

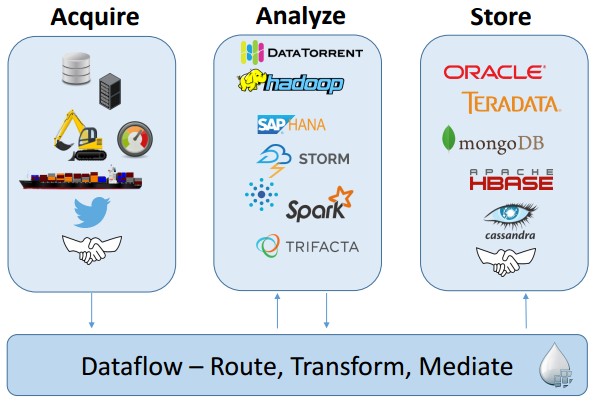
### NiFi was built to automate the flow of data between systems.

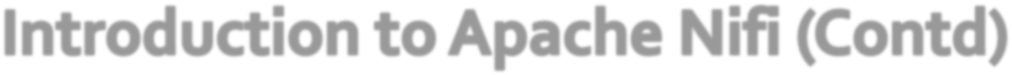
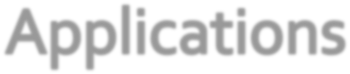
### While the term 'dataflow' is used in a variety of contexts, we use it here to mean the automated

### and managed flow of information between systems

### It supports powerful and scalable directed graphs of data routing,

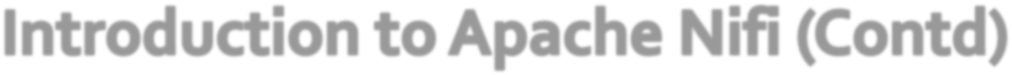
### transformation, and system mediation logic.



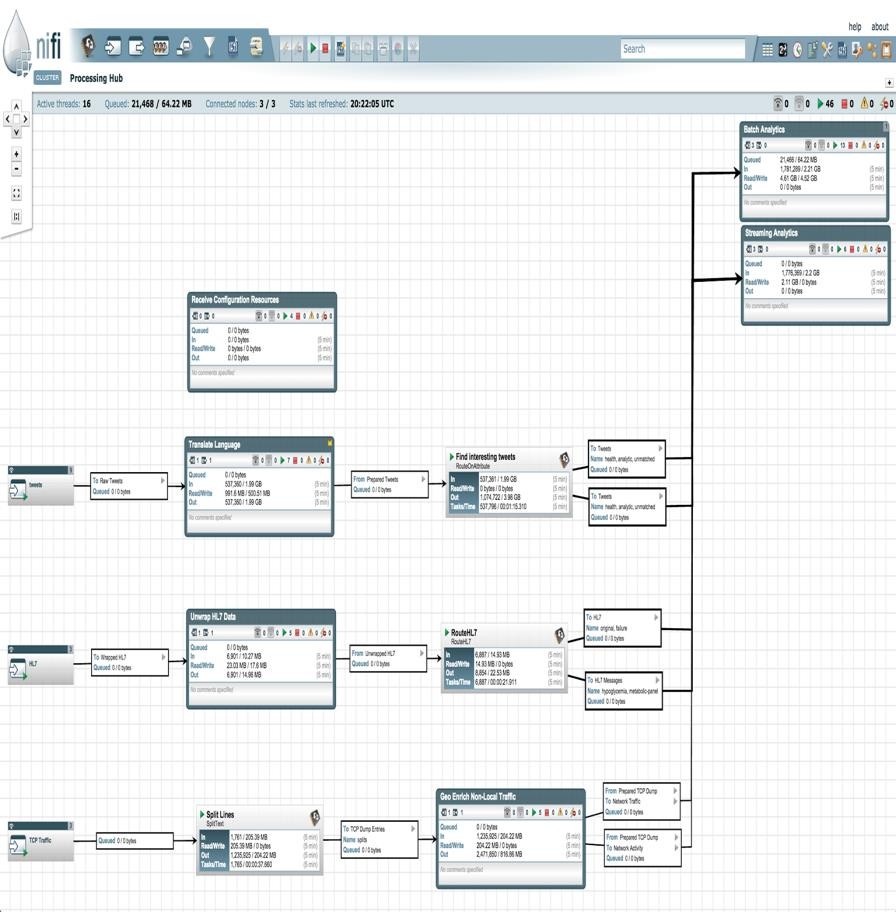


# Guaranteed Delivery

* + Data Buffering
  + Prioritized Queuing
  + Flow specific – Latency vs Throughput
  + Data Provenance
  + Visual plugins
  + Flow Templates
  + Dynamic Change of flow



* Powerful and reliable system to process and



**distribute** data

* **Directed graphs** of data routing and

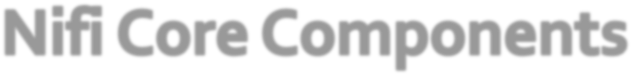
transformation

* **Web-based** User Interface for creating, monitoring, & controlling data flows
* **Modify data flow at runtime**, dynamically

prioritize data

* **Data Provenance** tracks data through entire system
* **Easily extensible** through development of custom

components



#### FlowFile

A FlowFile is a data record, which consists of a pointer to its content (payload) and attributes to support the content, that is associated with one or more provenance events.

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

#### Processor

Performs the work, can access FlowFiles

* + **Connection**

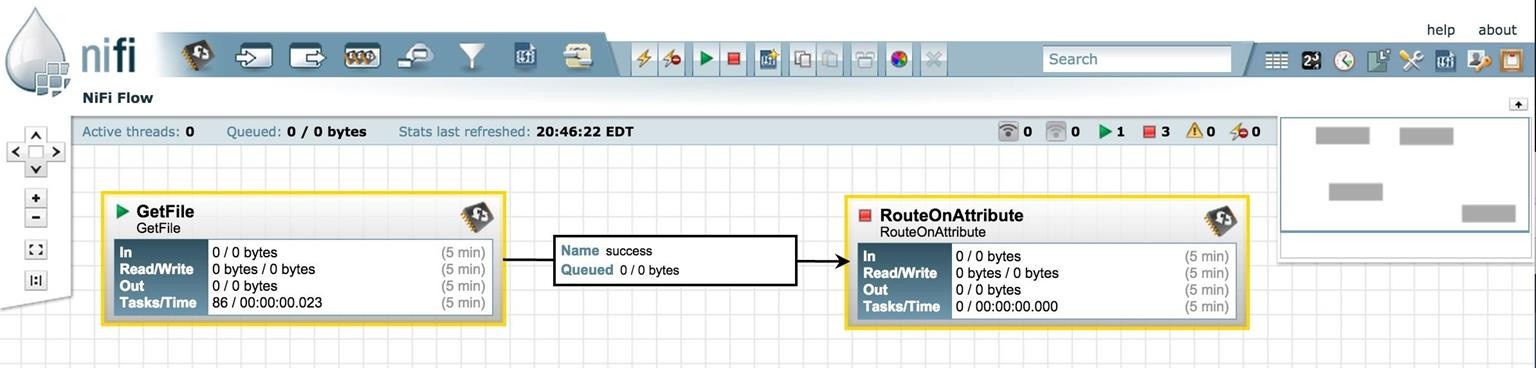
Links between processors, Queues that can be dynamically

prioritized

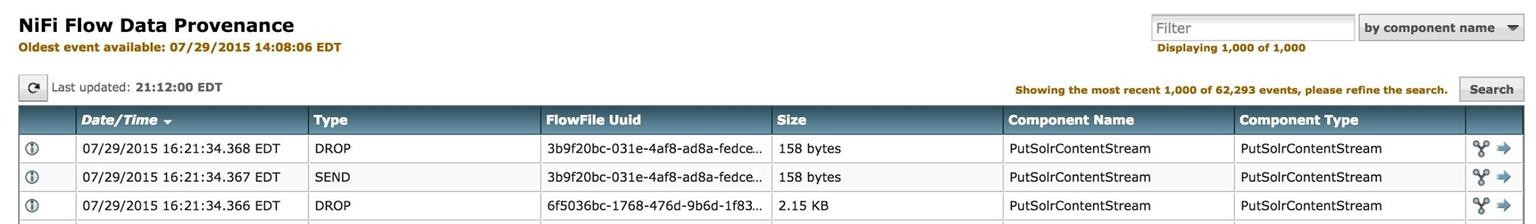
* + **Process Group**

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

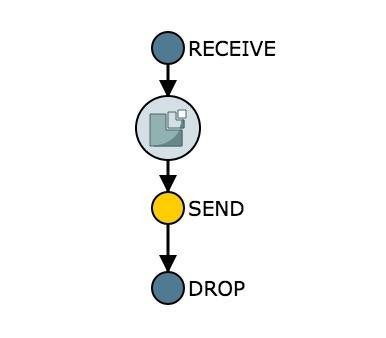
* + **Port**
  + **Funnel**



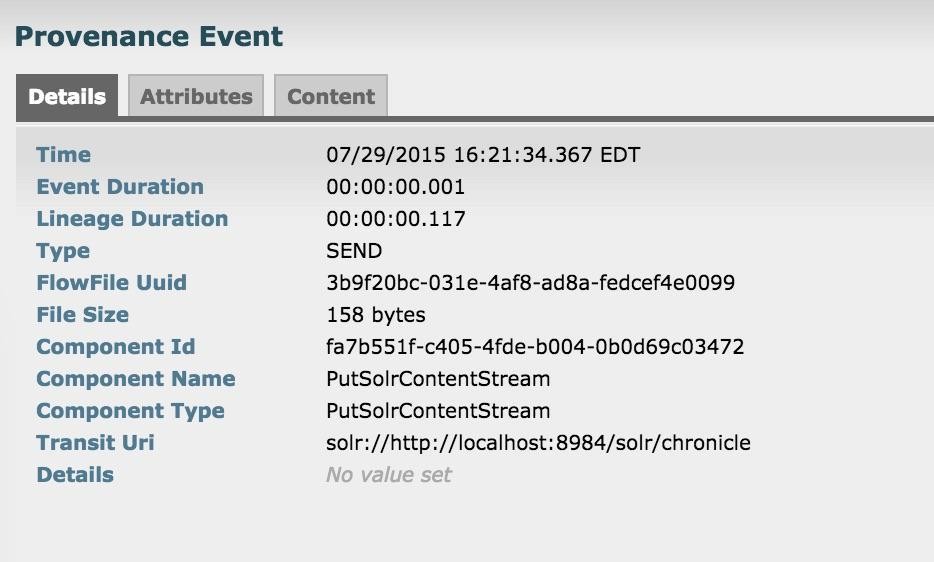
* **Drag and drop processors to build a flow**
* **Start, stop, and configure components in real time**
* **View errors and corresponding error messages**
* **Create templates of common processor & connections**



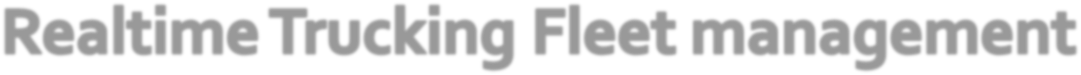
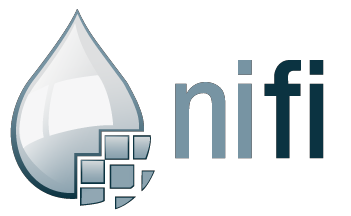
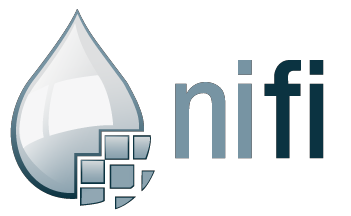
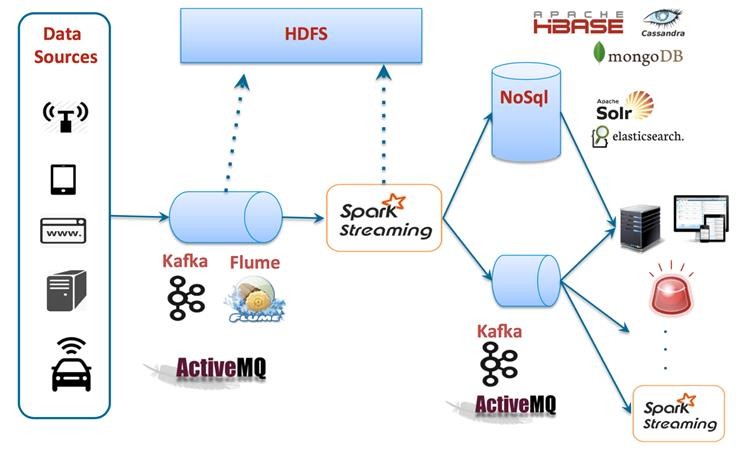
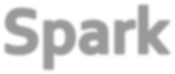
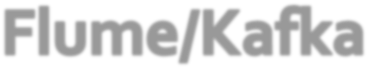
* **Tracks data at each point as it flows**

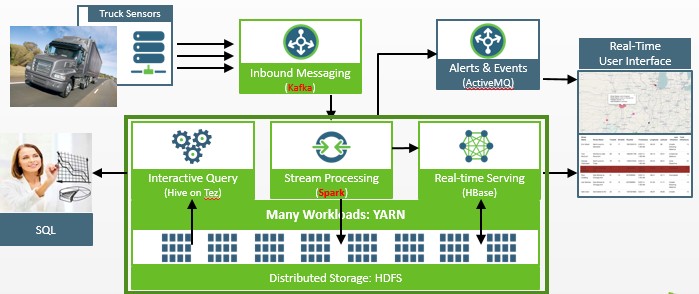


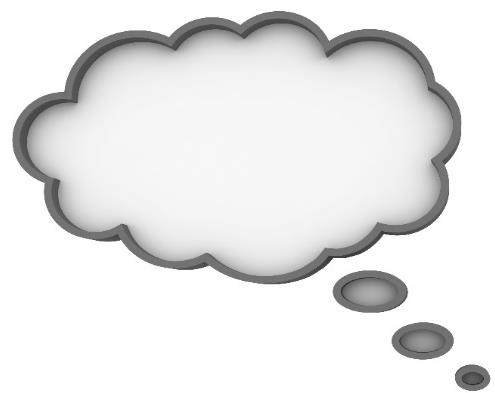
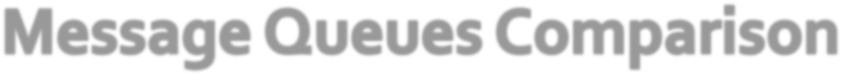
**through the system**



* **Records, indexes, and makes events available for display**







Why Kafka?

When we have other messaging systems Aren’t they Good?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Kafka** | **Flume** | **NIFI** | **RabbitMQ** |
| **Hadoop Plugins** | **Custom** | **Native** | **Medium** | **Least** |
| **Runtime Transformation** | **Yes** | **Yes** | **Yes** | **Yes** |
| **Data Sourcing** | **Push** | **Pull** | **Push and Pull** | **Pull** |
| **Distributed** | **High** | **Medium** | **Low** | **High** |
| **Persistence** | **Yes** | **No** | **No** | **Yes** |
| **Coding** | **API changes** | **Configuration changes** | **UI Plugins** | **API Changes** |
| **Coding Complexity** | **High** | **Medium** | **Low** | **High** |
| **Scalability** | **High** | **Medium** | **Medium** | **Medium** |
| **Runtime changes in the Data flow** | **No, Downtime needed** | **No, Downtime needed** | **Yes, No Downtime needed** | **No, Downtime needed** |
| **WorkFlow** | **No workflow**  **orchestration** | **No workflow**  **orchestration** | **Workflow orchestration**  **available** | **No workflow**  **orchestration** |