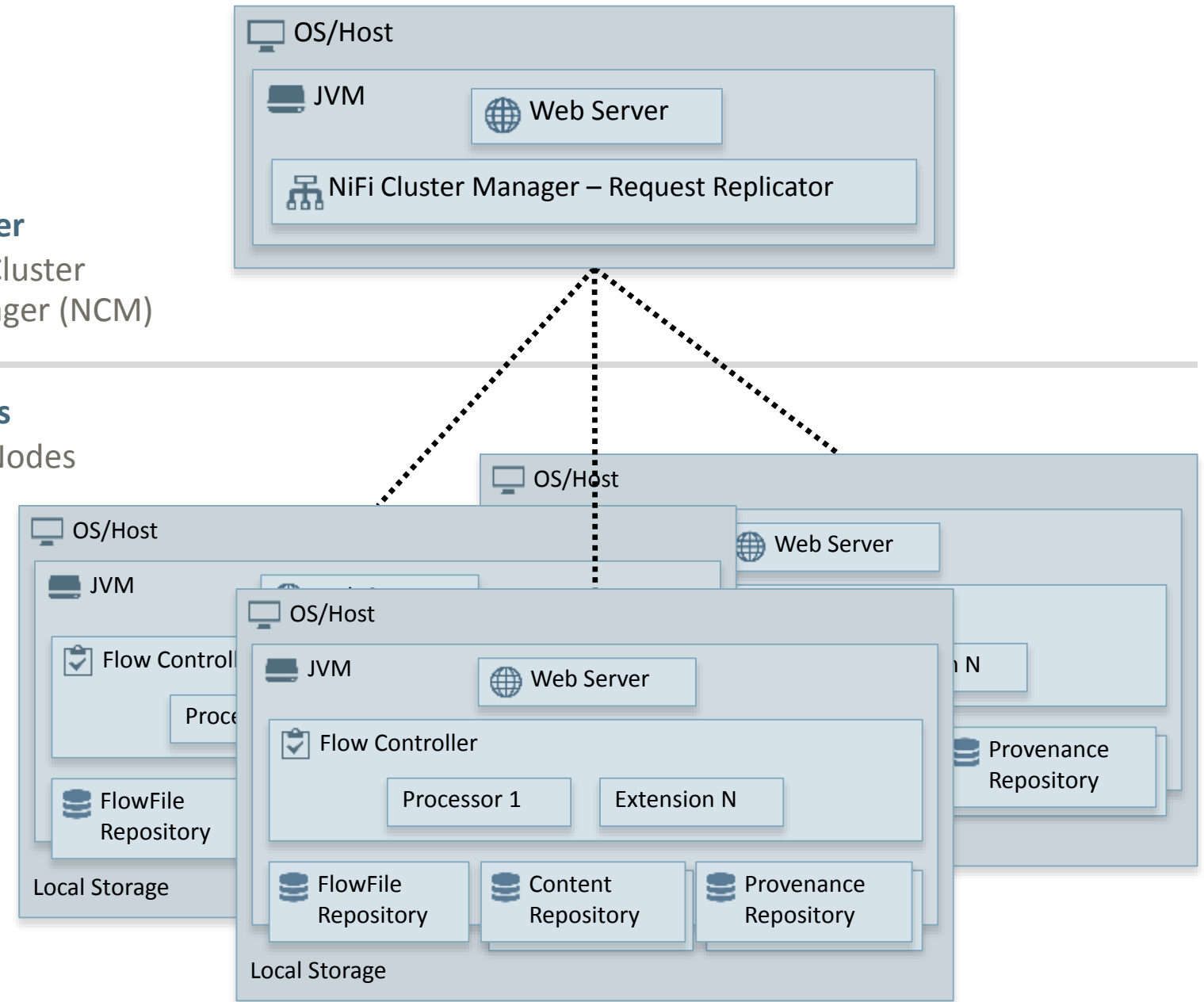


# Architecture



**Master**  
NiFi Cluster Manager (NCM)

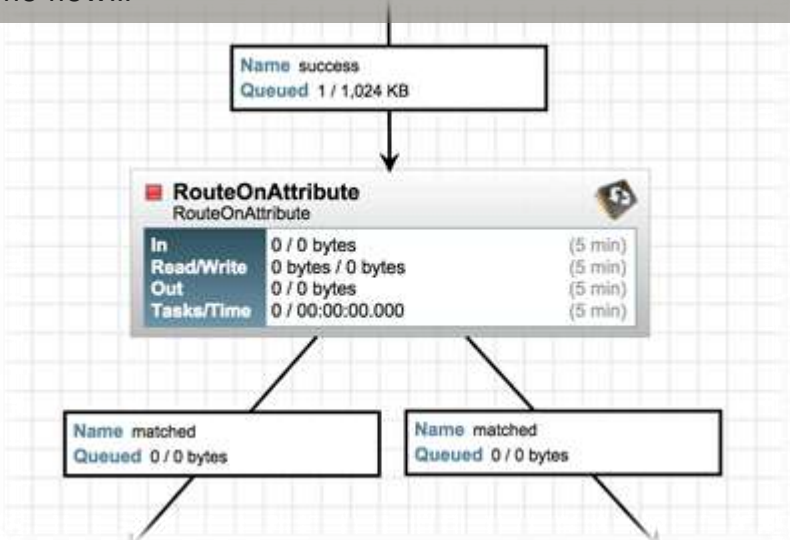
**Slaves**  
NiFi Nodes



# NiFi Architecture – Repositories - Pass by reference

Excerpt of demo flow...

BEFORE



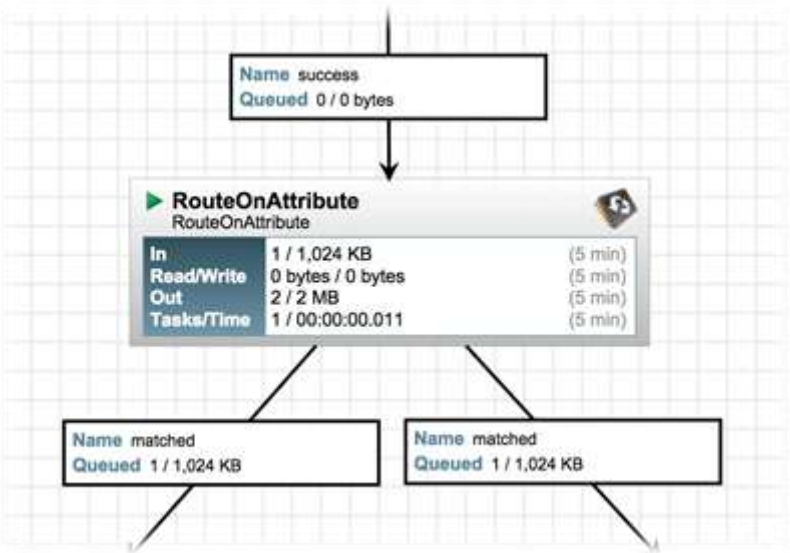
What's happening inside the repositories...

$F_1 \rightarrow C_1$

$C_1$

$P_1 \rightarrow F_1$

AFTER



$F_1 \rightarrow C_1$


$F_2 \rightarrow C_1$


$C_1$


$P_1 \rightarrow F_1$  – Create

$P_2 \rightarrow F_1$  – Route

$P_3 \rightarrow F_2$  – Clone ( $F_1$ )

 FlowFile

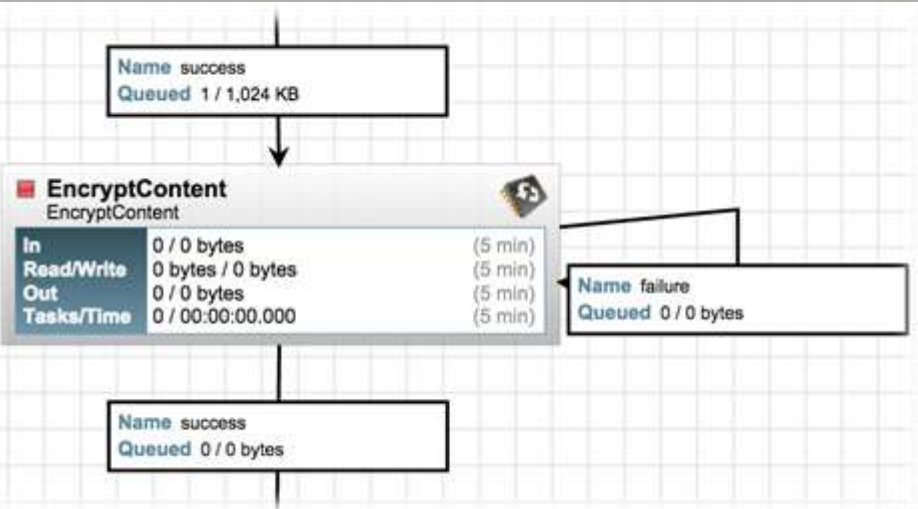
 Content

 Provenance

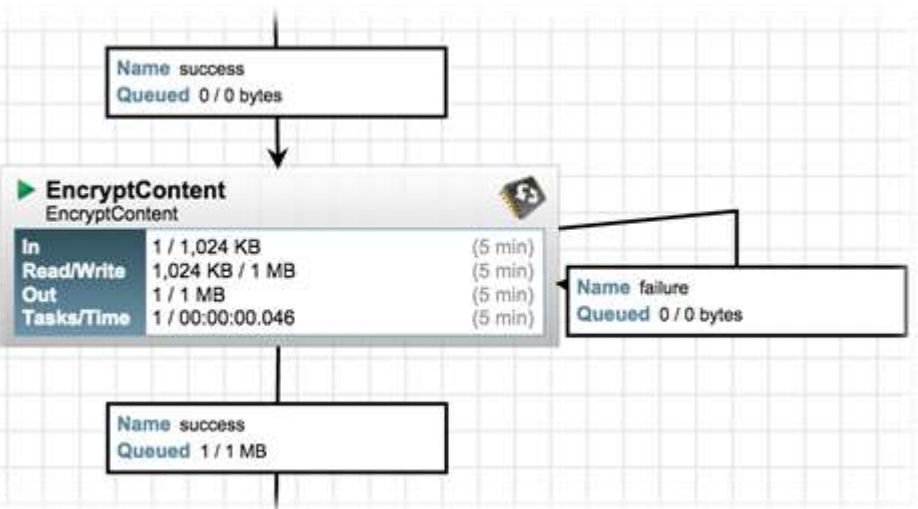
# NiFi Architecture – Repositories – Copy on Write

Excerpt of demo flow...

BEFORE



AFTER



What's happening inside the repositories...

$F_1 \rightarrow C_1$

$C_1$

$P_1 \rightarrow F_1$  - CREATE

$F_1 \rightarrow C_1$   
 $F_{1.1} \rightarrow C_2$

$C_1$  (plaintext)  
 $C_2$  (encrypted)

$P_1 \rightarrow F_1$  - CREATE  
 $P_2 \rightarrow F_{1.1}$  - MODIFY

FlowFile

Content

Provenance



# Performance & Scaling

- Optimize I/O...
  - Separate partition for each repository
  - Multiple partitions for content repository
  - RAID configurations for redundancy & striping
- Tune JVM Memory, GC, and # of threads
- Scale up with a cluster
  - 100s of thousands of events per second per node
- Scale down to a Raspberry Pi
  - 10s of thousands of events per second



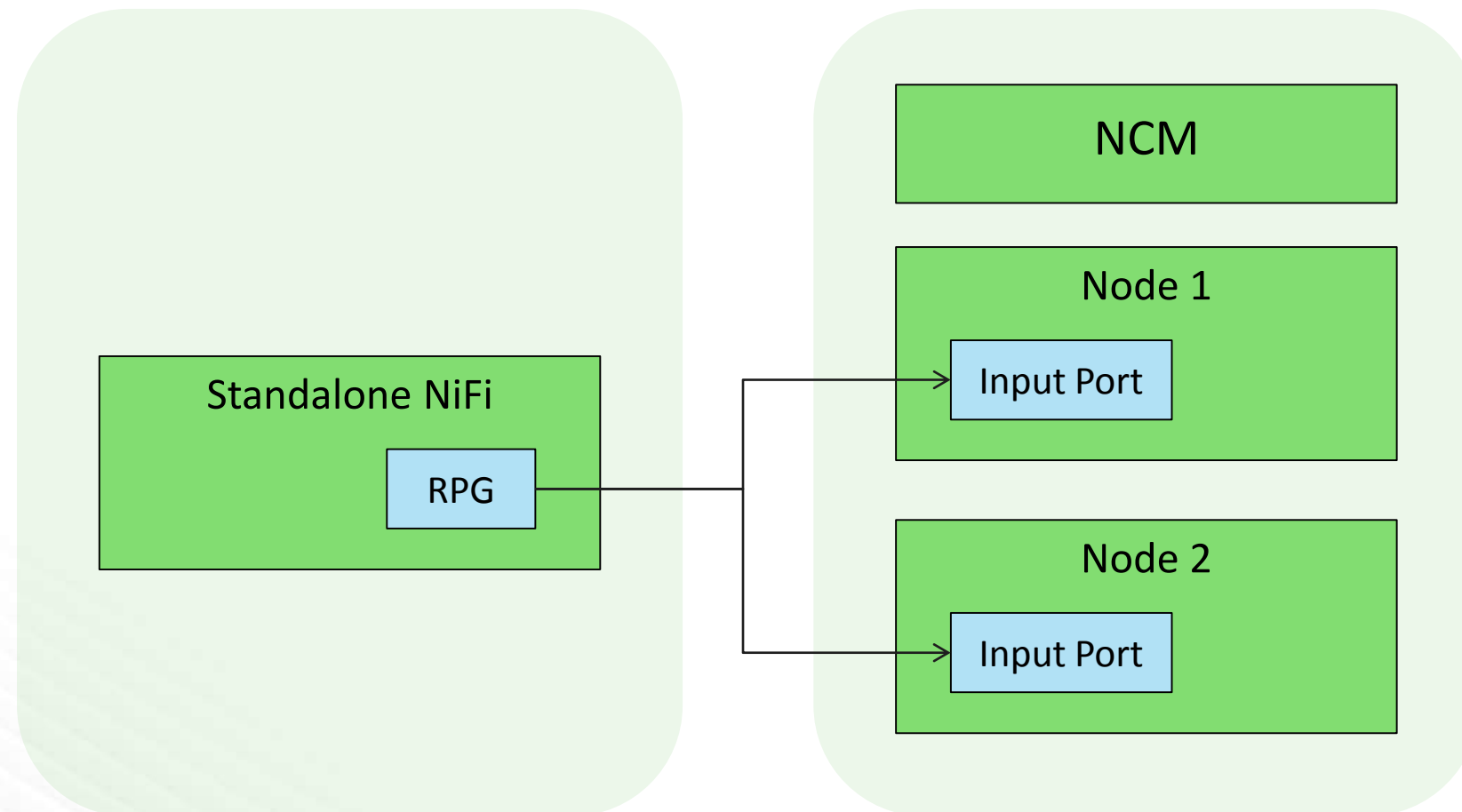
# Apache NiFi Site-To-Site

# Site-To-Site

- Direct communication between two NiFi instances
- Push to Input Port on receiver, or Pull from Output Port on source
- Communicate between clusters, standalone instances, or both
- Handles load balancing and reliable delivery
- Secure connections using certificates (optional)

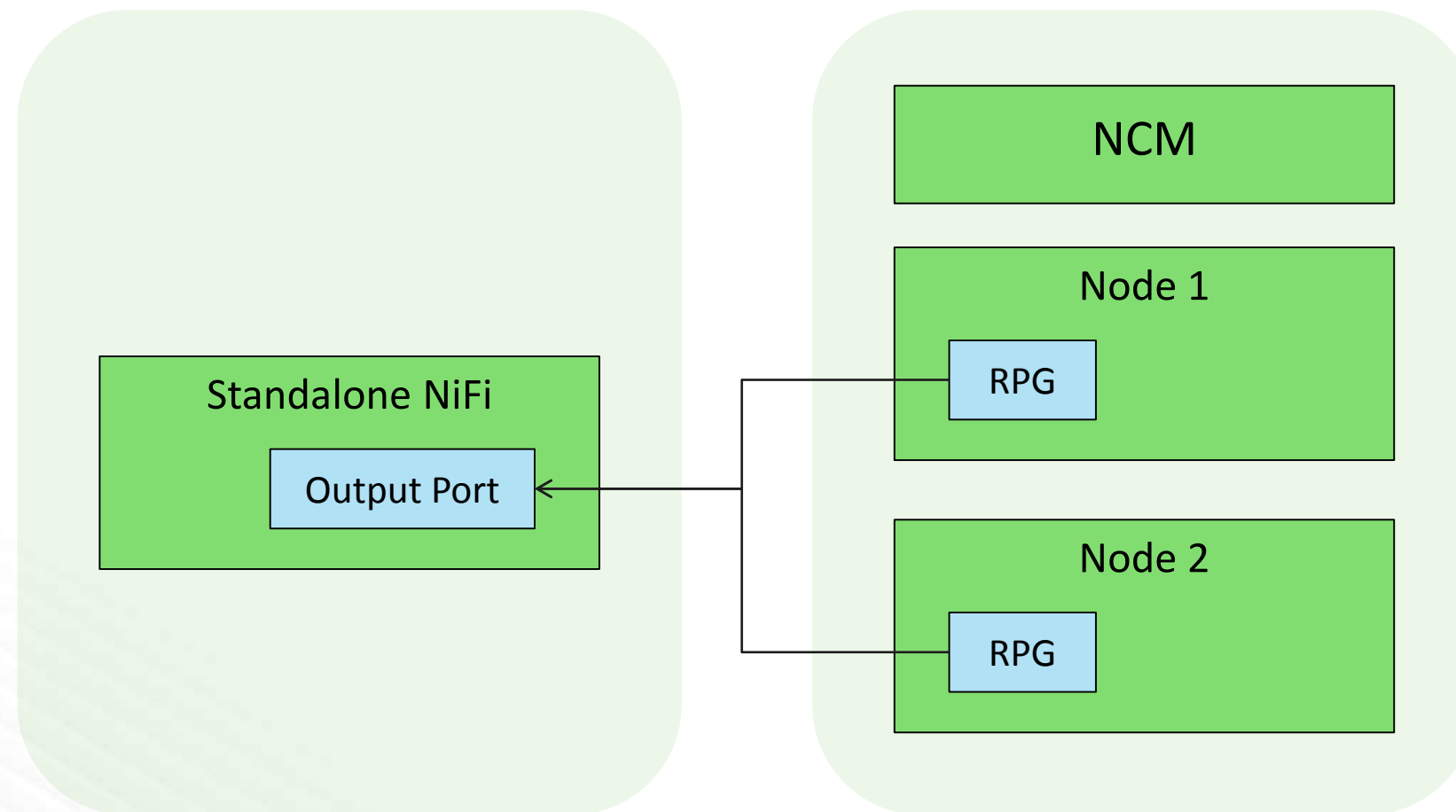
# Site-To-Site Push

- Source connects Remote Process Group to Input Port on destination
- Site-To-Site takes care of load balancing across the nodes in the cluster



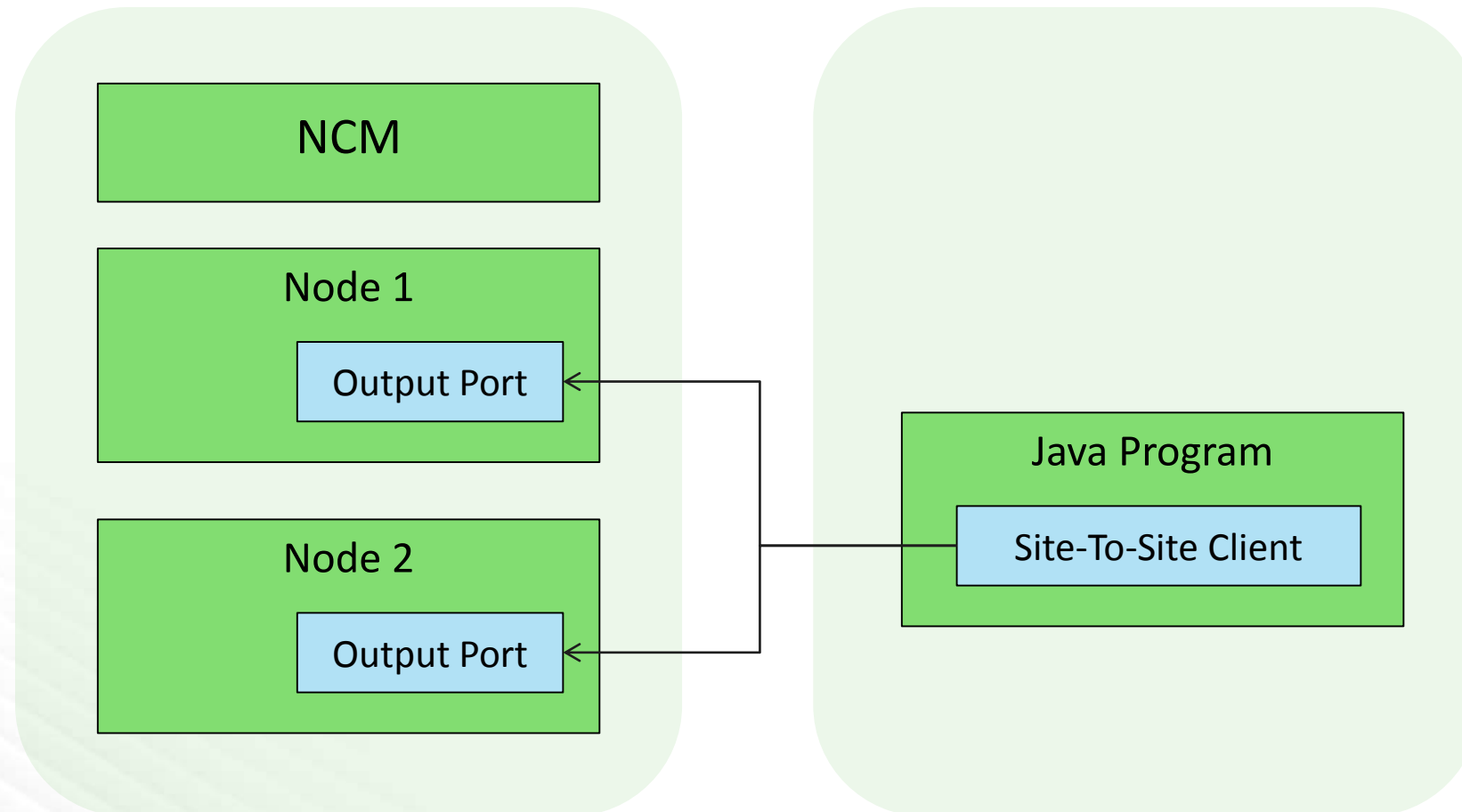
# Site-To-Site Pull

- Destination connects Remote Process Group to Output Port on the source
- If source was a cluster, each node would pull from each node in cluster



# Site-To-Site Client

- Code for Site-To-Site broken out into reusable module
- <https://github.com/apache/nifi/tree/master/nifi-commons/nifi-site-to-site-client>
- Foundation for integration with stream processing platforms



# Current Stream Processing Integrations

## Spark Streaming - NiFi Spark Receiver

- <https://github.com/apache/nifi/tree/master/nifi-external/nifi-spark-receiver>

## Storm – NiFi Spout & Bolt

- <https://github.com/apache/nifi/tree/master/nifi-external/nifi-storm-spout>

## Flink – NiFi Source & Sink

- <https://github.com/apache/flink/tree/master/flink-streaming-connectors/flink-connector-nifi>

## Apex - NiFi Input Operators & Output Operators

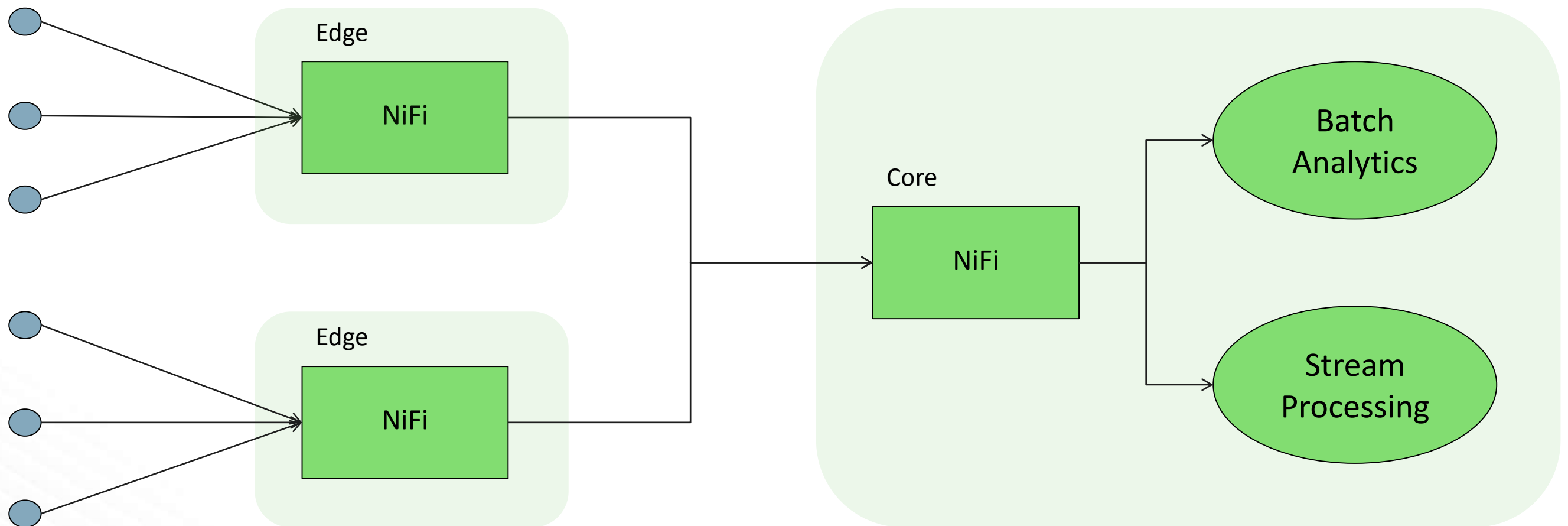
- <https://github.com/apache/incubator-apex-malhar/tree/master/contrib/src/main/java/com/datatorrent/contrib/nifi>



# Bi-Directional Data Flows

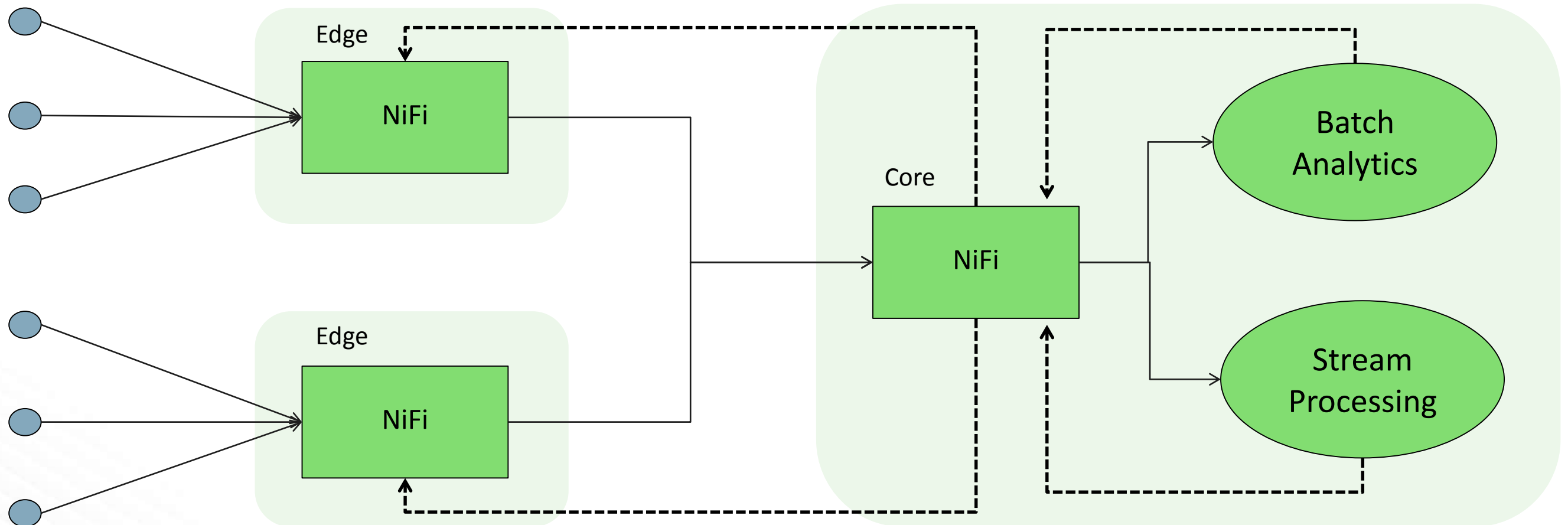
# Drive Data to Core for Analysis

- Drive data from sources to central data center for analysis
- Tiered collection approach at various locations, think regional data centers



# Dynamically Adjusting Data Flows

- Push analytic results back to core NiFi
- Push results back to edge locations/devices to change behavior



# Future Work

# Apache NiFi Roadmap

- HA Control Plane
  - Zero Master cluster, Web UI accessible from any node
  - Auto-Election of “Cluster Coordinator” and “Primary Node” through ZooKeeper
- HA Data Plane
  - Ability to replicate data across nodes in a cluster
- Multi-Tenancy
  - Restrict Access to portions of a flow
  - Allow people/groups within an organization to only access their portions of the flow
- Extension Registry
  - Create a central repository of NARs and Templates
  - Move most NARs out of Apache NiFi distribution, ship with a minimal set

# Apache NiFi Roadmap

- Variable Registry
  - Define environment specific variables through the UI, reference through EL
  - Make templates more portable across environments/instances
- Redesign of User Interface
  - Modernize look & feel, improve usability, support multi-tenancy
- Continued Development of Integration Points
  - New processors added continuously!
- MiNiFi
  - Complimentary data collection agent to NiFi's current approach
  - Small, lightweight, centrally managed agent that integrates with NiFi for follow-on dataflow management



# Thanks!

## Resources

- Apache NiFi Mailing Lists
  - [https://nifi.apache.org/mailling\\_lists.html](https://nifi.apache.org/mailling_lists.html)
- Apache NiFi Documentation
  - <https://nifi.apache.org/docs.html>
- Getting started developing extensions
  - <https://cwiki.apache.org/confluence/display/NIFI/Maven+Projects+for+Extensions>
  - <https://nifi.apache.org/developer-guide.html>

## Contact Info:

- Email: [bbende@hortonworks.com](mailto:bbende@hortonworks.com)
- Twitter: @bbende

# Thank you