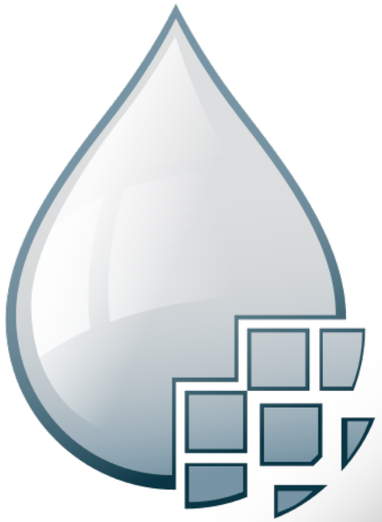


Let Your Data Flow with Apache NiFi

Bryan Bende – Staff Software Engineer @Hortonworks

DEVNEXUS 2018

HORTONWORKS®
POWERING THE FUTURE OF DATA™

About Me

- Staff Software Engineer @ Hortonworks
- Apache NiFi PMC & Committer
- Contact Info
  - Email - bbende@hortonworks.com
  - Twitter - @bbende
  - Blog - https://bryanbende.com

HORTONWORKS

Agenda

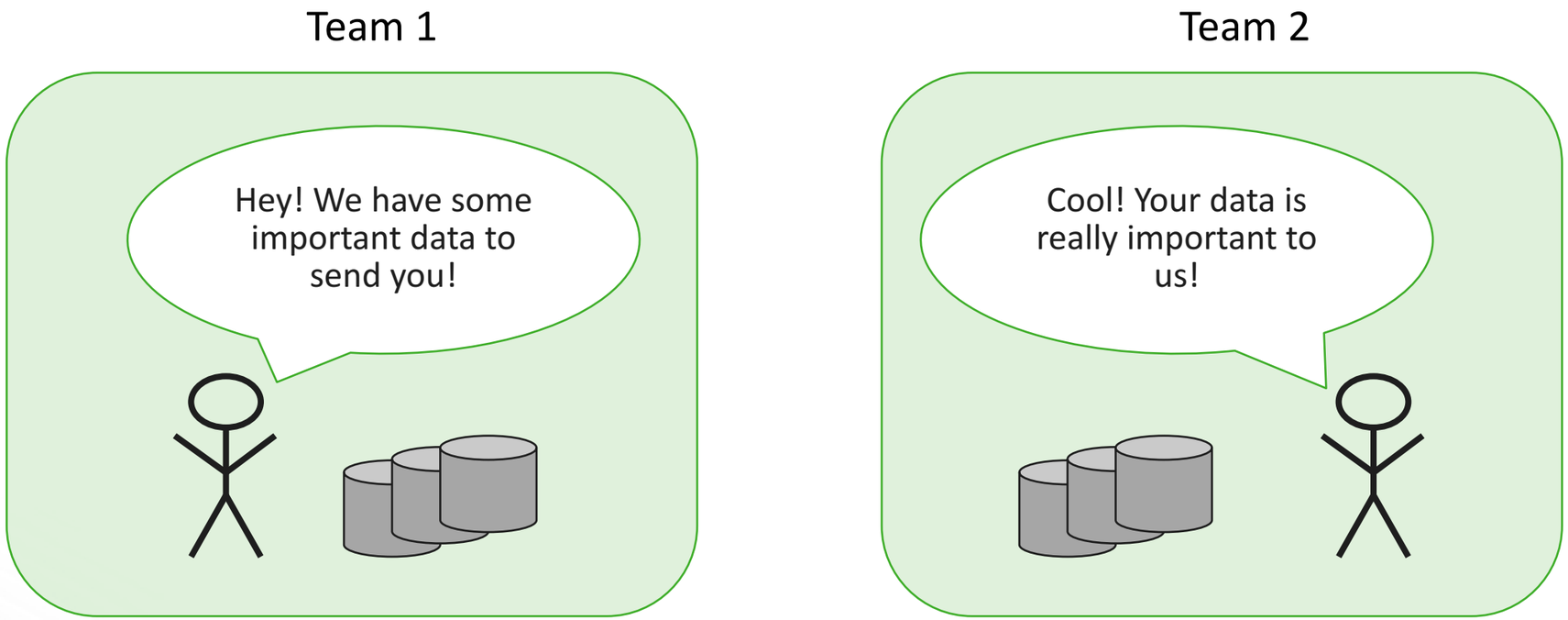
- The Problem
- Apache NiFi Overview
- Developing Extensions
- Demo Cool Stuff!
- Q&A

HORTONWORKS

The Problem

HORTONWORKS®

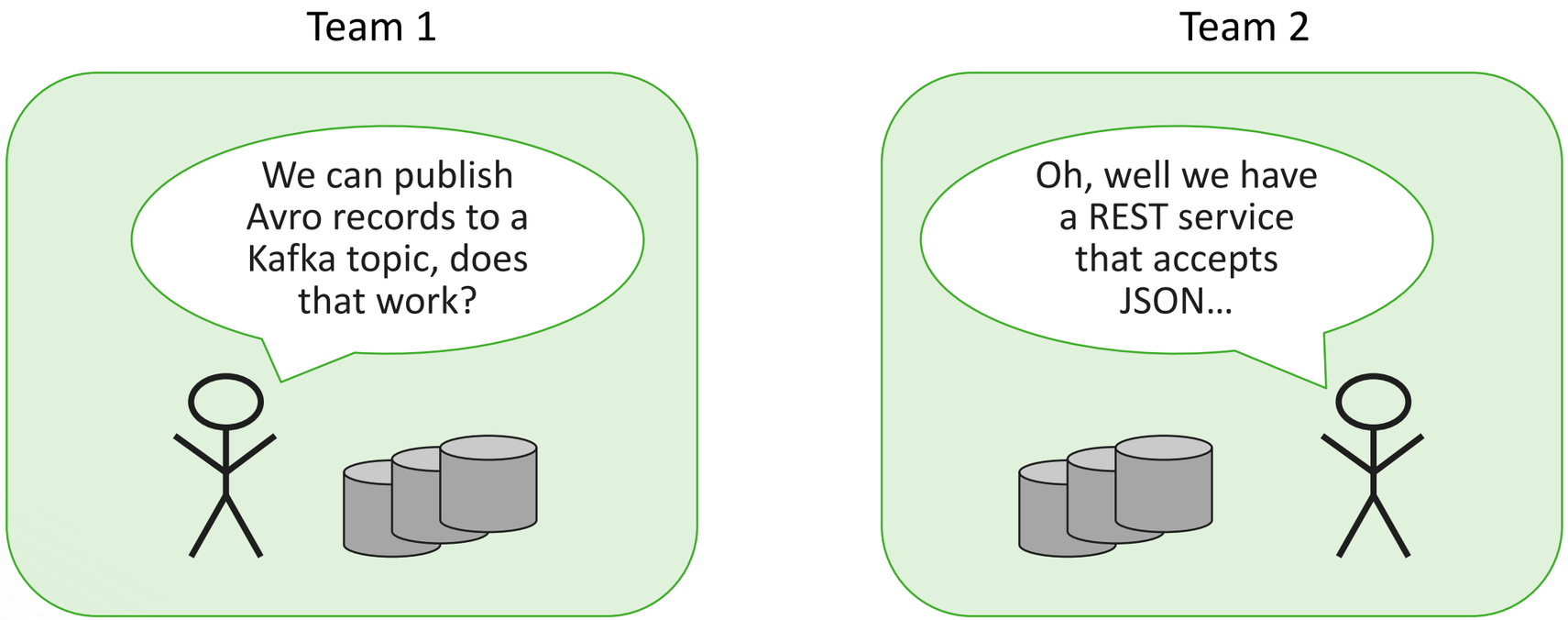
It starts out so simple…



This should be easy right?...

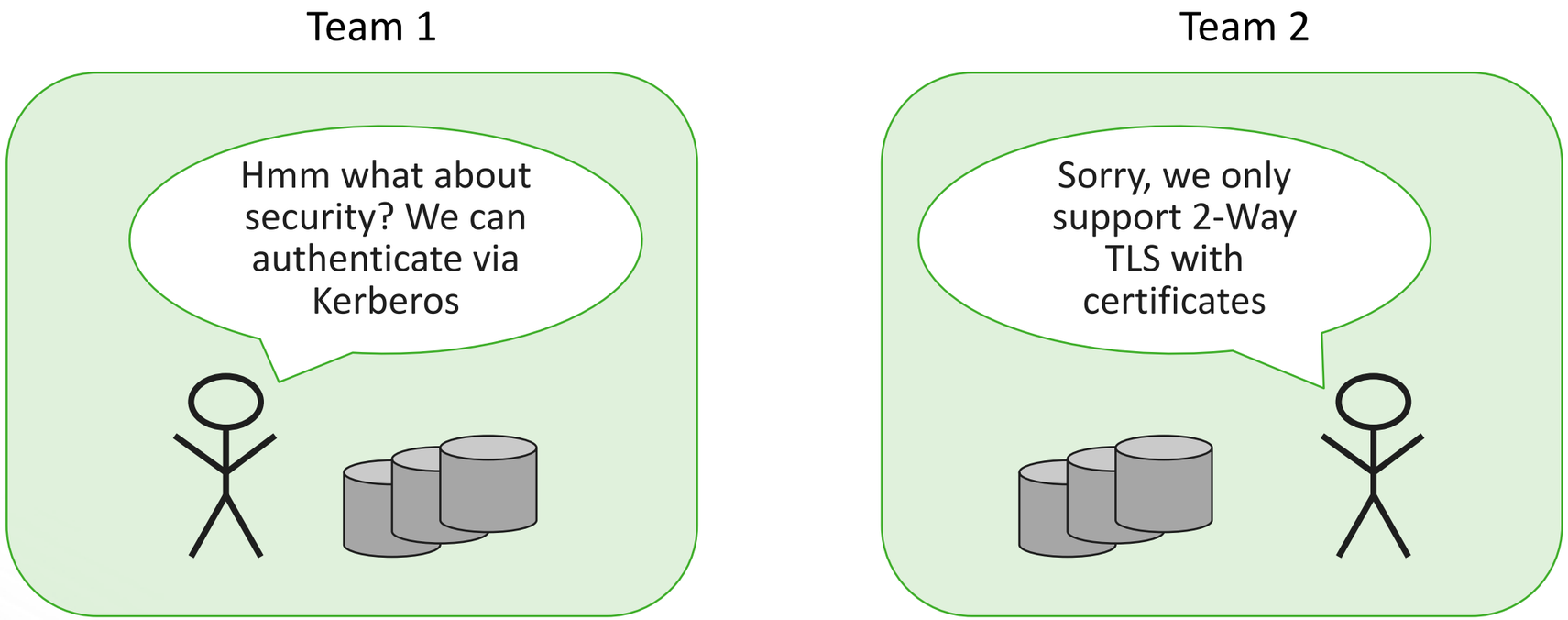
HORTONWORKS

But what about formats & protocols?



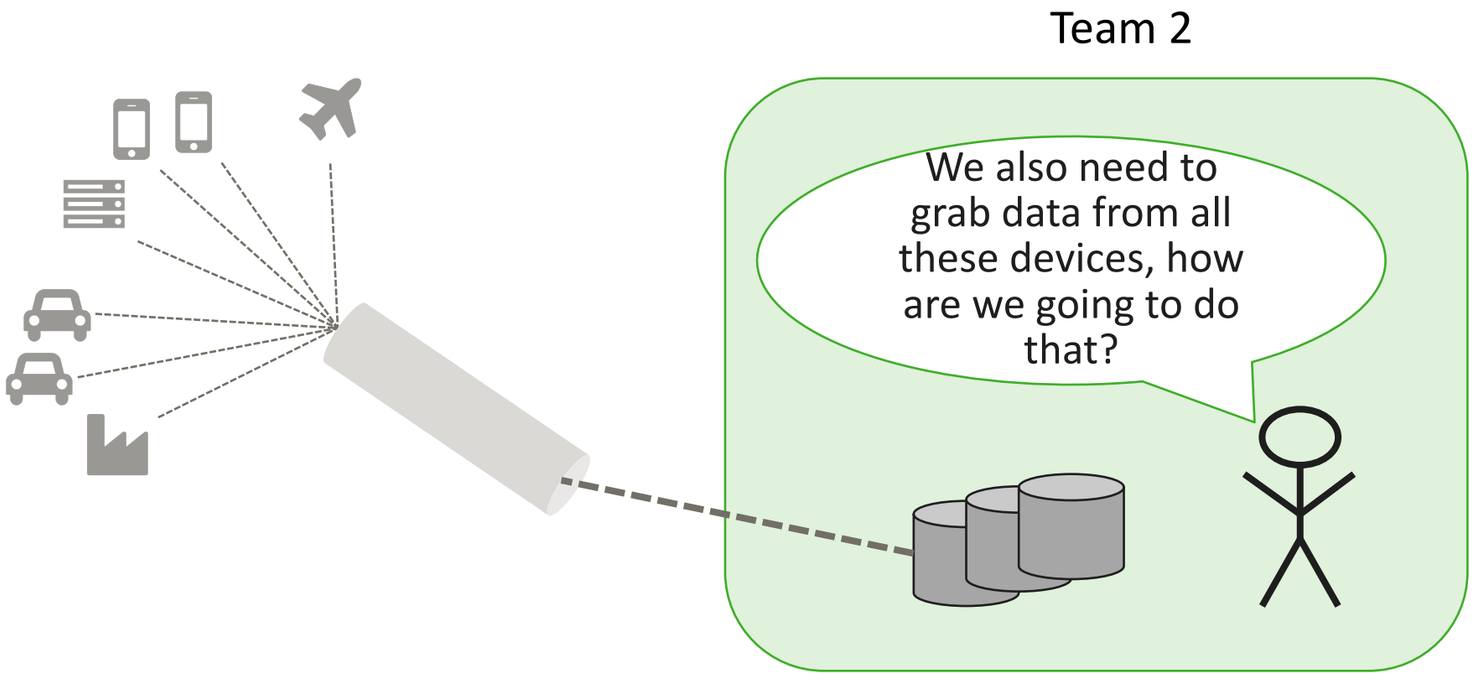
HORTONWORKS

And what about security & authentication?



HORTONWORKS

And what about all these devices at the edge?



HORTONWORKS

Wouldn't it be nice if there was a tool that could help these teams?

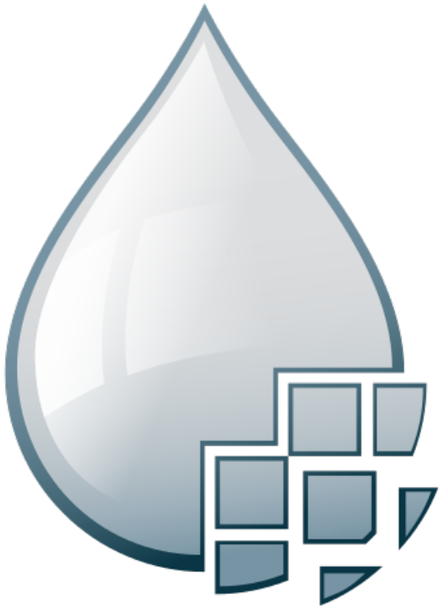
HORTONWORKS

Enter Apache NiFi…

HORTONWORKS

Apache NiFi

- Created to address the challenges of global enterprise dataflow
- Key features:
  - **Interactive Command and Control**
  - **Version Control & Deployment**
  - **Data Lineage (Provenance)**
  - **Data Prioritization**
  - **Data Buffering/Back-Pressure**
  - **Record Processing**
  - **Scale Out Clustering**
  - **Scale Down Agent**
  - **Extensibility**



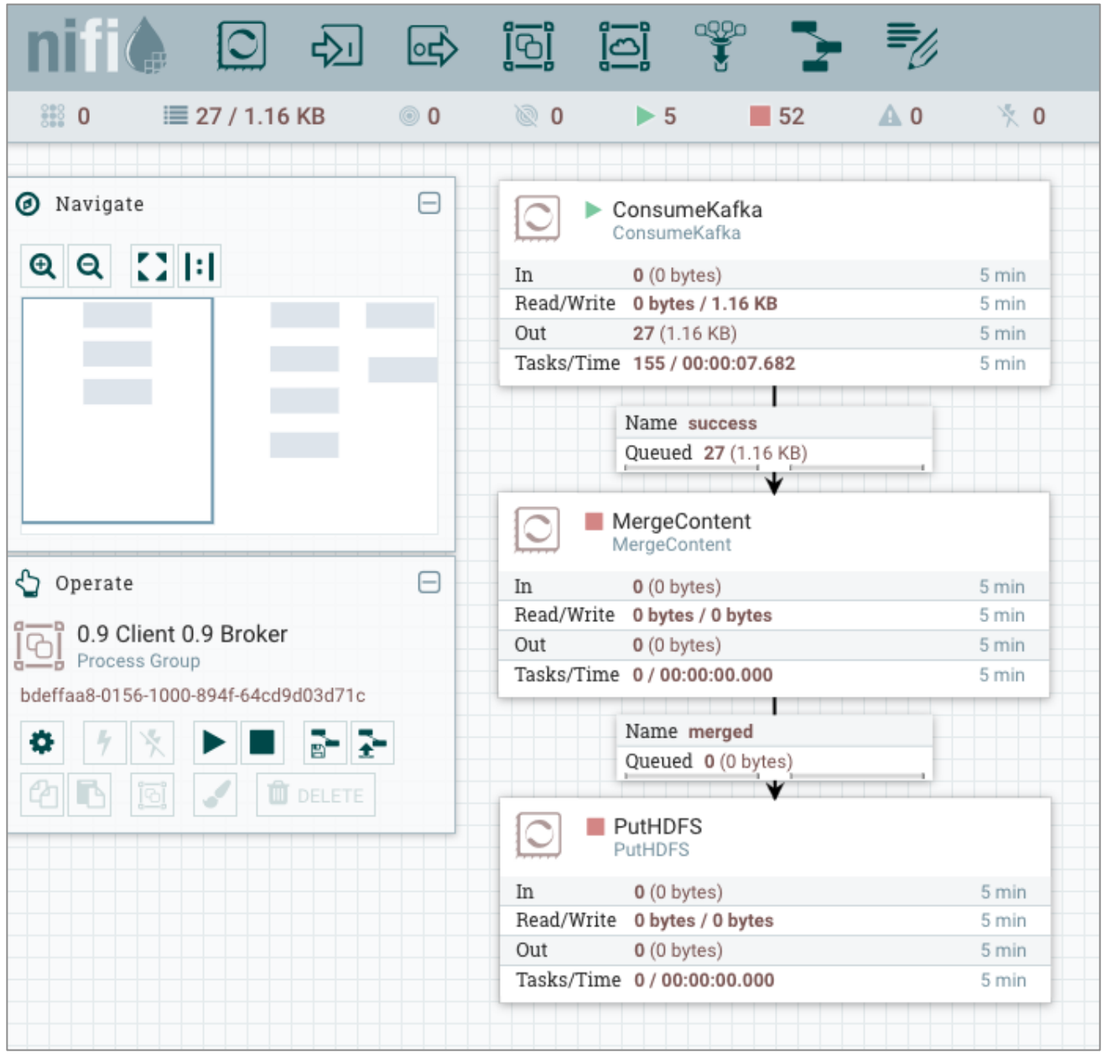
HORTONWORKS

NiFi Core Concepts

FBP Term	NiFi Term	Description
Information Packet	FlowFile	Each object moving through the system.
Black Box	FlowFile Processor	Performs the work, doing some combination of data routing, transformation, or mediation between systems.
Bounded Buffer	Connection	The linkage between processors, acting as queues and allowing various processes to interact at differing rates.
Scheduler	Flow Controller	Maintains the knowledge of how processes are connected, and manages the threads and allocations thereof which all processes use.
Subnet	Process Group	A set of processes and their connections, which can receive and send data via ports. A process group allows creation of entirely new component simply by composition of its components.

Interactive Command & Control

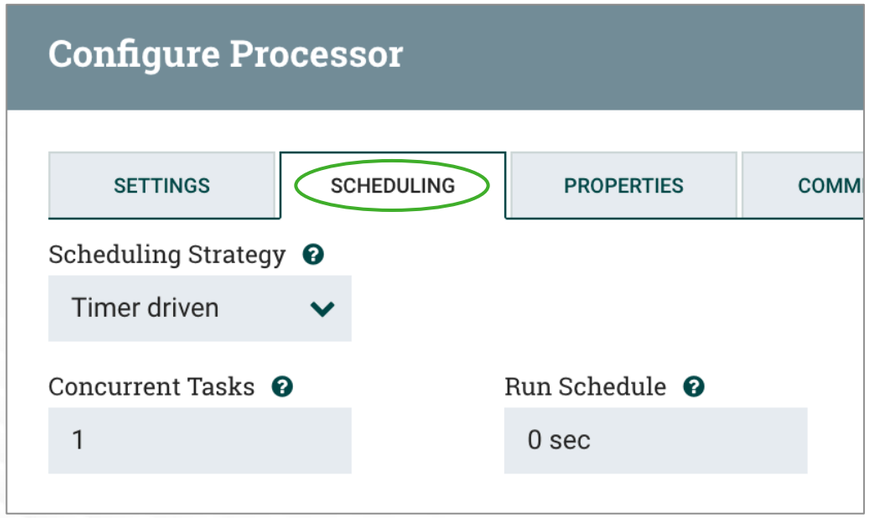
- Drag & drop processors to build a flow
- Start, stop, & configure components in real-time
- View errors & corresponding messages
- View statistics & health of the dataflow
- Create shareable templates of common flows

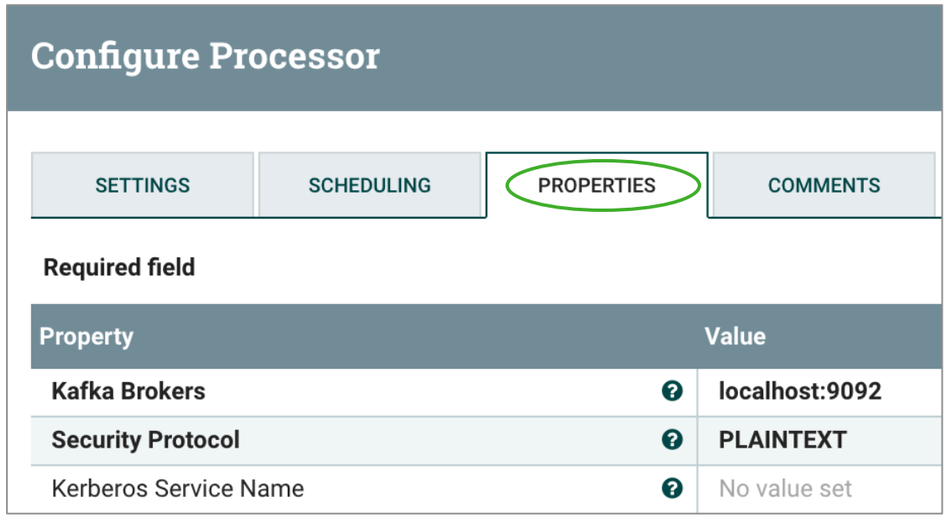


HORTONWORKS

Scheduling & Configuration

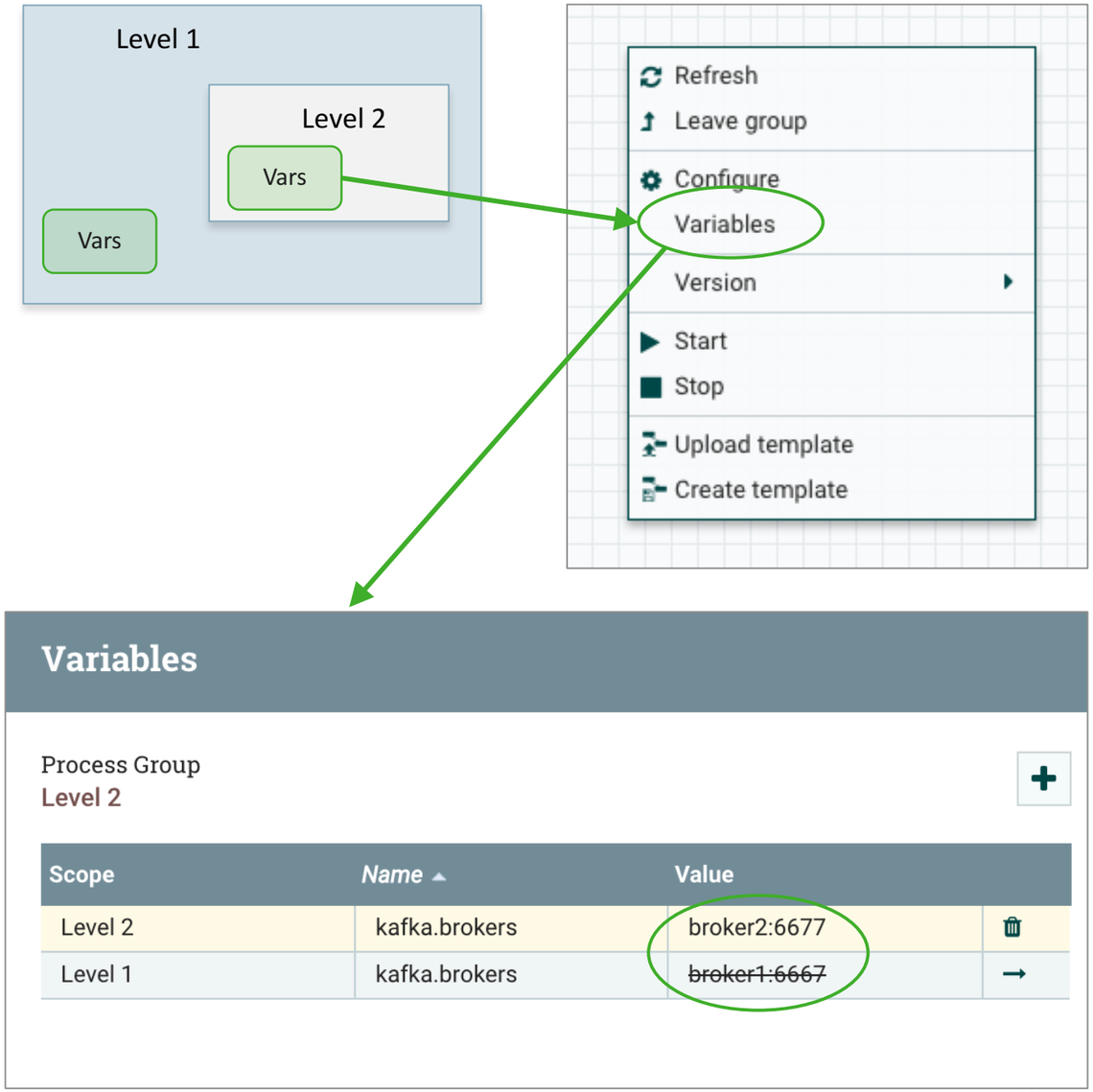
- Scheduling
  - Timer Driven or CRON
  - “Timer Driven - 0 sec” = go-fast when data is available
  - Concurrent tasks = # of threads executing the processor
- Properties
  - Enter processors specific configuration
  - Easily see required properties
  - Tooltips for descriptions





Variables

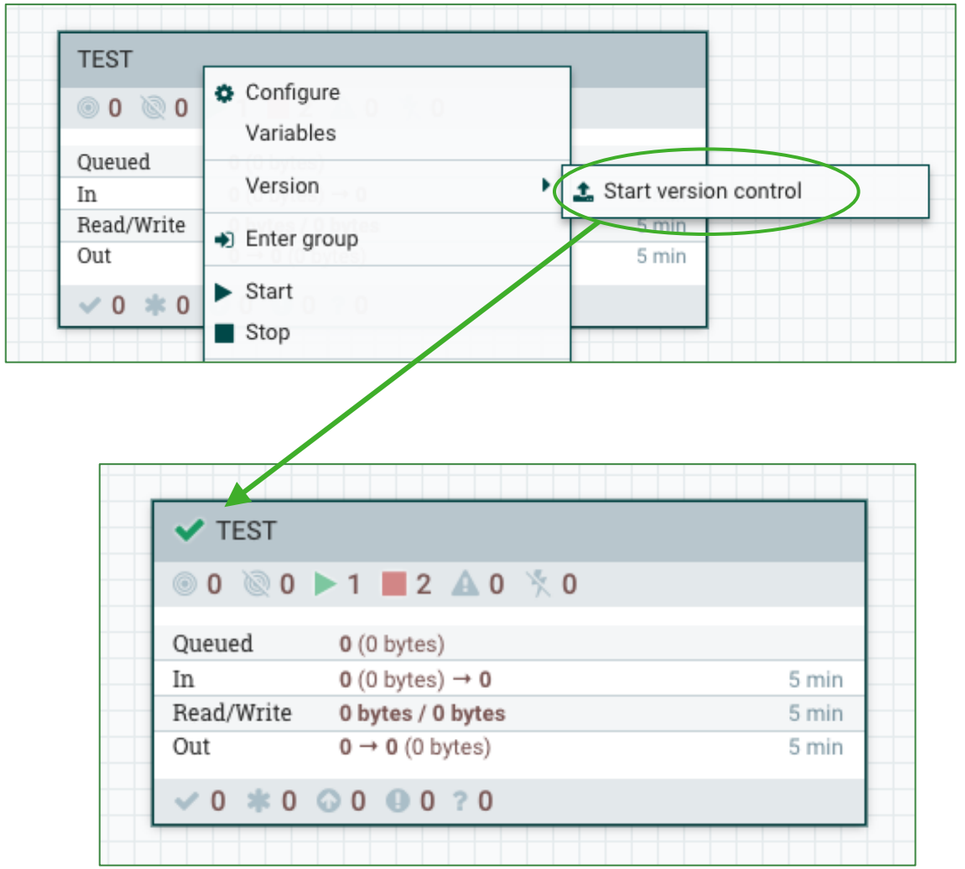
- Parameterize configuration like connection strings, file paths, etc.
- Referenced via Expression language
  - `Kafka Brokers = ${kafka.brokers}`
- Variables associated with a process group
- Right-click on canvas to view variables for current group
- Hierarchical order of precedence, resolve closest reference to component
- Editing variables automatically restarts any components referencing the variables



Version Control & Deployment

- NiFi Registry - sub-project of Apache NiFi
  - https://github.com/apache/nifi-registry
  - https://issues.apache.org/jira/projects/NIFIREG
- Complimentary application, central location for storage/management of “versioned” resources
- Initial capability to store and retrieve “versioned flows”
- Integration on NiFi side
  - Start/Stop version control of a process group
  - Change version (upgrade/downgrade)
  - Import new process group from a version

APACHE NIFI registry



HORTONWORKS

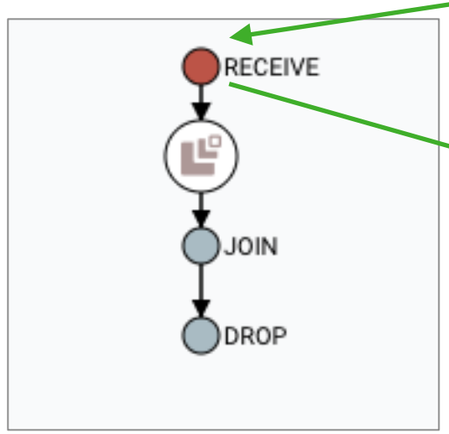
Provenance/Lineage

Displaying 13 of 104
Oldest event available: 11/15/2016 13:34:50 EST
Showing the most recent events.

ConsumeKafka | by component name

	Date/Time	Type	FlowFile Uuid	Size	Component Name	Component Type	
ⓘ	11/15/2016 13:35:03.8...	RECEIVE	379fc4f6-60e0-4151-9743-28...	44 bytes	ConsumeKafka	ConsumeKafka	⋔→
ⓘ	11/15/2016 13:35:02.7...	RECEIVE	78f8c38b-89fc-4d00-a8d8-51...	44 bytes	ConsumeKafka	ConsumeKafka	⋔→
ⓘ	11/15/2016 13:35:01.6...	RECEIVE	2bcd5124-bb78-489f-ad8a-7...	44 bytes	ConsumeKafka	ConsumeKafka	⋔→

- 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

Attribute Values

filename
328717796819631
No value previously set

kafka.offset
44815
No value previously set

kafka.partition
6
No value previously set

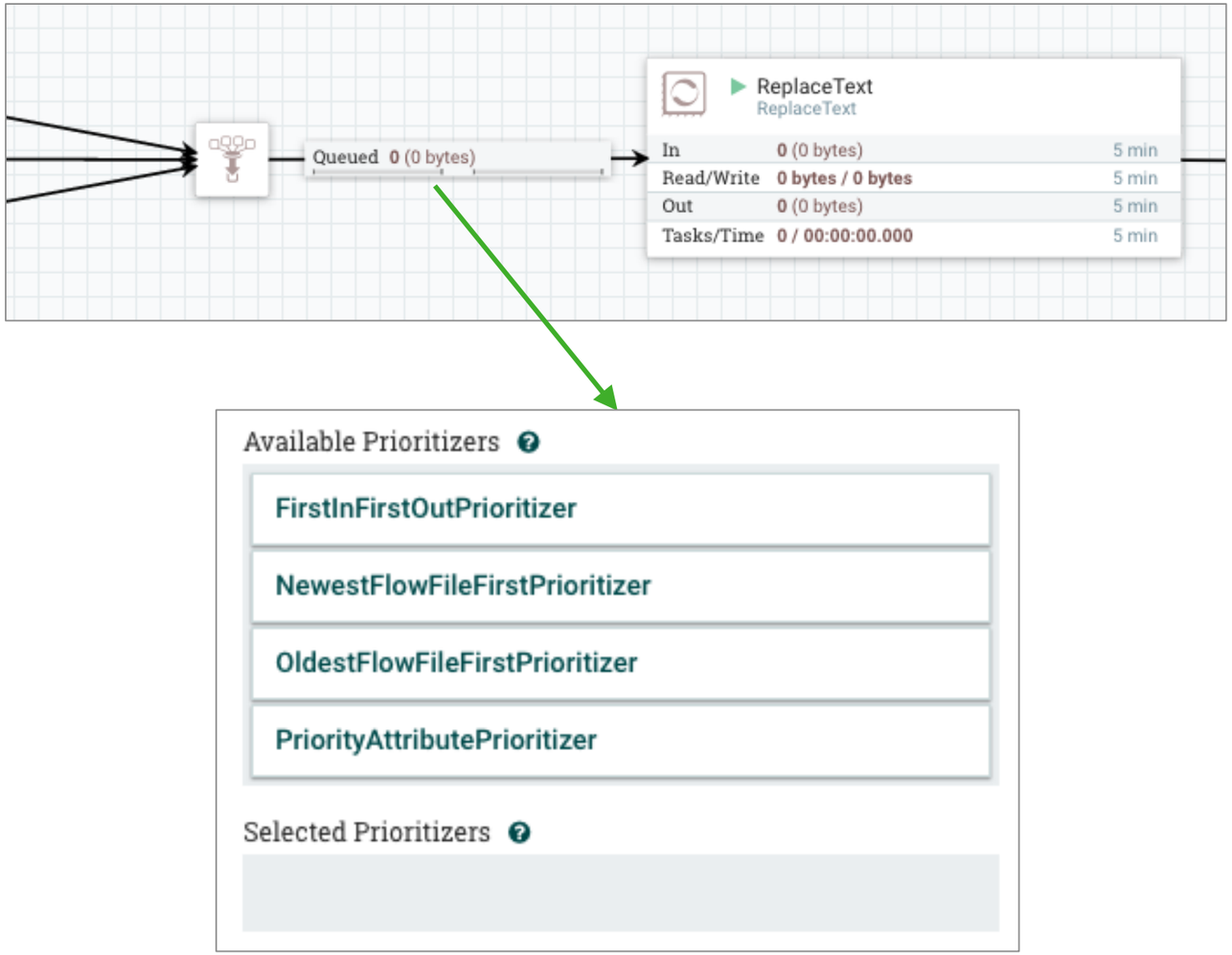
kafka.topic
nifi-testing
No value previously set

path
./
No value previously set

uuid
2057fd62-3853-4a66-8996-d95f2f073a05
No value previously set

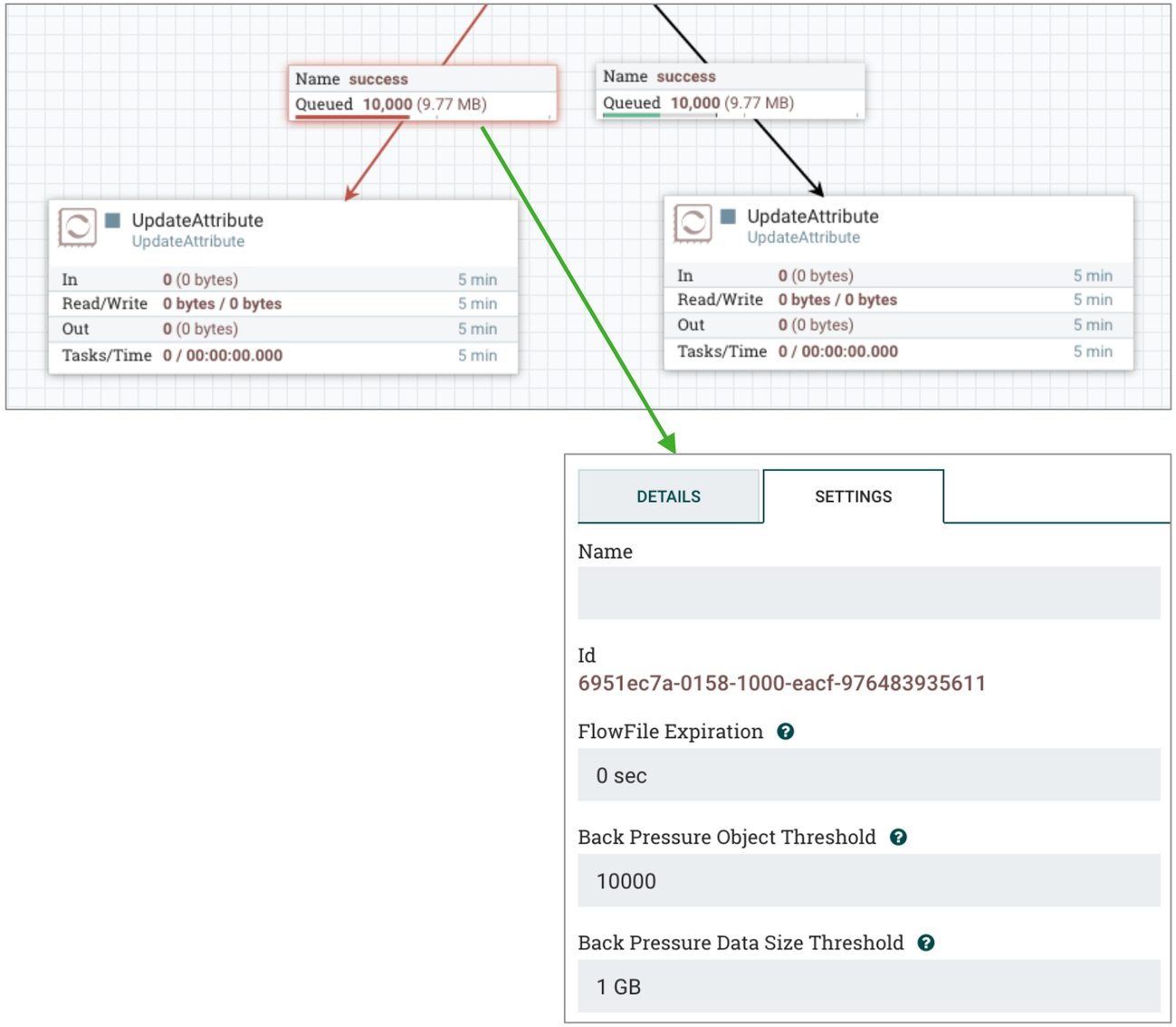
Prioritization

- 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



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

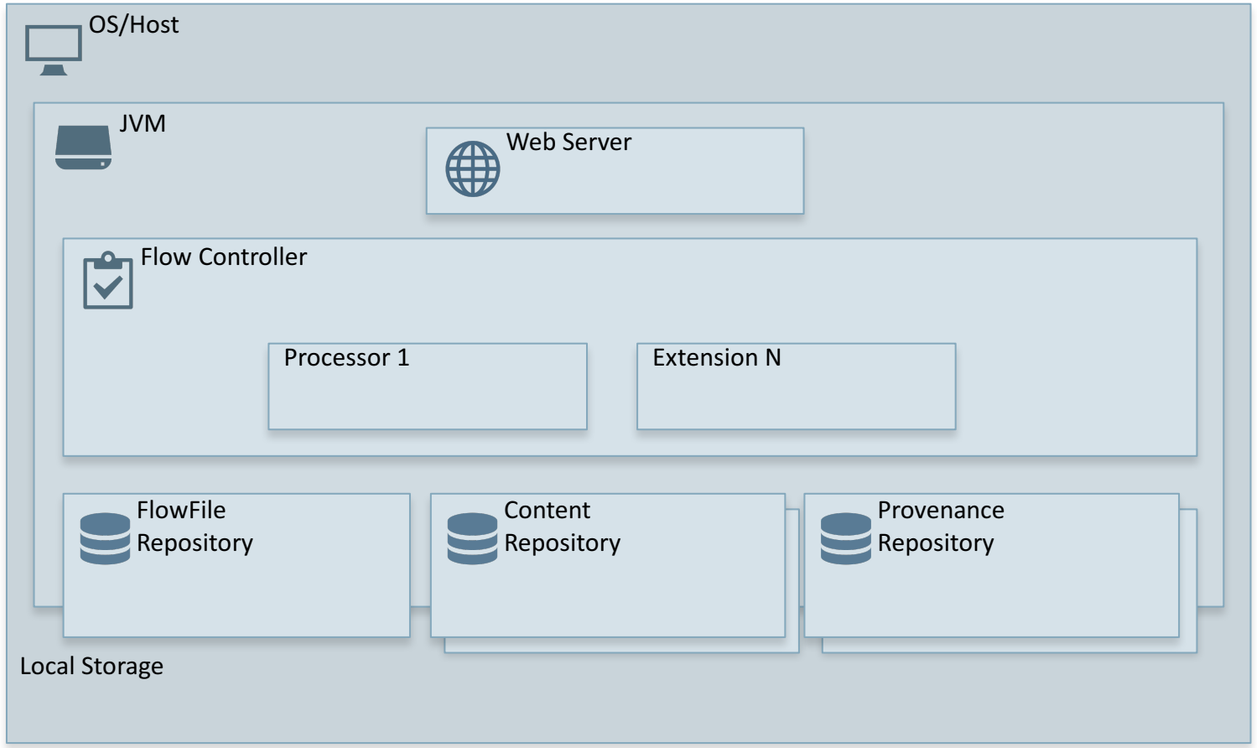


Record Processing

- Remain data agnostic, but process “records” when appropriate
  - Released in Apache NiFi 1.2.0 (May 2017), improvements in 1.3.0 and 1.4.0
- Centralize the logic for reading/writing records into controller services
  - *Readers/Writers for CSV, JSON, Avro, etc.*
- Provide standard processors that operate on records
  - *ConvertRecord, QueryRecord, PartitionRecord, UpdateRecord, etc.*
- Provide integration with schema registries
  - *Local Schema Registry*
  - *Hortonworks Schema Registry*
  - *Confluent Schema Registry*
- Significant performance improvements processing records in place vs. splitting

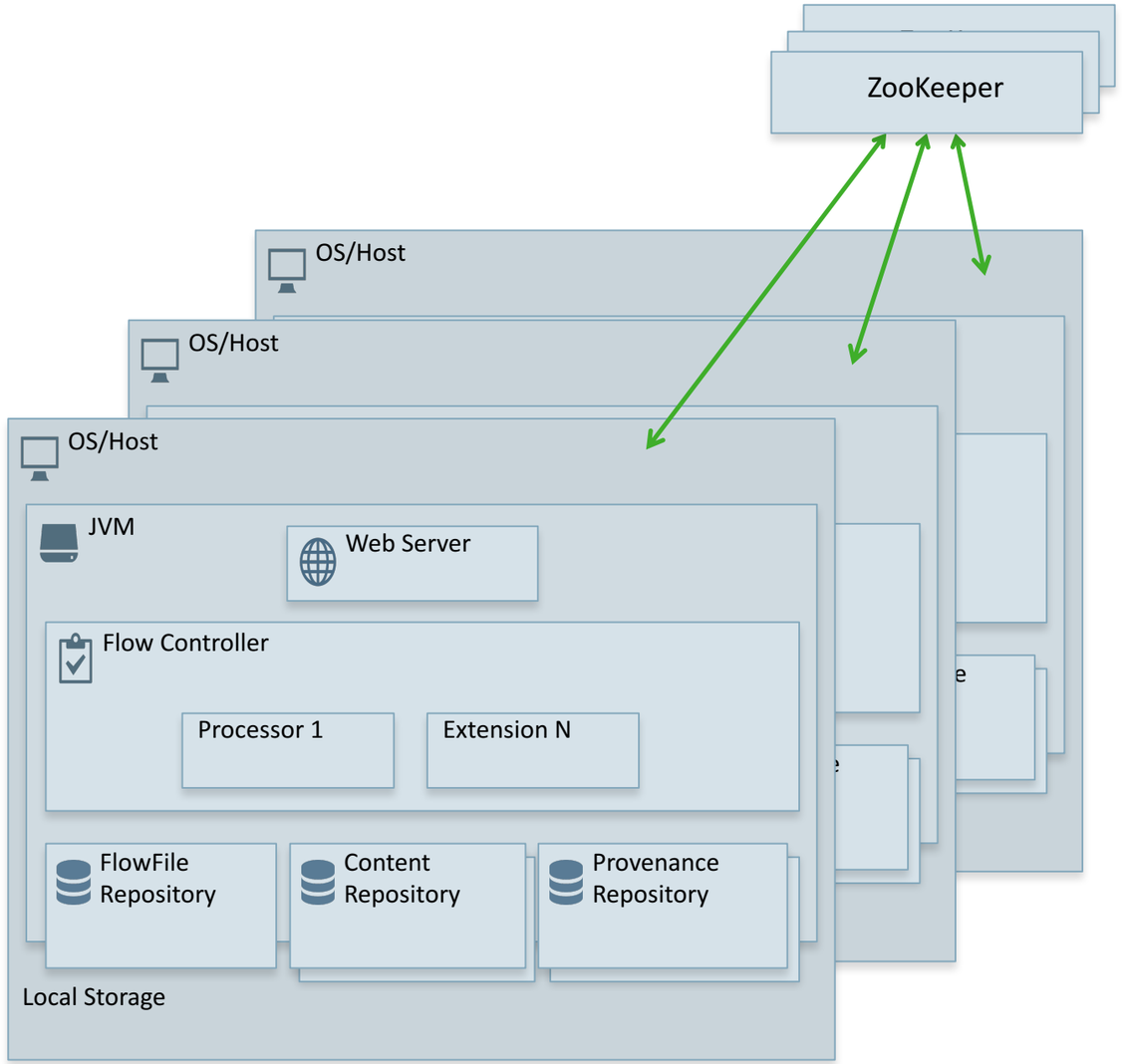
Standalone Architecture

- FlowFile Repository
  - Write Ahead Log
  - State of every FlowFile
  - Pointers to content repository (pass-by-reference)
- Content Repository
  - FlowFile content
  - Copy-on-write
- Provenance Repository
  - Write Ahead Log + Lucene Indexes
  - Store & search lineage events



Scaling Up – Cluster Architecture

- Same dataflow on each node, data partitioned across cluster
- Access the UI from any node
- ZooKeeper for auto-election of Cluster Coordinator & Primary Node
- Cluster Coordinator receives heartbeats from other nodes, manages joining/ disconnecting
- Primary Node for scheduling processors on a single node



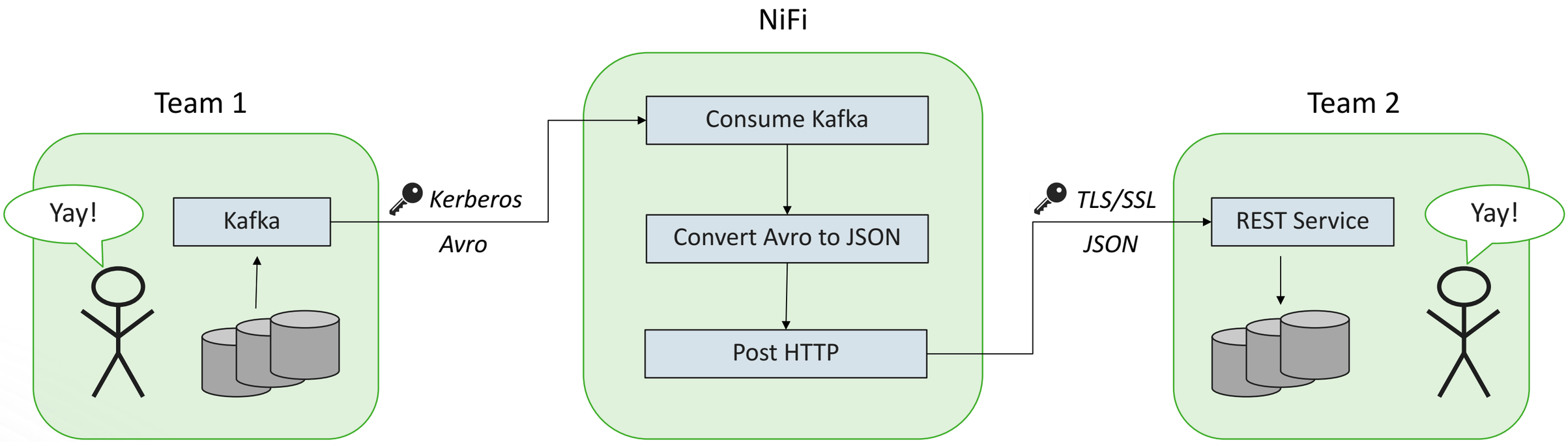
HORTONWORKS

Scaling Down

- MiNiFi - sub-project of Apache NiFi
  - Agent model to collect data at the edge
  - Design & deploy, instead of command & control
  - Transfer data back to a central NiFi via site-to-site
- Java & C++ Distributions
  - MiNiFi Java
    - Smaller headless version of NiFi
    - https://github.com/apache/nifi-minifi
  - MiNiFi C++
    - Run where the JVM can't, target resource constrained environments
    - https://github.com/apache/nifi-minifi-cpp
  - MiNiFi Toolkit
    - Convert NiFi Template XML to MiNiFI YAML

HORTONWORKS

Bringing this full circle…



HORTONWORKS

Building Extensions

HORTONWORKS

Extensibility

- Already 200+ processors, but someone will always need “just one more”
- Built from the ground up with extensions in mind
- Service-loader pattern for…
  - Processors
  - Controller Services
  - Reporting Tasks
- Extensions packaged as NiFi Archives (NARs)
  - Deploy NiFi lib directory and restart
  - Provides ClassLoader isolation
  - Same model as standard components

HORTONWORKS

Maven Archetype for NARs

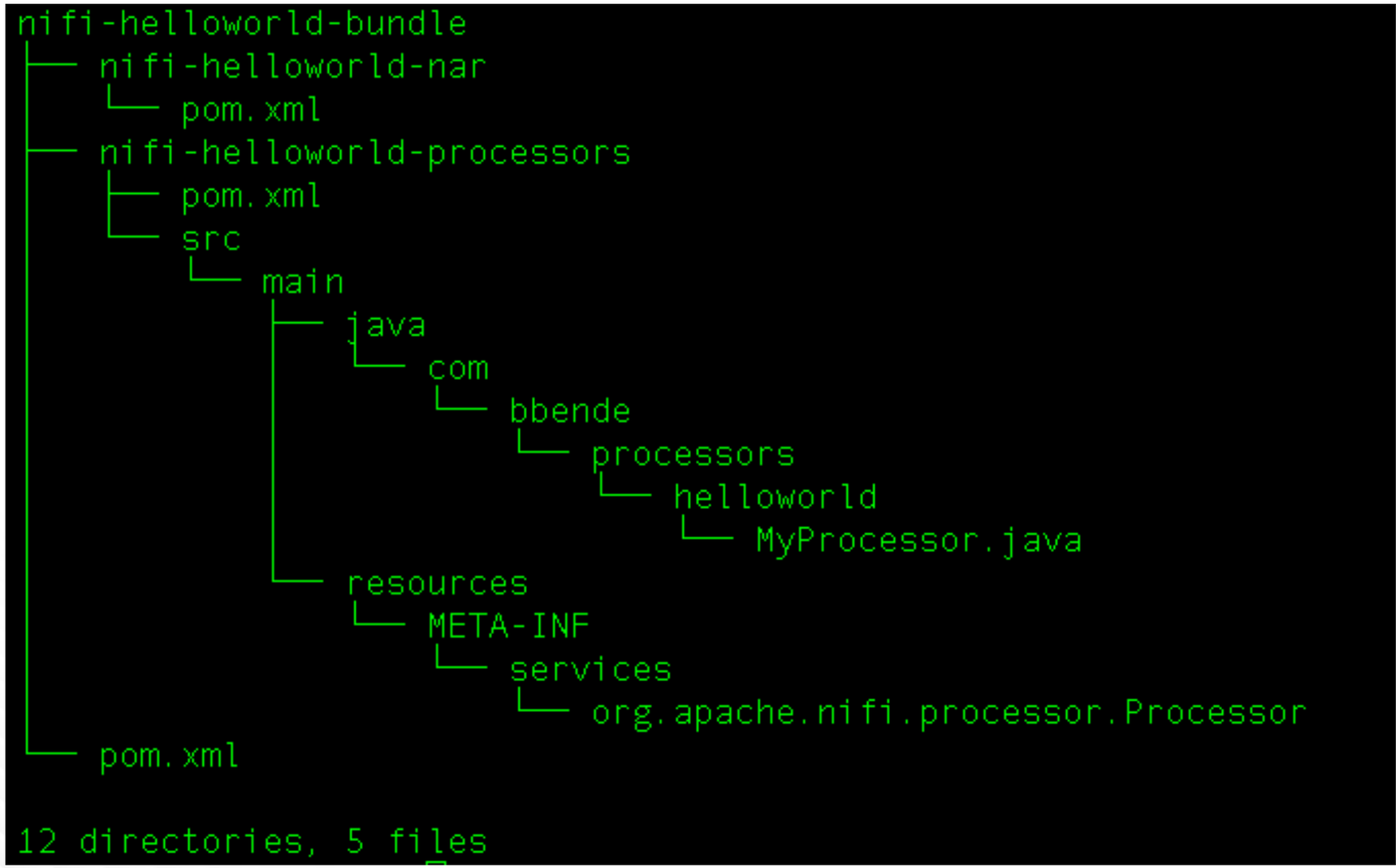
```
mvn archetype:generate \
     -DarchetypeGroupId=org.apache.nifi \
     -DarchetypeArtifactId=nifi-processor-bundle-archetype \
     -DarchetypeVersion=1.5.0 \
     -DnifiVersion=1.5.0
```

```
Define value for property 'groupId': : com.bbende
Define value for property 'artifactId': : nifi-helloworld-bundle
Define value for property 'version':  1.0-SNAPSHOT: :
Define value for property 'artifactBaseName': : helloworld
Define value for property 'package':  com.bbende.processors.helloworld: :
Using property: nifiVersion = 1.5.0
```

https://cwiki.apache.org/confluence/display/NIFI/Maven+Projects+for+Extensions

HORTONWORKS

Project Structure



HORTONWORKS

Implementing a Processor

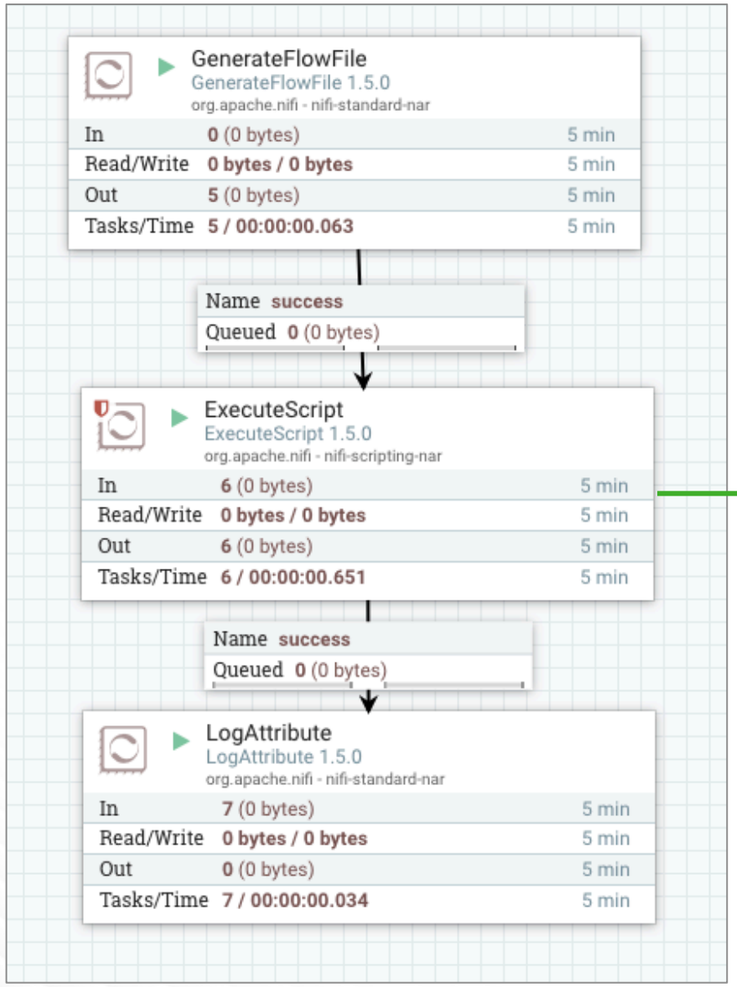
```
@Override
public void onTrigger(ProcessContext context, ProcessSession session)
     throws ProcessException {

    FlowFile flowFile = session.get();
    if (flowFile == null) {
        return;
    }

    flowFile = session.write(flowFile, (in, out) -> {
       // stream flow file in, write new data out
    });

    session.transfer(flowFile, MY_RELATIONSHIP);
}
```

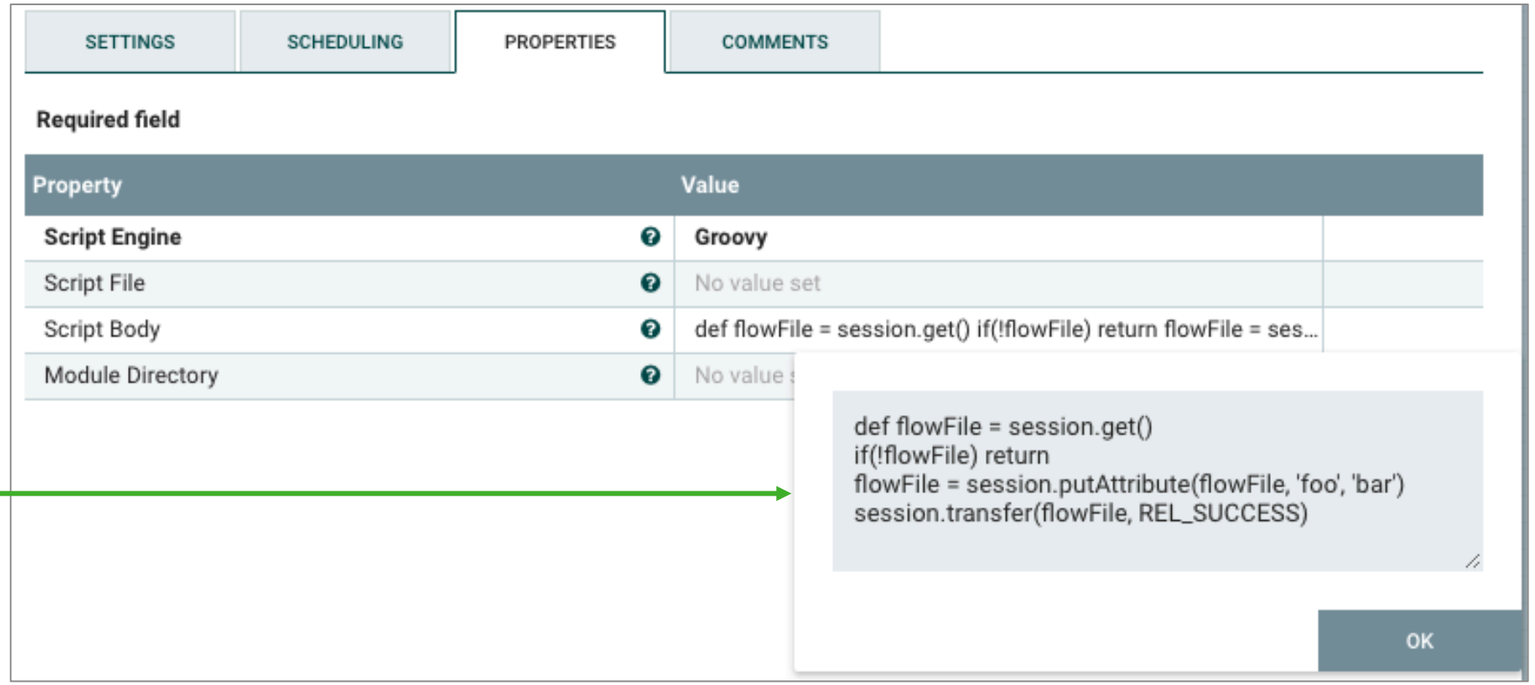

HORTONWORKS

Scripting Processors

- *"I don't want to deal with all that Maven & Java stuff!"*
- Embed scripts anywhere in the data flow, no compiling/deploying
- Supported scripting languages
  - Based on JSR-223 Java Scripting API
  - Groovy, Clojure, ECMAScript, Jython, Ruby, Lua
- *ExecuteScript*
  - Enter script code directly in processor property, or point to external script file
  - Script is passed specific API objects like session, context, and logger
- *InvokeScriptedProcessor*
  - Enter script code directly in processor property, or point to external script file
  - Implement an entire processor in a supported scripting language

ExecuteScript Example





DEMO!!

HORTONWORKS

Example Scenario

- User data
  - https://randomuser.me
- Initially in CSV format
  - `name.title,name.first,name.last,email,registered`
  - `mr,dennis,reyes,dennis.reyes@example.com,2012-04-10 01:54:19`
  - `miss,carole,gomez,carole.gomez@example.com,2002-12-17 22:15:49`
- Requirements
  - Convert CSV to JSON
  - Add a full_name field with first name + last name
  - Add a gender field based on title (i.e. if title == mr then MALE)
  - Ingest to different Solr collections depending on environment

Questions?

HORTONWORKS

Learn more and join us!

Apache NiFi site
http://nifi.apache.org

Subscribe to and collaborate at
dev@nifi.apache.org
users@nifi.apache.org

Submit Ideas or Issues
https://issues.apache.org/jira/browse/NIFI

Follow us on Twitter
@apachenifi

HORTONWORKS

Thank you!

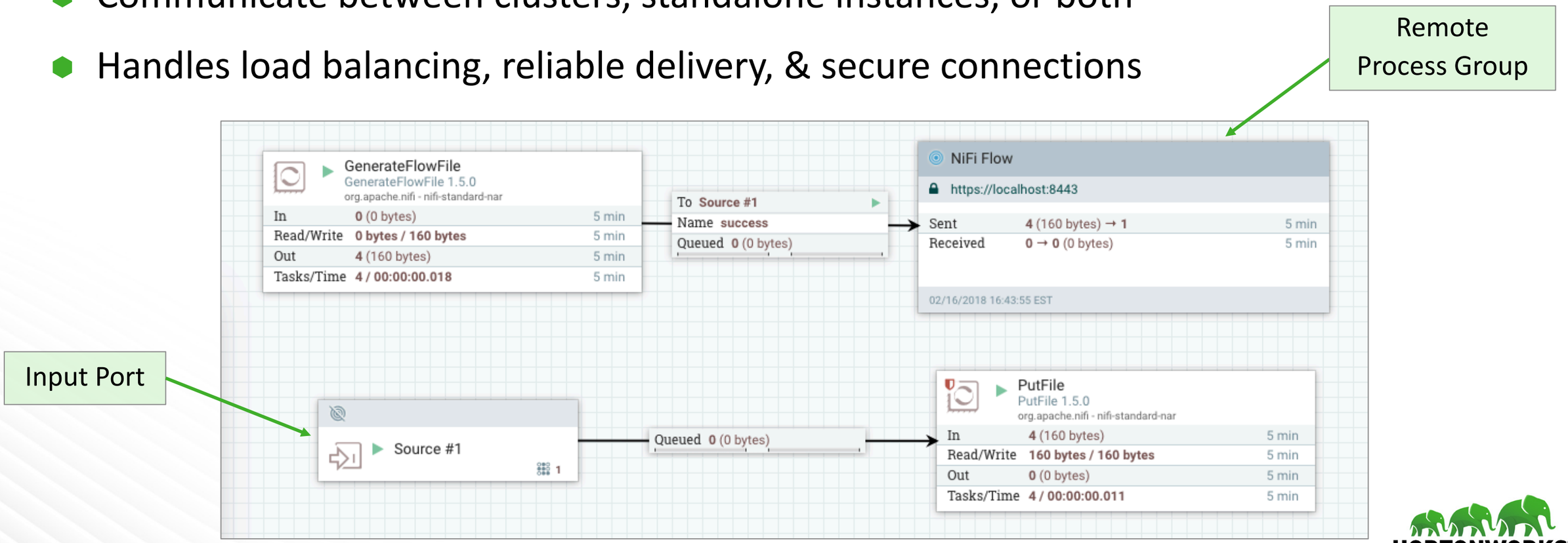
HORTONWORKS

Building NARs

```
cd nifi-helloworld-bundle

mvn clean package

cd nifi-helloworld-nar/target/

unzip nifi-helloworld-nar-1.0-SNAPSHOT.nar

Archive:  nifi-helloworld-nar-1.0-SNAPSHOT.nar
creating: META-INF/
inflating: META-INF/MANIFEST.MF
creating: META-INF/bundled-dependencies/
inflating: META-INF/LICENSE
inflating: META-INF/DEPENDENCIES
inflating: META-INF/NOTICE
inflating: META-INF/bundled-dependencies/nifi-helloworld-processors-1.0.jar
inflating: META-INF/bundled-dependencies/nifi-utils-1.5.0.jar
```

Site-To-Site

- Direct communication between two NiFi instances via TCP or HTTP
- Push to Input Port on receiver, or Pull from Output Port on source
- Communicate between clusters, standalone instances, or both
- Handles load balancing, reliable delivery, & secure connections



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