Apache NiFi

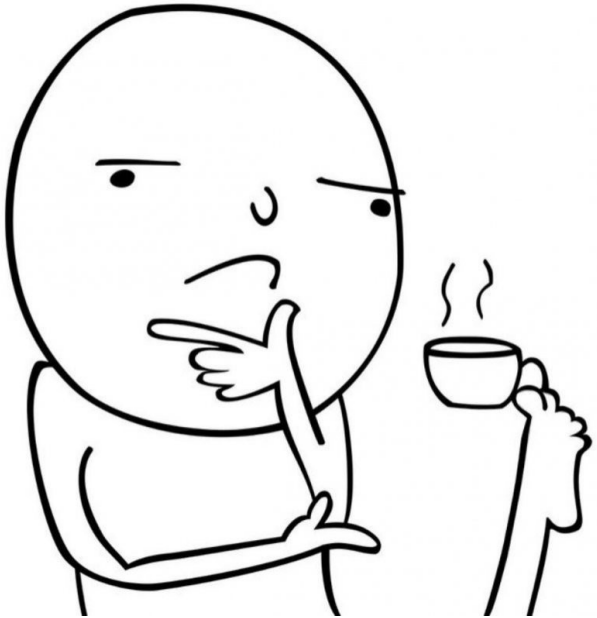
Brief Overview

Ihor Didyk
Software Engineer

GlobalLogic

Agenda

1. Data Flow vs Pipeline
2. NiFi Overview
3. NiFi Key Features
4. Demo



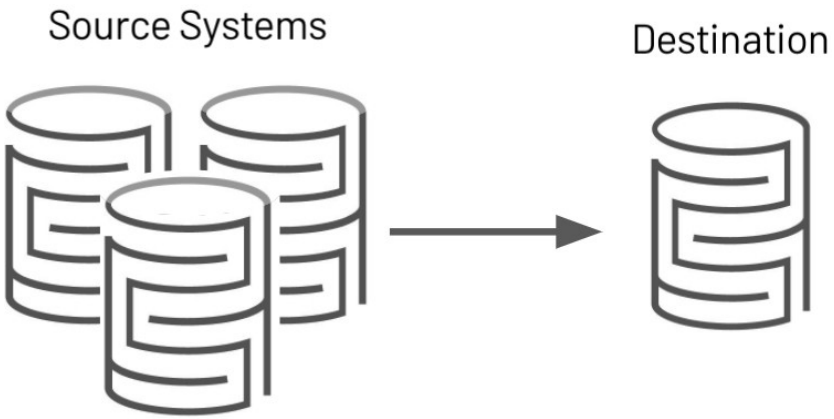
DISCLAIMER

Everything described here is true and complete to the best of author's knowledge. All recommendations and inferences are made without guarantee of the part of the author. The author disclaims any liability in connection with the use of this information.

Data Flow

Pipeline

What is Data Flow?



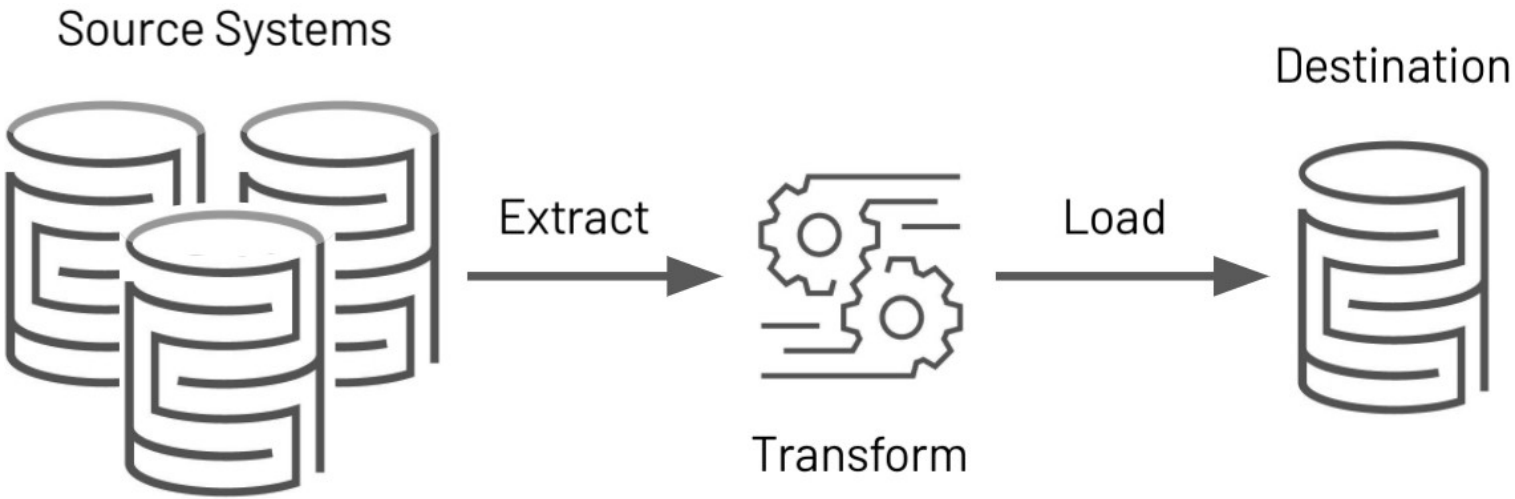
Source can be:

- CSV
- JSON
- HTTP data
- Images
- Video, etc

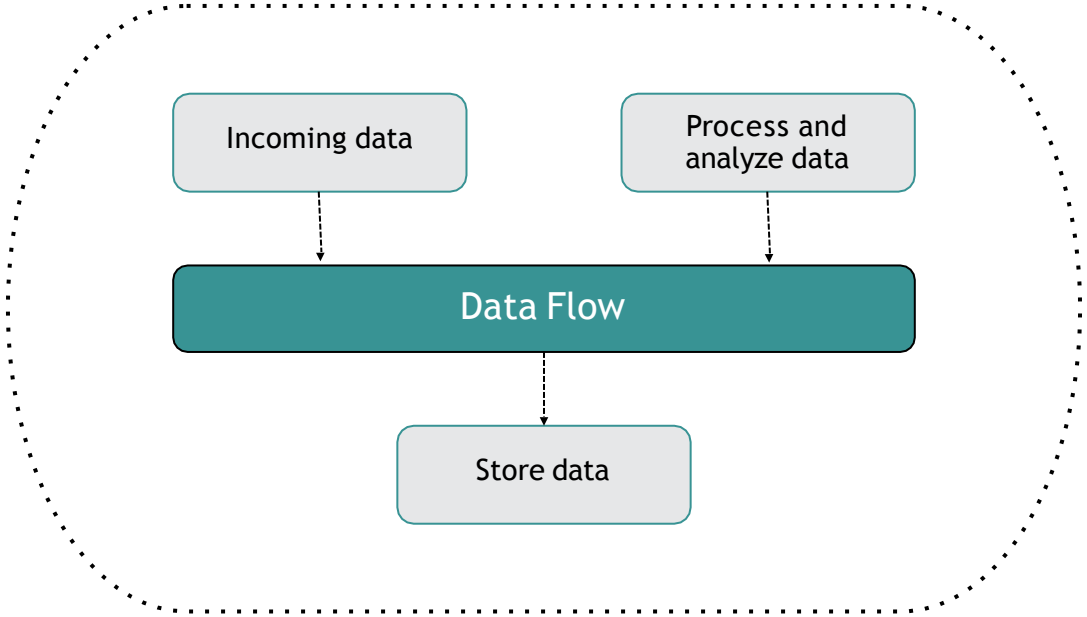
Destination can be:

- DB
- Brokers
- Storage
- Queues, etc

Data Pipeline/ETL

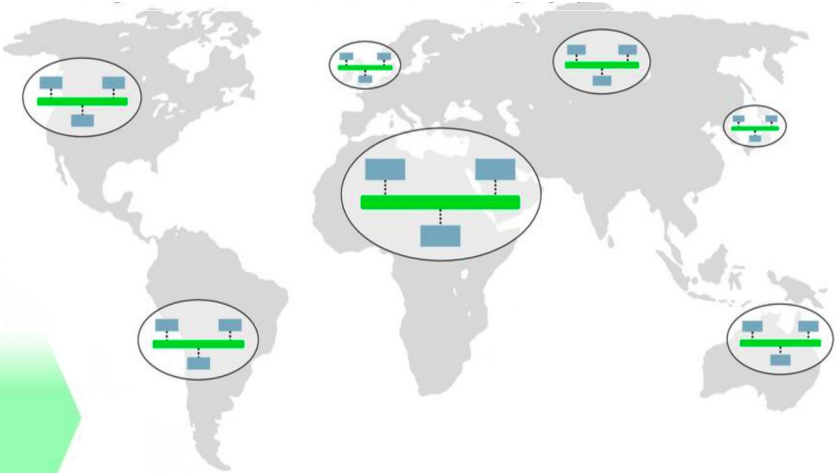


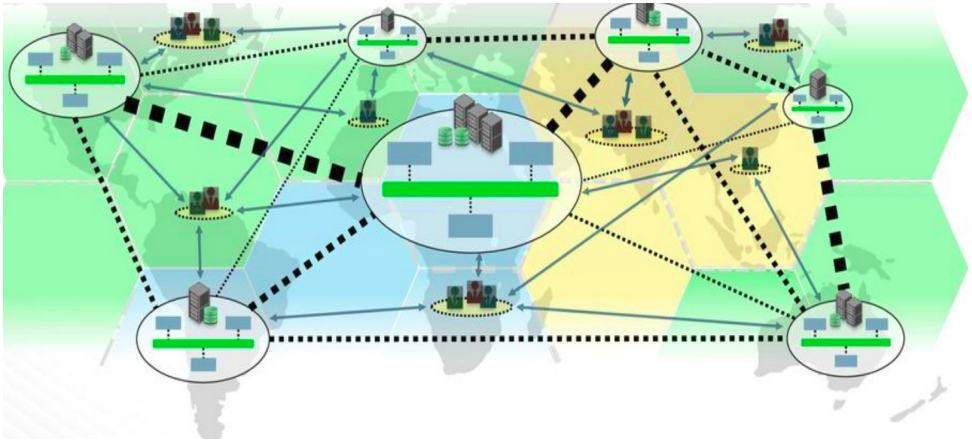
Simplified View of Enterprise Data Flow



Realistic View of Enterprise Data Flow

- Different organizations/business units across different locations
- Interacting with different business partners and customers
- Messaging problem at large scale





NiFi Overview

What Is NiFi?

Open-source software for automating and managing the flow of data between systems

Provides web-based User Interface to create, monitor, and control data flows

System with a highly configurable and modifiable data flow process to modify data at runtime

Why Use NiFi?

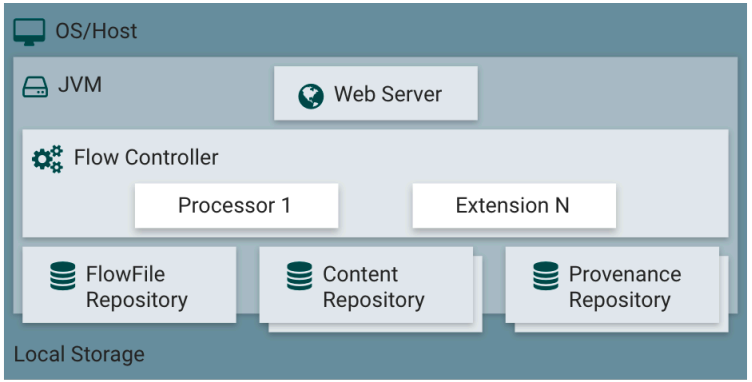
Reliable and secure transfer of data between systems

Delivery of data from sources to analytics platforms

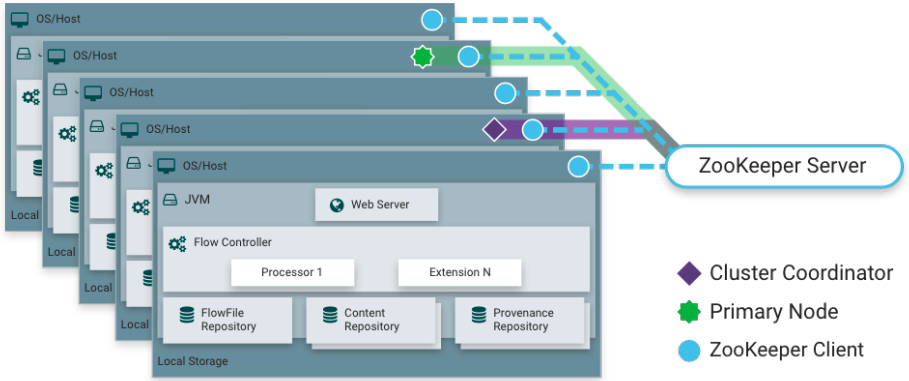
Enrichment and preparation of data:

- Conversion between formats
- Extraction
- Parsing
- Routing

NiFi Architecture



Standalone



Cluster

NiFi Use Cases

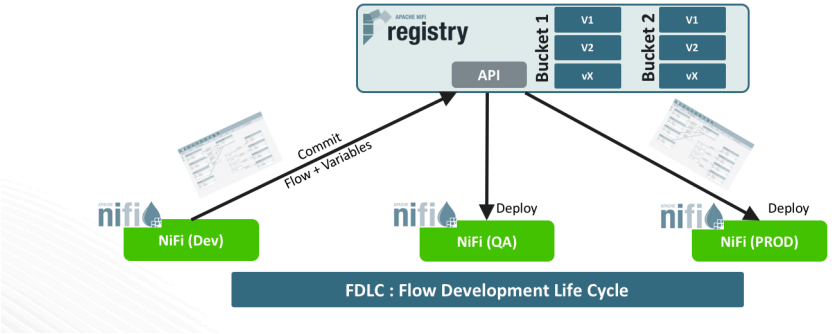
- **Insurance**
  - Risk & underwriting analysis
  - Claims analytics
  - Usage-based insurance
  - New product development
- **HealthCare**
  - Single view of Patient
  - Real-time vital sign monitoring
  - EMR optimization
  - Supply Chain Optimization
- **Telecommunication**
  - Single view of the customer
  - CDR analysis
  - Dynamic Bandwidth allocation
- **Oil & Gas- Industry**
  - Real-time monitoring
  - Single view of the Operation
  - Predictive Maintenance
  - Archive & Analytics
  - Unstructured data classification
- **Manufacturing**
  - Preventative Maintenance
  - Supply Chain Optimization
  - Quality Control
- **Financial Services**
  - Anti-money laundering
  - Fraud-Detection
  - Risk-data management

Companies Currently Using NiFi

COMPANY NAME	WEBSITE	HQ ADDRESS	CITY	STATE	ZIP	COUNTRY	TOP LEVEL INDUSTRY	SUB LEVEL INDUSTRY
Peraton	peraton.com	12975 Worldgate Dr	Herndon	VA	20170-6...	US	Telecommunications	Telephony & Wireless
Nike	nike.com	One Bowerman Drive	Beaverton	OR	97005-6...	US	Manufacturing	Textiles & Apparel
JPMorgan Chase	jpmorganchase.com	383 Madison Ave	New York	NY	10179-0...	US	Finance	Banking
T-Mobile	t-mobile.com	12920 Se 38th St	Bellevue	WA	98006	US	Telecommunications	Telephony & Wireless
U.S. Bank	usbank.com	800 Nicollet Mall	Minneapolis	MN	55402-7...	US	Finance	Banking

Best practices running NiFi

- Ideal to separate test/dev/production environments in NiFi
- You should break your flow into process groups
- Use a naming convention, use comments and labels
- Organize your projects into three parts ingestion, test & monitoring
- Use unique names for variable



Alternatives & Competitors

IBM InfoSphere DataStage

Azure Data Factory

Big Data Platform & Data Integration

Data Integration

AWS Glue

Intelligent Integration Platform (IIP)

Informatica PowerCenter

Fivetran

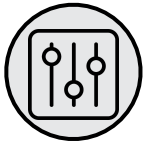
Qubole

Abstractions

- **FlowFile**
  - Data unit moving through the system
  - Contains context and attributes
- **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
  - Receives data via inputs
  - Sends data via outputs

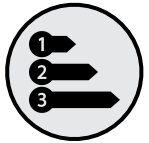
NiFi Key Features Overview

Key Features



Visual command and control

Data provenance



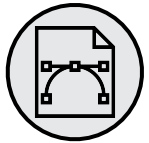
Data prioritization

Data buffering (Back Pressure)

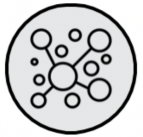
Control latency

Security

Input/output port in processing group

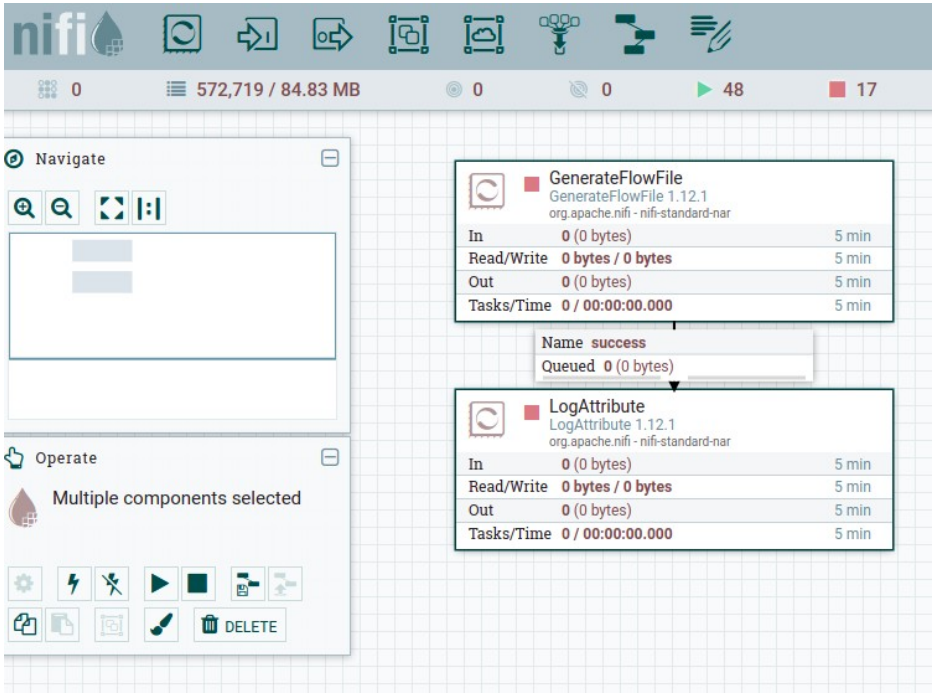


Extensibility



Scale-out clustering

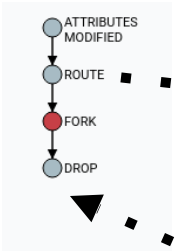
Visual Command and Control

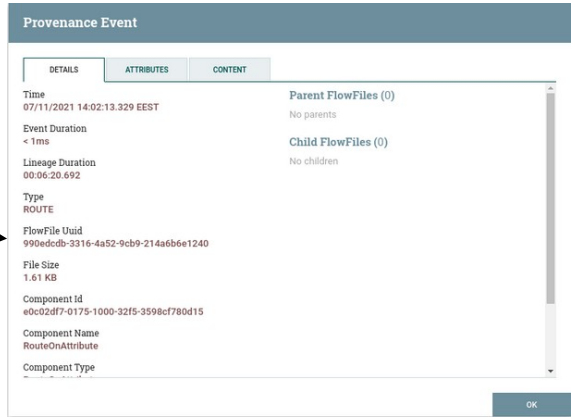


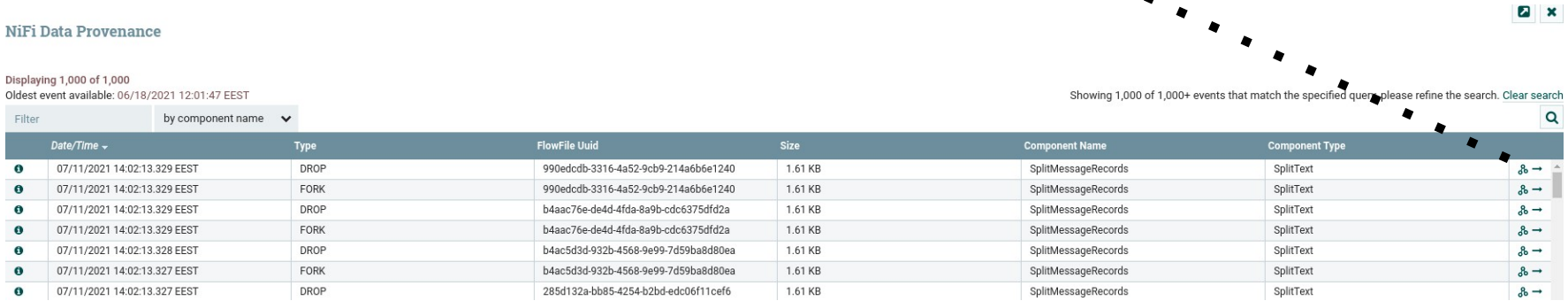
- Drag and drop processors to build a flow
- Start, stop, and configure components in real time
- View errors
- View corresponding error messages
- Create templates of processor with connections

Data Provenance/Lineage

- Track data at each point as it flows through a system
- View records, indexes, and events
- Handle fan-it/fan-out, merging, splitting data
- View attributes and content at given points in time



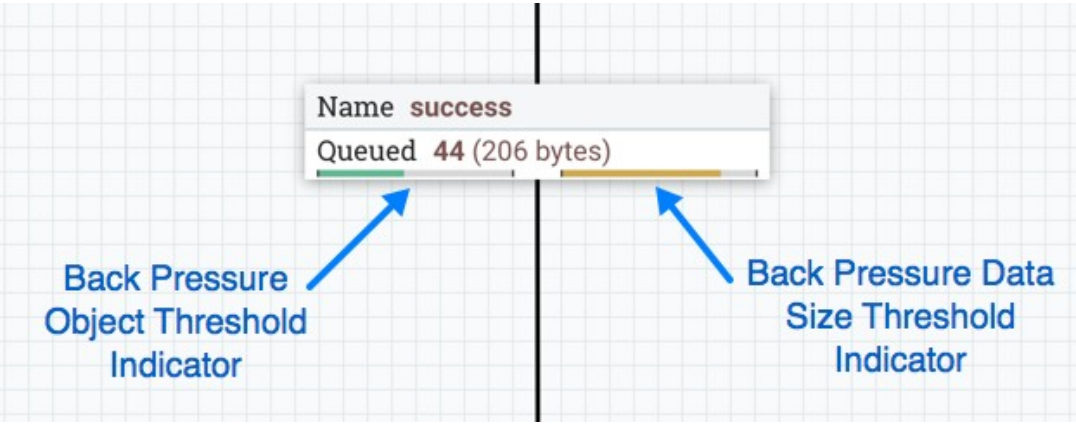




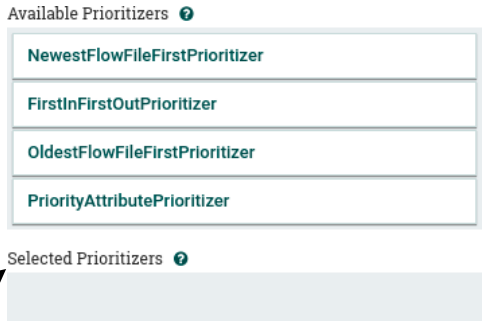
Date/Time	Type	FlowFile Uuid	Size	Component Name	Component Type
07/11/2021 14:02:13.329 EEST	DROP	990edcdb-3316-4a52-9cb9-214a6b6e1240	1.61 KB	SplitMessageRecords	SplitText
07/11/2021 14:02:13.329 EEST	FORK	990edcdb-3316-4a52-9cb9-214a6b6e1240	1.61 KB	SplitMessageRecords	SplitText
07/11/2021 14:02:13.329 EEST	DROP	b4aac76e-de4d-4fda-8a9b-cdc6375dfd2a	1.61 KB	SplitMessageRecords	SplitText
07/11/2021 14:02:13.329 EEST	FORK	b4aac76e-de4d-4fda-8a9b-cdc6375dfd2a	1.61 KB	SplitMessageRecords	SplitText
07/11/2021 14:02:13.328 EEST	DROP	b4ac5d3d-932b-4568-9e99-7d59ba8d80ea	1.61 KB	SplitMessageRecords	SplitText
07/11/2021 14:02:13.327 EEST	FORK	b4ac5d3d-932b-4568-9e99-7d59ba8d80ea	1.61 KB	SplitMessageRecords	SplitText
07/11/2021 14:02:13.327 EEST	DROP	285d132a-bb85-4254-b2bd-edc06f11cef6	1.61 KB	SplitMessageRecords	SplitText

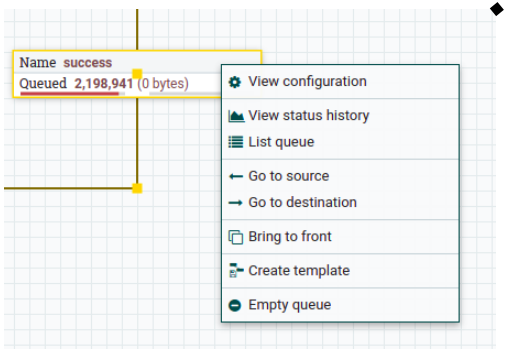
Back Pressure

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



Data Prioritization

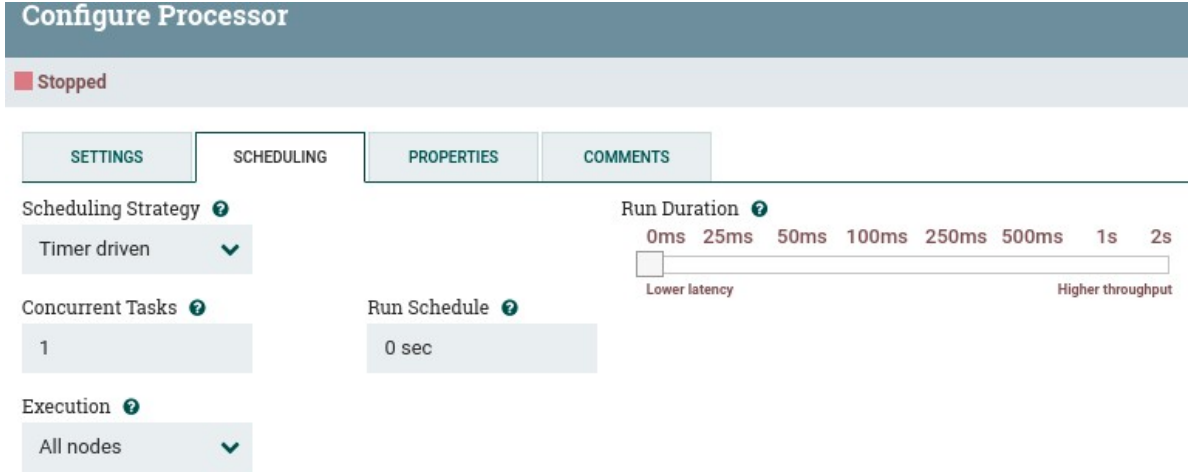




- Configure a prioritizer per connection
- Determine what is important for your data - time, arrival order, importance of a data set
- Funnel many connections down to a single connection to prioritize data sets
- Develop your custom prioritizer if needed

Latency vs Throughput

- Choose between lower latency or higher throughput on each processor
- Higher throughput allows framework to batch together all operations for selected amount of time to improve performance
- Processor developer determines whether to support this by using @SupportsBathching annotations

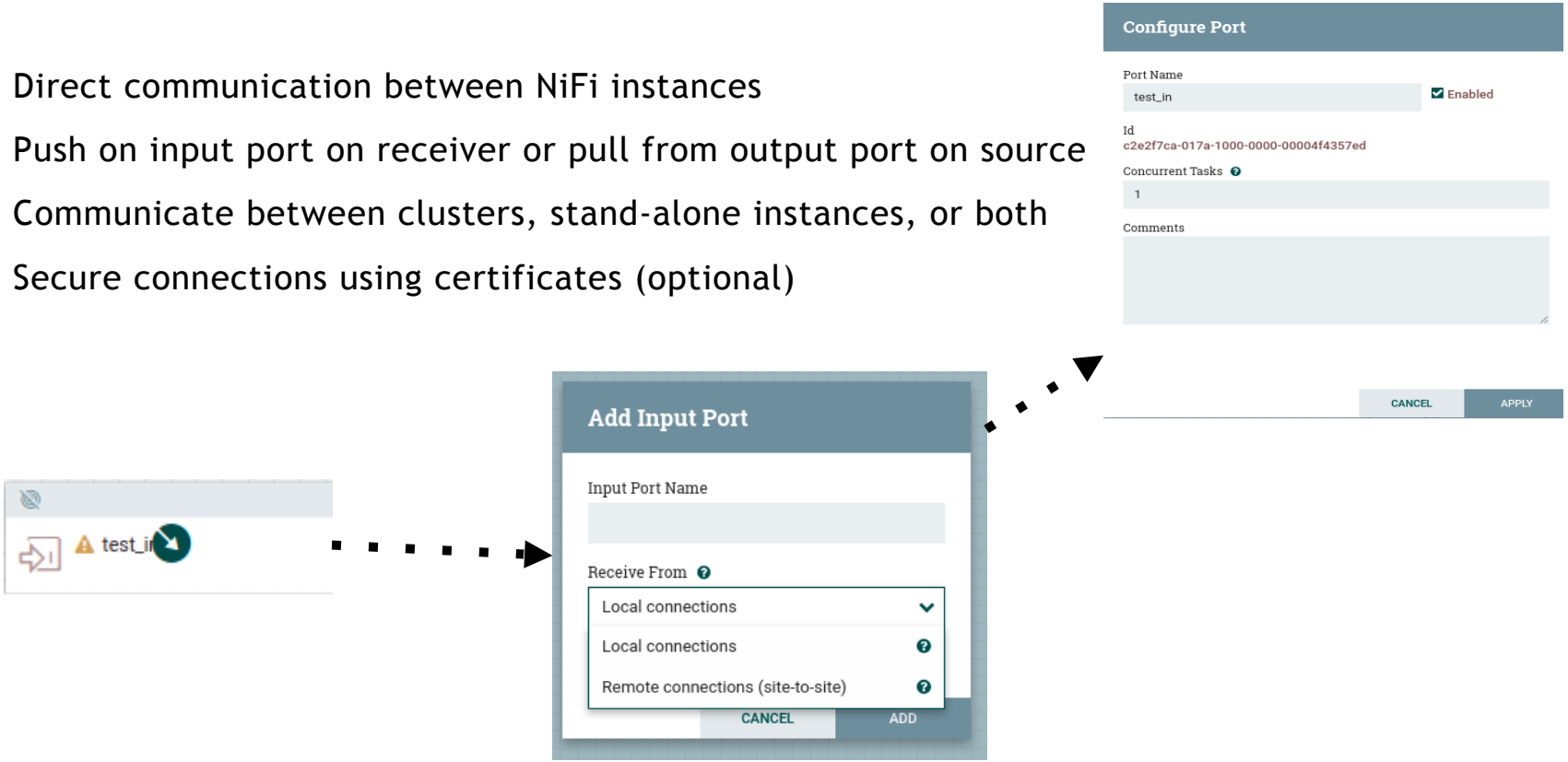


Security

- **Control plane**
  - Pluggable authentication
    - *2-Way SSL*
    - *LDAP*
    - *Kerberos*
  - Pluggable authorization
    - *File-based authority provider out of the box*
    - *Multiple roles to define 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 of audit trail of data

Input/Output Port in Processing Group

- Direct communication between NiFi instances
- Push on input port on receiver or pull from output port on source
- Communicate between clusters, stand-alone instances, or both
- Secure connections using certificates (optional)



Extensibility

- **Build from scratch with extensions in mind**
- **Service-loader pattern for**
  - Processors
  - Controller services
  - Reporting tasks
  - Prioritizers
- **Extensions packaged as NARs (NiFi Archives)**
  - Deploy NiFi lib directory and restart
  - Add NAR to extensions folder and pull NiFi up for 5 seconds
  - Same model as standard components

Demo

Thanks :)
Any questions?

ihor.didyk@globalllogic.com

Apache NiFi Community