



Secure Dataflow From Edge to Core with Apache NiFi and MiNiFi

Andy LoPresto | @yolopey

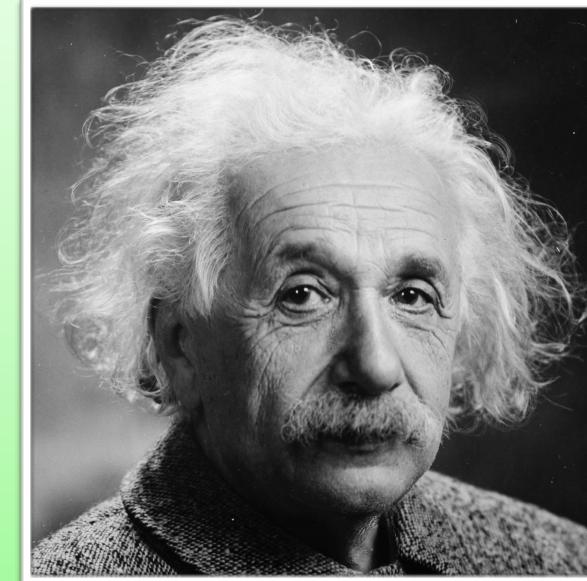
Sr. Member of Technical Staff at Hortonworks, Apache NiFi PMC & Committer

05 November 2018 Future of Data Sydney

Acknowledgement of Country

We acknowledge the traditional owners of the land on which we meet today, the Gadigal People of the Eora Nation and pay our respects to Elders past and present.

Gauging Audience Familiarity With NiFi



"What's a NeeFee?"

No experience with dataflow
No experience with NiFi

"I can pick this up pretty quickly"

Some experience with dataflow
Some experience with NiFi

"I refactored the Ambari integration endpoint to allow for mutual authentication TLS during my coffee break"

Forgotten more about NiFi than most of us will ever know

Agenda

- *What is dataflow and what are the challenges?*
- Apache NiFi
- Apache MiNiFi
- Apache NiFi Registry
- Security
- New Features
- Roadmap
- Q&A



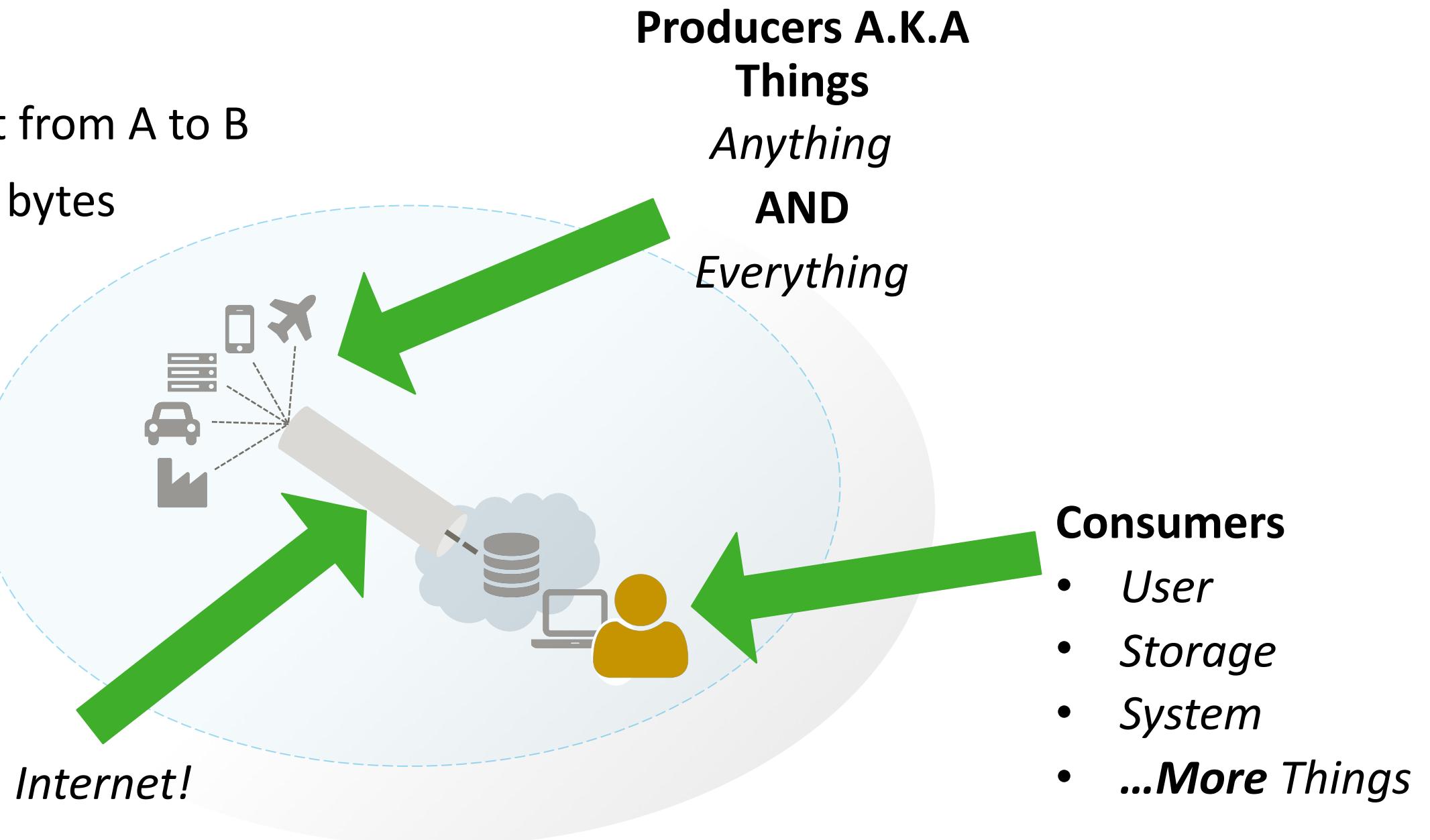
All slides provided online, no need to transcribe



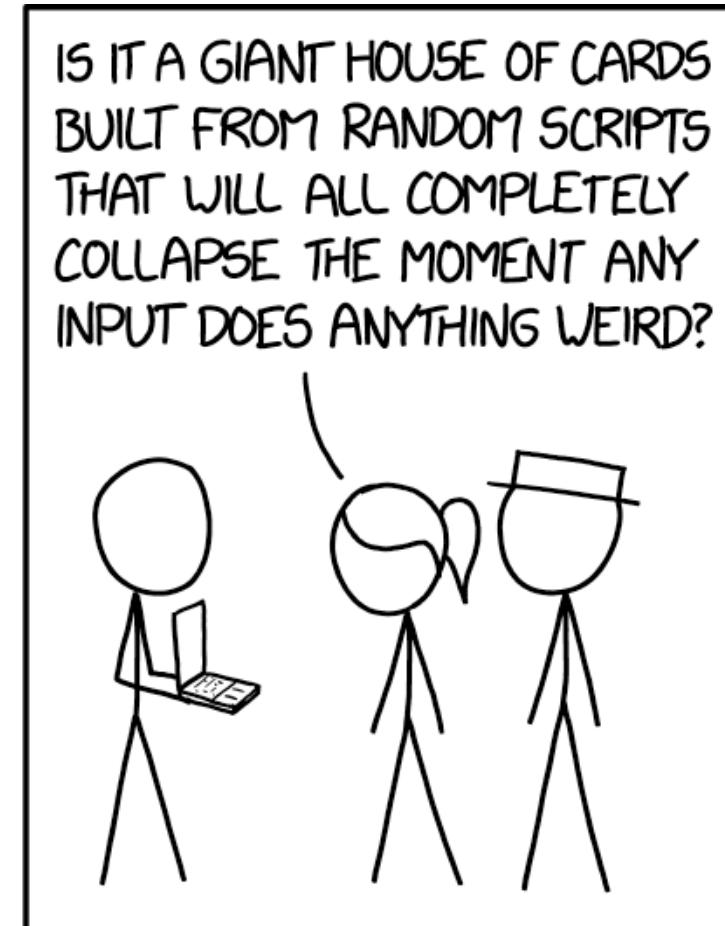
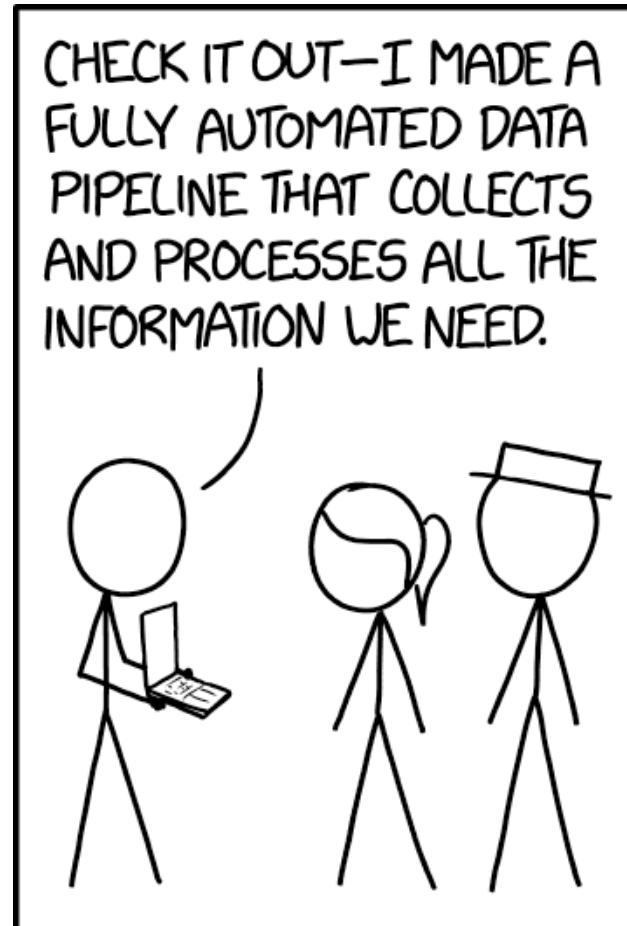
What is dataflow?

What is dataflow?

- Moving some content from A to B
- Content could be any bytes
 - Logs
 - HTTP
 - XML
 - CSV
 - Images
 - Video
 - Telemetry



Moving data *effectively* is hard



"Data Pipeline" <https://xkcd.com/2054/>

Dataflow Challenges In 3 Categories

Data

- Standards
- **Formats**
- Protocols
- Veracity
- Validity
- Schemas
- Partitioning/
Bundling

Infrastructure

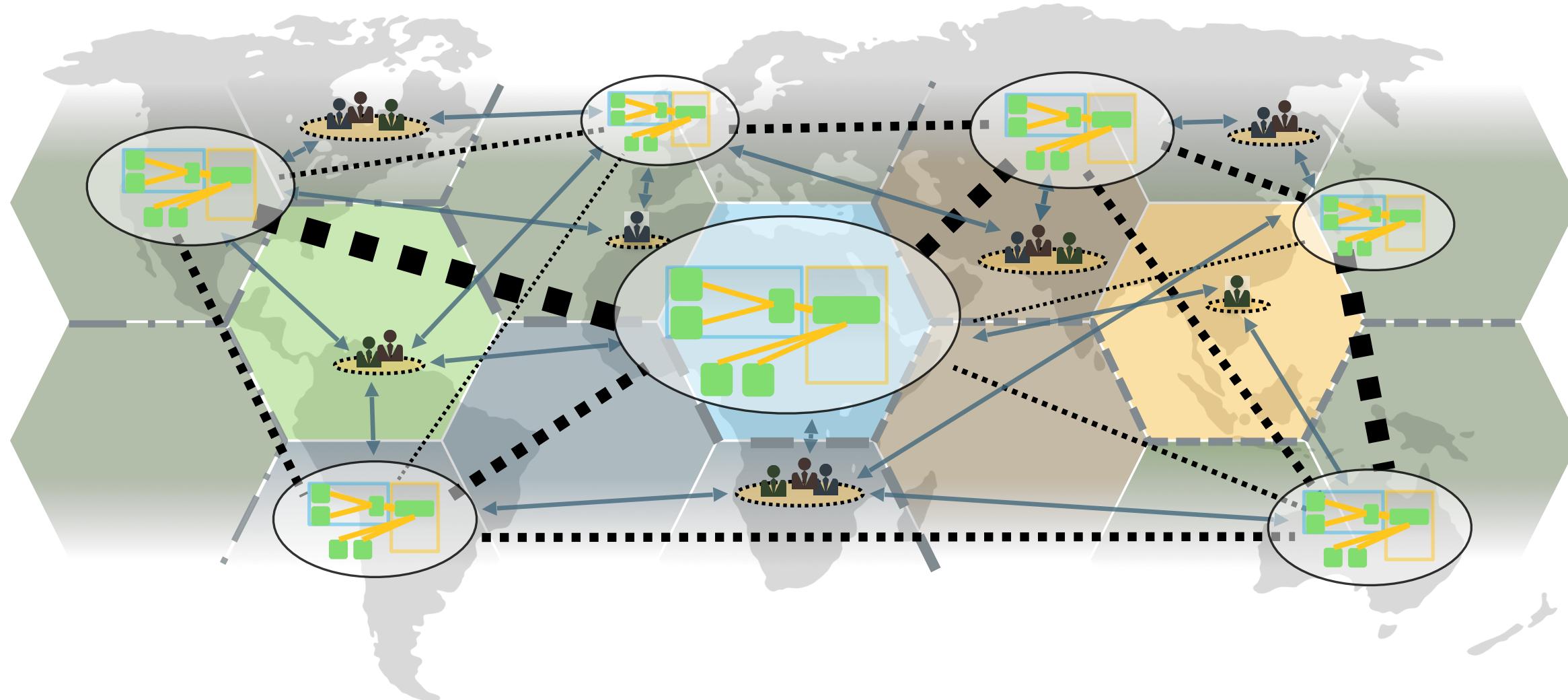
- “Exactly Once”
Delivery
- Ensuring
Security
- **Overcoming**
Security
- Credential
Management
- Network

People

- Compliance
- “**That** [person |
team | group]”
- **Consumers**
Change
- **Requirements**
Change
- “Exactly Once”
Delivery

Let's Connect Lots of As to Bs to As to Cs to Bs to Δ s to Cs to φ s

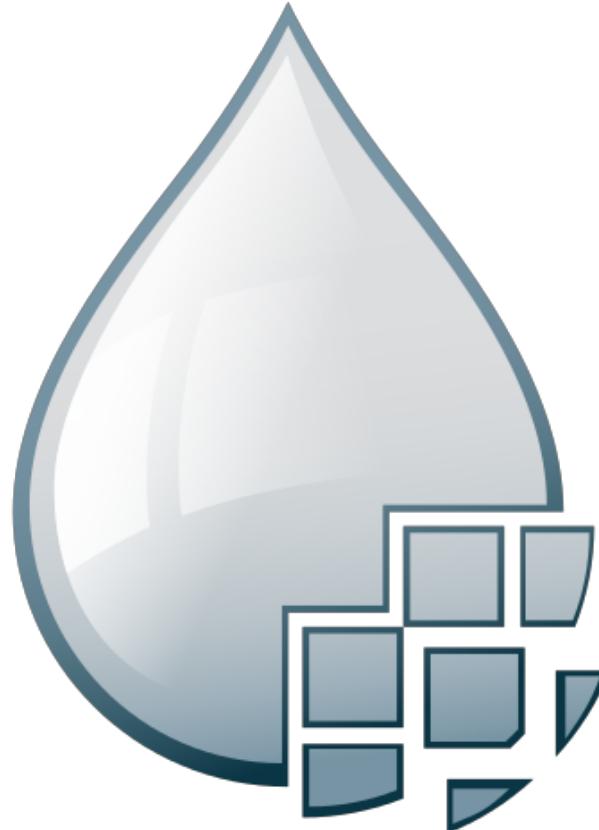
Raise your hand if you want to maintain Python scripts for the rest of your life



Apache NiFi

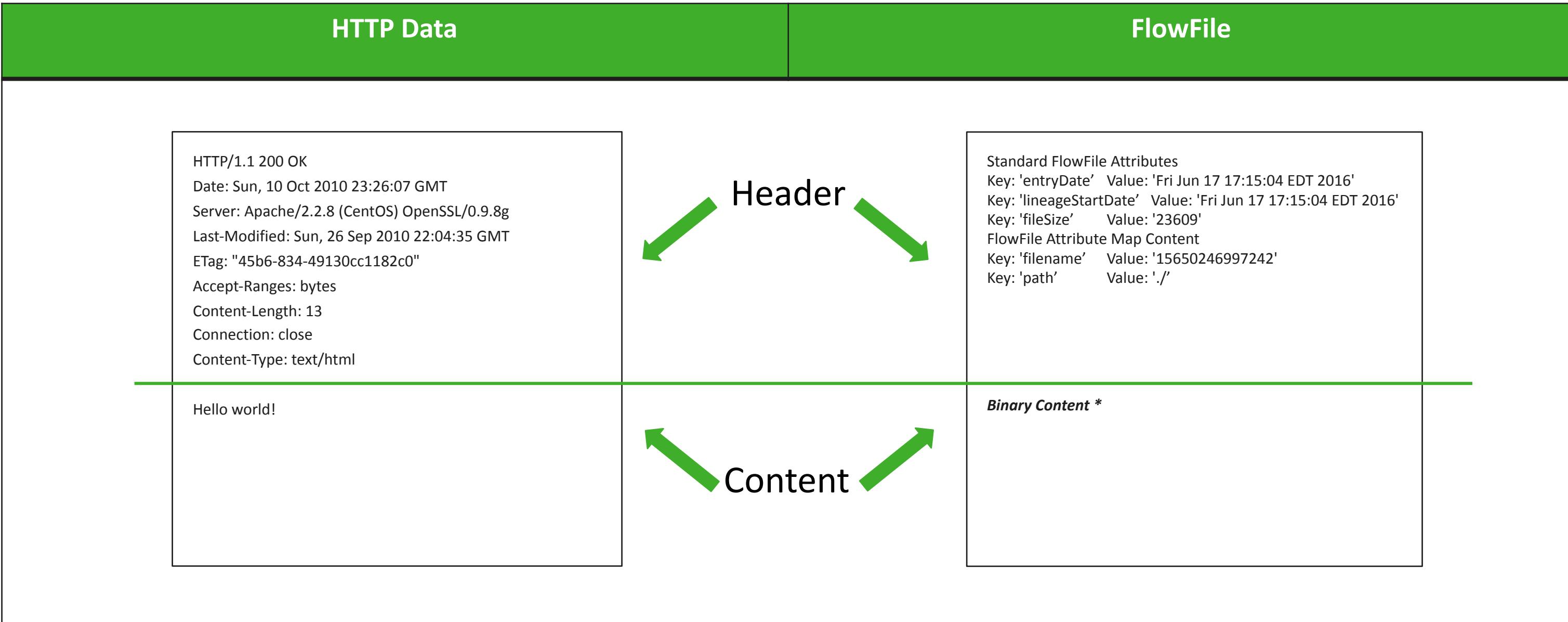
Apache NiFi

Key Features



- Guaranteed delivery
 - Data buffering
 - Backpressure
- Pressure release
- Prioritized queuing
- Flow specific QoS
 - Latency vs. throughput
 - Loss tolerance
- Data provenance
- Supports push and pull models
- Recovery/recording a rolling log of fine-grained history
- Visual command and control
- Flow templates
- Pluggable, multi-tenant security
- Designed for extension
- Clustering

Flowfiles Are Like HTTP Data



User Interface

Less of this... ... more of this

The screenshot shows a terminal window and a web browser side-by-side.

Terminal (Left):

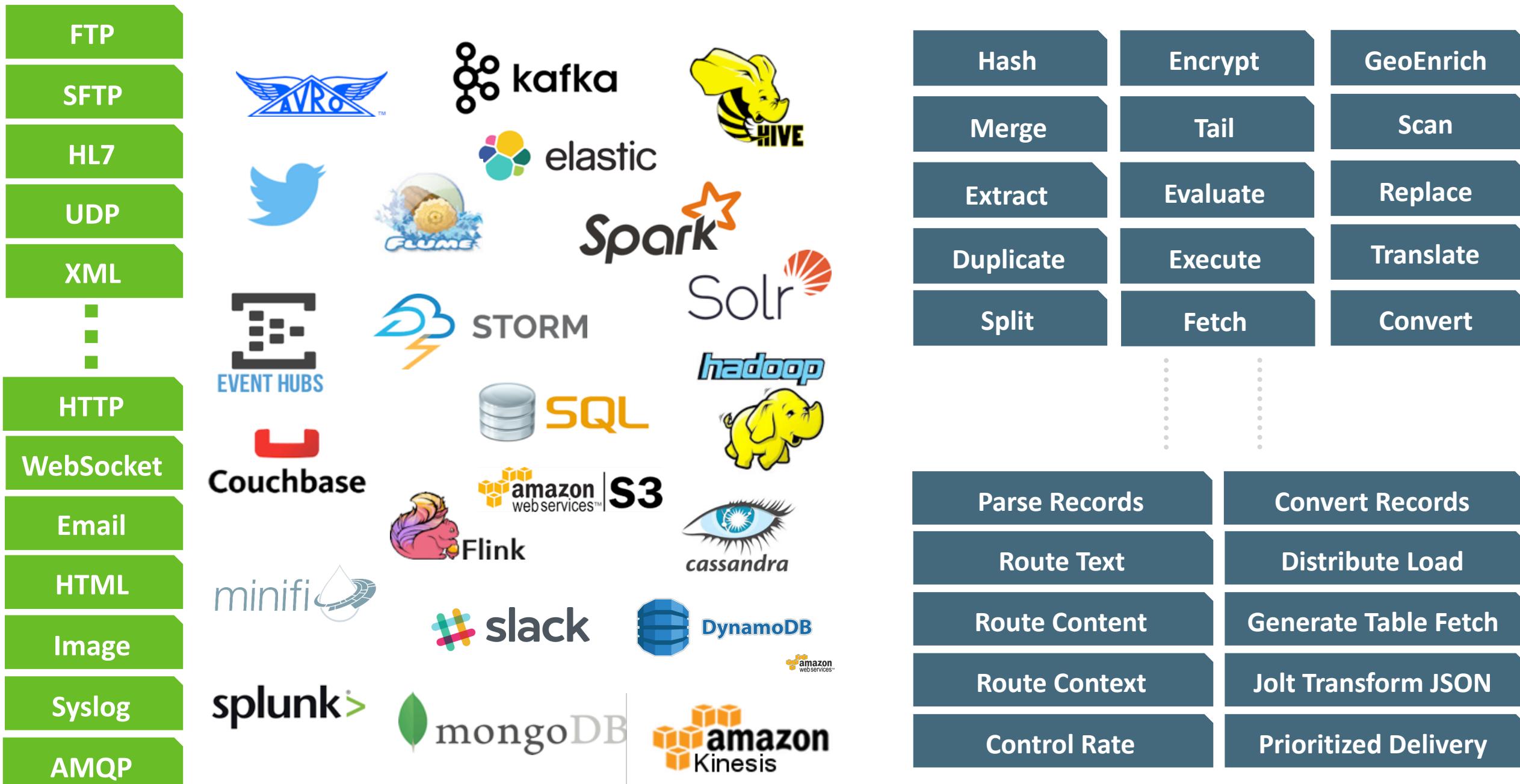
```
scratch/release_verification (master) alopreno
ls -l
total 144
drwxr-xr-x 16 alopreno staff 544B Nov 24 19:20 .
drwxr-xr-x 17 alopreno staff 578B Nov 24 14:25 ..
drwxr-xr-x 38 alopreno staff 1.3K Nov 24 14:33 archive/
-rw-r--r-- 1 alopreno staff 2.5K Nov 24 19:20 authorizations.xml
-rw-r--r-- 1 alopreno staff 3.7K Nov 24 19:19 authorizers.xml
-rw-rw-r-- 1 alopreno staff 2.1K Nov 23 23:36 bootstrap-notification-services.xml
-rw-r--r-- 1 alopreno staff 3.1K Nov 24 19:29 bootstrap.conf
-rw-r--r-- 1 alopreno staff 2.5K Nov 24 14:33 flow.xml.gz
-rw-r----- 1 alopreno staff 3.0K Nov 24 19:16 keystore.jks
-rw-rw-r-- 1 alopreno staff 8.0K Nov 23 23:36 logback.xml
-rw-r--r-- 1 alopreno staff 2.6K Nov 23 23:36 login-identity-providers.xml
-rw-r--r-- 1 alopreno staff 9.0K Nov 24 19:17 nifi.properties
-rw-r--r-- 1 alopreno staff 3.6K Nov 23 23:36 state-management.xml
-rw-r--r-- 1 alopreno staff 911B Nov 24 19:16 truststore.jks
-rw-r--r-- 1 alopreno staff 226B Nov 24 19:20 users.xml
-rw-r--r-- 1 alopreno staff 1.4K Nov 23 23:36 zookeeper.properties
h12203: /Users/alopreno/Workspace/scratch/release_verification (master) alopreno
21s @ 17:26:43 $ more nifi-1.1.0-RC1-failed/nifi-1.1.0/nifi-assembly/target/nifi-1.1.0/conf/authorizations.xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<authorizations>
    <policies>
        <policy identifier="b2c0320b-0384-38d2-8ff7-58a26dde3897" resource="/flow" action="R">
            <user identifier="9860132a-283e-b023393be562"/>
        </policy>
        <policy identifier="7e67308c-a837-3ba1-8b37-4a40ff8d43fe" resource="/data/process-groups/9871f1da-0158-1000-c0b2-42aaca57d800" action="R">
            <user identifier="9860132a-283e-370b-ba63-b023393be562"/>
        </policy>
        <policy identifier="b762618f-3d36-3b45-943e-bd38605e276e" resource="/data/process-groups/9871f1da-0158-1000-c0b2-42aaca57d800" action="R">
            <user identifier="9860132a-283e-370b-ba63-b023393be562"/>
        </policy>
        <policy identifier="61dfe35b-b37f-4b36-b2e7bc4989c3" resource="/process-groups/9871f1da-0158-1000-c0b2-42aaca57d800" action="R">
            <user identifier="9860132a-283e-370b-ba63-b023393be562"/>
        </policy>
        <policy identifier="ff9f93ce-1fda-3b08-9309-088ba6abc5a9" resource="/process-groups/9871f1da-0158-1000-c0b2-42aaca57d800" action="W">
            <user identifier="9860132a-283e-370b-ba63-b023393be562"/>
        </policy>
        <policy identifier="bc5512df-2a78-3bba-959e-3bc1ba5f781d" resource="/restricted-components" action="W">
            <user identifier="9860132a-283e-370b-ba63-b023393be562"/>
        </policy>
        <policy identifier="70197fb3-af4d-3938-b246-498db26032fa" resource="/tenants" action="R">
            <user identifier="9860132a-283e-370b-ba63-b023393be562"/>
        </policy>
        <policy identifier="30f46aab-baa8-3b50-8b9d-d768a83e719f" resource="/tenants" action="W">
            <user identifier="9860132a-283e-370b-ba63-b023393be562"/>
        </policy>
        <policy identifier="6c25353b-fc4a-3813-baa8-f879d9853e7e" resource="/policies" action="R">
            <user identifier="9860132a-283e-370b-ba63-b023393be562"/>
        </policy>
        <policy identifier="3906378a-8f9b-3a71-aa0b-44325b1723cf" resource="/policies" action="W">
            <user identifier="9860132a-283e-370b-ba63-b023393be562"/>
        </policy>
        <policy identifier="26c66358-8548-31c8-a38c-e0eb4bc3ddba" resource="/controller" action="R">
            <user identifier="9860132a-283e-370b-ba63-b023393be562"/>
        </policy>
        <policy identifier="4470abbe-3b1e-31cd-9daa-598eae5c59ef" resource="/controller" action="W">
            <user identifier="9860132a-283e-370b-ba63-b023393be562"/>
        </policy>
    </authorizations>

```

Browser (Right):

The browser window shows the Apache NiFi user interface. The title bar says "1. scratch/release_verification (bash)". The main content area displays the NiFi configuration file "nifi-1.2.0-SNAPSHOT-bin/nifi-1.2.0-SNAPSHOT (bash)". The configuration includes settings like "nifi.flowfile.repository.always.sync=false", "nifi.swap.in.period=5 sec", and "nifi.swap.out.period=5 sec". It also lists various NiFi components and their configurations, such as "Content Repository", "Bootstrap Config File", and "Java home". Log entries from the NiFi bootstrap process are visible at the bottom of the configuration page.

Deeper Ecosystem Integration: 286+ Processors, 61 Controller Services



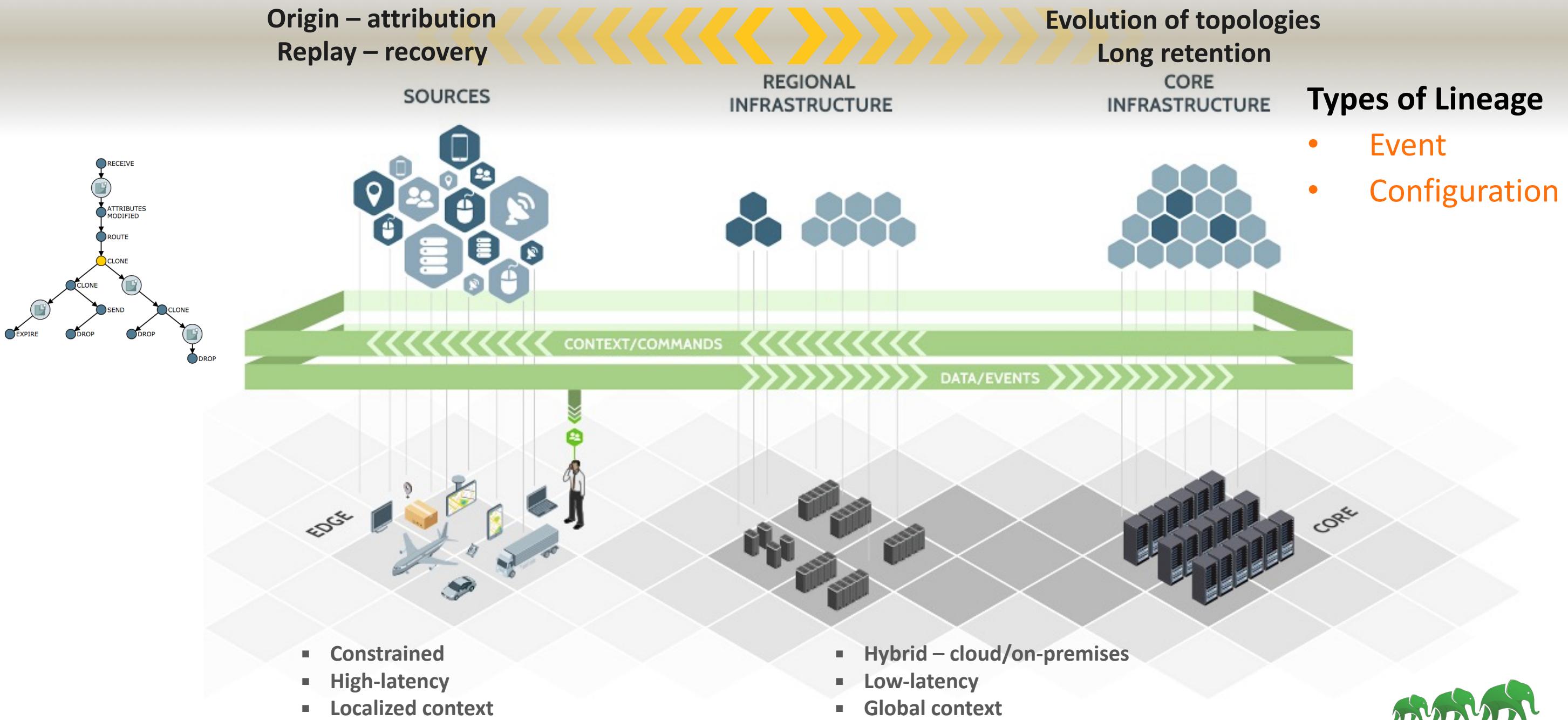
All Apache project logos are trademarks of the ASF and the respective projects.

“Where did this chicken come from?”



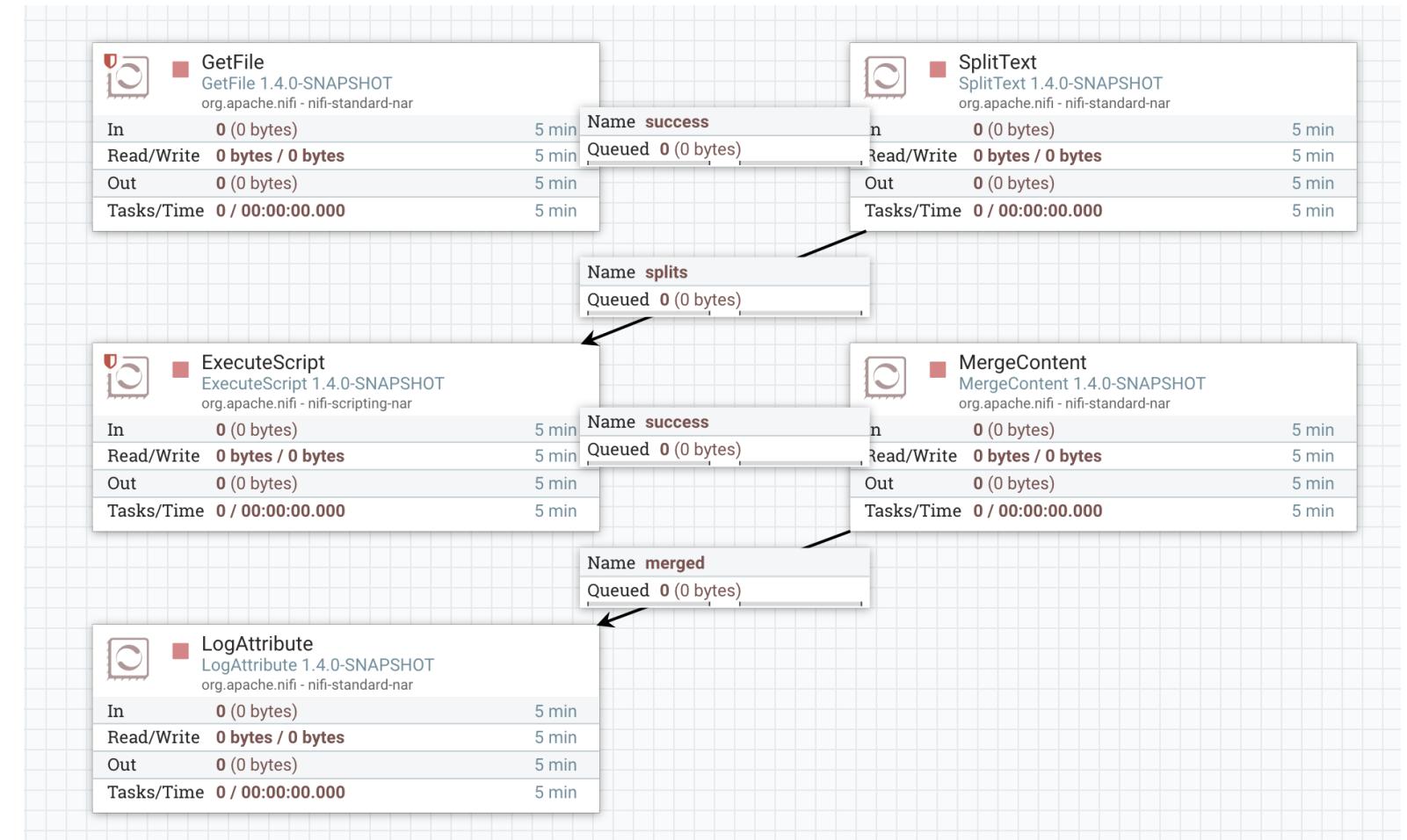
[Colin the Chicken | Portlandia | IFC](#)

Data Provenance



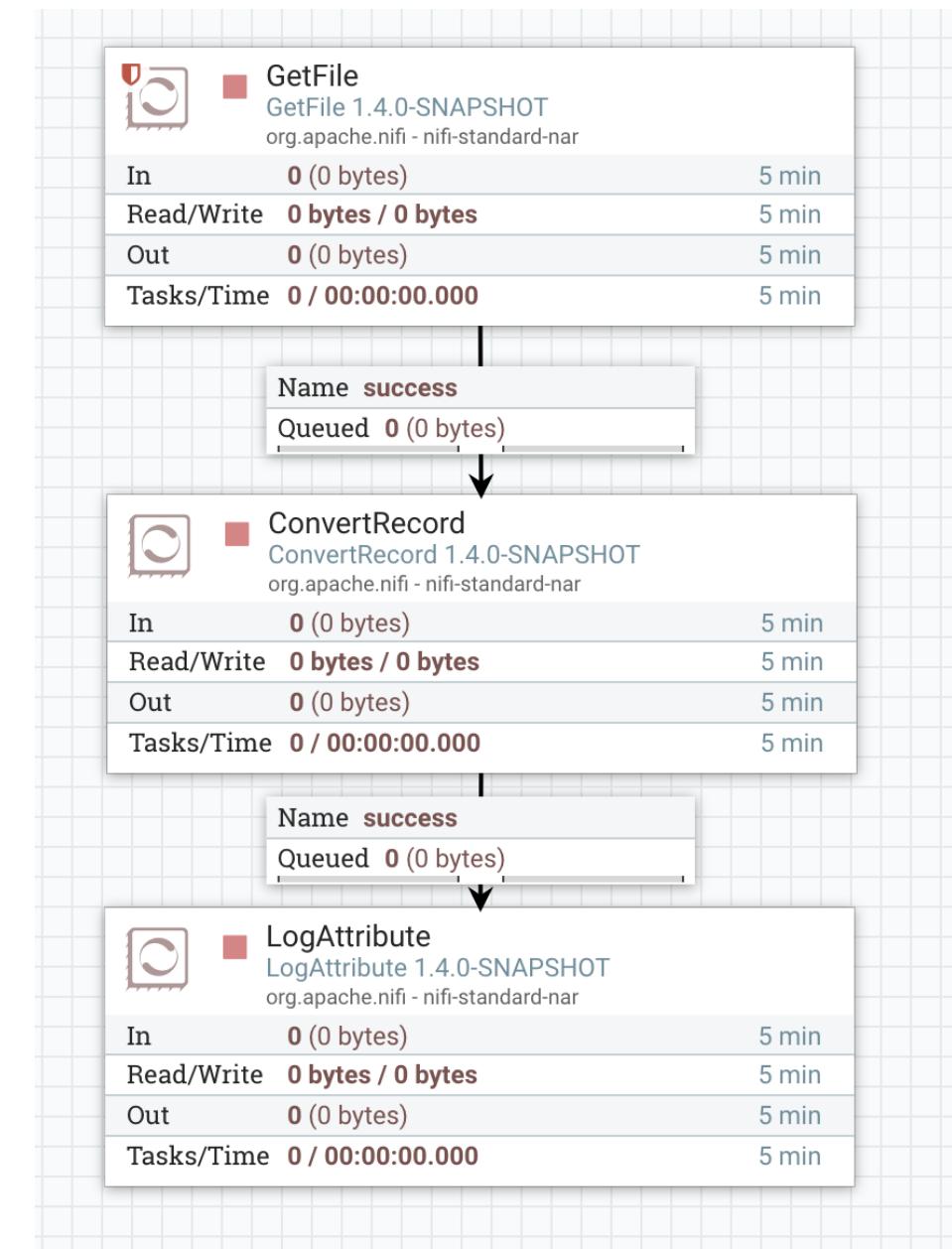
Record Parsing

- Previously, data had to be divided into individual flowfiles to perform work
- CSV output with 50k lines would need to be split, operated on, remerged
- $1 + 50k + 50k + 1$ flowfiles = 100k flowfiles



Record Parsing

- Now flowfile content can contain many “record” elements
- Read and write with **Reader* and **Writer* Controller Services
- Perform lookups, routing, conversion, SQL queries, validation, and more...
- 1 + 1 flowfiles = 2 flowfiles



Encrypted Provenance Repository

- Every provenance event record is encrypted with AES G/CM before being persisted to disk
 - Decrypted on deserialization for retrieval/query
 - Random access via offset seek
 - Handles key migration & rotation

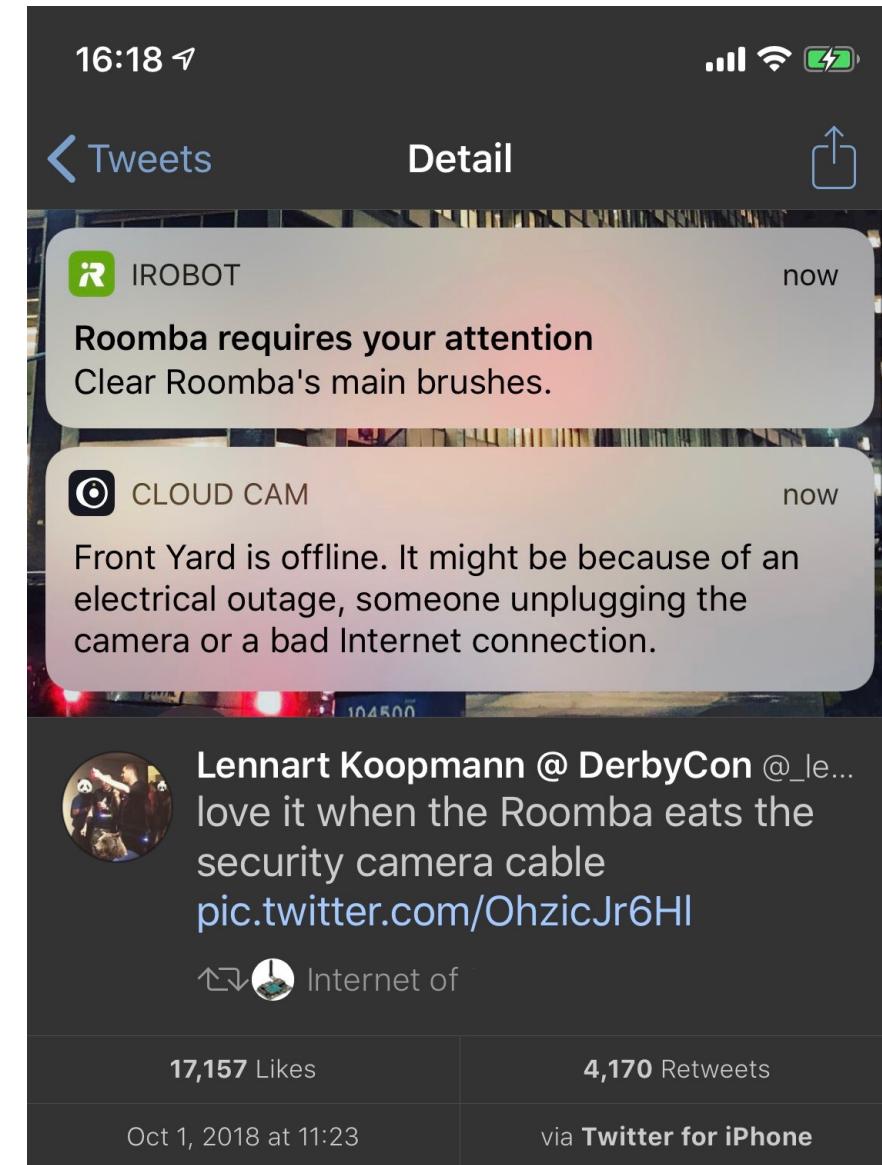
NiFi Data Provenance						
Displaying 3 of 3		Showing the events that match the specified query. Clear search				
Filter	by component name					
Date/Time ▾	Type	FlowFile Uuid	Size	Component Name	Component Type	Actions
06/05/2017 20:17:4...	CONTENT_MODIFIED	d602bdfd-9d14-4c2e...	77 bytes	ConvertRecord	ConvertRecord	
06/05/2017 20:17:4...	ROUTE	d602bdfd-9d14-4c2e...	46 bytes	LookupRecord	LookupRecord	
06/05/2017 20:17:4...	FORK	f540f7cf-1e41-4cb7...	40 bytes	LookupRecord	LookupRecord	

```
essors.standard.ConvertRecord
2017-06-05 20:17:31,885 INFO [StandardProcessScheduler Thread-5] o.a.n.c.s.TimerDrivenSchedulingAgent Stopped scheduling GenerateFlowFile
[elid=1] 0.prov
2017-0 46228 00000277 01ACECD00 05737200 2D6F7267 2E617061 63686526 6E696669
2017-0 46256 2E70726F 76656E61 6E63652E 456E6372 79707469 6F6E4D65 74616461
2017-0 46284 7461F1CD 98C1C961 1FED0200 05490010 63697068 65742479 74654C65
2017-0 46312 6677468 4C000961 6C676F72 6974686D 7400124C 6A617661 2F6C616E
2017-0 46340 67F25374 72696E67 38B5B007 69764279 67547374 0025B842 4C00056B
2017-0 46368 65794964 71007006 014C0007 76657273 696F6E71 007E0001 78700000
2017-0 46396 18A17400 11414553 2F47434D 2F4E6F50 61646469 6E677572 0025B842
2017-0 46424 ACF317F8 00680540 020000078 70000000 1022D425 38C2D858 C607F8B7
2017-0 46452 4A832876 F0740004 4B657931 74000276 31AA2005 6C21458B 6E08FD14
2017-0 46480 73C8CBF6 CC6AF975 75315970 8A8E98B4 456AC01 D007897F 289ECB3A
2017-0 46508 5F4327EC 91C523ED 2D418131 10C8DC61 8752C1E3 68254C87 C7CB3D8
2017-0 46536 DD998512 C85B8714 BE455FB4 F4673937 E1847863 73061529 808B3C3
2017-0 46564 03A6AB4E 55FFCC0A BA7AB8A4 199B4862 318E9988 A3BAE9F5 3CC663B8
2017-0 46592 87641F46 D63C9F4C BDE25008 4D24E82 26C1946A DC02D006 EB333351
2017-0 46620 A4A9C7D4 636A9EF8 864816D5 66D81457 55E16D76 3DF0962 C83F7B8E
2017-0 46648 C7C67EF1 D98563B8 A98EA071 3727D69C ACFC8159 C8D358C0 EBC2C7E2
2017-0 46676 06ED0AEE D6CA71E2 F082F2D6A B38E4F9F E4453D36 807919ED 5959788B
2017-0 46704 F4472F58 55EEE080 0F8AE3C3 E0C8FC83 7459DA91 515FC306 916B931E
2017-0 46732 26004327 571A72F1 38A5E2ZC 08A0A7A7 FF125651 70DFDBD14 9B994846
2017-0 46760 C178A555 3868931B DAF73A29 99634A00 74268E4A F0145602 49AD4774
2017-0 46788 276A3B23 FB4F2955 6318805D FA146380 9D10203F CB442103 7E40A085
2017-0 46816 D93EE58A 5425FEB2 819FA04C 0F4688A3 A9F550A2 73AC72C6 E01E078
2017-0 46844 63260E4E 085B4999 822FB2B8 DE40803F 1D806B000 00003900 000027701
2017-0 46872 ACED0005 7372002D 6F72672E 61706163 68652E6E 6966692E 70726F76
2017-0 46900 656E616E 63652E45 6637279 704696F 6E4D6574 6164174 611FCD9C
2017-0 46928 493161F 0D200005 49001063 69706865 72427974 654C5656 6774684C
2017-0 46956 0009616C 676F7269 74686D74 00124C6A 6176612F 6C616E67 2F537472
2017-0 46984 6966E73B 58000769 76427974 65737400 025B842C 00056B65 79494647
2017-0 47012 007E0001 4C000776 65727369 6F6E7100 7E001700 70000001 8A740011
2017-0 47040 4145532F 47434D2F 4E65F061 6464696E 67757200 025B842A F317F806
2017-0 47068 0854E002 00007870 00000010 01DD7B22 177C1811 42FBCB2A 37434662
2017-0 47096 7400044B 65793174 00276311 4606648A A598A4B0 82B474D 828C64E
2017-0 d=21c 47124 86AC8412 D18E7868 6D667BE4 6652E184 D75A0F69 1EDE140 E18E9385
2017-0 47152 18E78611 F7E9637 A058D602 11AD8B9D 91EA64B1 11EE632A B0049E8E
2017-0 47180 53371C38 30F9BCDE D0C2BC2B 5FD3DF1D 009480A5 14977F93 76946E02
[elid=21c] Signed Int ↴ [big ↴] (select some data) 0 out of 92697 bytes
2017-0 i.continueOnConversion=false; i.conversionTime=245860777; Another save pending
2017-06-05 20:17:48,933 INFO [NiFi Web Server-21] o.a.n.controller.StandardFlowFileQueue Cancelling ListFlowFile Request with ID 7b688ae
8-015c-2000-c565-f2e0249938715
2017-06-05 20:17:55,381 INFO [Provenance Query-1] o.a.nifi.properties.NiFiPropertiesLoader Determined default nifi.properties path to b
e '/Users/alo presto/WorkSpace/scratch/release_verification/nifi-1.3.0-RC1/nifi-1.3.0/nifi-assembly/target/nifi-1.3.0-bin/nifi-1.3.0/.c
onf/nifi.properties'
2017-06-05 20:17:55,382 INFO [Provenance Query-1] o.a.nifi.properties.NiFiPropertiesLoader Determined default nifi.properties path to b
e '/Users/alo presto/WorkSpace/scratch/release_verification/nifi-1.3.0-RC1/nifi-1.3.0/nifi-assembly/target/nifi-1.3.0-bin/nifi-1.3.0/.c
onf/nifi.properties'
2017-06-05 20:17:55,383 INFO [Provenance Query-1] o.a.nifi.properties.NiFiPropertiesLoader Loaded 125 properties from /Users/alo presto/
WorkSpace/scratch/release_verification/nifi-1.3.0-RC1/nifi-1.3.0/nifi-assembly/target/nifi-1.3.0-bin/nifi-1.3.0/.conf/nifi.properties
2017-06-05 20:17:55,397 INFO [Provenance Query-1] o.a.nifi.provenance.StandardQueryResult Completed Query! [ld602bffd-9d14-4c2e-804-56b
cc7ce9df67] I comprised of 1 steps in 164 millis
2017-06-05 20:17:55,397 INFO [Provenance Query-1] o.a.nifi.provenance.TaskQueryTask Successfully queried index ./provenance_repos
itory/index-1496719050076 for query ?uid=ld602bffd-9d14-4c2e-804-56bce79df67; retrieved 3 events with a total of 3 hits in 41 millis
```

Apache MiNiFi

IoT Challenges

- Limited computing capability
- Limited power/network
- **Restricted software library/platform availability**
- **No UI**
- Physically inaccessible
- Not frequently updated
- **Competing standards/protocols**
- Scalability
- **Privacy & Security**



@_lennart

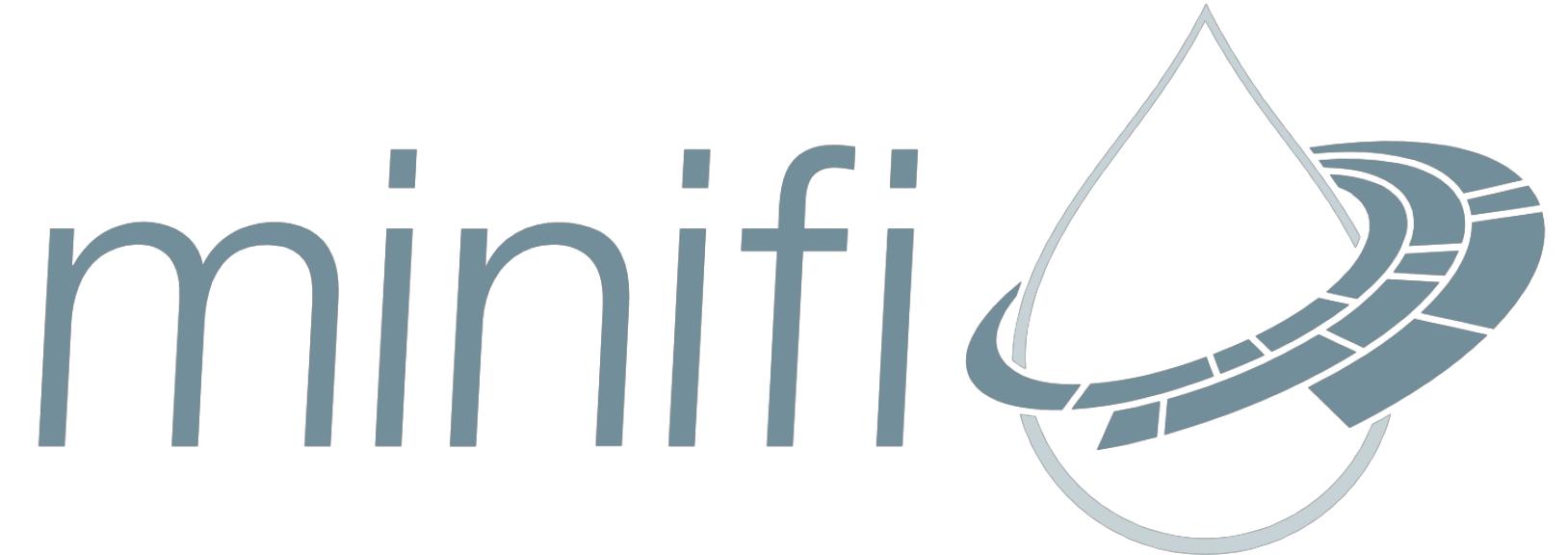
So Why Do We Need A Different Solution?

- NiFi is designed to “own the box”
- NiFi 0.7.x started up in about 10-15 minutes on RP3 (593 MB)
- NiFi 1.x started up in about 30 minutes on RP3 (760 MB)
 - 33 new processors
 - Rewrite for multi tenant authorization
 - Complete UI overhaul

```
▶hw12203:/Users/alopresto/Workspace/scratch/rp3b-demo (master) alopresto
 ━ 113s @ 17:09:05 $ ssh pi@my-raspberry-pi
 ^C
▶hw12203:/Users/alopresto/Workspace/scratch/rp3b-demo (master) alopresto
 ━ 145s @ 17:09:37 $ █
```

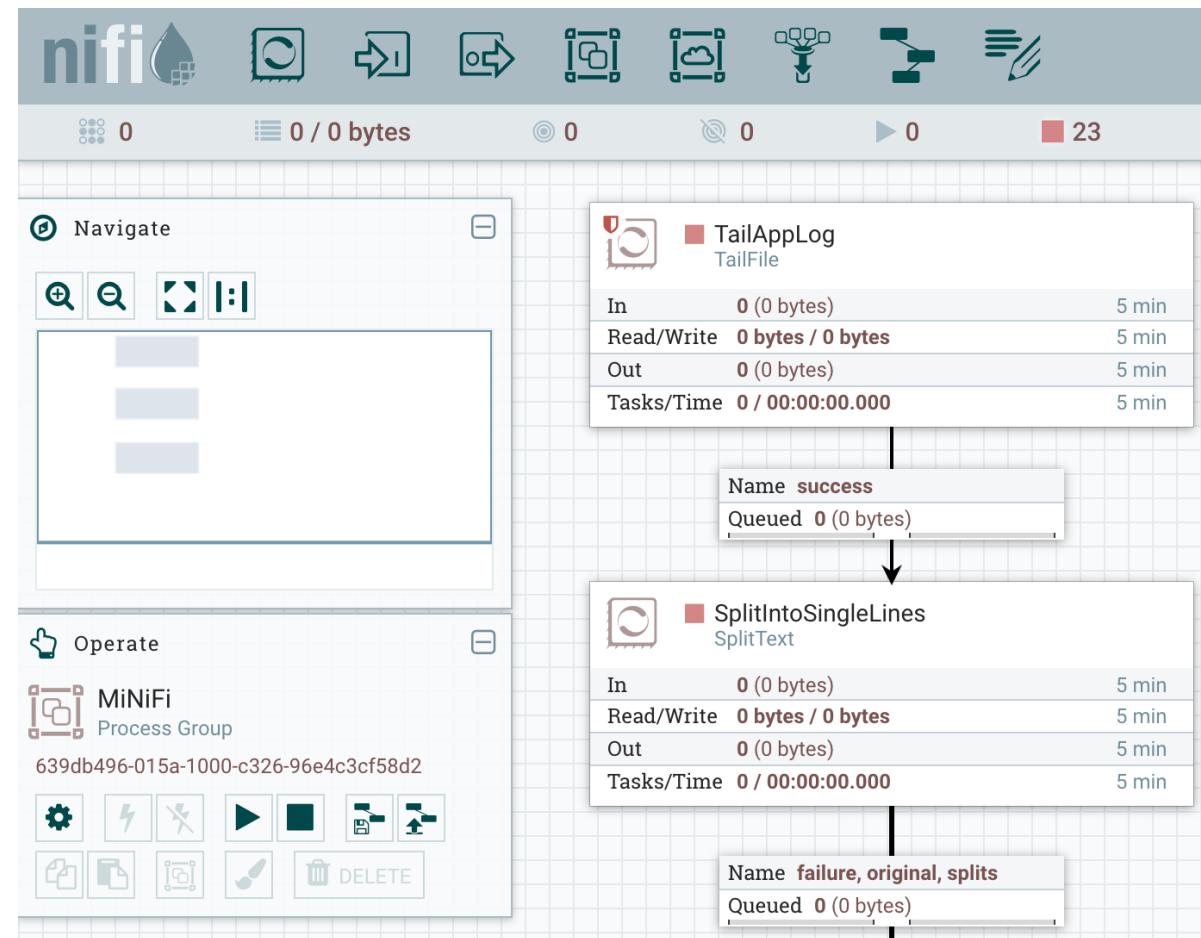
Apache NiFi Subproject: MiNiFi

- Get the key parts of NiFi close to where data begins and provide bidirectional communication
- NiFi lives in the data center — give it an enterprise server or a cluster of them
- MiNiFi lives as close to where data is born and is a guest on that device or system
 - IoT
 - Connected car
 - Legacy hardware

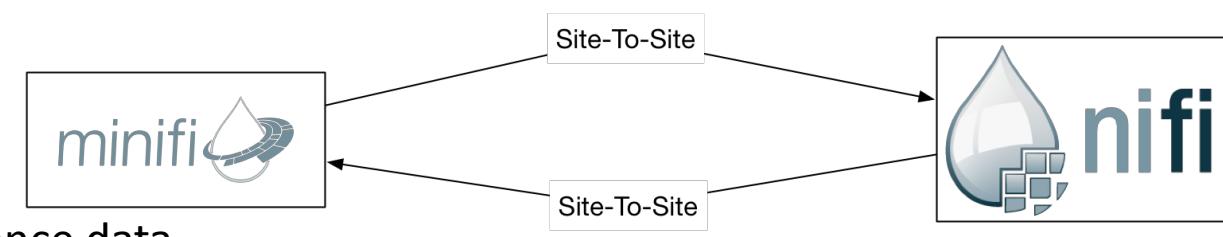


Flavors of MiNiFi

- MiNiFi Java (v0.5.0)
 - Modified version of NiFi
 - No UI
 - YAML configuration
 - Reduced processor count
 - 63+ by default, more available with additional NARs
- MiNiFi C++ (v0.5.0)
 - Written from scratch
 - 33 processors by default
 - Bi-directional site-to-site & provenance data

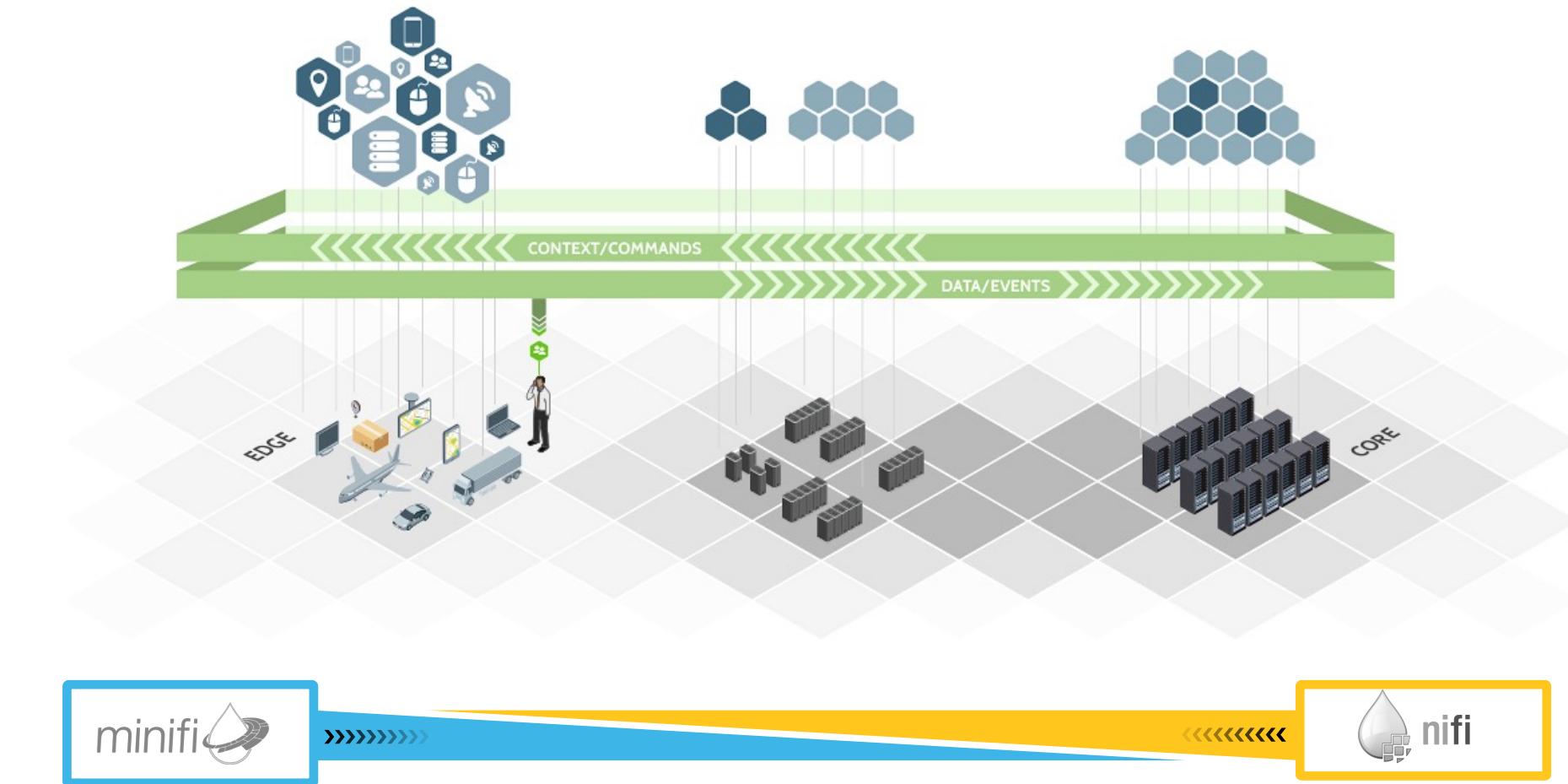


```
Security Properties:  
keystore: /tmp/ssl/localhost-ks.jks  
keystore type: JKS  
keystore password: localtest  
key password: localtest  
truststore: /tmp/ssl/localhost-ts.jks  
truststore type: JKS  
truststore password: localtest  
ssl protocol: TLS  
Sensitive Props:  
key:  
algorithm: PBWEWITHMD5AND256BITAES-CBC-OPENSSL  
provider: BC  
  
Processors:  
- name: TailAppLog  
  class: org.apache.nifi.processors.standard.TailFile  
  max concurrent tasks: 1  
  scheduling strategy: TIMER_DRIVEN  
  scheduling period: 10 sec  
  penalization period: 30 sec  
  yield period: 1 sec  
  run duration nanos: 0  
  auto-terminated relationships list:  
    Properties:  
      File to Tail: logs/minifi-app.log  
      Rolling Filename Pattern: minifi-app*  
      Initial Start Position: Beginning of File  
- name: SplitIntoSingleLines  
  class: org.apache.nifi.processors.standard.SplitText  
  max concurrent tasks: 1  
  scheduling strategy: TIMER_DRIVEN  
  scheduling period: 0 sec  
  penalization period: 30 sec  
  yield period: 1 sec  
  run duration nanos: 0  
  auto-terminated relationships list:  
    - failure  
    - original  
  Properties:  
    Line Split Count: 1  
    Header Line Count: 0  
    Remove Trailing Newlines: true
```



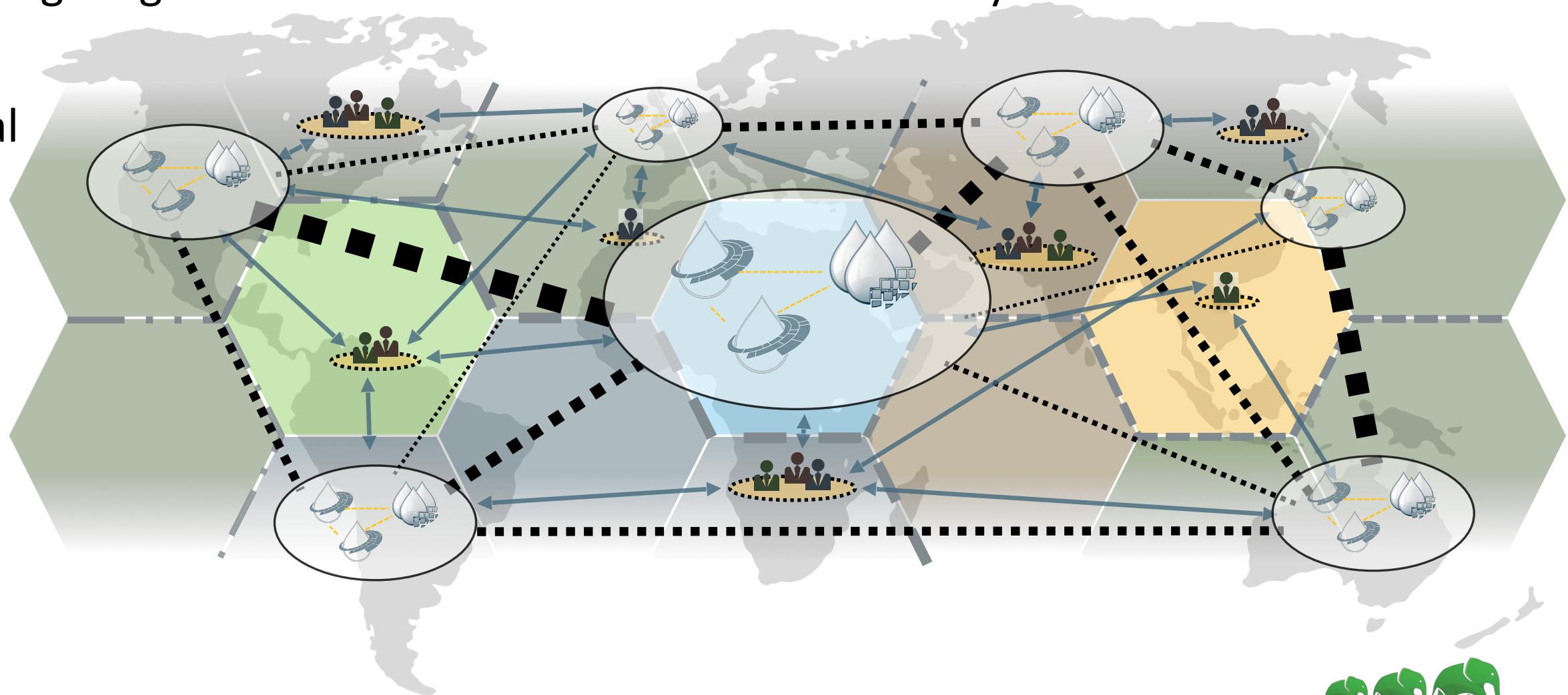
How Does MiNiFi Interact With NiFi?

- NiFi
 - Design flows
 - Aggregate data from many sources
 - Perform routing/analysis/SEP
- MiNiFi
 - Receive flows
 - Collect data
 - Send for processing



Let's Add Dimensionality

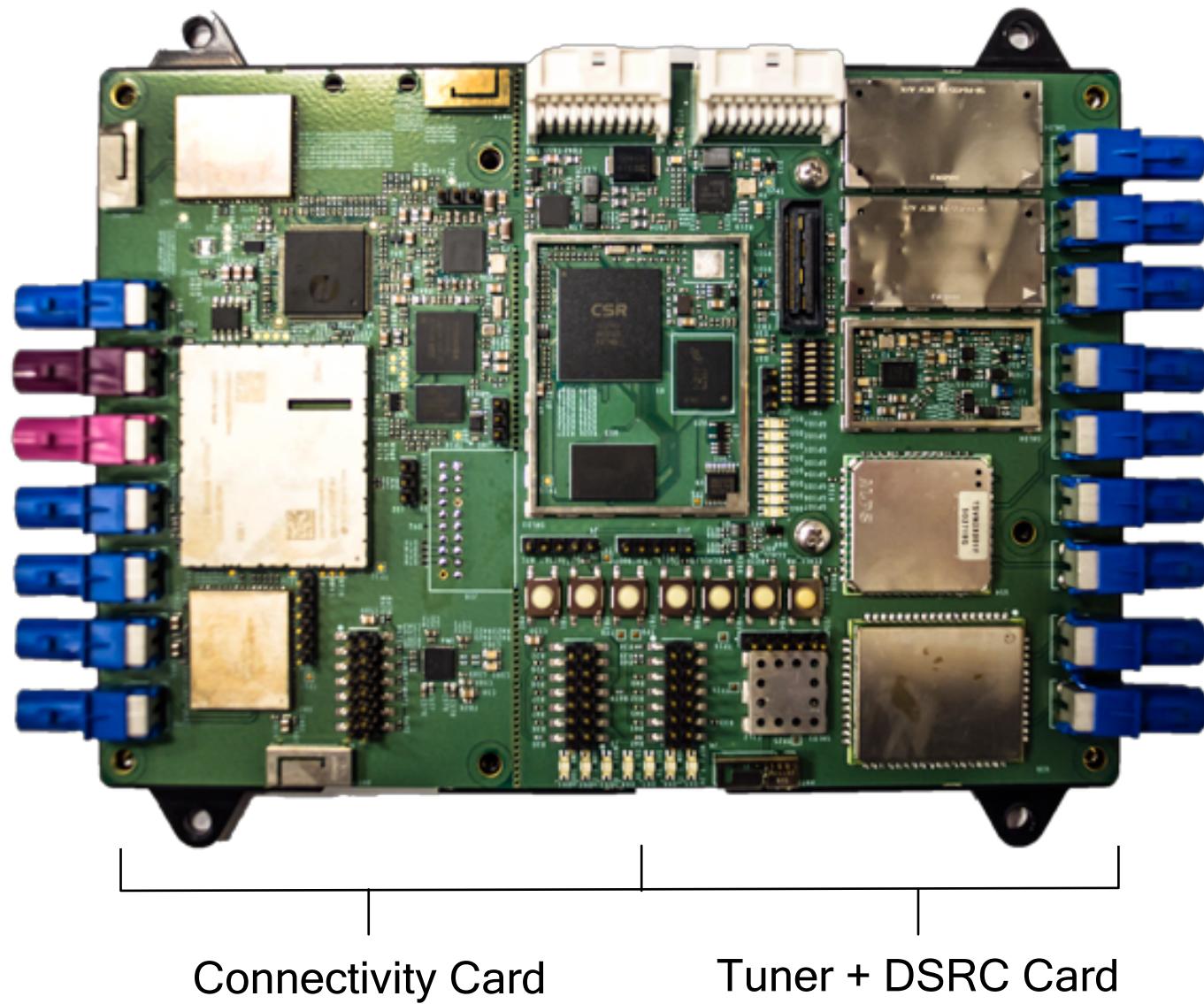
- We've been imagining EDGE to CORE as a bi-directional linear system
- Let's expand that to the real world



What does MiNiFi provide?

- Data tagging/provenance
- Governance from edge (geopolitical restrictions)
- Security (encryption, certificate-based authentication)
- Low latency (immediate reactions & decision-making)

Connected Car Reference Platform Box



Connectivity Card

Tuner + DSRC Card

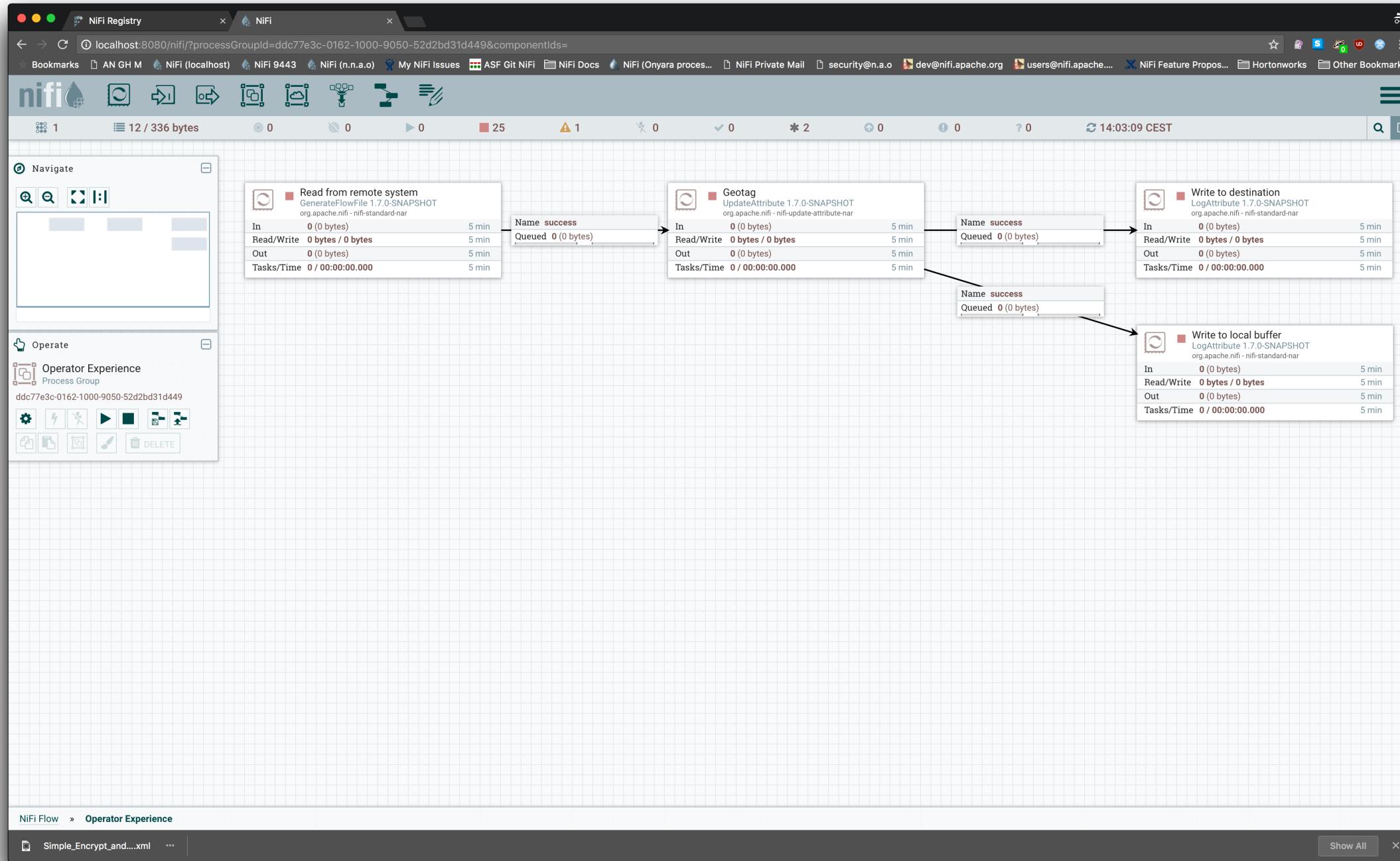
Apache NiFi Registry

Flow Development Lifecycle (FDLC)

- Origins of NiFi
- Operator Experience
 - MC data, don't drop, mitigate temporarily
- Version Control
- Environment Promotion



Operator Experience



Challenges

- Templates
 - Updates/replacement
 - Sensitive property replacement
- Flow.xml.gz migration
 - Key synchronization
- Environment promotion
 - Approval processes
 - Verifiability

Template Replacement

- Export a new version of template
- Transfer (somehow)
- Verify?
- Import onto canvas side-by-side existing flow
- Stop processors
- Empty queues
- Reconnect queues
- Start
- Pray?

```
-rw-r--r-- 1 alopresto staff 381B Apr 19 15:02 flow_20170626-171416_label_description.xml  
-rw-r--r-- 1 alopresto staff 1.4K Apr 19 15:02 flow_20170628-111620_xxe_transform_xml.xml  
-rw-r--r-- 1 alopresto staff 1.4K Apr 19 15:02 flow_20170628-155452_xxe_with_patch.xml  
-rw-r--r-- 1 alopresto staff 1.4K Apr 19 15:02 flow_20170629-103436_xxe_password_leak.xml  
-rw-r--r-- 1 alopresto staff 1.9K Apr 19 15:02 flow_20170717-192339_trusted_hostname.xml  
-rw-r--r-- 1 alopresto staff 2.7K Apr 19 15:02 flow_20170718-101751_evaluate_json_path.xml  
-rw-r--r-- 1 alopresto staff 5.2K Apr 19 15:02 flow_20170721-175745_site_to_site_secure.xml  
-rw-r--r-- 1 alopresto staff 5.2K Apr 19 15:02 flow_20170724-161539.xml  
-rw-r--r-- 1 alopresto staff 5.7K Apr 19 15:02 flow_20170724-181056_listen_http_sslv3.xml  
-rw-r--r-- 1 alopresto staff 2.3K Apr 19 15:02 flow_20170726-155905_encrypted_spk.xml  
-rw-r--r-- 1 alopresto staff 5.2K Apr 19 15:02 flow_20170802-172619_simple_gen_and_log.xml  
-rw-r--r-- 1 alopresto staff 5.2K Apr 19 15:02 flow_20170804-164011_simple_gen_and_log.xml  
-rw-r--r-- 1 alopresto staff 1.7K Apr 19 15:02 flow_20170807-154118_groovy_json_uppercase.xml  
-rw-r--r-- 1 alopresto staff 1.8K Apr 19 15:02 flow_20170807-220536_put_mongo_record.xml  
-rw-r--r-- 1 alopresto staff 6.1K Apr 19 15:02 flow_20170809-105304_date_difference.xml  
-rw-r--r-- 1 alopresto staff 6.4K Apr 19 15:02 flow_20170809-174400_json_record_path.xml  
-rw-r--r-- 1 alopresto staff 1.3K Apr 19 15:02 flow_20170809-210516_validate_xml.xml  
-rw-r--r-- 1 alopresto staff 1.4K Apr 19 15:02 flow_20170815-102522_json_to_sql.xml  
-rw-r--r-- 1 alopresto staff 1.5K Apr 19 15:02 flow_20170822-104407_json_formatter.xml  
-rw-r--r-- 1 alopresto staff 1.7K Apr 19 15:02 flow_20170823-123729_evaluatexpath.xml  
-rw-r--r-- 1 alopresto staff 2.5K Apr 19 15:02 flow_20170825-115758_encrypt_and_decrypt.xml  
-rw-r--r-- 1 alopresto staff 2.0K Apr 19 15:02 flow_20170825-121351_encrypt_decrypt_clean.xml  
-rw-r--r-- 1 alopresto staff 2.0K Apr 19 15:02 flow_20170825-121656_encrypt_decrypt_aligned.xml  
-rw-r--r-- 1 alopresto staff 2.3K Apr 19 15:02 flow_20170825-190225_listen_http_and_invoke_http_with_tls1_2.xml  
-rw-r--r-- 1 alopresto staff 5.0K Apr 19 15:02 flow_20170831-153220_psaltis_meetup_template_local.xml  
-rw-r--r-- 1 alopresto staff 4.0K Apr 19 15:02 flow_20170905-175343_xxe_template.xml  
-rw-r--r-- 1 alopresto staff 4.3K Apr 19 15:02 flow_20170905-185907_funnel_bus.xml  
-rw-r--r-- 1 alopresto staff 8.1K Apr 19 15:02 flow_20170906-111801_chained_get_html_element_NIFI-4356.xml  
-rw-r--r-- 1 alopresto staff 2.5K Apr 19 15:02 flow_20170909-121848_sydney_demo.xml  
-rw-r--r-- 1 alopresto staff 3.2K Apr 19 15:02 flow_20170919-202215_meetup_sydney.xml  
-rw-r--r-- 1 alopresto staff 3.8K Apr 19 15:02 flow_20170920-190328_crash_course.xml  
-rw-r--r-- 1 alopresto staff 2.9K Apr 19 15:02 flow_20170926-180548_handle_http_with_tls1_2.xml  
-rw-r--r-- 1 alopresto staff 1.7K Apr 19 15:02 flow_20170929-172333_encrypt_decrypt_1_4_0_verification.xml  
-rw-r--r-- 1 alopresto staff 6.3K Apr 19 15:02 flow_20171003-192928_update_record_and_replace_text.xml  
-rw-r--r-- 1 alopresto staff 8.9K Apr 19 15:02 flow_20171004-180529_web_server_and_json.xml  
-rw-r--r-- 1 alopresto staff 3.8K Apr 19 15:02 flow_20171103-183334_groovy_json_to_sql_srsw.xml  
-rw-r--r-- 1 alopresto staff 1.5K Apr 19 15:02 flow_20171127-190651_local_site_to_site.xml  
-rw-r--r-- 1 alopresto staff 2.1K Apr 19 15:02 flow_20180101-145059_registry_rc_verification.xml  
-rw-r--r-- 1 alopresto staff 1.4K Apr 19 15:02 flow_20180104-104211_test_count_text.xml  
-rw-r--r-- 1 alopresto staff 2.0K Apr 19 15:02 flow_20180104-112138_test_count_text_remote.xml  
-rw-r--r-- 1 alopresto staff 1.7K Apr 19 15:02 flow_20180202-100218_consume_jms_test.xml  
-rw-r--r-- 1 alopresto staff 1.7K Apr 19 15:02 flow_20180202-105514.xml  
-rw-r--r-- 1 alopresto staff 1.7K Apr 19 15:02 flow_20180202-131629_consume_jms_10_threads.xml  
-rw-r--r-- 1 alopresto staff 1.7K Apr 19 15:02 flow_20180202-160620_consume_jms_multithread.xml  
-rw-r--r-- 1 alopresto staff 1.6K Apr 19 15:02 flow_20180208-200731_simple_demo_route_on_time_parity.xml  
-rw-r--r-- 1 alopresto staff 5.2K Apr 19 15:02 flow_20180214-182110_jms_deserialization_test.xml  
-rw-r--r-- 1 alopresto staff 2.6K Apr 19 15:02 flow_20180216-205336_attribute_evaluator.xml  
-rw-r--r-- 1 alopresto staff 5.7K Apr 19 15:02 flow_20180228-130457_attribute_sqo.xml  
-rw-r--r-- 1 alopresto staff 8.3K Apr 19 15:02 flow_20180307-171740_NIFI-4928.xml  
-rw-r--r-- 1 alopresto staff 1.5K Apr 19 15:02 flow_20180313-171310_python_process_attribute.xml  
-rw-r--r-- 1 alopresto staff 2.4K Apr 19 15:02 flow_20180315-113634_NIFI-4246-0Auth.xml  
-rw-r--r-- 1 alopresto staff 1.9K Apr 19 15:02 flow_20180328-115056.xml  
-rw-r--r-- 1 alopresto staff 2.5K Apr 19 15:02 flow_20180405-111419_prioritizers.xml
```

NiFi Registry for Dataflows

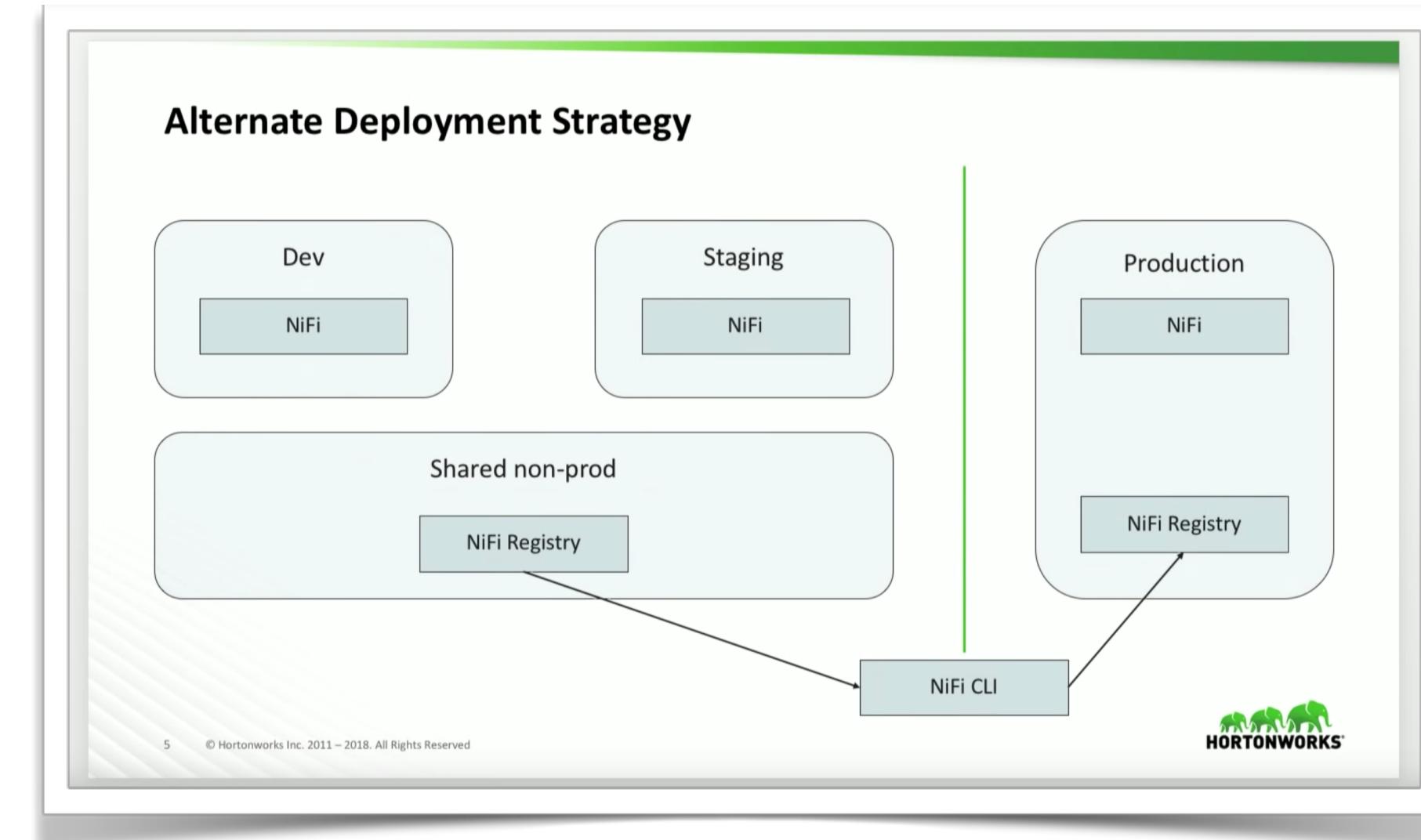
Introducing Apache NiFi Registry 0.3.0

- Previously, flows were exported via XML templates
 - Didn't contain sensitive values
 - Couldn't be updated in-place
 - No tracking system
- NiFi Registry brings asset management as first-class citizen to NiFi
- Flows can be versioned

The screenshot shows the Apache NiFi Registry interface. At the top, there's a header with the NiFi Registry logo, a dropdown menu labeled 'NiFi Registry / All ▾', and user information like 'registry_user' and 'LOGOUT'. Below the header, there's a search bar and a sorting dropdown set to 'Sort by: Name (a - z)'. The main area displays two flows: 'Flow 1 - Bucket 1' (1 version) and 'Flow 2 - Bucket 2' (2 versions). For 'Flow 2 - Bucket 2', a detailed view is shown with a 'DESCRIPTION' field containing 'Description 2'. To the right, a 'CHANGE LOG' section shows two entries: 'Version 2 - 40 minutes ago by registry_user' and 'Add processors Dec-26-2017 at 11:23 PM'. Another entry, 'Version 1 - 41 minutes ago by registry_user', is also listed.

Flows can be promoted between environments

- Connect multiple NiFi instances to a NiFi Registry instance
- Communicate between multiple NiFi Registry instances
 - via multiple Registry Clients
 - via *NiFi CLI*



Extensibility

- Git-backed persistence
 - Share flows via GitHub, etc.
- Commit hooks
 - Register a hook & action
 - “When a new version of the flow is committed to QA Registry, email the QA team and post in the QA Deploy Slack channel”
- Pluggable DB implementations

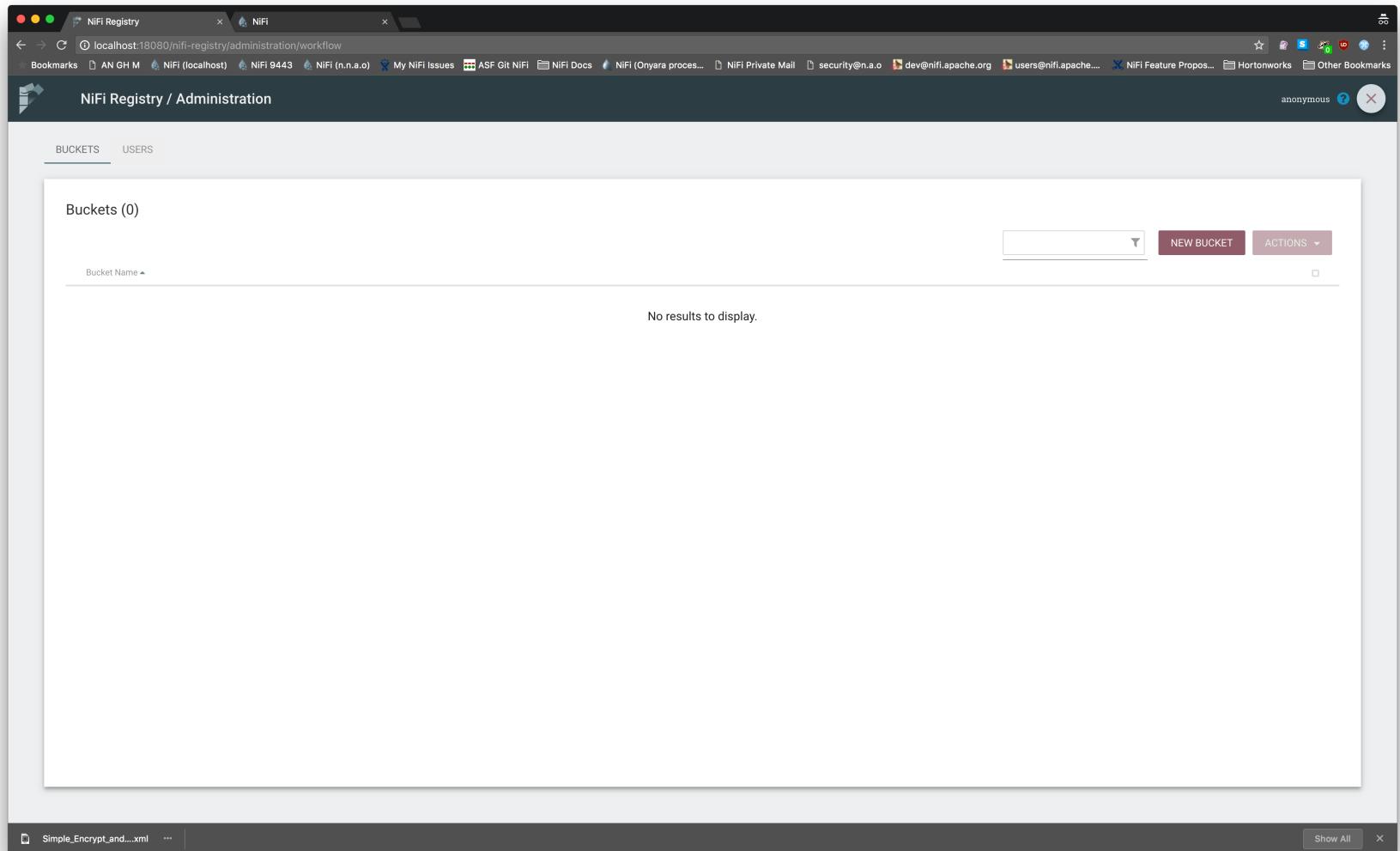
Event Hooks

Event hooks are an integration point that allows for custom code to be triggered when NiFi Registry application events occur.

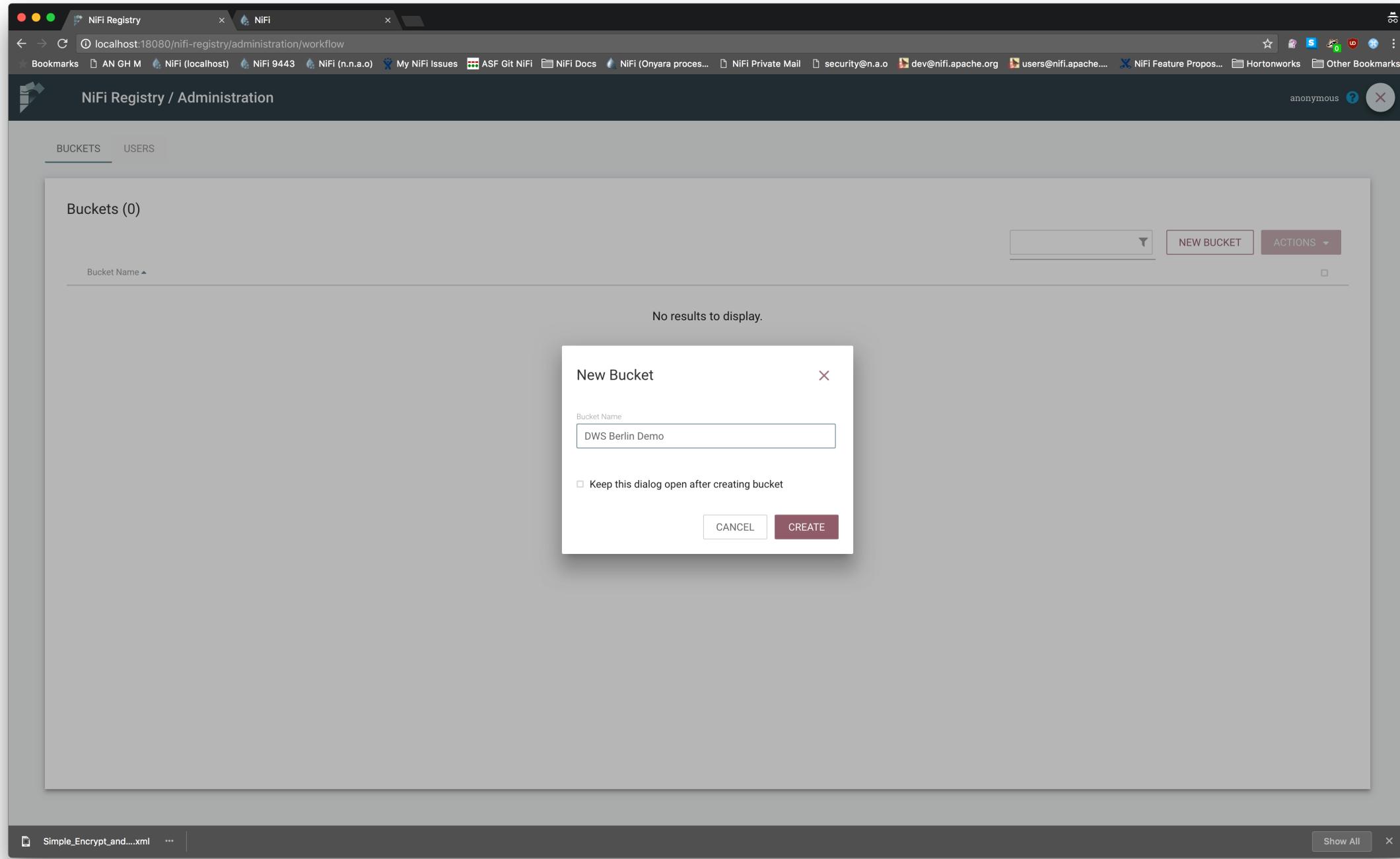
Event Name	Description
CREATE_BUCKET	A new registry bucket is created.
CREATE_FLOW	A new flow is created in a specified bucket. Only triggered on first time creation of a flow with a given name.
CREATE_FLOW_VERSION	A new version for a flow has been saved in the registry.
UPDATE_BUCKET	A bucket has been updated.
UPDATE_FLOW	A flow that exists in a bucket has been updated.
DELETE_BUCKET	An existing bucket in the registry is deleted.
DELETE_FLOW	An existing flow in the registry is deleted.
REGISTRY_START	Invoked once the NiFi Registry application has been successfully started. This is only invoked after a complete and successful start.

Create Registry

- Install nifi-registry
- \$ mvn clean install
- \$./bin/nifi-registry.sh start
- Browse to <http://localhost:18080>



Create Bucket



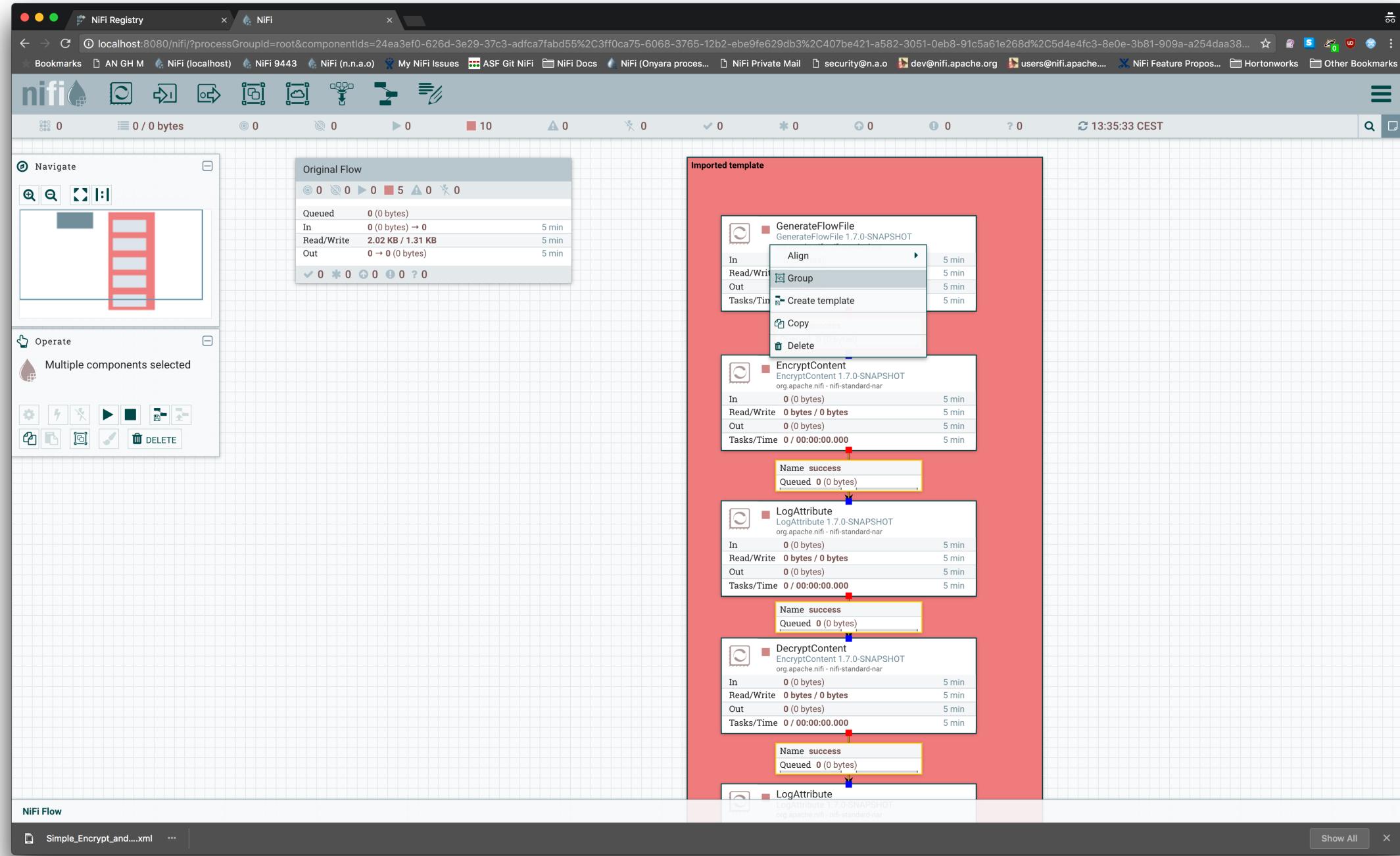
Connect to NiFi

The screenshot shows the NiFi Registry interface in a web browser. The title bar indicates the window is titled "NiFi Registry" and the address bar shows "localhost:8080/nifi/". The main content area is titled "NiFi Settings" and displays a table of "REGISTRY CLIENTS". The table has columns for "Name", "Location", and "Description". A single entry is listed:

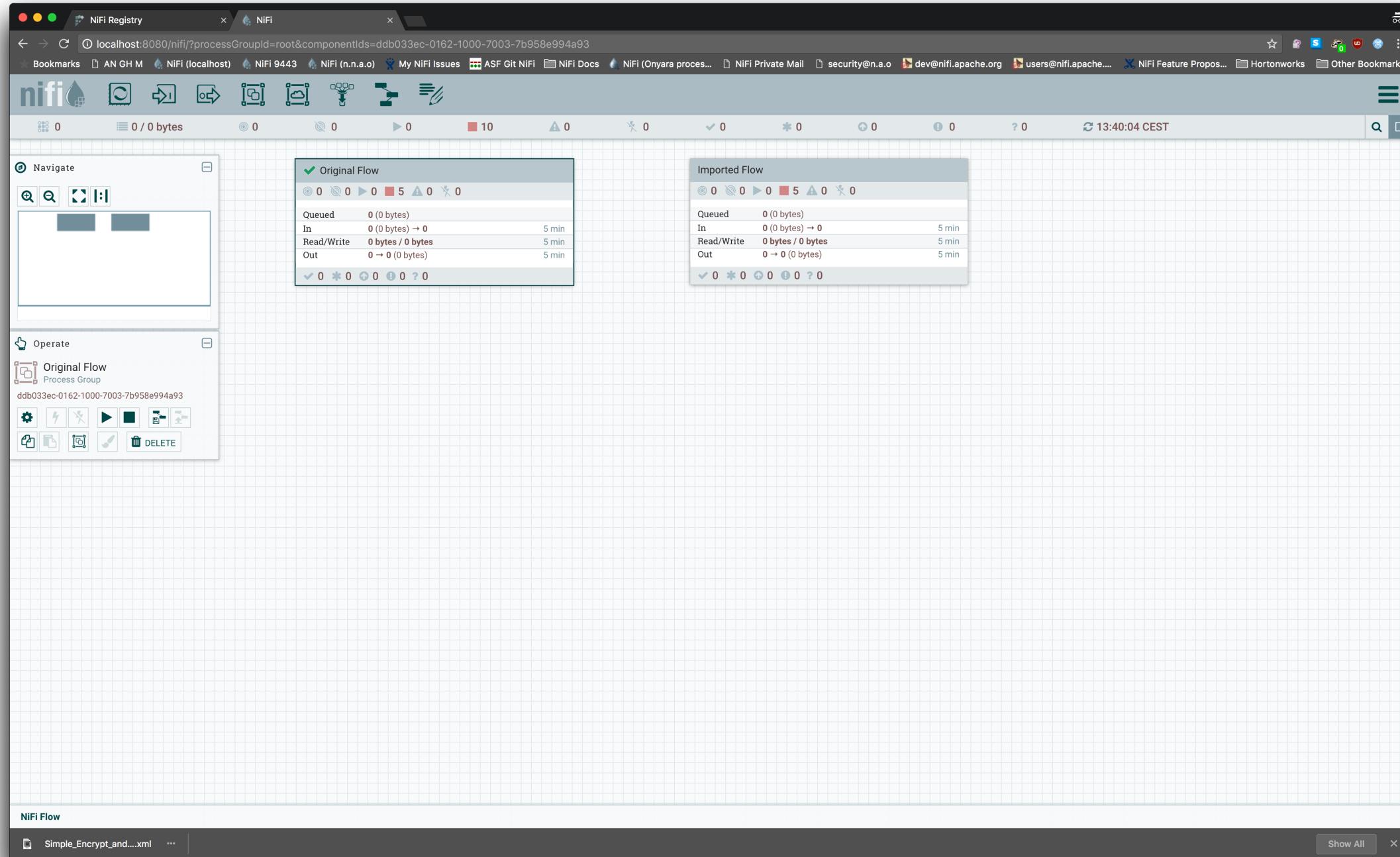
Name	Location	Description
DWS Berlin Registry	http://localhost:18080	A NiFi Registry instance used during the DWS Berlin FDLC demo.

At the bottom left of the main content area, there is a message: "Last updated: 13:36:00 CEST". Below the main content area, a "NIFI FLOW" tab is visible with the file name "Simple_Encrypt_and...xml".

Create Process Group



Commit Version



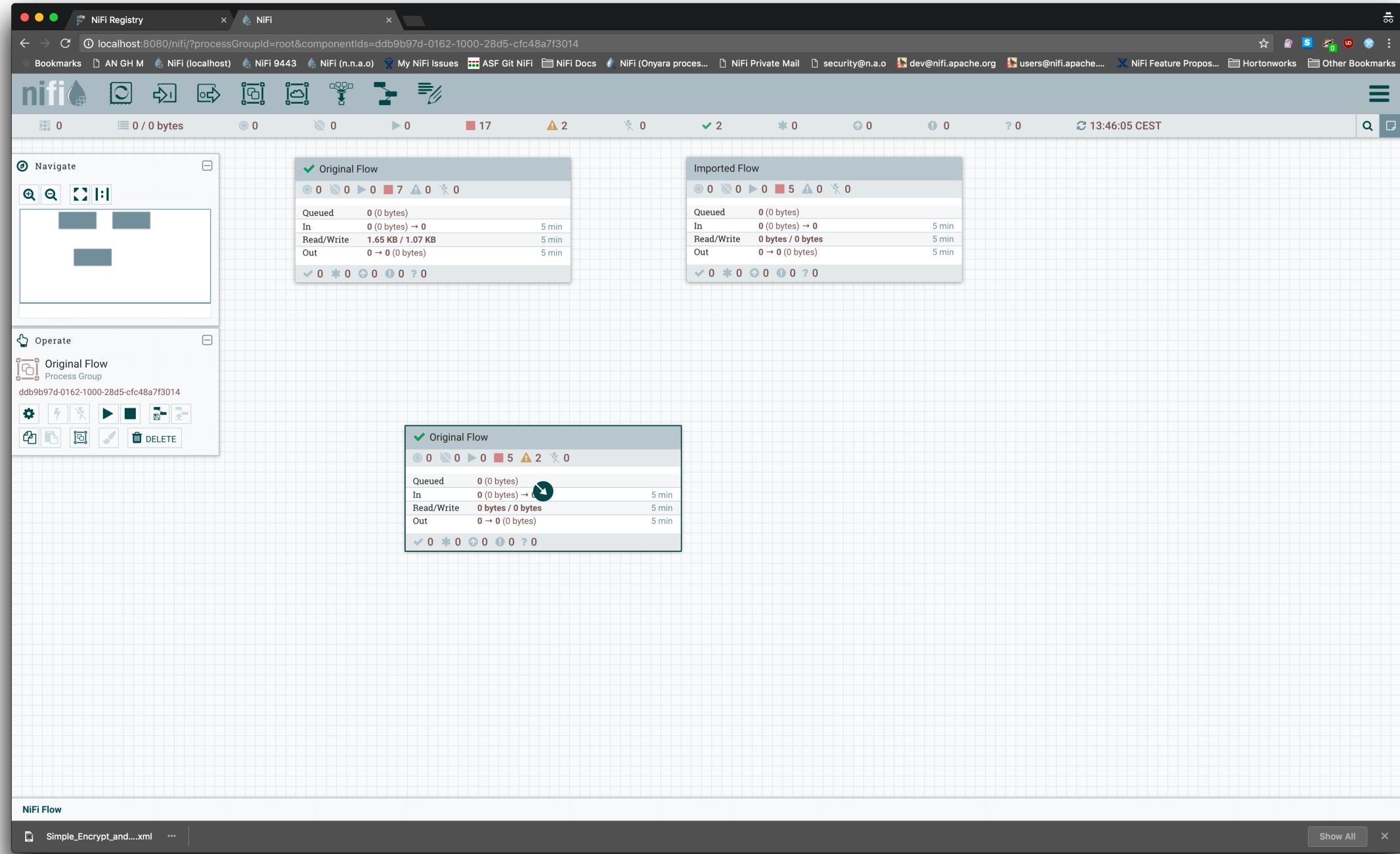
View flow in Registry

The screenshot shows a web browser window for the NiFi Registry at localhost:18080/nifi-registry/explorer/grid-list. The title bar says "NiFi Registry" and "NiFi". The address bar shows the URL. The page header includes a search bar, a sort dropdown set to "Name (a-z)", and a user status "anonymous". The main content area displays a flow named "Simple Encrypt And Decrypt Flow - DWS Berlin Demo". To the right of the flow name is a "VERSIONS" section showing "2". Below this, there are sections for "BUCKET IDENTIFIER" (d7df1fc1-d14b-4bb5-a8a6-ea57a60c5c98), "FLOW IDENTIFIER" (b288f8eb-8adf-455a-b887-785239e97d33), and "DESCRIPTION" (This flow generates plaintext data, encrypts it using AES GCM, logs the cipher text, decrypts it, and logs the plaintext.). On the right side of the flow details, there is a "CHANGE LOG" section with two entries:

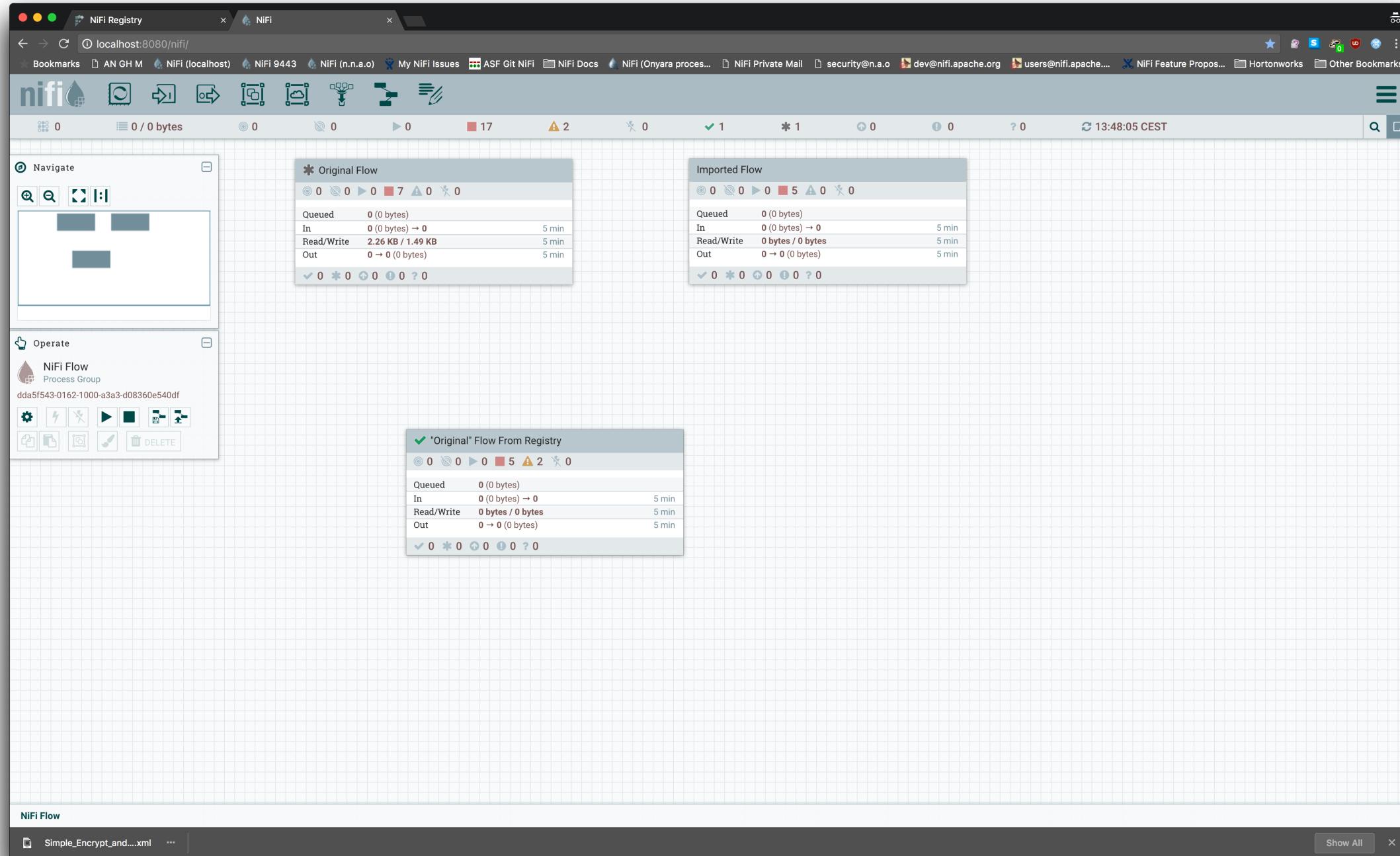
- Version 2 - a few seconds ago by anonymous: Added UpdateAttribute processors to indicate when content was plaintext/encrypted. (Apr-19-2018 at 1:45 PM)
- Version 1 - 5 minutes ago by anonymous: (no details provided)

At the bottom left of the main content area, there is a file icon followed by "Simple_Encrypt_and....xml" and three dots (...). At the bottom right, there are "Show All" and "X" buttons.

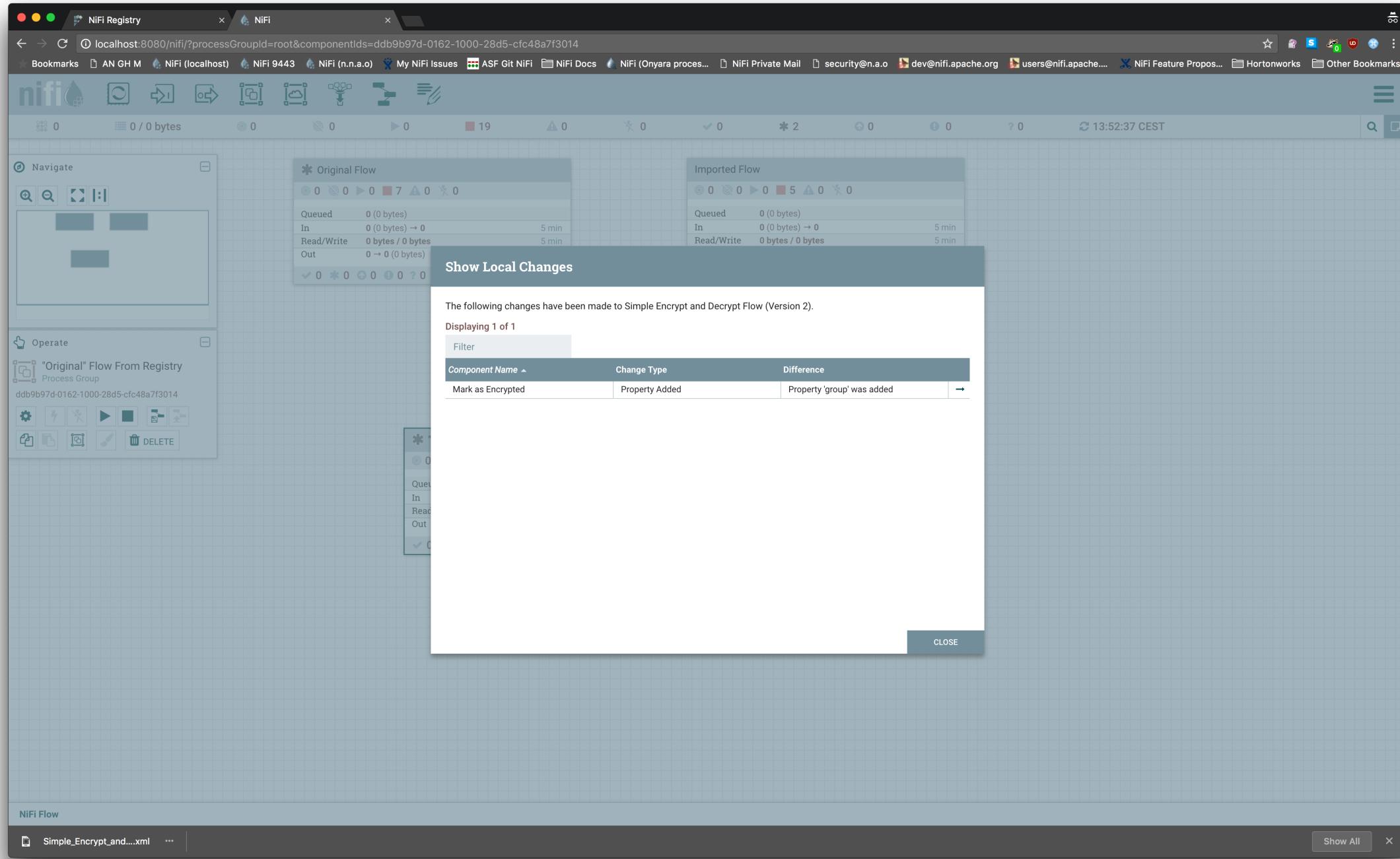
Import new instance into NiFi



Modify the original flow



See local changes before committing



Commit

The screenshot shows a web browser window for the NiFi Registry at localhost:18080/nifi-registry/explorer/grid-list. The page displays a single flow entry:

Simple Encrypt And Decrypt Flow - DWS Berlin Demo (Flow) | VERSIONS 2

BUCKET IDENTIFIER: d7df1fc1-d14b-4bb5-a8a6-ea57a60c5c98

FLOW IDENTIFIER: b288f8eb-8adf-455a-b887-785239e97d33

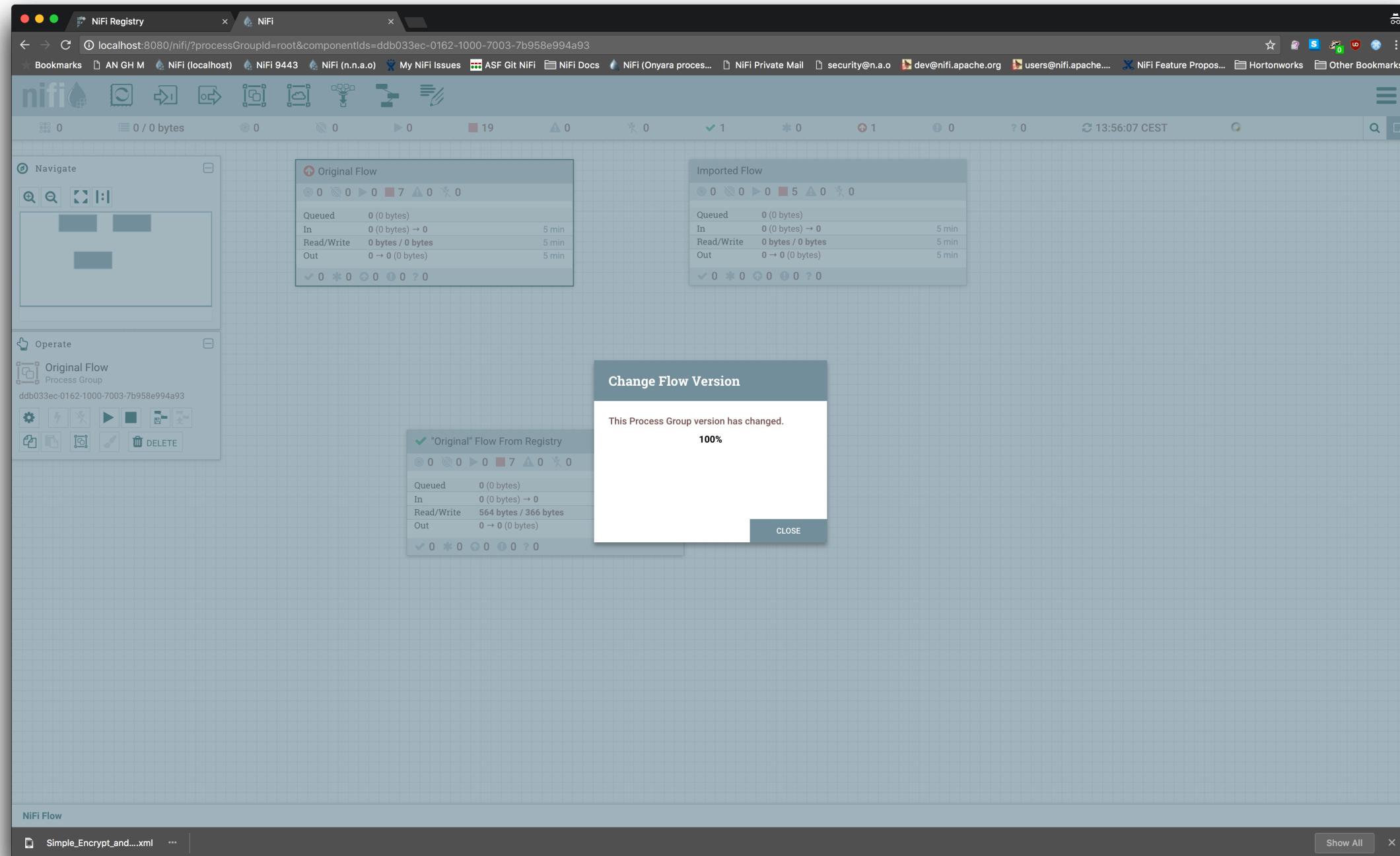
DESCRIPTION: This flow generates plaintext data, encrypts it using AES GCM, logs the cipher text, decrypts it, and logs the plaintext.

CHANGE LOG (3 entries):

- Version 3 - a few seconds ago (by anonymous): Changed the cipher used to AES CTR, added the group as an attribute, and added a local variable for group. (Apr-19-2018 at 1:53 PM)
- Version 2 - 8 minutes ago (by anonymous):
- Version 1 - 13 minutes ago (by anonymous):

ACTIONS button is visible in the top right corner of the flow card.

Update new instance from Registry



Complementary Tools

NiFi Toolkit

- TLS Toolkit
 - Generates, signs, and packages keys and certificates for NiFi services (node/cluster, clients)
- Encrypt Config
 - Protects sensitive configuration values like passwords
- CLI
 - Interacts with NiFi & NiFi Registry to operate on flows

The default output type in interactive mode is `simple`, and the default output type in standalone mode is `json`.

Example of simple output for list-buckets:

```
#> registry list-buckets -ot simple  
  
My Bucket - 3c7b7467-0012-4d8f-a918-6aa42b6b9d39
```

Example of json output for list-buckets:

```
#> registry list-buckets -ot json  
[ {  
  "identifier" : "3c7b7467-0012-4d8f-a918-6aa42b6b9d39",  
  "name" : "My Bucket",  
  "createdTimestamp" : 1516718733854,  
  "permissions" : {  
    "canRead" : true,  
    "canWrite" : true,  
    "canDelete" : true  
  },  
  "link" : {  
    "params" : {  
      "rel" : "self"  
    },  
    "href" : "buckets/3c7b7467-0012-4d8f-a918-6aa42b6b9d39"  
  }  
} ]
```

Security

Security

- Secure the instance
 - Sensitive properties
 - HTTPS
 - Authentication & Authorization
- Secure the configuration
 - Encrypt the configs
- Secure the data
 - EncryptContent
 - Encrypted repositories

Configure Processor

SETTINGS SCHEDULING PROPERTIES COMMENTS

Required field +

Property	Value
Mode	Encrypt
Key Derivation Function	None
Encryption Algorithm	AES_GCM
Allow insecure cryptographic modes	Not Allowed
Password	No value set
Raw Key (hexadecimal)	Sensitive value set
Public Keyring File	In keyed encryption, this is the raw key, encoded in hexadecimal
Public Key User Id	Supports expression language: false
Private Keyring File	History: • ***** - 02/21/2017 20:30:46 PST (CN=alopresto, OU=Apache NiFi)
Private Keyring Passphrase	No value set No value set

CANCEL APPLY

Sensitive Component Properties

- NiFi encrypts all sensitive component properties (*database password, FTP password, etc.*) with a configurable algorithm
 - Default is PBEWITHMD5AND256BITAES-CBC-OPENSSL
- If no key material is provided by the admin, a default is used
 - TBC in 2.0.0
 - **Populate this with a random, unique value when deploying NiFi** `nifi.sensitive.props.key=thisIsBadButWayBetterThanNothing`

```
148 # security properties #
149 nifi.sensitive.props.key=
150 nifi.sensitive.props.key.protected=
151 nifi.sensitive.props.algorithm=PBEWITHMD5AND256BITAES-CBC-OPENSSL
152 nifi.sensitive.props.provider=BC
153 nifi.sensitive.props.additional.keys=
```

```
/***
 * This constructor creates an encryptor using <em>Password-Based Encryption</em> (PBE). The <em>key</em> value is
 * the direct value provided in <code>nifi.sensitive.props.key</code> in
 * <code>nifi.properties</code>, which is a <em>PASSWORD</em> rather than a <em>KEY</em>, but is named such for
 * backward/legacy logical compatibility throughout the rest of the codebase.
 * <p>
 * For actual raw key provision, see {@link #StringEncryptor(String, String, byte[])}.
 *
 * @param algorithm the PBE cipher algorithm ({@link EncryptionMethod#algorithm})
 * @param provider the JCA Security provider ({@link EncryptionMethod#provider})
 * @param key      the UTF-8 characters from nifi.properties -- nifi.sensitive.props.key
 */
public StringEncryptor(final String algorithm, final String provider, final String key) {
    this.algorithm = algorithm;
    this.provider = provider;
    this.key = null;
    this.password = new PBEKeySpec(key == null
        ? DEFAULT_SENSITIVE_PROPS_KEY.toCharArray()
        : key.toCharArray());
    initialize();
}
```

First Step After Uncompressing NiFi

- Change the default key material
- Input to KDF to generate the symmetric key used for sensitive processor property value encryption

```
139 # security properties #
140 nifi.sensitive.props.key=This is not the default NiFi key material.
141 nifi.sensitive.props.key.protected=
142 nifi.sensitive.props.algorithm=PBEWITHMD5AND256BITAES-CBC-OPENSSL
143 nifi.sensitive.props.provider=BC
144 nifi.sensitive.props.additional.keys=
145
```

```
125     <name>Encryption Algorithm</name>
126     <value>AES_GCM</value>
127   </property>
128   <property>
129     <name>allow-weak-crypto</name>
130     <value>not-allowed</value>
131   </property>
132   <property>
133     <name>Password</name>
134   </property>
135   <property>
136     <name>raw-key-hex</name>
137     <value>enc{9D6F3EE98D4D04CD7875ED77E9EDBC30526820CDD2C764C45DCBBC0CBBC839FB2D80B
B0477779BFB49DD8FC262E05C9F2EB6BDED343BA8D116395BC4C34B4DA41C3978DEC4CA939D08552
A8DE9E089481959180E3DEAE537BE1BA278FA8F90C9}</value>
138   </property>
```

Configure HTTPS

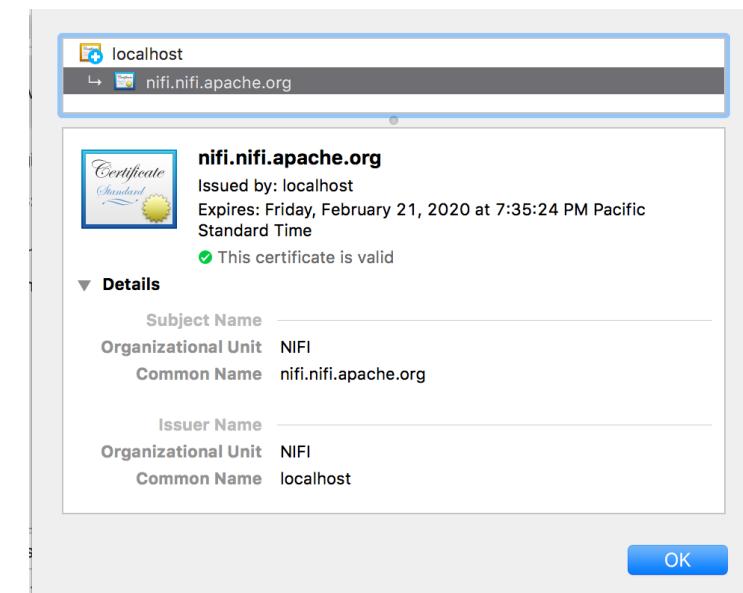
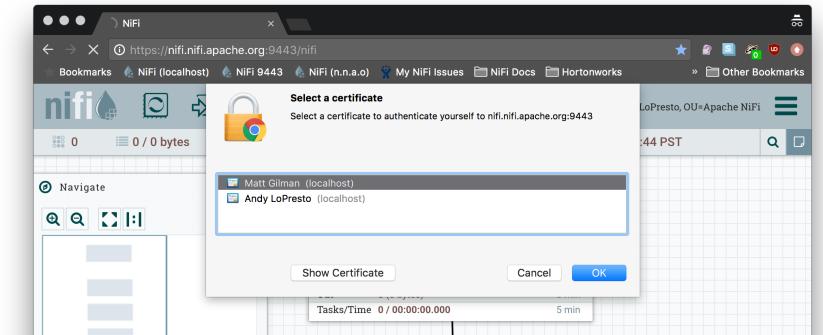
- Deploying a keystore & truststore secures the network connections and enables authentication
 - Keystore = “who am I?”
 - Truststore = “who should I listen to?”

```
nifi.properties (nifi) | nifi.properties (scratch)
A >> > nifi-1.4.0-SNAPSHOT > conf > nifi.properties <- B >> > Workspace > scratch > secure_nifi > nifi.properties <-
Kaleidoscope
nifi.properties ...Workspace/scratch/secure_nifi
nifi.properties ...bin/nifi-1.4.0-SNAPSHOT/conf

120 # Site to Site properties
121 nifi.remote.input.host=-
122 nifi.remote.input.secure=false
123 nifi.remote.input.socket.port=-
124 nifi.remote.input.http.enabled=true
125 nifi.remote.input.http.transaction.ttl=30 sec
126
127 # web properties #
128 nifi.web.war.directory=./lib
129 nifi.web.http.host=
130 nifi.web.http.port=8080
131 nifi.web.http.network.interface.default=-
132 nifi.web.https.host=-
133 nifi.web.https.port=-
134 nifi.web.https.network.interface.default=-
135 nifi.web.jetty.working.directory=./work/jetty
136 nifi.web.jetty.threads=200
137
138 # security properties #
139 nifi.sensitive.props.key=
140 nifi.sensitive.props.key.protected=
141 nifi.sensitive.props.algorithm=PBEWITHMD5AND256BITAES-CBC-OPENSSL
142 nifi.sensitive.props.provider=BC
143 nifi.sensitive.props.additional.keys=
144
145 nifi.security.keystore=-
146 nifi.security.keystoreType=jks
147 nifi.security.keystorePasswd=password
148 nifi.security.keyPasswd=password
149 nifi.security.truststore=-
150 nifi.security.truststoreType=jks
151 nifi.security.truststorePasswd=password
152 nifi.security.truststorePasswd=password
153 nifi.security.needClientAuth=
115 # Site to Site properties
116 nifi.remote.input.host=nifi.nifi.apache.org
117 nifi.remote.input.secure=true
118 nifi.remote.input.socket.port=10443
119 nifi.remote.input.http.enabled=true
120 nifi.remote.input.http.transaction.ttl=30 sec
121
122 # web properties #
123 nifi.web.war.directory=./lib
124 nifi.web.http.host=
125 nifi.web.http.port=-
126 nifi.web.https.host=nifi.nifi.apache.org
127 nifi.web.https.port=9443
128
129 nifi.web.jetty.working.directory=./work/jetty
130 nifi.web.jetty.threads=200
131
132 # security properties #
133 nifi.sensitive.props.key=
134 nifi.sensitive.props.key.protected=
135 nifi.sensitive.props.algorithm=PBEWITHMD5AND256BITAES-CBC-OPENSSL
136 nifi.sensitive.props.provider=BC
137 nifi.sensitive.props.additional.keys=
138
139 nifi.security.keystore=/conf/nifi.nifi.apache.org/keystore.jks
140 nifi.security.keystoreType=jks
141 nifi.security.keystorePasswd=password
142 nifi.security.keyPasswd=password
143 nifi.security.truststore=/conf/nifi.nifi.apache.org/truststore.jks
144 nifi.security.truststoreType=jks
145 nifi.security.truststorePasswd=password
146 nifi.security.needClientAuth=
```

Enable TLS (SSL)

- Three ways to do this
 - Self-signed certificate generation
(manual / hard)
 - Existing enterprise certificates
(manual / hard)
 - NiFi TLS Toolkit
(automatic / easy)



■ Secure Connection

The connection to this site is encrypted and authenticated using a strong protocol (TLS 1.2), a strong key exchange (ECDHE_RSA with P-256), and a strong cipher (AES_256_GCM).

NiFi TLS Settings

- HDF 3.0 (NiFi 1.2.0+) API/UI supports only TLSv1.2+
 - Strong cipher suites
 - No SHA-1*
 - Prevents downgrade attacks
- Can still support legacy protocols for external service ingest

NiFi TLS Toolkit

- Can run in *standalone* or *client/server* mode
- Automatically generates NiFi Certificate Authority (CA)
- Generates certificates for n nodes
- Signed by CA via CSR with shared-secret token
- Builds keystore & truststore
- Populates passwords in `nifi.properties`
- Generates client certificate for authentication
- Strong random passwords generated if not provided

```
Run toolkit in standalone mode ./bin/tls-toolkit.sh standalone
Generate a certificate for this hostname -n 'nifi.nifi.apache.org'
Generate a client certificate for this user -C 'CN=alopresto, OU=Apache NiFi'
Set the truststore password -P someLongTruststorePassword
Set the keystore password -S someLongKeystorePassword
Set the client cert password -B someLongClientCertPassword
Provide the input/output nifi.properties -f ../../../../../../nifi-assembly/target/
nifi-1.3.0-bin/nifi-1.3.0/conf/nifi.properties
Provide the output directory for the keys -o ../../../../../../nifi-assembly/target/
nifi-1.3.0-bin/nifi-1.3.0/conf/
```

```
hw12203:nifi-toolkit-1.3.0 (master) alopresto
└─ 15s @ 16:51:23 $ ./bin/tls-toolkit.sh standalone -n 'nifi.nifi.apache.org'
-C 'CN=alopresto, OU=Apache NiFi' -P someLongTruststorePassword -S
someLongKeystorePassword -B someLongClientCertPassword -f .../conf/
nifi.properties -o .../conf/
o.a.n.t: CommandLine: Using .../conf/nifi.properties as template.
o.a.n.t: Running standalone certificate generation with output directory ...
conf
o.a.n.t: Using existing CA certificate .../conf/nifi-cert.pem and key .../conf/
nifi-key.key
o.a.n.t: Writing new ssl configuration to .../conf/nifi.nifi.apache.org
o.a.n.t: Successfully generated TLS configuration for nifi.nifi.apache.org 1 in
.../conf/nifi.nifi.apache.org
o.a.n.t: Generating new client certificate .../conf/
CN=alopresto_OU=Apache_NiFi.p12
o.a.n.t: Successfully generated client certificate .../conf/
CN=alopresto_OU=Apache_NiFi.p12
o.a.n.t: tls-toolkit standalone completed successfully
hw12203:nifi-toolkit-1.3.0 (master) alopresto
└─ 6s @ 16:51:29 $
```

Standalone Examples

Create 4 sets of keystore, truststore, nifi.properties for localhost along with a client certificate with the given DN:

```
bin/tls-toolkit.sh standalone -n 'localhost(4)' -C 'CN=username,OU=NIFI'
```

Create keystore, truststore, nifi.properties for 10 NiFi hostnames in each of 4 subdomains:

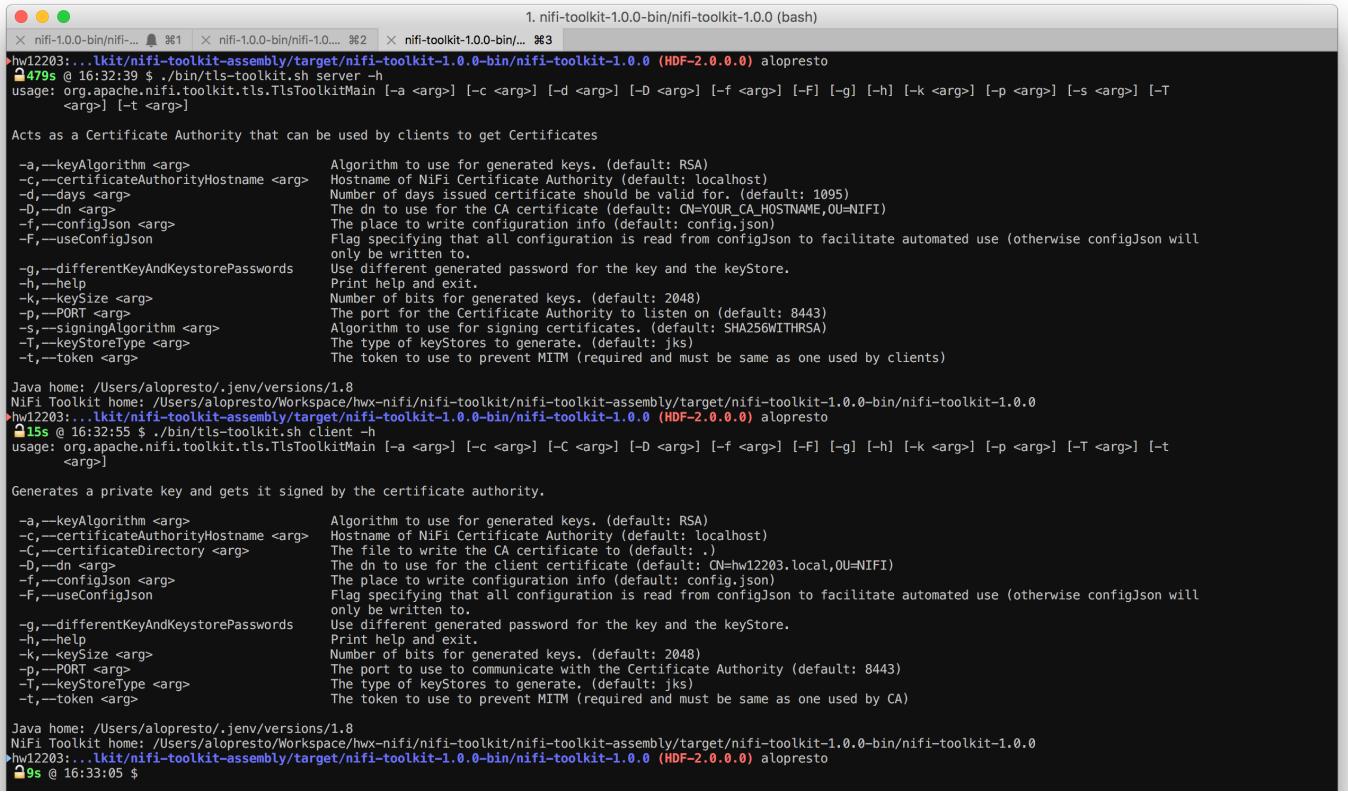
```
bin/tls-toolkit.sh standalone -n 'nifi[01-10].subdomain[1-4].domain'
```

Create 2 sets of keystore, truststore, nifi.properties for 10 NiFi hostnames in each of 4 subdomains along with a client certificate with the given DN:

```
bin/tls-toolkit.sh standalone -n 'nifi[01-10].subdomain[1-4].domain(2)' -C 'CN=username,OU=NIFI'
```

Client/Server Mode

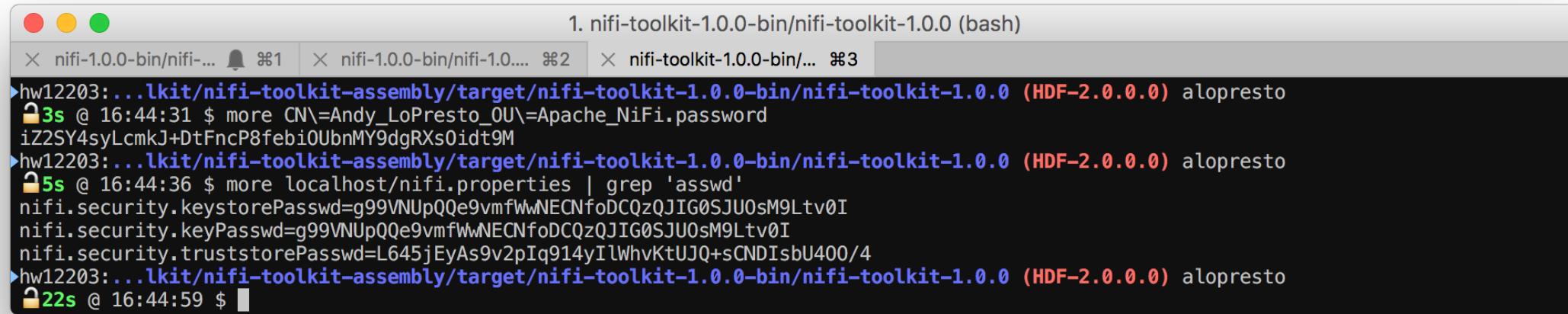
- Long running CA
- Multiple clients can generate a certificate locally and send CSR to CA to be signed



The screenshot shows a terminal window with three tabs open. The active tab is titled 'nifi-toolkit-1.0.0-bin/nifi-toolkit-1.0.0 (bash)'. The command entered is 'nifi-toolkit-assembly/target/nifi-toolkit-1.0.0-bin/nifi-toolkit-1.0.0 (HDF-2.0.0.0) alopresto'. The output shows usage information for the 'server' command, which generates a certificate authority. It includes options for key algorithm (-a), certificate authority hostname (-c), days (-d), dn (-D), config json (-f), use config json (-F), different key and keystore passwords (-g), help (-h), key size (-k), port (-P), signing algorithm (-s), key store type (-T), and token (-t). The output also includes Java home and NiFi Toolkit home paths.

Security Mechanisms

- Private key is not communicated from client to CA; CSR is issued and signed
- Token and HMAC/SHA-256 is used to prevent MitM
- Default passwords generated for keystores, certificates are 32 bytes of secure random, Base64-encoded (~43 chars)



The screenshot shows a macOS terminal window titled "1. nifi-toolkit-1.0.0-bin/nifi-toolkit-1.0.0 (bash)". It has three tabs open: "nifi-1.0.0-bin/nifi...", "nifi-1.0.0-bin/nifi-1....", and "nifi-toolkit-1.0.0-bin/...". The current tab displays the following command-line session:

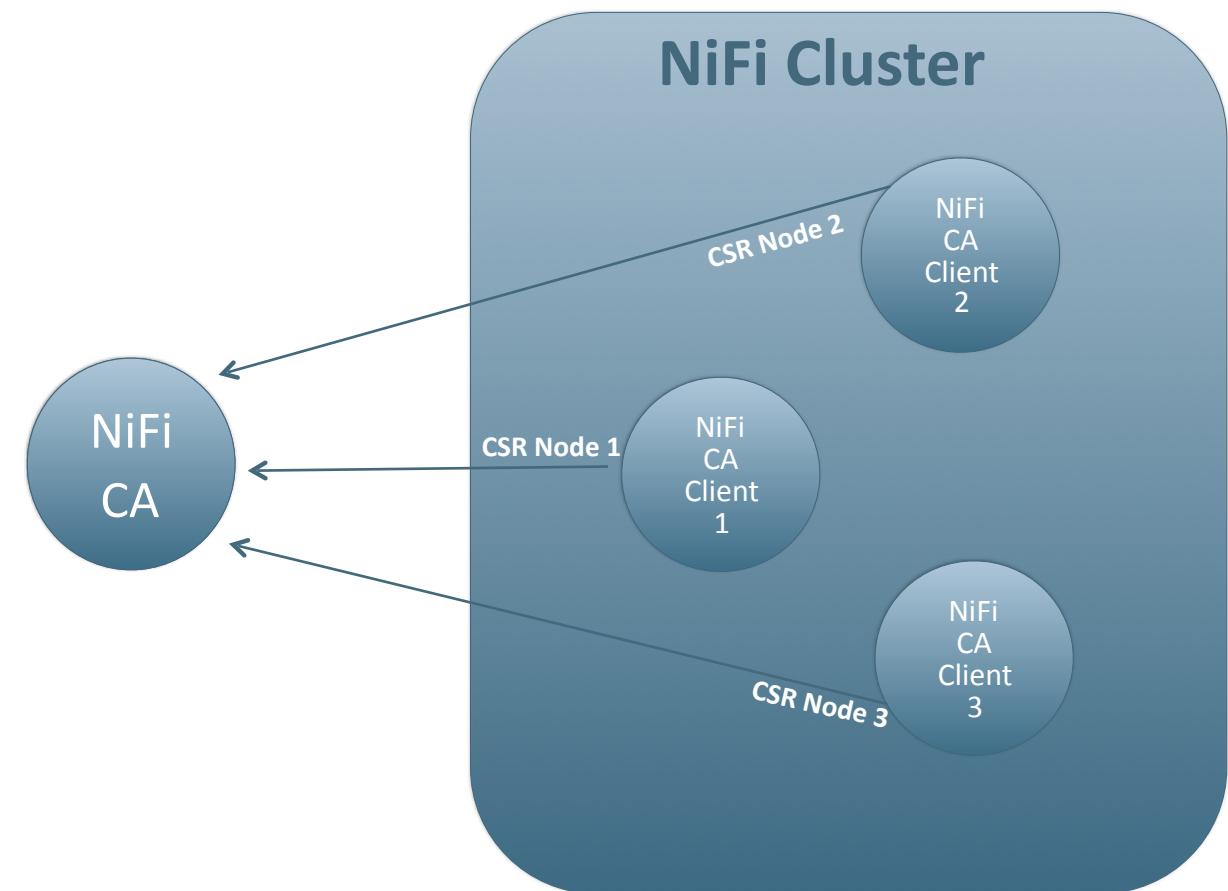
```
▶ hw12203:...lkit/nifi-toolkit-assembly/target/nifi-toolkit-1.0.0-bin/nifi-toolkit-1.0.0 (HDF-2.0.0.0) alopresto
 3s @ 16:44:31 $ more CN\=Andy_LoPresto_0U\=Apache_NiFi.password
 iZ2SY4syLcmkJ+DtFncP8feb0UbnMY9dgRXs0idt9M
▶ hw12203:...lkit/nifi-toolkit-assembly/target/nifi-toolkit-1.0.0-bin/nifi-toolkit-1.0.0 (HDF-2.0.0.0) alopresto
 5s @ 16:44:36 $ more localhost/nifi.properties | grep 'asswd'
 nifi.security.keystorePasswd=g99VNUpQqe9vmfWwNECNfoDCQzQJIG0SJu0sM9Ltv0I
 nifi.security.keyPasswd=g99VNUpQqe9vmfWwNECNfoDCQzQJIG0SJu0sM9Ltv0I
 nifi.security.truststorePasswd=L645jEyAs9v2pIq914yIlWhvKtUJQ+sCNDIsbU400/4
▶ hw12203:...lkit/nifi-toolkit-assembly/target/nifi-toolkit-1.0.0-bin/nifi-toolkit-1.0.0 (HDF-2.0.0.0) alopresto
 22s @ 16:44:59 $
```

NiFi Certificate Authority - Background

- Driving Factors
 - NiFi nodes in a secured cluster need a way to trust and securely communicate with each other (in addition to the user)
 - Generating and deploying trusted certificates manually for nodes can be very time consuming (especially in large clusters)
 - Ambari did not previously have a method to centralize certificate generation & distribution
- Quick Facts
 - Uses NiFi TLS Toolkit – Client / Server Mode
 - Uses a “Shared Secret” scheme to establish trust
 - Installation is NOT required
 - Can be managed similarly to other components (offered as additional option when adding NiFi service)
 - Can be installed on separate host from NiFi nodes
 - **1.8.0+** can use external CA certificate to sign certs (*rather than NiFi-generated only*)

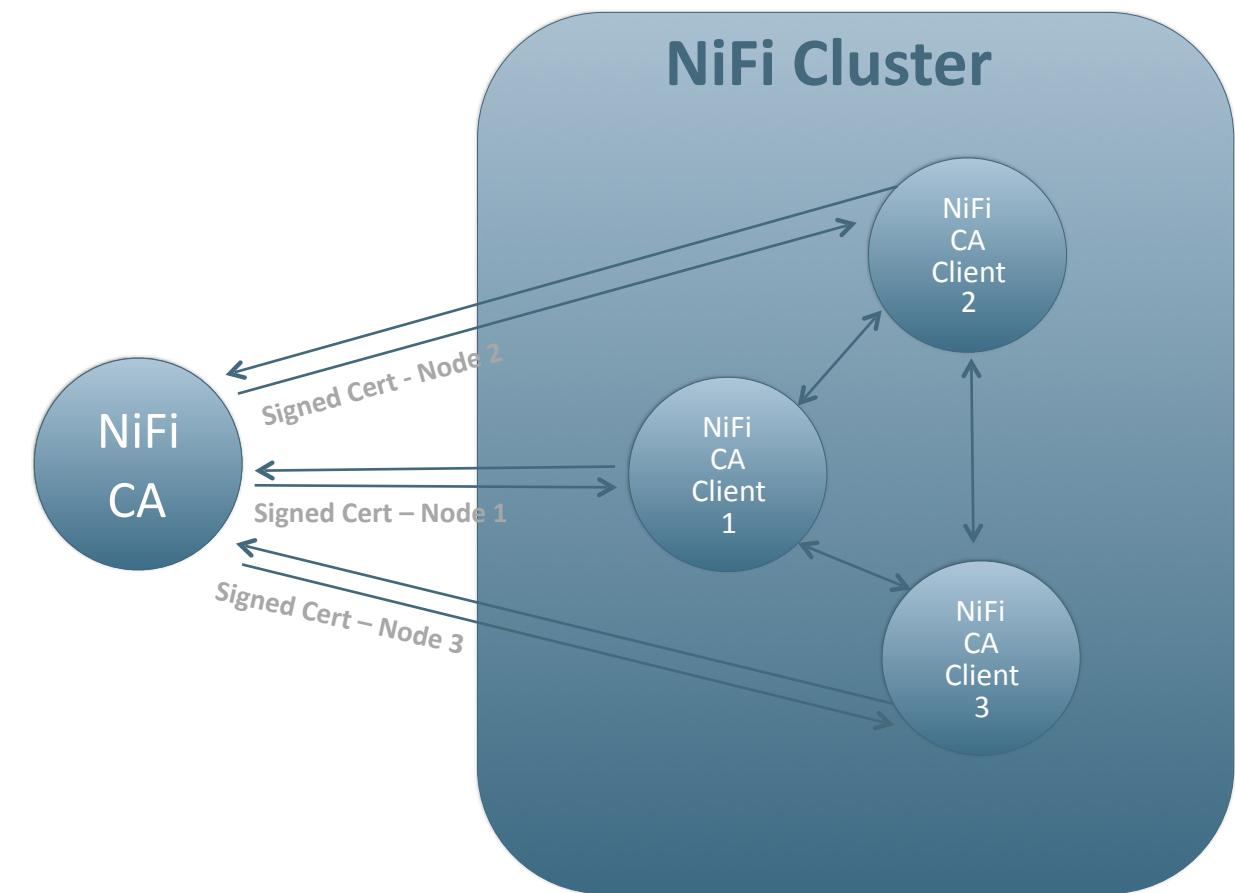
NiFi Certificate Authority – Communication Workflow

1. CA generates a self signed certificate on startup (default port 10443)
2. NiFi CA clients (running on NiFi node hosts), on startup, generate a key pair and a Certificate Signing Request (CSR). The CSR contains the identity of the client, the public key (from the key pair), and a hash generated using the **NiFi CA Token*** and the public key's signature
3. CA clients send the CSR to the CA
4. CA verifies that the CSR is from a trusted NiFi CA Client by generating a hash using its copy of the token with the public key signature in the CSR and compares with the hash included in the request



NiFi Certificate Authority – Communication Workflow

5. When CA has confirmed node is trusted, it will sign the CSR and generate a hash of its public key fingerprint using the same token. Both are sent back to the node
6. Once the client receives the response from the CA, to verify it is the trusted CA, it will obtain the CA's public key (from the TLS connection) and generate a hash using the secret token and the CA's public key signature
7. Upon validation the client will accept the signed certificate. Ambari will then update NiFi properties settings for the newly signed cert to start the NiFi node in SSL mode

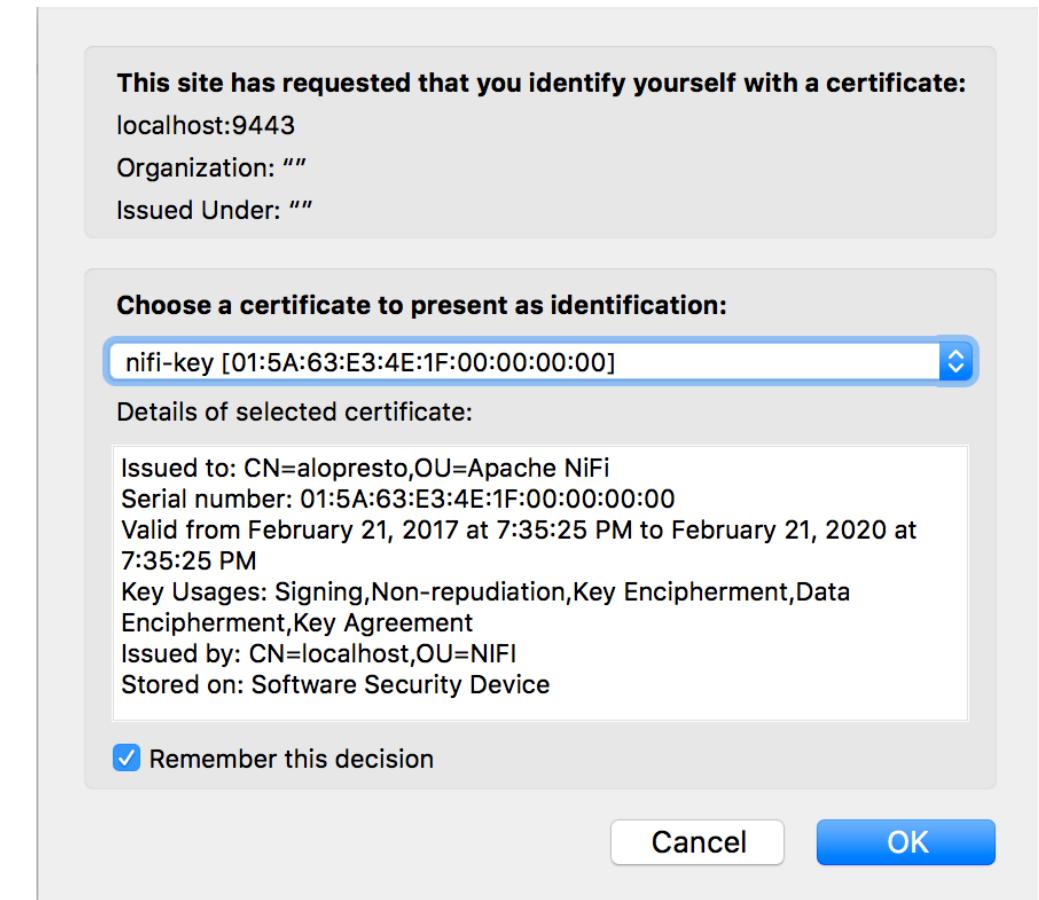


Authentication

- Supported authentication mechanisms
 - Client certificate
 - LDAP
 - Kerberos
 - OpenID Connect
 - KnoxSSO
 - OAuth 2.0 *under review

Client Certificate Authentication

- Always available (original auth mechanism for NiFi)
- Client certificate signed by a CA in the server truststore
- Issuer DN becomes username
- Authenticated on every request



LDAP Authentication

- Configured via `login-identity-providers.xml`
- Supports LDAP w/ StartTLS and LDAPS
- After initial authentication, encrypted JWT is issued via response and placed in local storage
- JWT substituted for LDAP credentials on following requests

```
nifi.security.keystore=./conf/keystore.jks
nifi.security.keystoreType=jks
nifi.security.keystorePasswd=FcD7b/eywqwicpl4ibC2PvvFm2vlz00BZf00blrBd9c
nifi.security.keyPasswd=FcD7b/eywqwicpl4ibC2PvvFm2vlz00BZf00blrBd9c
nifi.security.truststore=./conf/truststore.jks
nifi.security.truststoreType=jks
nifi.security.truststorePasswd=iQ5ZqLZtH2fskrUfXDcpFc/IbVRgidbbvaEvHz5Q3BE
nifi.security.needClientAuth=
nifi.security.user.authorizer=file-provider
nifi.security.user.login.identity.provider=ldap-provider
nifi.security.ocsp.responder.url=
nifi.security.ocsp.responder.certificate=
```

```
<provider>
  <identifier>ldap-provider</identifier>
  <class>org.apache.nifi.ldap.LdapProvider</class>
  <property name="Authentication Strategy">START_TLS</property>
  <property name="Manager DN">CN=admin, OU=NiFi</property>
  <property name="Manager Password">jkhsd7s0bdl1dk3asdk113</property>
  <property name="TLS - Keystore">./conf/keystore.jks</property>
  <property name="TLS - Keystore Password">FcD7b/
eywqwicpl4ibC2PvvFm2vlz00BZf00blrBd9c</property>
  <property name="TLS - Keystore Type">JKS</property>
  <property name="TLS - Truststore">./conf/truststore.jks</property>
  <property name="TLS - Truststore Password">iQ5ZqLZtH2fskrUfXDcpFc/
IbVRgidbbvaEvHz5Q3BE</property>
  <property name="TLS - Truststore Type">JKS</property>
  <property name="TLS - Client Auth">WANT</property>
  <property name="TLS - Protocol">TLS</property>
  <property name="TLS - Shutdown Gracefully"></property>
  <property name="Url">ldap://ldap.nifi.apache.org:389</property>
  <property name="User Search Base">dc=nifi,dc=apache,dc=org</property>
  <property name="User Search Filter">cn={0}</property>
...
</provider>
```

Kerberos Authentication

- Configured via `login-identity-providers.xml`
- Supports SPNEGO via GSS-API and form login as backup
 - *May be removed in future release*
- After initial authentication, encrypted JWT is issued via response and placed in local storage
- JWT substituted for Kerberos credentials on following requests

```
klist
Credentials cache: API:5D669ADD-9028-441F-A6E0-C2C78DA04CC5
Principal: alopresto@NIFI.APACHE.ORG

Issued           Expires          Principal
Jun 2 10:12:29 2017 Jun 22 20:12:27 2017 krbtgt/NIFI.APACHE.ORG@NIFI.APACHE.ORG
```

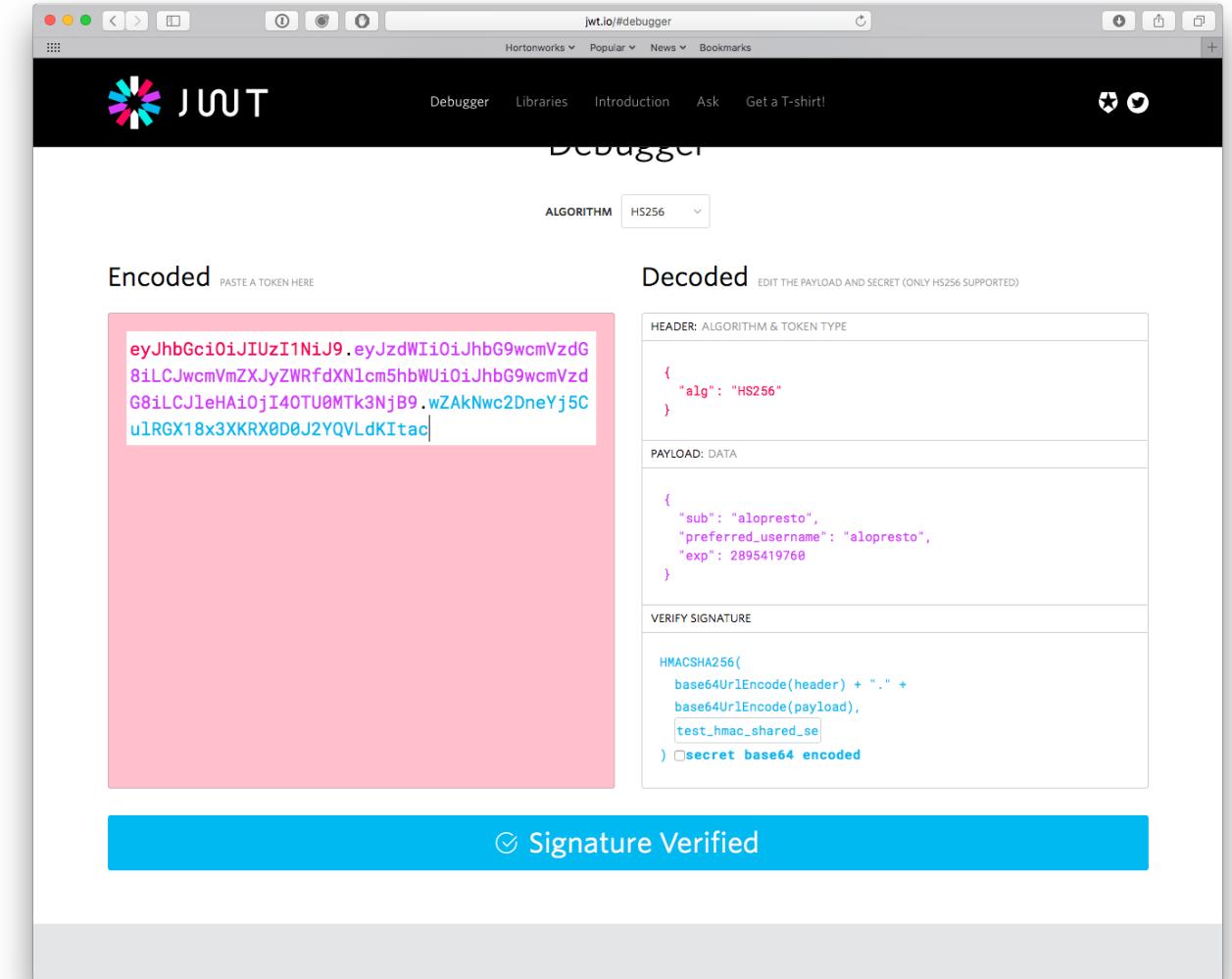
```
nifi.security.user.login.identity.provider=kerberos-provider
nifi.kerberos.krb5.file=/path/to/docker-kdc/krb5.conf
nifi.kerberos.spnego.principal=HTTP/nifi.apache.org@NIFI.APACHE.ORG
nifi.kerberos.spnego.keytab.location=/path/to/docker-kdc/krb5.keytab
nifi.kerberos.spnego.authentication.expiration=12 hours
```



Preference Name	Status	Type	Value
network.negotiate-auth.allow-non-fqdn	default	boolean	false
network.negotiate-auth.allow-proxies	default	boolean	true
network.negotiate-auth.delegation-uris	user set	string	https://nifi.apache.org:9445
network.negotiate-auth.gsslib	default	string	
network.negotiate-auth.trusted-uris	user set	string	https://nifi.apache.org:9445
network.negotiate-auth.using-native-gsslib	default	boolean	true

JWT - JSON Web Tokens

- Encrypted & signed token containing data passed as Base64-encoded **Authorization** header in HTTP request
- Allows for stateless web applications
- Encrypted with per-user key stored by NiFi



Access Policies Enhancements

- Cleaner UI and more convenient for common tasks
 - Collect all user policies in single view (policy-centric vs. user-centric)
 - Fewer clicks to add policies

The screenshot shows a browser window for the NiFi web interface at <https://localhost:9443/nifi/>. The main navigation bar includes links for Bookmarks, NIFI (localhost), NIFI 9443, NIFI (n.n.a.o), My NIFI Issues, NIFI Docs, Hortonworks, and Other Bookmarks. A modal dialog box is open, titled 'NiFi Users' with the sub-section 'User Policies'. The dialog displays two users: 'CN=Andy LoPresto, OU=Apache NiFi' and 'CN=Matt Gilman, OU=Apache NiFi'. Below this, a table lists various policies for the selected user, categorized by policy type (e.g., Component policy, Data policy, Global policy) and action (read, write). Some policies are highlighted in yellow. A note at the bottom states: 'Some policies may be inherited by descendant components unless explicitly overridden.' A 'CLOSE' button is at the bottom right of the dialog.

Restricted Components

- Controller services, reporting tasks, and processors which could maliciously affect the host system are now **RESTRICTED**
- Special annotation and access policy required

Add Processor

Source	Type	Version	Tags
all groups	Type ▾		
amazon	DeleteHDFS	1.4.0-SNAPSHOT	restricted, HDFS, hadoop, delete...
attributes	ExecuteFlumeSink	1.4.0-SNAPSHOT	sink, restricted, flume, hadoop, p...
avro	ExecuteFlumeSource	1.4.0-SNAPSHOT	restricted, flume, get, hadoop, s...
aws	ExecuteProcess	1.4.0-SNAPSHOT	process, external, restricted, inv...
consume	ExecuteScript	1.4.0-SNAPSHOT	luaj, python, jython, clojure, js, ex...
csv	ExecuteStreamCommand	1.4.0-SNAPSHOT	command execution, stream, re...
database	FetchFile	1.4.0-SNAPSHOT	ingress, input, restricted, get, file...
fetch	FetchHDFS	1.4.0-SNAPSHOT	restricted, get, fetch, hdfs, hado...
files	FetchParquet	1.4.0-SNAPSHOT	restricted, get, fetch, HDFS, had...
get	GetFile	1.4.0-SNAPSHOT	ingress, input, restricted, get, file...
hadoop	GetHDFS	1.4.0-SNAPSHOT	restricted, get, fetch, HDFS, had...
ingest	InvokeScriptedProcessor	1.4.0-SNAPSHOT	luaj, python, jython, ivthon, restr...
insert			
json			
listen			
logs			
message			
put			
remote			
restricted			
source			
split			
sql			
text			
update			

Filter

DeleteHDFS 1.4.0-SNAPSHOT org.apache.nifi - nifi-hadoop-nar

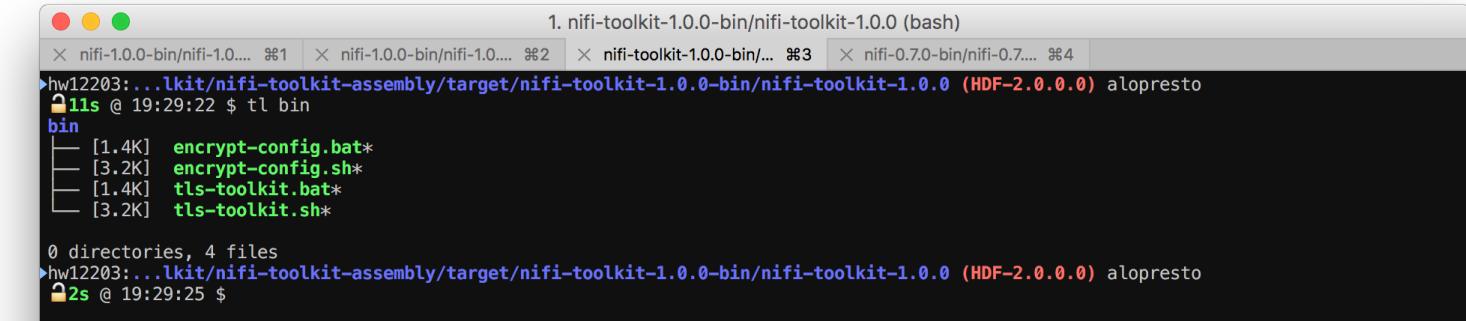
Deletes one or more files or directories from HDFS. The path can be provided as an attribute from an incoming FlowFile, or a statically set path that is periodically removed. If this processor has an incoming connection, it will ignore running on a periodic basis and instead rely on incoming FlowFiles to trigger a delete. Note that you may use a wildcard character to match multiple files o...

CANCEL ADD



Secure Configuration

- Previously, sensitive configuration values were exposed in **plaintext**
 - Recommendation for POSIX file permissions to prevent unauthorized access
- Pluggable architecture which allows **NiFiProperties** to be secured until application load
- Passwords protected by default; arbitrary values can be protected via **nifi.sensitive.props.additional.keys**



```
1. nifi-toolkit-1.0.0-bin/nifi-toolkit-1.0.0 (bash)
hw12203:...lkit/nifi-toolkit-assembly/target/nifi-toolkit-1.0.0-bin/nifi-toolkit-1.0.0 (HDF-2.0.0.0) alopresto
1ls @ 19:29:22 $ tl bin
bin
[1.4K] encrypt-config.bat*
[3.2K] encrypt-config.sh*
[1.4K] tls-toolkit.bat*
[3.2K] tls-toolkit.sh*
0 directories, 4 files
hw12203:...lkit/nifi-toolkit-assembly/target/nifi-toolkit-1.0.0-bin/nifi-toolkit-1.0.0 (HDF-2.0.0.0) alopresto
2s @ 19:29:25 $
```



```
73 # security properties #
74 nifi.sensitive.props.key=n2z+tTbHuZ4V4V2 | jwWhdasxvXD4ZG2lMAes/vqh6u4vaz4xgL4aEbF4Y/dXevgk3uLRc0wf1vc4RDQ
75 nifi.sensitive.props.key.protected=aes/gcm/256
76 nifi.sensitive.props.algorithm=PBEWITHMD5AND256BITAES-CBC-OPENSSL
77 nifi.sensitive.props.provider=BC
78 nifi.sensitive.props.additional.keys=
79
80 nifi.security.keystore=/path/to/keystore.jks
81 nifi.security.keystoreType=JKS
82 nifi.security.keystorePasswd=oBjT92hIGRElIG0h | JMZ6uYuWNBBr0A6usq/Jt3DaD2e4otNirZDytac/w/KFe0H0krJR03ycho
83 nifi.security.keystorePasswd.protected=aes/gcm/256
84 nifi.security.keyPasswd=ac/BaE35SL/esLiJ | +ULrvRLYdIDA2VqpE0eQXDEMjaLBMG2kbK0d0wBk/hGebDKlVg
85 nifi.security.keyPasswd.protected=aes/gcm/256
86 nifi.security.truststore=
87 nifi.security.truststoreType=
88 nifi.security.truststorePasswd=X/RSINr20CJ1Kwe | dENJevX5P61ix+97airrtoB0oyasMFS6DG6fHbX+SZtw2VAMlISSnDeT97Q
89 nifi.security.truststorePasswd.protected=aes/gcm/256
90 nifi.security.needClientAuth=
91 nifi.security.user.authorizer=
```

Config Encryption Tool

- First implementation
 - Future options: HSM integration, Vault, KeyWhiz, etc.
- Command-line tool which accepts plaintext `nifi.properties` file and encrypts values using AES/GCM
 - 128 or 256-bit key depending on system JCE unlimited strength cryptographic jurisdiction policies
- Persists master key in `bootstrap.conf`
- Can also handle `login-identity-providers.xml`, `authorizers.xml`, and `flow.xml.gz`

If We Really Have TLS, Why Encrypt Data?

- All data transmitted over TLS is encrypted
- On NiFi, automatically decrypted
- Attributes visible
- Content still encrypted because of EncryptContent processor
- Can serve as secure route for follow-on systems

Configure Processor

SETTINGS SCHEDULING PROPERTIES COMMENTS

Required field

Property	Value
Mode	Encrypt
Key Derivation Function	None
Encryption Algorithm	AES_GCM
Allow insecure cryptographic modes	Not Allowed
Password	No value set
Raw Key (hexadecimal)	Sensitive value set
Public Keyring File	In keyed encryption, this is the raw key, encoded in hexadecimal
Public Key User Id	Supports expression language: false
Private Keyring File	History: • ***** - 02/21/2017 20:30:46 PST (CN=alopresto, OU=Apache NiFi)
Private Keyring Passphrase	No value set No Value set

CANCEL APPLY

Does It Work?

The screenshot shows a debugger interface with two main sections. The top section, labeled "View as: original", displays memory dump data. The bottom section, labeled "View as: hex", shows assembly code.

Memory Dump (View as: original):

1	駒옙↳棱口擊? ? 檸櫻噉𠮾口𦵽口惹 + 萊駁邝銓姪既口口口丂𦵽𢹔塔 / 講擘瓊𦵽𦵽口𦵽瓈𦵽
---	--

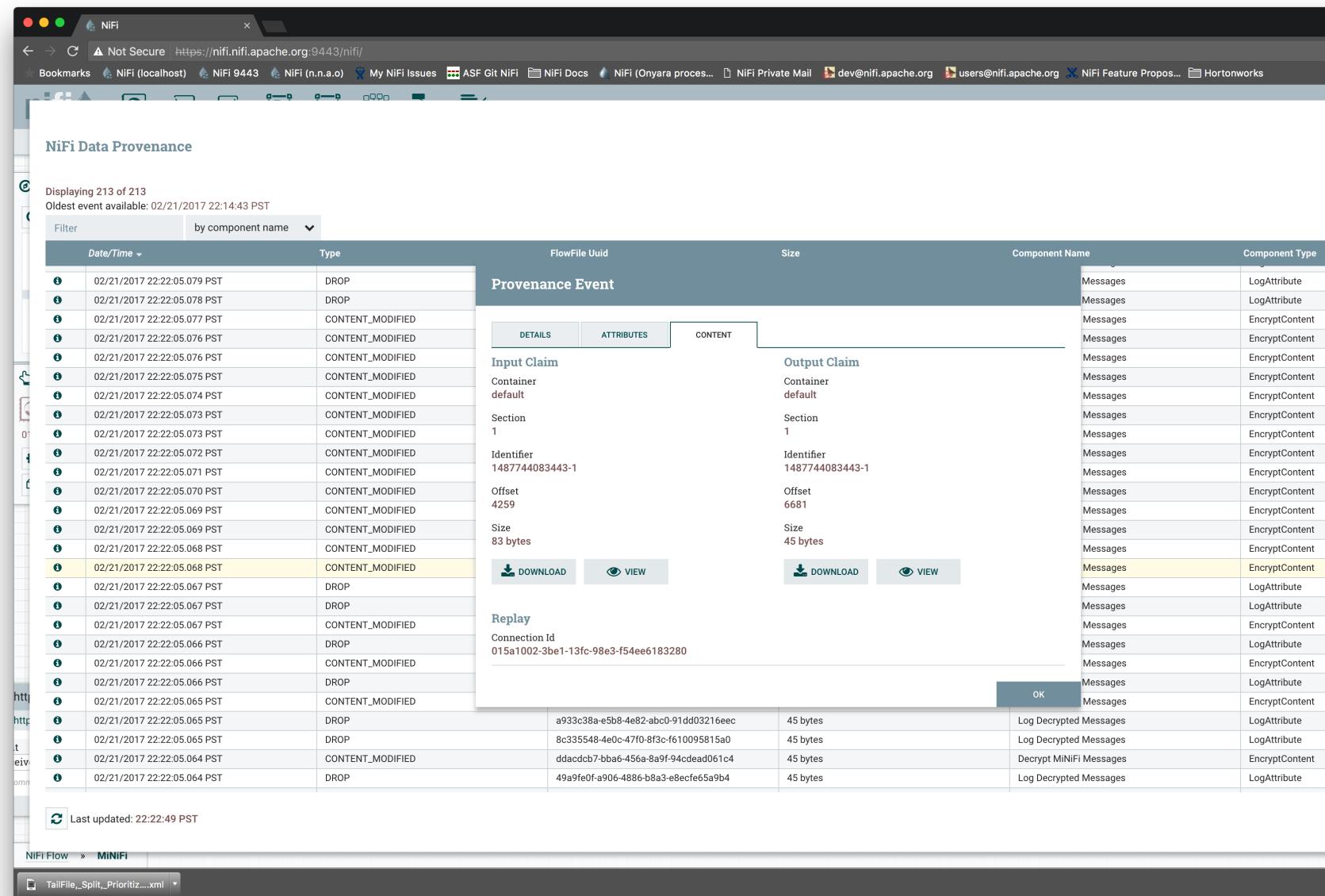
Assembly Code (View as: hex):

0x00000000 CB 54 29 E4 BA 64 3B 1B 55 85 84 70 22 9B 77 03 .T)...d;.U..p".w.
0x00000010 4E 69 46 69 49 56 2C 19 3C B1 DA AF 45 65 F9 A9 NiFiIV,.<...Ee..
0x00000020 E8 D6 F7 B1 53 79 A1 11 81 83 96 FA E6 48 20 B5Sy.....H ..
0x00000030 CD AE 71 66 03 C7 20 11 C0 DD 65 9E 4A 54 59 4B ..qf... ...e.JTYK
0x00000040 4E 6A CD D7 5D 85 7D 68 C6 AF EF 70 F8 23 E0 D6 Nj...].}h...p.#... =..
0x00000050 3D 95 8E

A large yellow arrow points downwards from the assembly code section towards the bottom of the screen.

Bottom Status Bar:

1 This is a message at 2017/02/21 22:16:27.986Z



Encrypted Provenance Repository

- Every provenance event record is encrypted with AES G/CM before being persisted to disk
 - Decrypted on deserialization for retrieval/query
 - Random access via offset seek
 - Handles key migration & rotation

NiFi Data Provenance						
Displaying 3 of 3			Showing the events that match the specified query. Clear search			
Filter	by component name					
Date/Time ▾	Type	FlowFileUuid	Size	Component Name	Component Type	
 06/05/2017 20:17:4...	CONTENT_MODIFIED	d602bdf...d14-4c2e...	77 bytes	ConvertRecord	ConvertRecord	
 06/05/2017 20:17:4...	ROUTE	d602bdf...d14-4c2e...	46 bytes	LookupRecord	LookupRecord	
 06/05/2017 20:17:4...	FORK	f540f7cf-1e41-4cb7...	40 bytes	LookupRecord	LookupRecord	

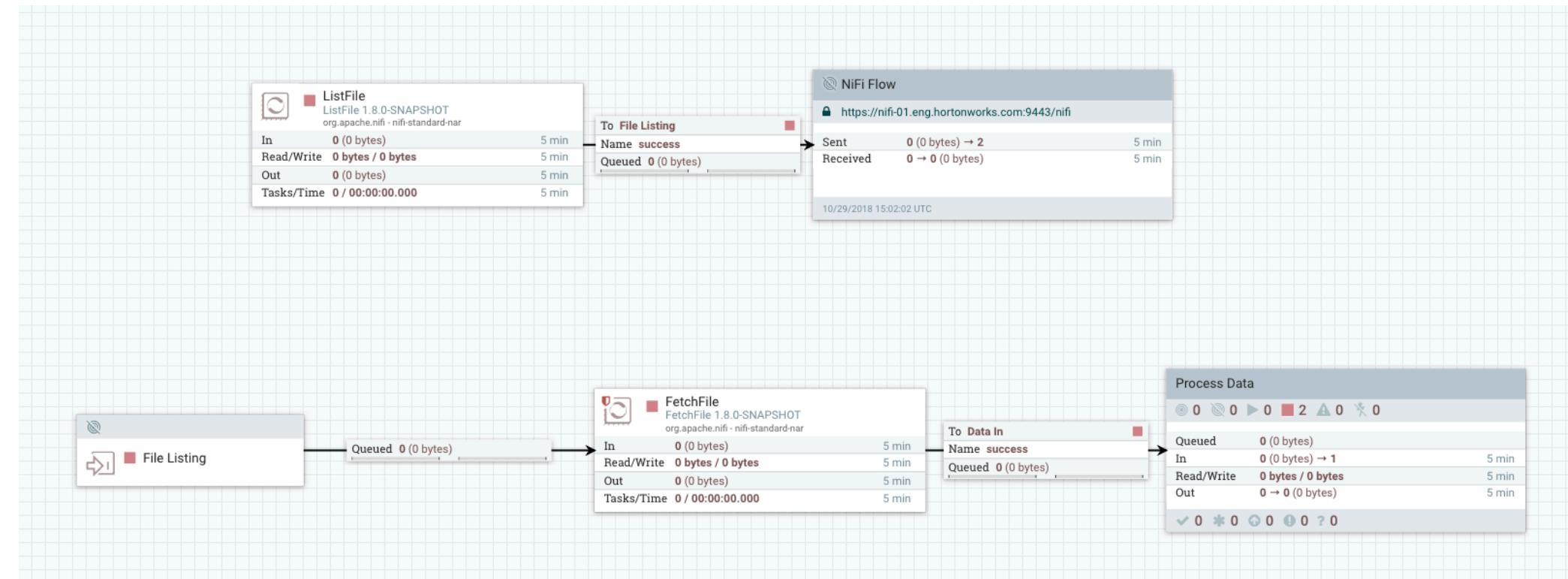
New Features

New Features in 1.8.0

- Cluster Data Management
 - Load-Balanced Connections
 - Node Decommissioning
- SQL results to record format
- Elasticsearch lookup service
- Docker improvements
- TLS Toolkit signing w/ external CA (standalone only)

Previously, on *NiFi Clusters*...

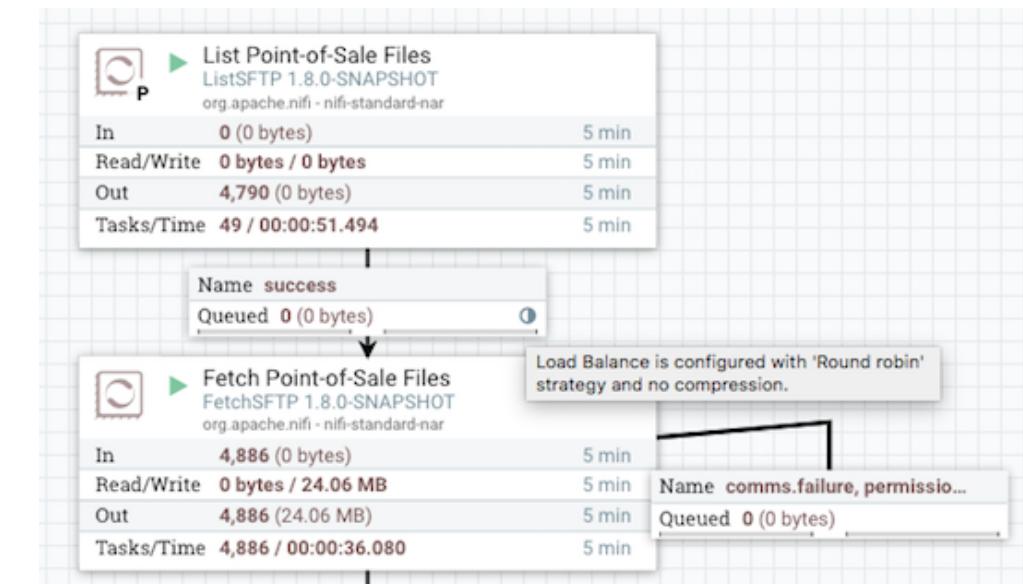
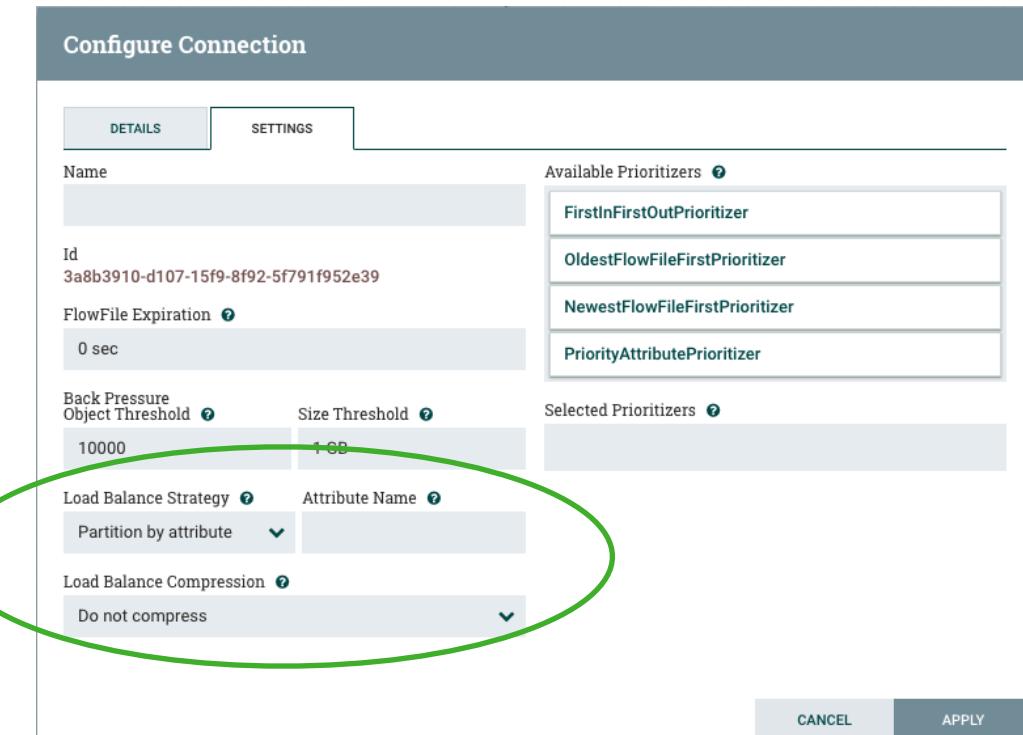
- Ingest activity (FTP, file read, etc.) performed on single node
- Routed via RPG to Input Port in cluster
- Flowfiles distributed across all nodes



[Mark Payne, Load Balancing Data Across a NiFi Cluster](#)

Load-Balanced Connections

- Individual connections can be load-balanced
 - None
 - Round Robin
 - Attribute node affinity
 - Single node



[Mark Payne, Load Balancing Data Across a NiFi Cluster](#)

Roadmap

Roadmap

- Encrypted content and flowfile repositories
- Encryption component enhancements
- Cryptographic proof on sequences
- Sensitive properties & variables
- Sensitive Attributes
- TLS Improvements
- Encrypted Logs

Encrypted Content & Flowfile Repositories

- Similar to encrypted provenance repository
- Data stored on disk encrypted with referenceable keys & metadata
- Still allow random access (offset seek vs. linear read)
- Use framework-level **KeyProvider** interface
- HDF/NiFi ready for deployment on untrusted hardware w/o writing sensitive data* to disk
 - TLS certs and encryption keys still persisted

Encryption Component Improvements

- **EncryptContent**
 - Improvements to key management
 - Enhanced metadata in attributes
- **EncryptAttribute**
- **EncryptRecord**
- Use framework-level **KeyProvider** controller service

Cryptographic Proof on Sequences

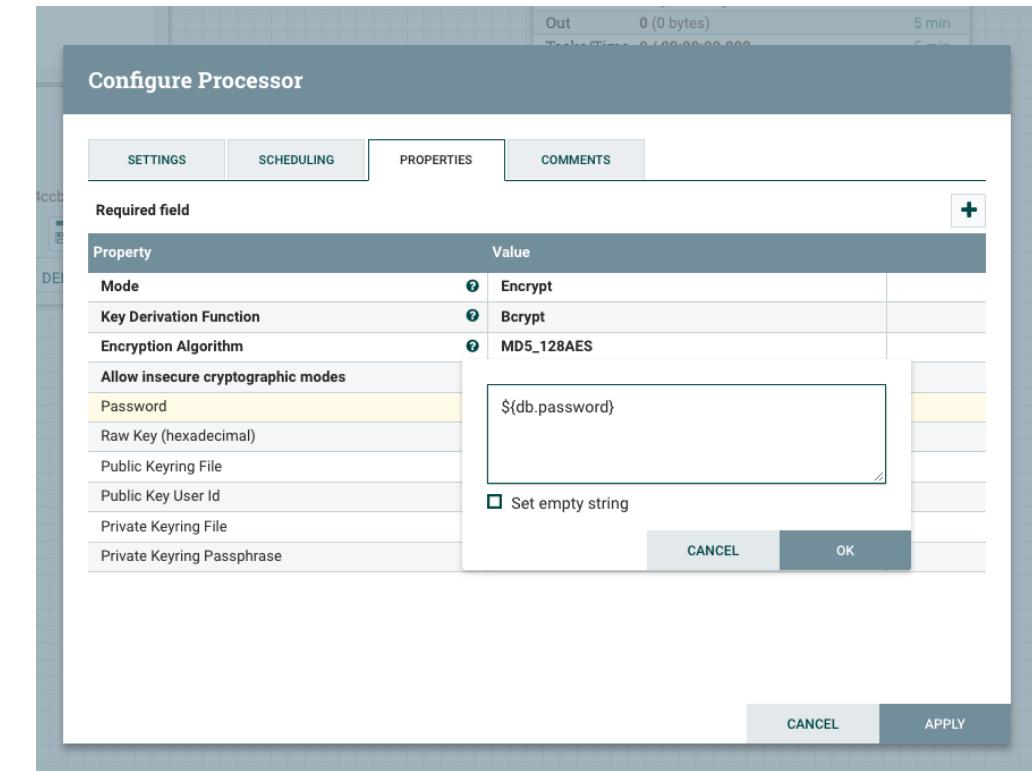
- Tired of hearing “blockchain” - signatures vs. distributed ledger
- Providing cascading cryptographic signatures that cover the previous event allows for provable provenance/chain of custody
- Can be applied to:
 - Provenance
 - Flow versioning / `flow.xml.gz` changes
 - Replicated requests in cluster operations
 - User/group authorization/authentication changes

Sensitive Properties & Variables

- Users want to promote flows between environments
- NiFi Registry enables this, but templates and versioned flows do not export sensitive properties
- Not all properties (and no passwords) support Expression Language
 - If `def = 123`, is `abc${def}` a 9 char password, or “`abc`” + `value(def)` = `abc123` ?
- Should enable:
 - Secure storage of sensitive variables on disk (i.e. encryption)
 - Secure export of sensitive variables to shared storage (i.e. encryption via Registry)
 - Key management (i.e. localized, derived, unique keys & encryption translation)
 - Access control for sensitive variables
 - Enable on-demand sensitive protection for dynamic properties

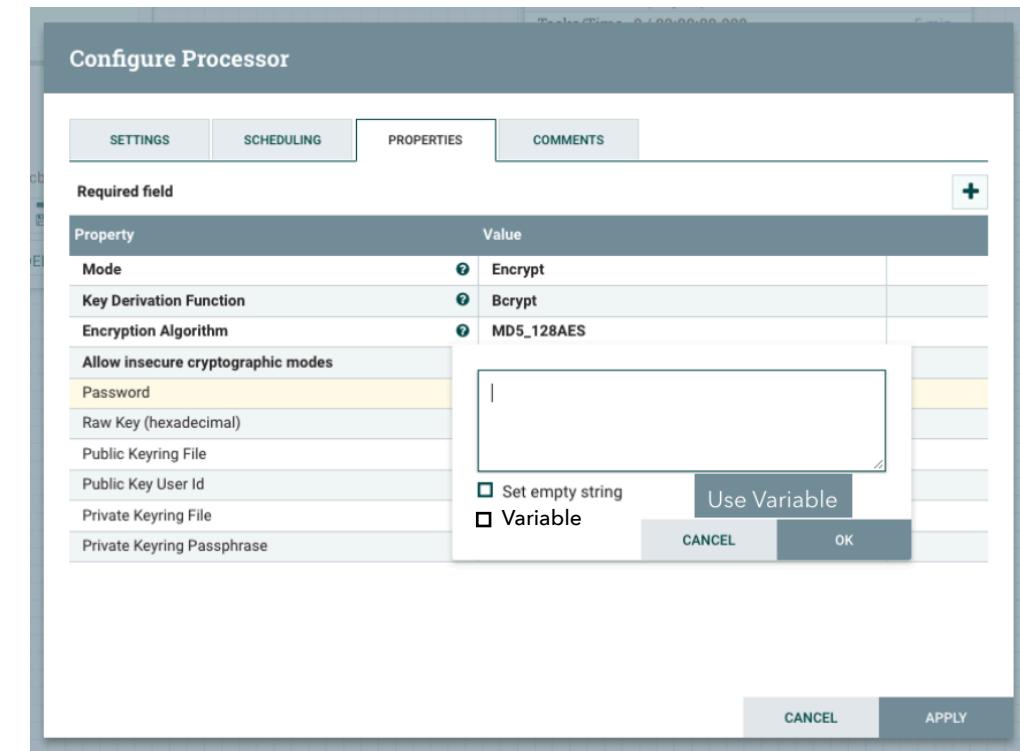
Current Situation

- No way to determine EL evaluation
- No way to protect dynamic properties



Proposed Changes

- Add “Variable” checkbox
 - Checked by default if property descriptor supports EL evaluation
 - If user un-checks, PD does not interpret EL (used for literal values containing `$\{...\}`)
- Add “Use Variable” button (auto-complete)



Use Variable

- Opens text field
- Allows free typing

Enter Variable Name...

Use Variable

- Typing triggers AJAX request to return variables
- New API endpoint
 - `GET /process-groups/abcd.../variables?q=d`
 - Text query filter
 - Permissions applied on server
 - Scope applied on server

d

data.endpoint

db.password

db.url

db.username

Variable Export

- Options
 - Never allow
 - Allow based on Registry admin setting (per-bucket?)
- Key management policies
 - Export in plaintext (**not secure**)
 - Export encrypted with original key (can't be deployed to other NiFi instances/rolled-over instances)
 - Export encrypted with per-Registry key (maintains independence of NiFi instances; preferred)
 - Import operation would then re-encrypt with destination NiFi's **nifi.sensitive.props.key**
 - Also possible to export with derived key to isolate buckets

Sensitive Attributes

- Some attributes contain sensitive data
- While components currently have access control policies, flowfiles/attributes do not
- Restricting access to sensitive attributes (PII, confidential, etc.) allows DFM/domain specialists to build and monitor flows without viewing data
- Prevents examination via provenance viewing as well

TLS Improvements

- Handle keystores with multiple certificates to allow for varied identification by role
- Automatic Protocol/Cipher Suite Upgrades
 - Custom list of cipher suites and protocol versions loaded into Jetty at startup (work done; PR to be opened)
 - Integration with Mozilla TLS Labs tools
 - Plug n' Play Simplicity - “Set it and forget it”
 - Automatically sets and continuously analyzes supported cipher suites to ensure compatibility & risk minimization
- Extend TLS Toolkit
 - Better handle pre-existing certificates/certificate authorities
 - Help enterprise users securely deploy NiFi with less friction
 - Provide “sanity check” feature to test connectivity with provided keystore/truststore combos & identify potential issues
 - Refactor underway

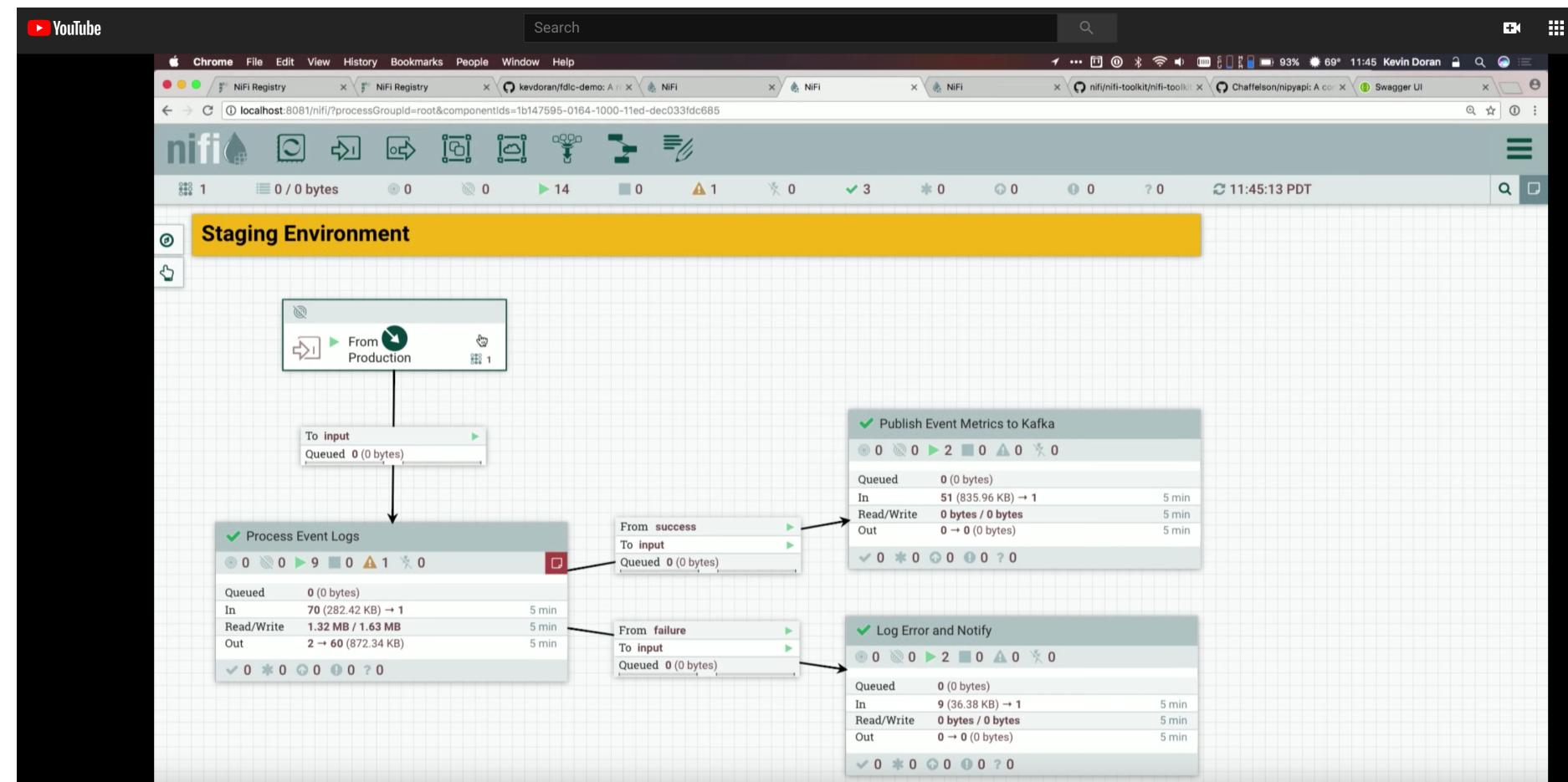
Encrypted Logs

- NiFi Provides Alternatives to Log Grepping
 - Provenance Data
 - Processor Stats
- Pluggable Log Interceptors
 - Accept messages, key/value pairs
 - Sensitive keys are registered in a lookup
 - Values for these keys are encrypted (using log-specific key) before output
- Log File Encryption
 - More difficult to debug, but more complete protection
 - Restrict user/client access logs, logs of encryption/lookup operations, etc.
 - Batching/latency trade-off
- Alternative method is hashing
 - Searching logs only requires executing hash function on query to generate “real” search term

Community

More Resources

- [FDLC with Apache NiFi](#), Kevin Doran
- [NiPyAPI Docs](#), Daniel Chaffelson
- [DevOps Tips](#), Tim Spann
- [Automate Workflow](#), Pierre Villard



New Announcements

- NiFi 1.8.0 — 26 Oct 2018 (212+ Jiras)
 - Jetty, DB improvements
 - Auto load-balancing queues
 - TLS Toolkit w/ external CA
 - Record processor improvements
- MiNiFi C++ 0.5.0 — 6 June 2018
- MiNiFi Java 0.5.0 — 7 July 2018
- NiFi Registry 0.3.0 — 25 Sept 2018



Community Health

apache / nifi

Unwatch ▾ 154 Unstar 1,103 Fork 1,041

Code Pull requests 145 Projects 0 Insights

Mirror of Apache NiFi

4,720 commits

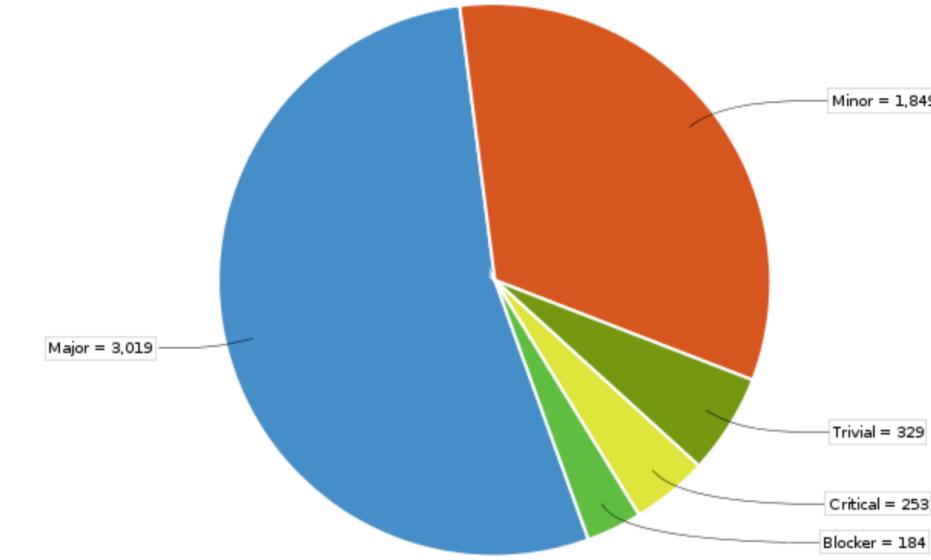
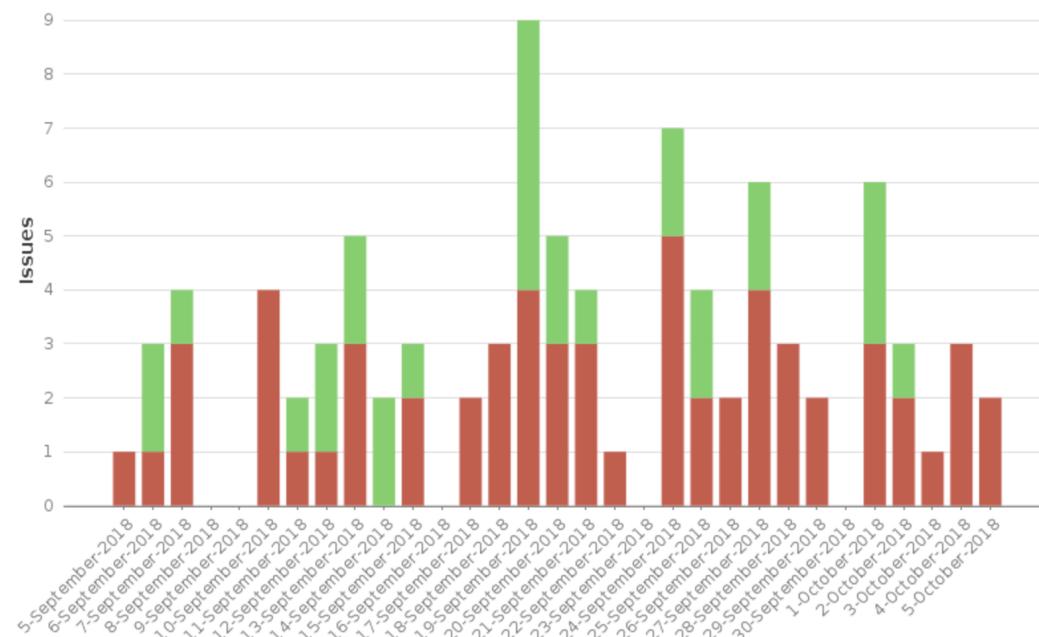
52 branches

63 releases

200 contributors

Apache-2.0

This chart shows the issues created in the last 30 days



Learn more and join us

Apache NiFi site

<https://nifi.apache.org>

Subproject MiNiFi site

<https://nifi.apache.org/minifi/>

Subscribe to and collaborate at

dev@nifi.apache.org

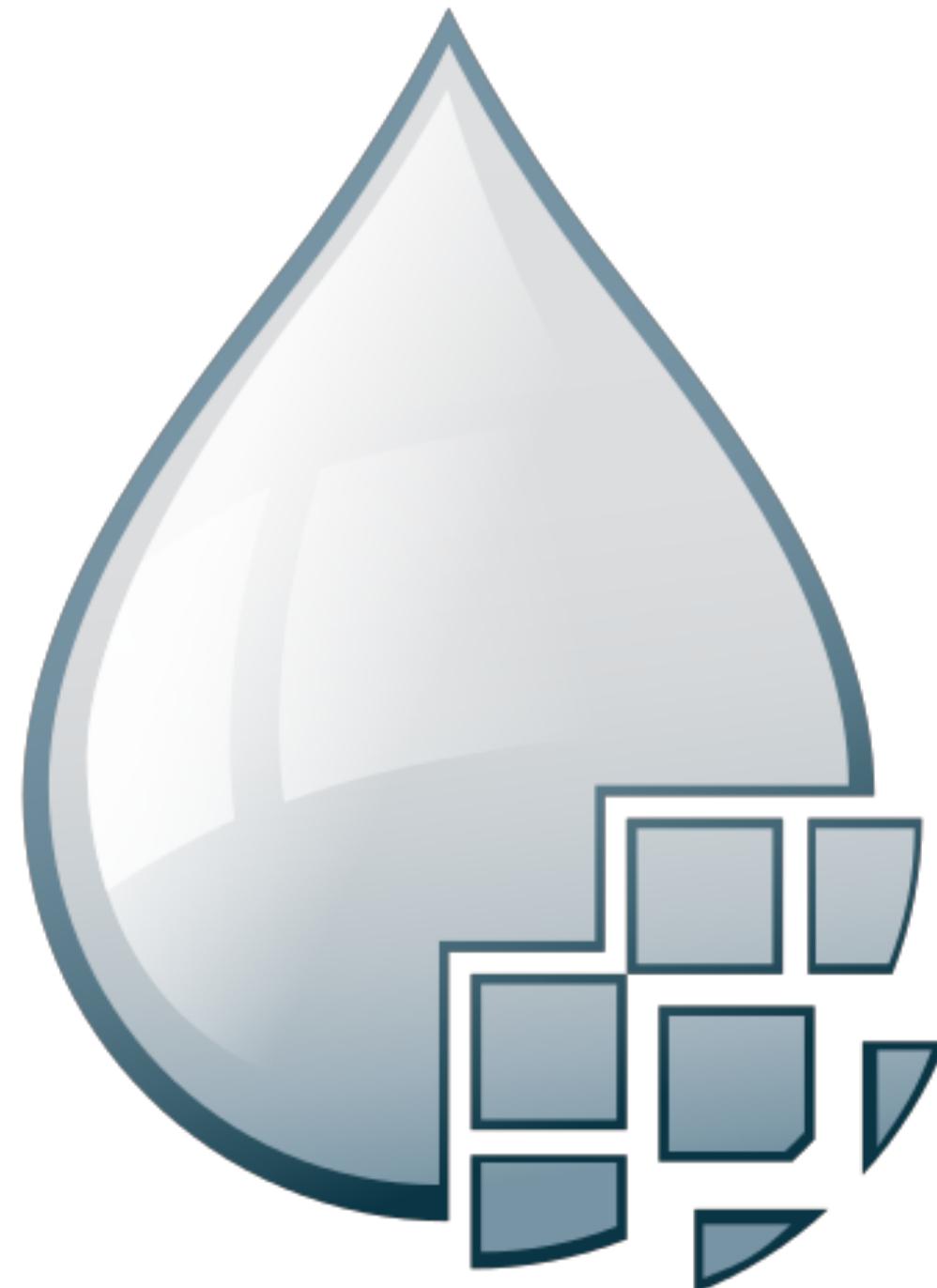
users@nifi.apache.org

Submit Ideas or Issues

<https://issues.apache.org/jira/browse/NIFI>

Follow us on Twitter

[@apachenifi](https://twitter.com/apachenifi)





Thank you

alopresto@hortonworks.com | alopresto@apache.org | [@yolopey](https://twitter.com/yolopey)
github.com/alopresto/slides