



*The SWIP Project is supported by the National Science Foundation under Grant 1642070, 1642053, and 1642090.*



# ***Integrity Protection for Scientific Workflow Data: Motivation and Initial Experiences***

**Mats Rynge, Karan Vahi, Ewa Deelman**

**Information Sciences Institute - University of Southern California**

**Anirban Mandal, Ilya Baldin**

**RENCI - University of North Carolina, Chapel Hill**

**Omkar Bhide, Randy Heiland, Von Welch, Raquel Hill**

**Indiana University**

**William L. Poehlman, F. Alex Feltus**

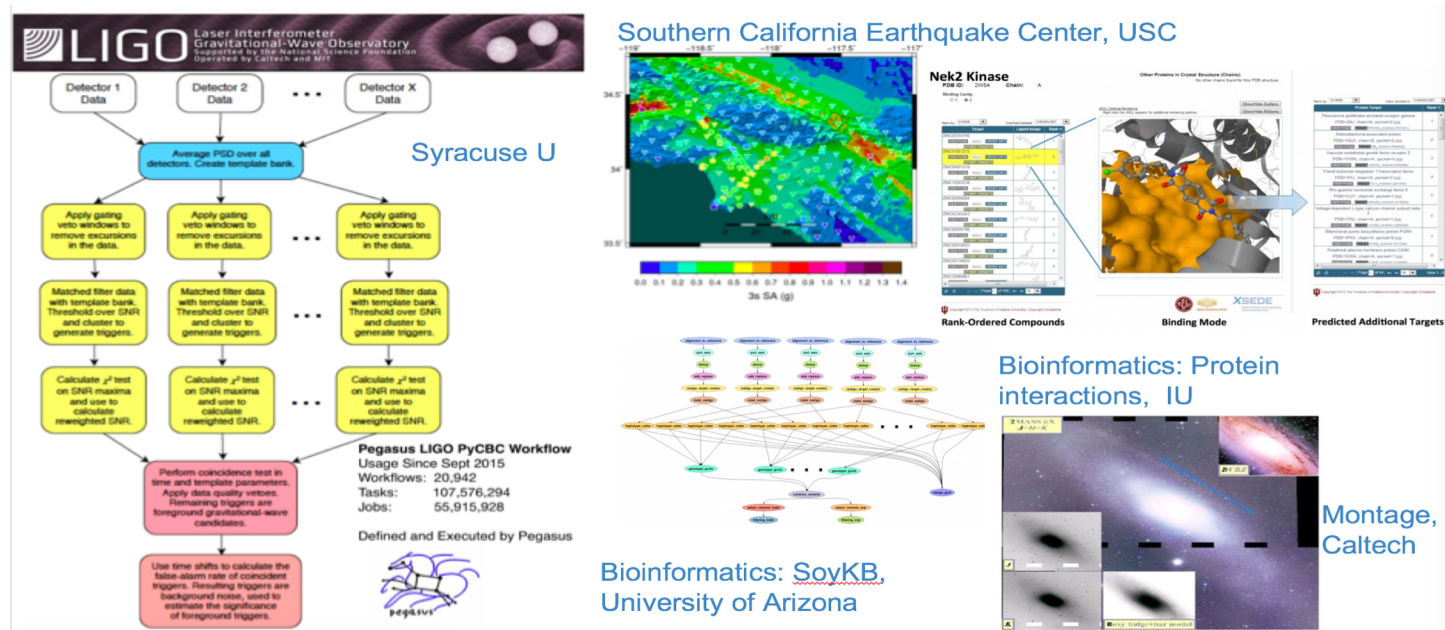
**Clemson University**



# Pegasus Workflow Management System, Production Use

Last 12 months: Pegasus users ran **240K** workflows, **145M** jobs

Majority of these include data transfers, using LAN, the Internet, local and remote storage



<https://pegasus.isi.edu/>



## Goals:

Provide additional assurances that a scientific workflow is not accidentally or maliciously tampered with during its execution.

Allow for detection of modification to its data or executables at later dates to facilitate reproducibility.

Integrate cryptographic support for data integrity into the Pegasus Workflow Management System.



PIs: Von Welch, Ilya Baldin, Ewa Deelman, Raquel Hill

Team: Omkar Bhide, Rafael Ferrieira da Silva, Randy Heiland, Anirban Mandal, Rajiv Mayani, Mats Rynge, Karan Vahi



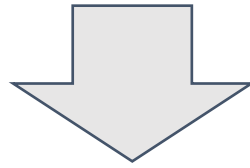
# Our Talk

- **Introduction and Motivations**
- Our Approach
- Current Status
- Welcome to the Jungle
- Integrity Issues in the Wild
- Future Work



## Data Integrity

010110101010



010100101010

# Challenges to Scientific Data Integrity

Modern IT systems are not perfect - errors creep in.

At modern “Big Data” sizes we are starting to see checksums breaking down.

Plus there is the threat of intentional changes: malicious attackers, insider threats, etc.

User Perception: “Am I not already protected? I have heard about TCP checksums, encrypted transfers, checksum validation, RAID and erasure coding – is that not enough?”

# Motivation:

## CERN/NEC Studies of Disk Errors

Examined Disk, Memory, RAID 5 errors.

“The error rates are at the  $10^{-7}$  level, but with complicated patterns.” E.g. 80% of disk errors were 64k regions of corruption.

Explored many fixes and their often significant performance trade-offs.

A similar study by NEC found that 1 in 90 SATA drives will experience silent data corruption.

## Data integrity

Bernd Panzer-Steindel, CERN/IT  
Draft 1.3 8. April 2007

### Executive Summary

We have established that low level data corruptions exist and that they have several origins. The error rates are at the  $10^{-7}$  level, but with complicated patterns. To cope with the problem one has to implement a variety of measures on the IT part and also on the experiment side. Checksum mechanisms have to be implemented and deployed everywhere. This will lead to additional operational work and the need for more hardware.

### Introduction

During January and February 2007 we have done a systematic analysis of data corruption cases in the CERN computer center. The major work in the implementation of probes and automatic running schemes were done by Tim Bell, Olof Barring and Peter Kelemen from the IT/FIO group. There have been similar problems reported in Fermilab and Desy and information exchange with them was done.

The following paper will provide results from this analysis, a judgment of the situation and a catalogue of measures needed to get the problem under control.

It is also to be seen as a starting point for further discussions with IT, the experiments and the T1 sites.

[https://indico.cern.ch/event/13797/contributions/1362288/attachments/115080/163419/Data\\_integrity\\_v3.pdf](https://indico.cern.ch/event/13797/contributions/1362288/attachments/115080/163419/Data_integrity_v3.pdf)  
<https://www.necam.com/docs/?id=54157ff5-5de8-4966-a99d-341cf2cb27d3>

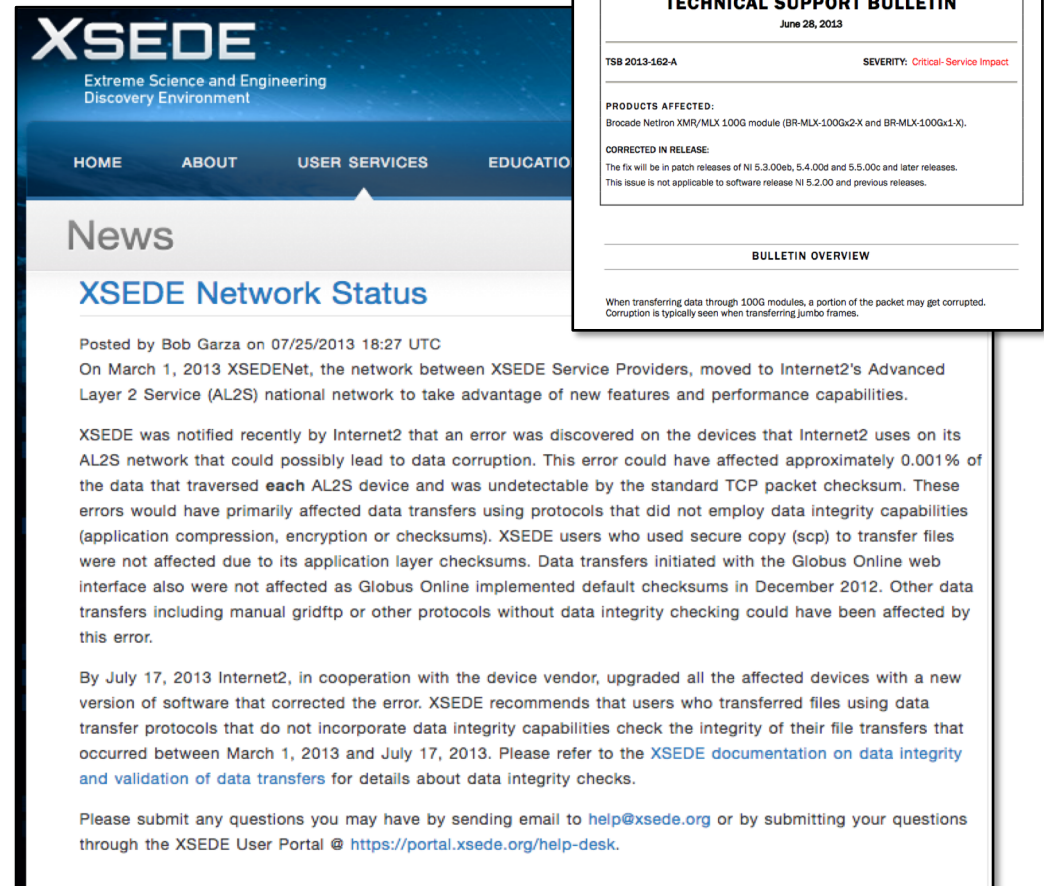
# Motivation: Network Corruption

Network router software  
inadvertently corrupts TCP **data**  
**and/or checksum!**

XSEDE and Internet2 example  
from 2013.

Second similar case in 2017:  
University of Chicago network  
upgrade caused data corruption  
for the FreeSurfer/Fsurf project.

Brocade TSB 2013-162-A



The image shows a screenshot of the XSEDE website's 'News' section and a Brocade Technical Support Bulletin (TSB) for 2013-162-A. The XSEDE website header includes the logo and navigation links: HOME, ABOUT, USER SERVICES, and EDUCATION. The news article, titled 'XSEDE Network Status', is dated 07/25/2013 18:27 UTC and is posted by Bob Garza. It discusses a network migration to Internet2's Advanced Layer 2 Service (AL2S) and a data corruption issue discovered on the devices. The article mentions that the error could affect approximately 0.001% of data transfers and that the fix was implemented by July 17, 2013. The Brocade TSB, titled 'TECHNICAL SUPPORT BULLETIN' for June 28, 2013, details the issue with Brocade Netiron XMR/MLX 100G modules, stating that a portion of the packet may get corrupted when transferring data through 100G modules. It also lists the products affected and the corrected release versions.

**XSEDE**  
Extreme Science and Engineering  
Discovery Environment

HOME ABOUT USER SERVICES EDUCATION

## News

### XSEDE Network Status

Posted by Bob Garza on 07/25/2013 18:27 UTC

On March 1, 2013 XSEDENet, the network between XSEDE Service Providers, moved to Internet2's Advanced Layer 2 Service (AL2S) national network to take advantage of new features and performance capabilities.

XSEDE was notified recently by Internet2 that an error was discovered on the devices that Internet2 uses on its AL2S network that could possibly lead to data corruption. This error could have affected approximately 0.001% of the data that traversed **each** AL2S device and was undetectable by the standard TCP packet checksum. These errors would have primarily affected data transfers using protocols that did not employ data integrity capabilities (application compression, encryption or checksums). XSEDE users who used secure copy (scp) to transfer files were not affected due to its application layer checksums. Data transfers initiated with the Globus Online web interface also were not affected as Globus Online implemented default checksums in December 2012. Other data transfers including manual gridftp or other protocols without data integrity checking could have been affected by this error.

By July 17, 2013 Internet2, in cooperation with the device vendor, upgraded all the affected devices with a new version of software that corrected the error. XSEDE recommends that users who transferred files using data transfer protocols that do not incorporate data integrity capabilities check the integrity of their file transfers that occurred between March 1, 2013 and July 17, 2013. Please refer to the [XSEDE documentation on data integrity and validation of data transfers](#) for details about data integrity checks.

Please submit any questions you may have by sending email to [help@xsede.org](mailto:help@xsede.org) or by submitting your questions through the XSEDE User Portal @ <https://portal.xsede.org/help-desk>.

**BROCADE**

### TECHNICAL SUPPORT BULLETIN

June 28, 2013

TSB 2013-162-A SEVERITY: **Critical-Service Impact**

**PRODUCTS AFFECTED:**  
Brocade Netiron XMR/MLX 100G module (BR-MLX-100Gx2-K and BR-MLX-100Gx1-X).

**CORRECTED IN RELEASE:**  
The fix will be in patch releases of NI 5.3.00eb, 5.4.00d and 5.5.00c and later releases. This issue is not applicable to software release NI 5.2.00 and previous releases.

#### BULLETIN OVERVIEW

When transferring data through 100G modules, a portion of the packet may get corrupted. Corruption is typically seen when transferring Jumbo frames.

<https://www.xsede.org/news/-/news/item/6390>



## Motivation:

### Software failures

Bug in StashCache data transfer software would occasionally cause silent failure (failed but returned zero).

Failures in the final staging out of data were not detected.

The workflow management system, believing workflow was complete, cleaned up. With the final data being incomplete and all intermediary data lost, ten CPU-years of computing came to naught.

How is this an data integrity issue? The workflow system should have verified that the data at the storage system after the transfer, is the expected data.

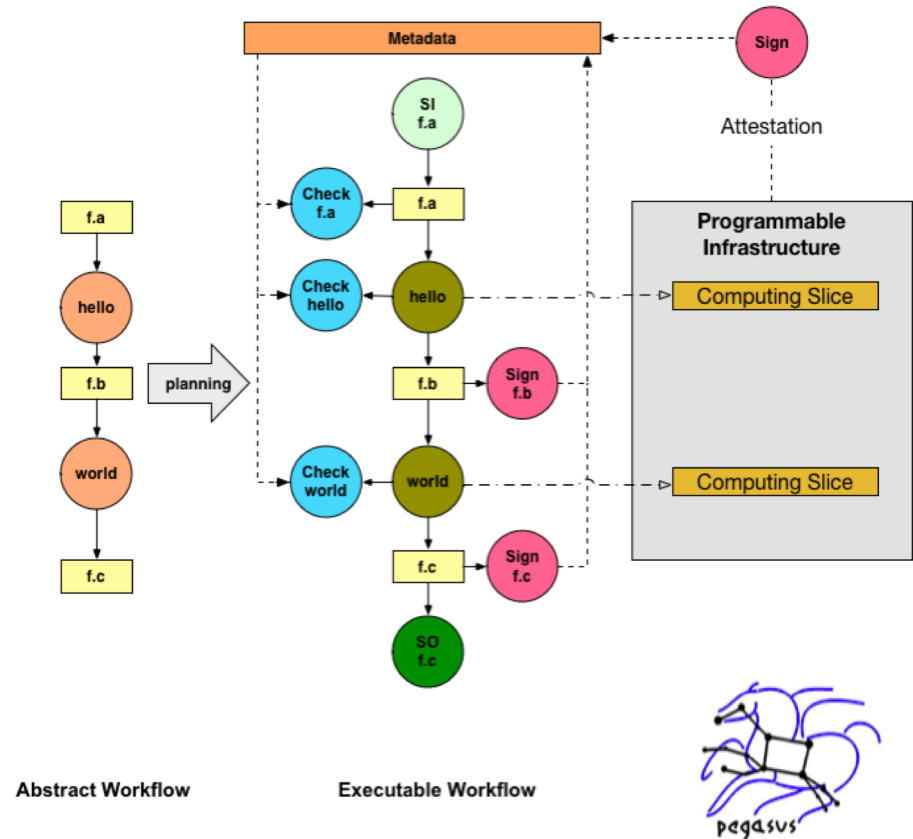
# Our Talk

- Introduction and Motivations
- **Our Approach**
- Current Status
- Welcome to the Jungle
- Integrity Issues in the Wild
- Future Work



# Our High Level Plan...

- Workflow Management Systems (WMS) are great places to tackle data integrity.
- They understand what data is created and ingested and do not mind tedious tasks such as generating and checking checksums.
- Placement is important within the workflow of generate/validate checksums
- Pegasus WMS is widely used (LIGO, SCEC, SoyKB, Montage, etc.) by the scientific community and is the target of our improvements.



# Application-level Checksums – SHA256

- Application-level checksums (hashes) allow for detection of changes.
- Explored some more advanced solutions, but at the end simplicity won
- Checksums already in use by many data transfer applications: scp, Globus/GridFTP, some parts of HTCondor, etc, but SWIP is focusing on end-to-end as well as over longer time periods

e.g. using a SHA in Python:

```
>>> hashlib.sha256(b"The Answer to the Ultimate Question of Life, the Universe, and Everything is 42").hexdigest()  
'8a72856cf94464dd641f0a2620ab604dd7a3f50293784a3a399acf6dc5b651cb'
```

```
>>> hashlib.sha256(b"The Answer To the Ultimate Question of Life, the Universe, and Everything is 42").hexdigest()  
'a39be9fd272f2569aa95a07134a55f032ecb5c51cef6d66fe4032ec30bf4f1b6'
```

```
>>> hashlib.sha256(b"The Answer is 42").hexdigest()  
'cbf296e175f02156cd60d6bf93aebd92893e72a0c4c48eade092d0dc7e28fc1'
```

# Our Talk

- Introduction and Motivations
- Our Approach
- **Current Status**
- Welcome to the Jungle
- Integrity Issues in the Wild
- Future Work





## Pegasus 4.9.0 Released

on OCTOBER 31, 2018

We are pleased to announce release of Pegasus 4.9.0 Pegasus 4.9.0 is be a major release of Pegasus. Highlights of new features: Integrity Checking – Pegasus now performs integrity checking on files in a workflow for non shared filesystem deployments. More details can be found in the documentation at [https://pegasus.isi.edu/documentation/integrity\\_checking.php](https://pegasus.isi.edu/documentation/integrity_checking.php) ... [Read More](#)

Integrity validation is on by default since the Pegasus 4.9.0 release (Oct 31<sup>st</sup>, 2018). Users who upgrade will automatically get the protection, but can opt out.

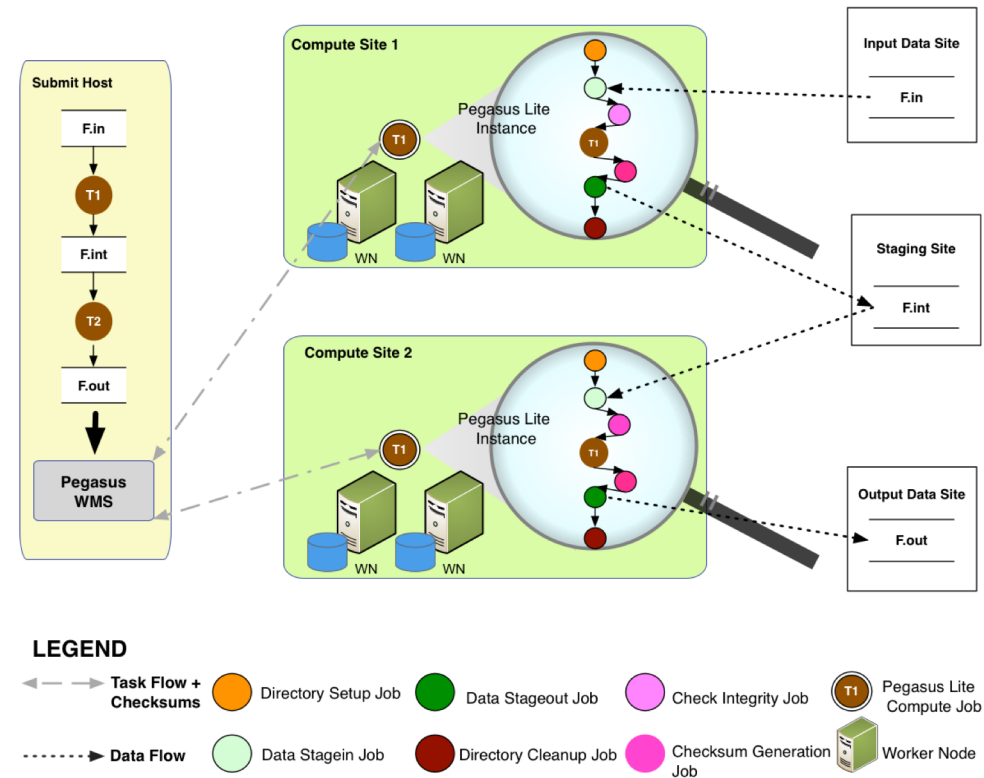
Sharing of detailed monitoring data with the Pegasus team is off by default. Users can opt-in. (We will come back to this at the end of the talk)



# Automatic Integrity Checking in Pegasus

Pegasus performs integrity checksums on input files right before a job starts on the remote node.

- For raw inputs, checksums specified in the input replica catalog along with file locations
- All intermediate and output files checksums are generated and tracked within the system.
- Support for sha256 checksums



**Job failure** is triggered if checksums fail

# Our Talk

- Introduction and Motivations
- Our Approach
- Current Status
- **Welcome to the Jungle**
- Integrity Issues in the Wild
- Future Work



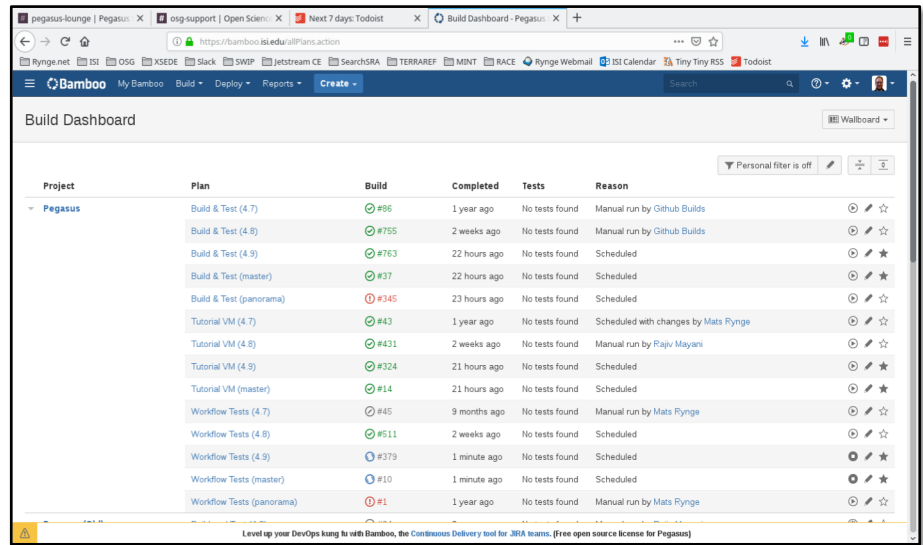
# How do you know your integrity protection is working?

- Imagine the following:  
You finish adding integrity protection to your software. You run a workflow and all goes smoothly.
- Was there no integrity problem or did you just fail to detect it?
- How do you reliably and repeatedly test integrity protection?



# Confidence in the implementation: Bamboo

- **At commit, for each target platform:**
  1. Build binary, workers, RPMs, DEBs, ....
  2. Run unit tests for Java, Python, and C components
  3. ~ 100 unit tests
- **Nightly:**
  1. Run functional tests. These are full workflows, configured to provide good code coverage
  2. ~ 85 workflows



The screenshot shows the Bamboo Build Dashboard for the 'Pegasus' project. The dashboard displays a table of builds with columns for Project, Plan, Build, Completed, Tests, and Reason. The builds are listed in descending order of completion time.

Project	Plan	Build	Completed	Tests	Reason
Pegasus	Build & Test (4.7)	#86	1 year ago	No tests found	Manual run by Github Builds
Pegasus	Build & Test (4.8)	#755	2 weeks ago	No tests found	Manual run by Github Builds
Pegasus	Build & Test (4.9)	#763	22 hours ago	No tests found	Scheduled
Pegasus	Build & Test (master)	#37	22 hours ago	No tests found	Scheduled
Pegasus	Build & Test (panorama)	#345	23 hours ago	No tests found	Scheduled
Pegasus	Tutorial VM (4.7)	#43	1 year ago	No tests found	Scheduled with changes by Mats Rynge
Pegasus	Tutorial VM (4.8)	#431	2 weeks ago	No tests found	Manual run by Rajiv Mayani
Pegasus	Tutorial VM (4.9)	#324	21 hours ago	No tests found	Scheduled
Pegasus	Tutorial VM (master)	#14	21 hours ago	No tests found	Scheduled
Pegasus	Workflow Tests (4.7)	#45	9 months ago	No tests found	Manual run by Mats Rynge
Pegasus	Workflow Tests (4.8)	#511	2 weeks ago	No tests found	Scheduled
Pegasus	Workflow Tests (4.9)	#379	1 minute ago	No tests found	Scheduled
Pegasus	Workflow Tests (master)	#10	1 minute ago	No tests found	Scheduled
Pegasus	Workflow Tests (panorama)	#1	1 year ago	No tests found	Manual run by Mats Rynge

# Enter the Chaos Jungle!

<https://github.com/RENCI-NRIG/chaos-jungle>

Inspired by Netflix's Chaos Monkey.

<https://github.com/Netflix/chaosmonkey>

Goal of Chaos Jungle (CJ) is to introduce different kinds of impairments into the virtual infrastructure - network, compute, storage.

The RENCi ORCA software creates virtual infrastructure on ExoGENi testbed. CJ software introduces impairments into data transfers.

We get virtual infrastructure that intentionally corrupts data

Randomly or predictably?

Now we can test how software runs under bad conditions.



[https://commons.wikimedia.org/wiki/File:Tioman\\_Rainforest.JPG](https://commons.wikimedia.org/wiki/File:Tioman_Rainforest.JPG)

# Chaos Jungle

Uses Linux eBPF (extended Berkeley Packet Filters) functionality

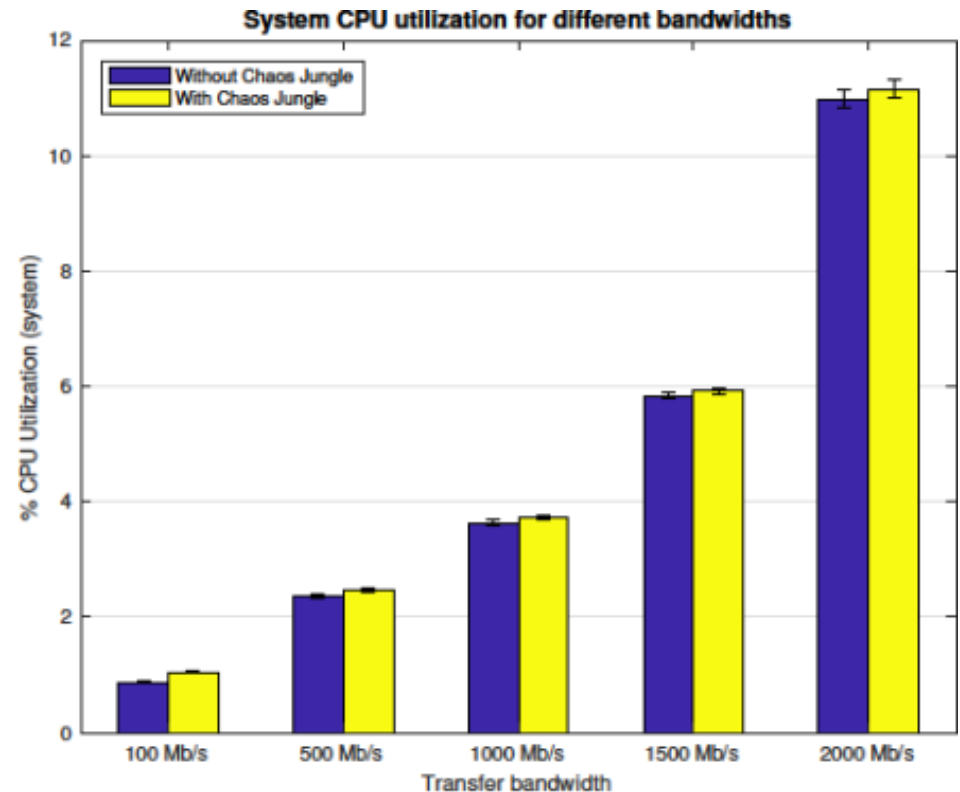
Introduces a small eBPF program into the kernel attaching to either TC filter or XDP hooks

Inspects received packets and modifies some of those that match flow descriptors without affecting the appropriate checksums.

The packets thus look valid on the receiving end, however contain invalid data.

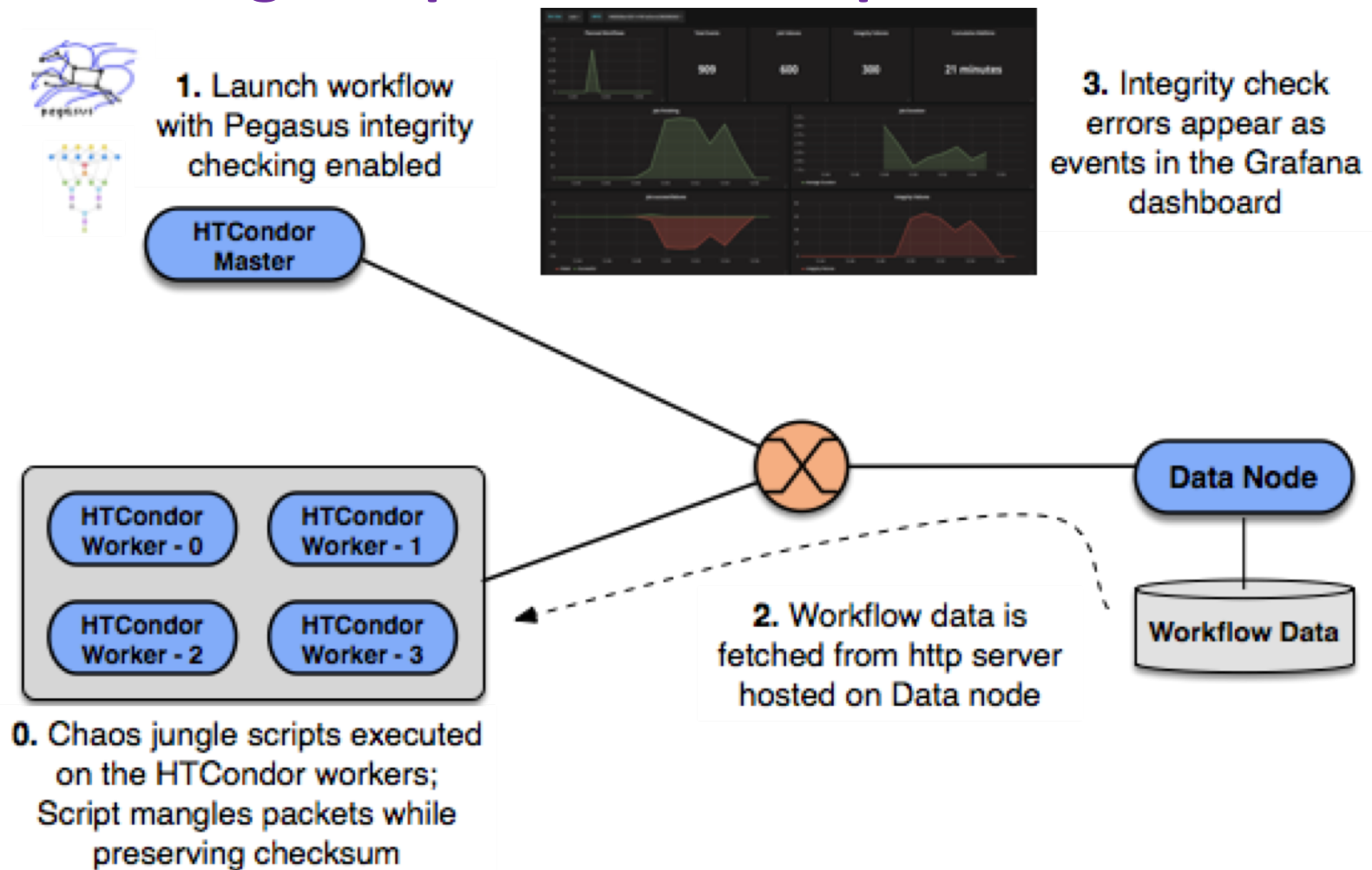
Fast and performant.

<https://github.com/RENCI-NRIG/chaos-jungle>



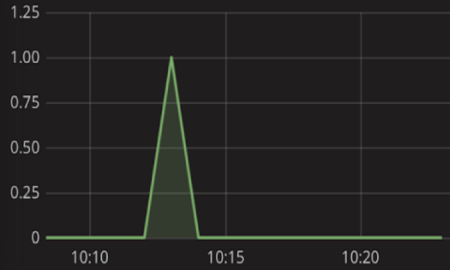


# Chaos Jungle Experiment Setup



Bin Size auto WFID bc19faa4-8ad6-4abf-b2d2-08e58290e347

Planned Workflows



Total Events

722

Job Failures

20

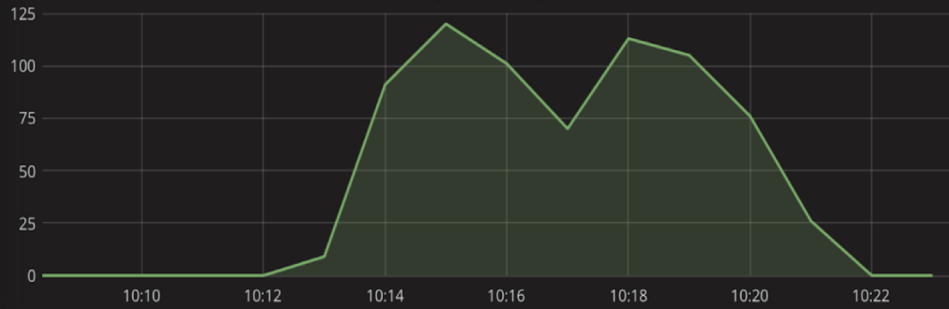
Integrity Failures

10

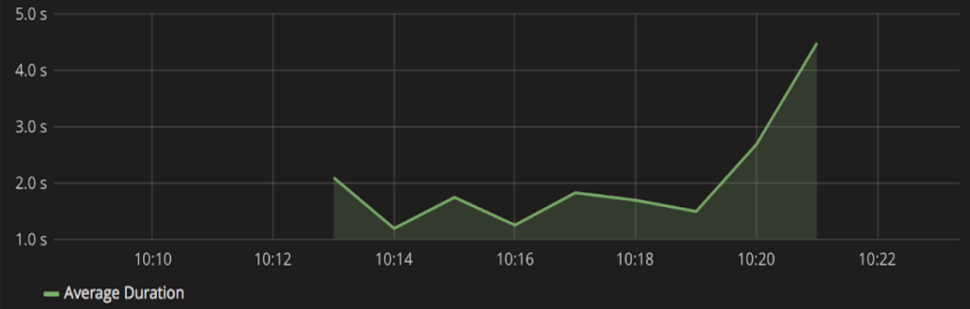
Cumulative Walltime

20 minutes

Job Finishing

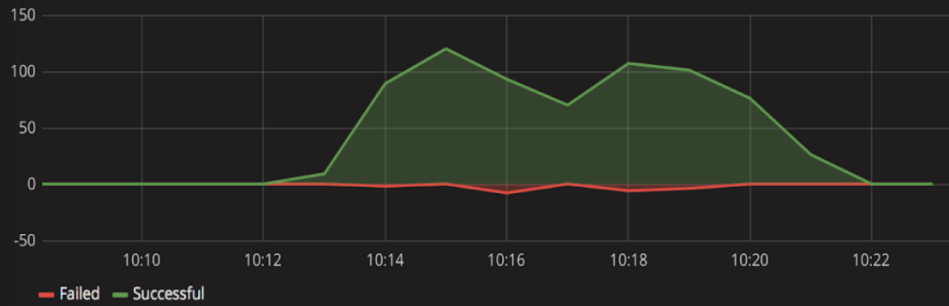


Job Duration



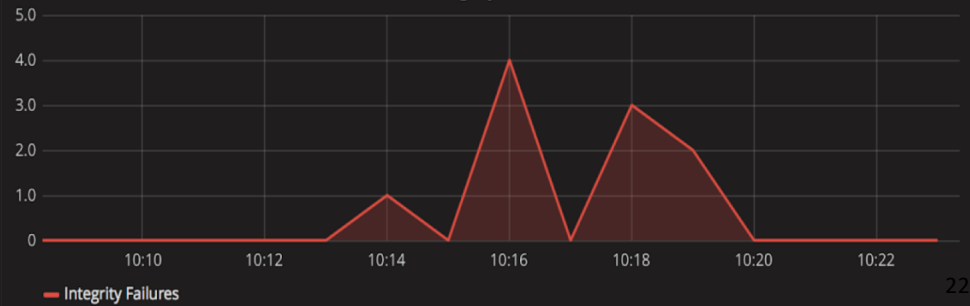
Average Duration

Job success/failures



Failed Successful

Integrity Failures



Integrity Failures

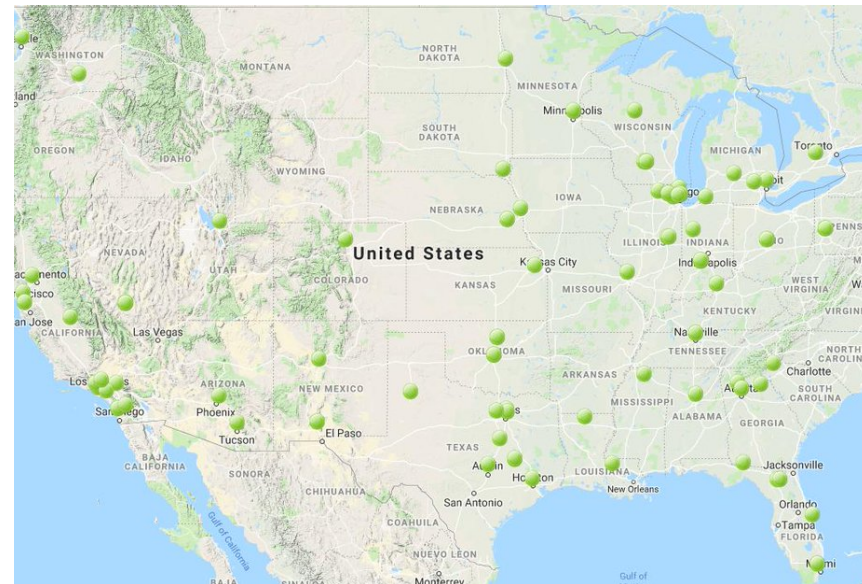
# Our Talk

- Introduction and Motivations
- Our Approach
- Current Status
- Welcome to the Jungle
- **Integrity Issues in the Wild**
- Future Work



# Production Workflows

- Large workflows with lots of data transfers
- “Unprotected” protocols - no SSL or other protocol level protections
- Open Science Grid - WAN transfers
- Collecting the data is on an opt-in basis



# Initial Results with Integrity Checking on

- OSG-KINC workflow (50,606 jobs) encountered **60 integrity errors** in the wild (production OSG). The problematic jobs were **automatically retried** and the workflow finished successfully.
- The 60 errors took place on 3 different hosts. The first one at UColorado, and group 2 and 3 at UNL hosts.
- Error Analysis (by hand)
  - 1 input file error at University of Colorado.
  - 3 input file (kinc executable) errors on one node at University of Nebraska. The timespan across the failures was 16 seconds. We suspect that the **node level cache got corrupted**.
  - 56 input file errors on a different compute nodes at University of Nebraska. The timespan across the failures was 1,752 seconds. We suspect that **the site level cache got corrupted**.

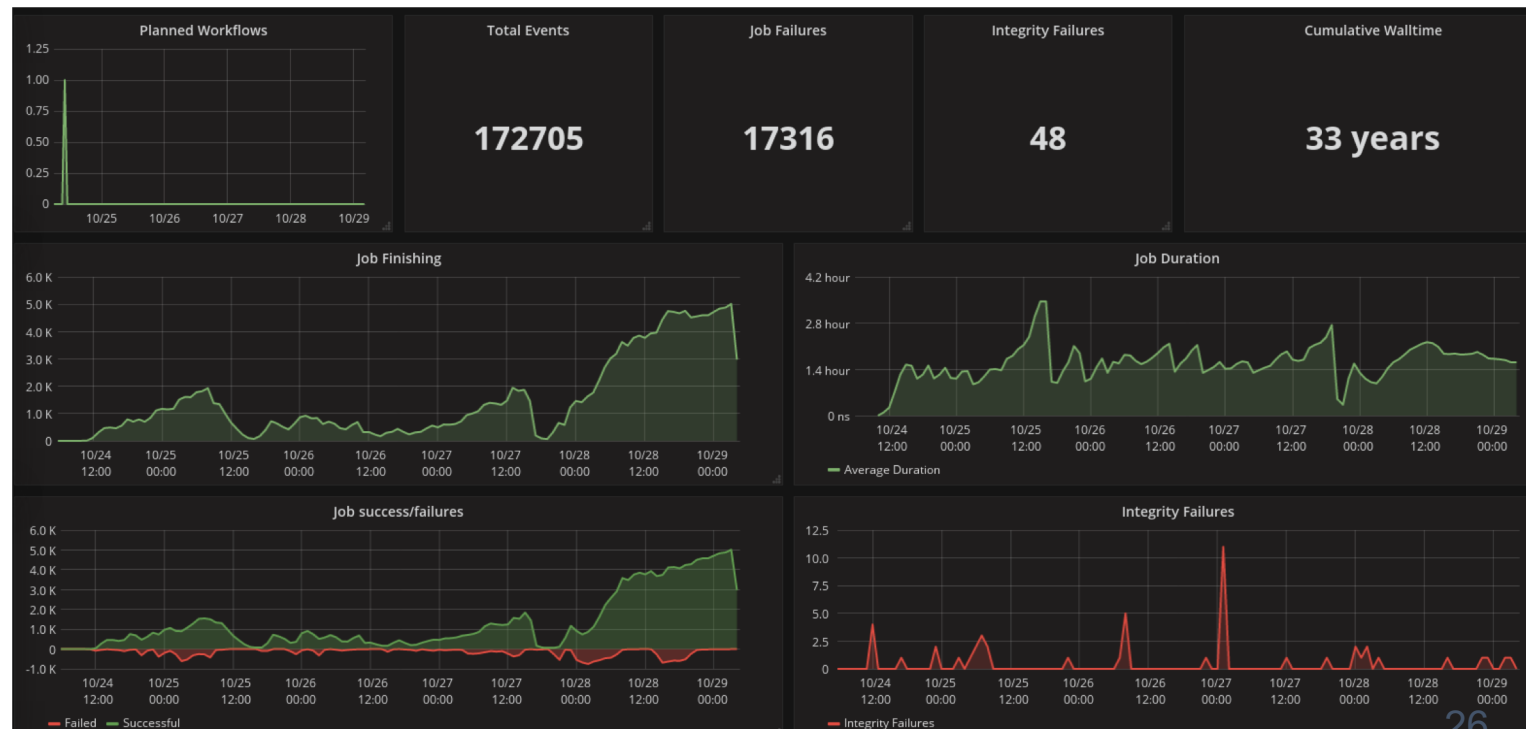
# Initial Results – VERITAS / Nepomuk Otte, GATech

Seeing very small, but steady stream of corrected integrity errors from reporting back to Pegasus dashboard.

For VERITAS, **~.04%** of transfers fail with integrity errors. (**~1 / 2500** transfers)

Cause uncertain  
(diagnosis is harder  
than detection).

Possibly errors in  
http based transfers  
(s3 protocol against  
CEPH)





# Checksum Overheads

- We have instrumented overheads and are available to end users via pegasus-statistics.

Type	Succeeded	Failed	Incomplete	Total	Retries	Total+Retries
Jobs	1606	0	0	1606	31	1637
Workflow wall time						: 7 hrs , 59 mins
Cumulative job wall time						: 17 days , 23 hrs
# Integrity Metrics						
3944 files checksums compared with total duration of 9 mins , 18 secs						
1947 files checksums generated with total duration of 4 mins , 37 secs						
# Integrity Errors						
Failures: 0 jobs encountered integrity errors						

- Other sample overheads on real world workflows

**1000 Node OSG Kinc Workflow**  
**Overhead of 0.054 % incurred**

- Ariella Gladstein's population modeling workflow
  - A 5,000 job workflow used up 167 days and 16 hours of core hours, while spending 2 hours and 42 minutes doing checksum verification, with an overhead of 0.068%.
- A smaller example is the Dark Energy Survey Weak Lensing Pipeline with 131 jobs.
  - It used up 2 hours and 19 minutes of cumulative core hours, and 8 minutes and 43 seconds of checksum verification. The overhead was 0.062%.

# Our Talk


- Introduction and Motivations
- Our Approach
- Current Status
- Welcome to the Jungle
- Integrity Issues in the Wild
- **Future Work**



# Challenges

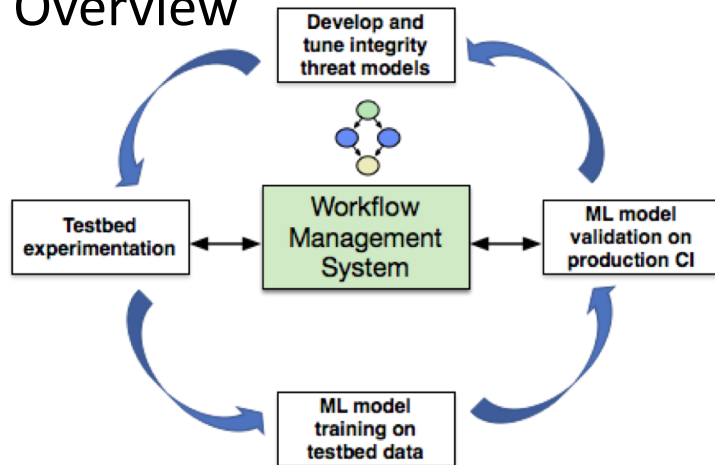
- Can we do more than know “something changed?”
- Detecting error easier than diagnosing error.
- Balance performance / integrity trade-off?
- How do we handle storage without compute capabilities?
- Long data life: today’s cryptographic algorithms will probably not last as long as we need the science data.  
E.g. what threats will Quantum computing bring?
- When do we hit limits of cryptographic algorithms (collisions)?
- Are all errors in all types of data of equal concern?

## Going Forward: Integrity Introspection for Scientific Workflows (IRIS)

- National Science Foundation CICI IRIS Grant #1839900 
- SWIP addresses **integrity checking** making sure that workflow computations are protected from integrity errors, but
  - Doesn't address analysis of integrity errors discovered, i.e. tracing the source of error or doing root cause analysis to remedy the problem.
- IRIS goal: Detect, diagnose, and pinpoint the source of unintentional integrity anomalies in scientific workflow executions on distributed cyberinfrastructure. (**integrity analysis**)

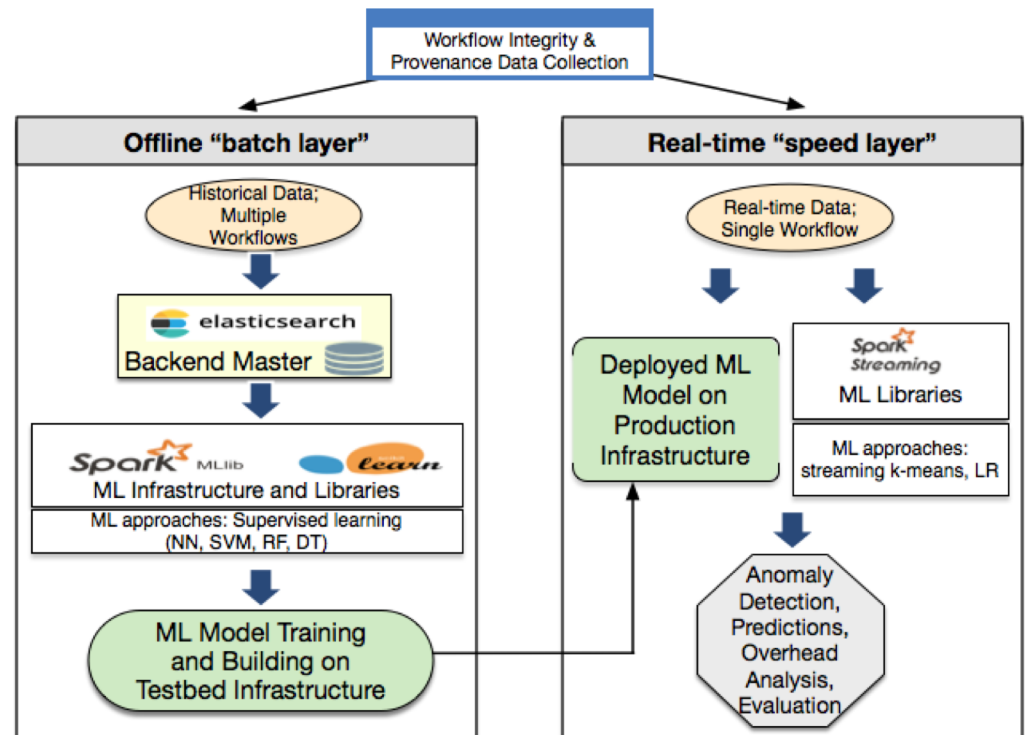
# IRIS Overall Approach

## IRIS Overview



Train ML algorithms on controlled testbeds and validate on national CI by integrating framework with Pegasus.

Engage with science application partners to deploy the analysis framework.



IRIS proposed framework

# Thanks!



We thank the National Science Foundation for funding this work (Grants 1642070, 1642053, 1642090). Views expressed may not necessarily be the views of the NSF. Thanks to Eli Dart for Brocade TSB details.