Tapis-CHORDS Integration: Time-Series Data Support in Science Gateway Infrastructure

Sean B. Cleveland*, Anagha Jamthe†, Smruti Padhy†, Je'aime Powell†, Joe Stubbs†, Michael D. Daniels§, Suzanne A. Pierce†, and Gwen A. Jacobs*,
*University of Hawaii-System, Honolulu, HI, USA
seanbc, gwenj@hawaii.edu
†Texas Advanced Computing Center, Austin, TX, USA
ajamthe, spadhy, jpowell, jstubbs, spierce@tacc.utexas.edu
§Colorado State University, Boulder, CO, USA
303miked@gmail.com

***Abstract*—The explosion of IoT devices and sensors in recent years has led to a demand for efficiently storing, processing and analyzing time-series data. Geoscience researchers use time-series data stores such as Hydroserver, VOEIS and CHORDS. Many of these tools require a great deal of infrastructure to deploy and expertise to manage and scale. Tapis's (formerly known as Agave) platform as a service provides a way to support researchers in a way that they are not responsible for the infrastructure and can focus on the science. The University of Hawaii (UH) and Texas Advanced Computing Center (TACC) have collaborated to develop a new API integration that combines Tapis with the CHORDS time series data service to support projects at both institutions for storing, annotating and querying time-series data. This new Streams API leverages the strengths of both the Tapis platform and CHORDS service to enable capabilities for supporting time-series data streams not available in either tool alone. These new capabilities may be leveraged by Tapis powered science gateways with needs for handling spatially indexed time-series data-sets for their researchers as they have been at UH and TACC.**

***Index Terms*—API, CHORDS, science gateway, streaming data, Tapis, time-series.**

I. Introduction

The explosion of IoT devices and sensors in recent years has lead to a demand for support of storing, processing and analyzing time-series data. Further, as more ML and AI systems are developed and come online the need for reenforcement data and dataset that can be used for training as well as input for driving events is increasing. A number of technologies have sprung up for storing large amounts of log data (Elasticsearch, Splunk,Apache Flume) and process data streams (Apache Kafka, Apache Storm, Apache Fink). Geoscience researchers have been using time-series data stores such as the Hydroserver [1],Virtual Observatory and Ecological Informatics System (VOEIS) [2] and the Cloud-Hosted Real-time Data Service (CHORDS) [3] for years. Many of these tools can be powerful but also require a great deal of infrastructure overhead to deploy and expertise to manage and scale.

Presented at Gateways 2019, San Diego, CA, USA, September 2325, 2019. https://osf.io/meetings/gateways2019/

To support streaming data for science, UH and TACC have developed APIs and infrastructure, the Streams API, to enable the integration of streaming data workflows, storage and retrieval with temporal and spatial support. These features pave the way to automated data stream processing workflows upon ingestion and integrate metadata for organization and robust data curation needed for dissemination and wider research community discovery and re-use. In this paper, we present our work to integrate existing CHORDS services to support streaming sensor data in Tapis [4], [5]. We start with an overview of Tapis metadata/data features and CHORDS and look at the synergy that can be enabled. We then present our new proof of concept for the Streams (Tapis-CHORDS integration) API and discuss the performance, challenges and opportunities.

II. Background

A. Tapis

Tapis is an open source, platform-as-a-service for hybrid cloud computing, data management and reproducible science. Tapis uses standards-based technologies and community promoted best practices to enable users to run code, manage data, collaborate meaningfully, and integrate anywhere. Tapis has been in production as the middleware that currently powers a number of community science gateways. Tapis is a multi-tenant, cloud-native distributed system. All services within the platform run as Docker containers, orchestrated as a single microservice architecture. The platform can be viewed as three logical tiers: platform services, science APIs, and support services. Platform Services contain services providing identity and API management, client registration, tenant admin services, and documentation. Science APIs contain the primary Science as a Service (ScaaS) functionality used to power science gateways such as data and job management, app publishing, notifications, etc. This tier is further divided into a frontend collection of loosely coupled services, microservices, and backend service workers. The microservices serve the user HTTP requests to the REST APIs, and service workers handle processing of asynchronous requests such as data movement and app publishing. Support services include databases, message queues, caches, object stores, service discovery, notification relays, and websocket proxies. A tenant represents a group of users, applications, and entitlements. Each instance of Tapis can support one or more tenants. Individual tiers and microservices can be replicated, configured, and scaled in support of a specific tenant, or shared to better leverage a single, consolidated resource footprint.

B. CHORDS

CHORDS is a real-time data services infrastructure that provides an easy-to-use system for acquiring, navigating and distributing real-time data streams via cloud services and the Internet. CHORDS can lower the barrier to these services for small instrument teams, employ data and metadata formats that adhere to community accepted standards, and broaden access to real-time data for the geosciences community. The CHORDS Portal is a Ruby on Rails web application and database that accepts real-time data from distributed instruments, and serves the measurements to anyone on the Internet. The data streams are pushed to and pulled from the CHORDS portal InfluxDB store using simple HTTP requests. Management tool allow users to monitor remote instruments, ensure correct operation, and maximize data collection. A rolling archive enables scientists and analysts to easily fetch the data in real-time, delivered directly to browsers, programs and mobile apps.

C. Motivation

The need to integrate existing CHORDS services to support streaming sensor data in Tapis, emerged at the 2018 workshop for the EarthCube Research Coordination Network for Intelligent Systems for Geosciences (IS-GEO RCN) to address the use cases from various domain scientists in data science, geoscience and informatics, particularly at UH and TACC. UH data scientists want the ability to develop reproducible novel ML/AI models that leverage advanced CI to analyze big data, streaming data and produce secondary data and knowledge products. Similarly, the Planet Texas 2050 group at TACC wish to leverage spatially dense and ever-increasing temporal datasets generated by Arduino-based microcontrollers as ground-truth inputs for integrated models of water-land-atmosphere-urban systems. Deployments collect hydrological and atmospheric measurements to feed models describing various processes related to flooding and aquifer recharge. Tapis-Chords integration stemmed as common solution to both institutional use cases.

III. Early Adopters/Use Cases

A. Ike Wai Project

The Ike Wai Gateway, water science gateway developed at the University of Hawaii (UH), Ike means knowledge and Wai means water in Hawaiian [7]. The gateway supports research in hydrology and water management, providing tools to address questions of water sustainability in Hawaii. The gateway provides a centralized web based user interfaces and APIs supporting multi-domain data management, computation, analysis and visualization tools to support reproducible science, modeling, data discovery and decision support for the Hawaii EPSCoR Ike Wai research team and wider Hawaii hydrology community. By leveraging the Tapis platform, UH has constructed a gateway that ties data and advanced computing resources together to support diverse research domains including microbiology, geochemistry, geophysics, economics and humanities, coupled with computational and modeling workflows delivered in a user friendly web interface and REST APIs (Fig 1). The Ike Wai project has deployed a number of sensors for measuring wells, rainfall and submarine groundwater discharge. Researchers desire the ability to access some of the data in real-time and in a serialized, queryable manner. Current support for these data is via annotated log files and spreadsheets and data extraction of particular measurement types, for instance chloride concentration, requires manually performing ETL workflows to create the desired dataset that spans multiple spatial locations. The ability to subset and combine data across distributed spatial locations is a must for groundwater modeling and investigations.

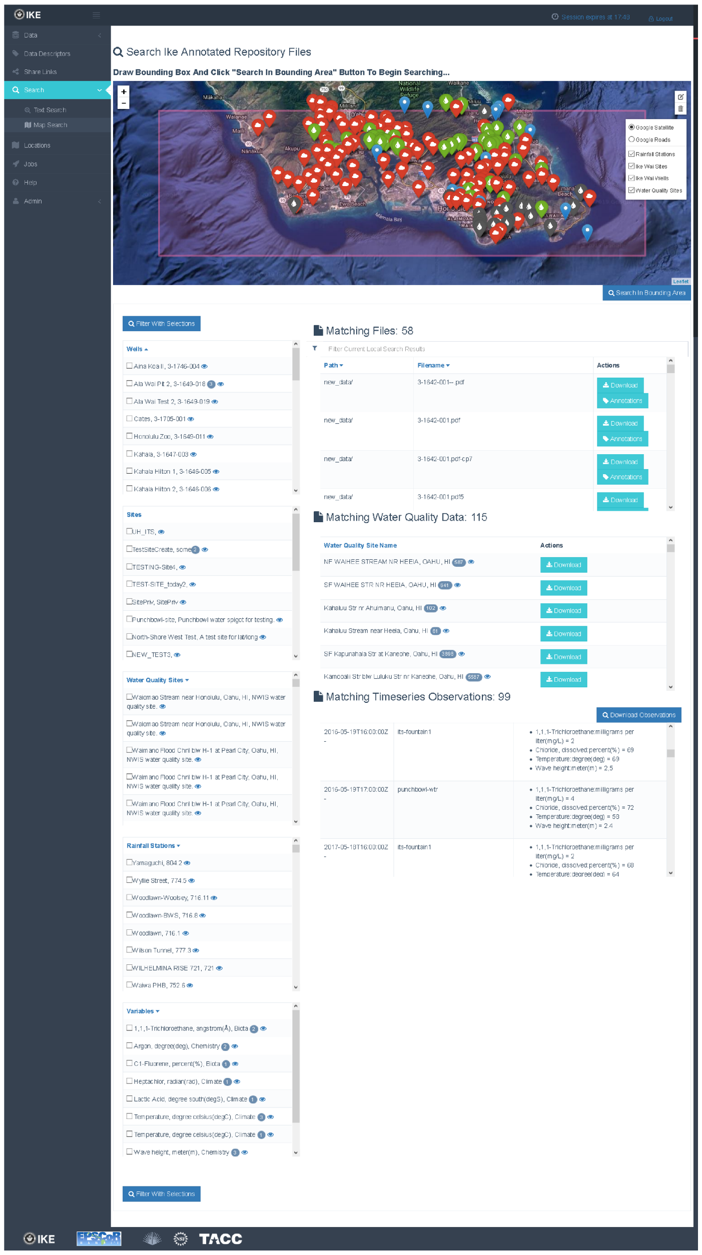
B. Planet Texas 2050

By the year 2050, the state of Texas is forecast to increase in population from 28 million to nearly 55 million residents. As a result, the effects of present utilization in the sustainability of natural resources (water, energy, and land-use) must be modeled and made available to policymakers. The Planet Texas 2050 (PT2050) project is designed to address knowledge and information needed to inform and support resilient responses in the face of identified vulnerabilities. The DataX Science Gateway, built on Tapis, is in development as part of the PT2050 initiative, to provide a platform through which scientists, data analysts, and policymakers collaborate to generate cross-disciplinary environmental models. The scientists and analysts creating the hybridized models will have unique access to both datasets, workflow generation tools, and collaborators historically partitioned across disciplines. The DataX Gateway enables the ingestion, data transformations and composition of integrated models. Core capabilities within the data portal include tools for assimilating disparate datasets, pre-processing data sources for inclusion in integrated models, and sharing through the community with access to large scale resources including storage, and computational capabilities at the Texas Advanced Computing Center. Generally, integrated models use static datasets. The purpose of this research was to explore a method by which real-time in-situ environmental edge monitoring systems could stream data into backend models for processing. The real-time data serves as a groundtruth source of information for models and expands the spectrum of possible use cases the DataX Gateway could support.

IV. Implementation

A. Tapis Metadata Changes

Tapis's Metadata service utilizes MongoDB as the database for storing metadata JSON documents in a Mongo collection



Search Ike Annotated Repository Files

Draw Bounding Box And Click "Search In Bounding Area" Button To Begin Searching...

Wells
- Aina Koa II, 3-1746-004
- Ala Wai Pit 2, 3-1649-018 3
- Ala Wai Test 2, 3-1649-019
- Cates, 3-1705-001
- Honolulu Zoo, 3-1649-011
- Kahala, 3-1647-003
- Kahala Hilton 1, 3-1646-005
- Kahala Hilton 2, 3-1646-006

Sites
- UH_ITS,
- TestSiteCreate, some 2
- TESTING-Site4,
- TEST-SITE_today2,
- SitePriv, SitePriv
- Punchbowl-site, Punchbowl water spigot for testing.
- North-Shore West Test, A test site for lat/long
- NEW_TEST3,

Water Quality Sites
- Waiomao Stream near Honolulu, Oahu, HI, NWIS water quality site.
- Waiomao Stream near Honolulu, Oahu, HI, NWIS water quality site.
- Waimano Flood Chnl blw H-1 at Pearl City, Oahu, HI, NWIS water quality site.
- Waimano Flood Chnl blw H-1 at Pearl City, Oahu, HI, NWIS water quality site.
- Waimano Flood Chnl blw H-1 at Pearl City, Oahu, HI, NWIS water quality site.

Rainfall Stations
- Yamaguchi, 804.2
- Wyllie Street, 774.5
- Woodlawn-Woolsey, 716.11
- Woodlawn-BWS, 716.8
- Woodlawn, 716.1
- Wilson Tunnel, 777.3
- WILHELMINA RISE 721, 721
- Waiwa PHB, 752.6

Variables
- 1,1,1-Trichloroethane, angstrom(Å), Biota 2
- Argon, degree(deg), Chemistry 2
- C1-Fluorene, percent(%), Biota 1
- Heptachlor, radian(rad), Climate 1
- Lactic Acid, degree south(degS), Climate 1
- Temperature, degree celsius(degC), Climate 3
- Temperature, degree celsius(degC), Climate 1
- Wave height, meter(m), Chemistry 3

Matching Files: 58

Path	Filename	Actions
new_data/	3-1642-001--.pdf	Download Annotations
new_data/	3-1642-001.pdf	Download Annotations
new_data/	3-1642-001.pdf-cp7	Download Annotations
new_data/	3-1642-001.pdf5	Download

Matching Water Quality Data: 115

Water Quality Site Name	Actions
NF WAIHEE STREAM NR HEEIA, OAHU, HI 587	Download
SF WAIHEE STR NR HEEIA, OAHU, HI 641	Download
Kahaluu Str nr Ahuimanu, Oahu, HI 102	Download
Kahaluu Stream near Heeia, Oahu, HI 61	Download
SF Kapunahala Str at Kaneohe, Oahu, HI 3898	Download
Kamooalii Str blw Luluku Str nr Kaneohe, Oahu, HI 5587	Download

Matching Timeseries Observations: 99

2016-05-19T16:00:00Z -	its-fountain1	• 1,1,1-Trichloroethane:milligrams per liter(mg/L) = 2 • Chloride, dissolved:percent(%) = 69 • Temperature:degree(deg) = 69 • Wave height:meter(m) = 2,5
2016-05-19T17:00:00Z -	punchbowl-wtr	• 1,1,1-Trichloroethane:milligrams per liter(mg/L) = 4 • Chloride, dissolved:percent(%) = 72 • Temperature:degree(deg) = 58 • Wave height:meter(m) = 2.4
2017-05-18T16:00:00Z -	its-fountain1	• 1,1,1-Trichloroethane:milligrams per liter(mg/L) = 2 • Chloride, dissolved:percent(%) = 68 • Temperature:degree(deg) = 64

Fig. 1. 'Ike Wai Gateway Example Screenshot - Spatial Search example with the integration of time-series data results in the "Timeseries Observations" using the Streams API.

called "Metadata". MongoDB offers geospatial support for data stored as GeoJSON objects. In order to leverage this capability for Tapis we needed to do two things. First a JSON field that would contain the information needed to be established and standardized on for storing any GeoJSON information for a piece of metadata/data. Second, a 2dsphere index needed to be created within the Tapis MongoDB "Metadata" collection that uses the established field, thus allowing MongoDB to support complex geospatial queries across all JSON documents that contain the field. For the Tapis platform instance at UH we standardized on a field id of "value.loc", "loc" being short for location. The value field is what Tapis uses to store user defined data information within a Metadata document, so all user defined fields are child fields of "value". With this straightforward modifications to the Tapis MongoDB the existing Metadata API functions and queries worked with no further changes necessary. The queries for searching on spatial Metadata conform to standard MongoDB spatial queries and could easily be utilized in the Tapis Metadata API GET requests.

B. Streams API

To integrate the CHORDS streaming data services into Tapis an additional API, called the Streams API was built using NodeJS to provide a REST service and the logic to combine the strengths of Tapis and CHORDS [6] (Table I). This new API, wraps the CHORDS data object APIs with logic that leverage the Tapis authentication and authorization services to allow multi-user, multi-permission and multi-tenant capabilities for the streaming data (Fig 2). The Streams API is designed to be integrated directly into Tapis utilizing the Authentication and Authorization services and requiring a Tapis API token to accompany any request. The Streams API service utilizes an administrative account to the CHORDS service which acts as the service account for all metadata and data. The segregation of metadata and data across Tapis tenants and users is enabled by using the Tapis metadata services internal permissions to restrict access, allowing a single CHORDS service to serve multiple tenants.

C. Metadata and Data Management

The Streams API service checks Tapis's internal permissions service to ensure the user is authorized for the requested actions. Once a user is authenticated and authorized the Streams API service moves forward with the requested actions, in the case of metadata object creations (i.e. site, instrument, variable) the service creates a CHORDS metadata object and then creates a Tapis metadata object that references the CHORDS object (Table I). The metadata objects are hierarchical with the site being a the top of the hierarchy and instruments belonging to a site and the variables belonging to an instrument. This linked metadata architecture enables the simple expansion of metadata using the Tapis metadata JSON document objects as well as advanced Tapis querying that includes spatial queries for the spatially indexed site metadata objects. Insertion of streaming data measurements uses the Tapis authentication and authorization services and metadata permission checks to ensure the user has been authorized for write access to the requested instrument. Upon verification the data is inserted into the CHORDS database. The Streams API support queries for measurements once authorization is established the the CHORDS service is accessed to return data in one of the supported GeoJSON or GeoCSV formats.

D. Local site implementation

Deploying the Streams API service requires an existing Tapis platform deployment to utilize, which does not have

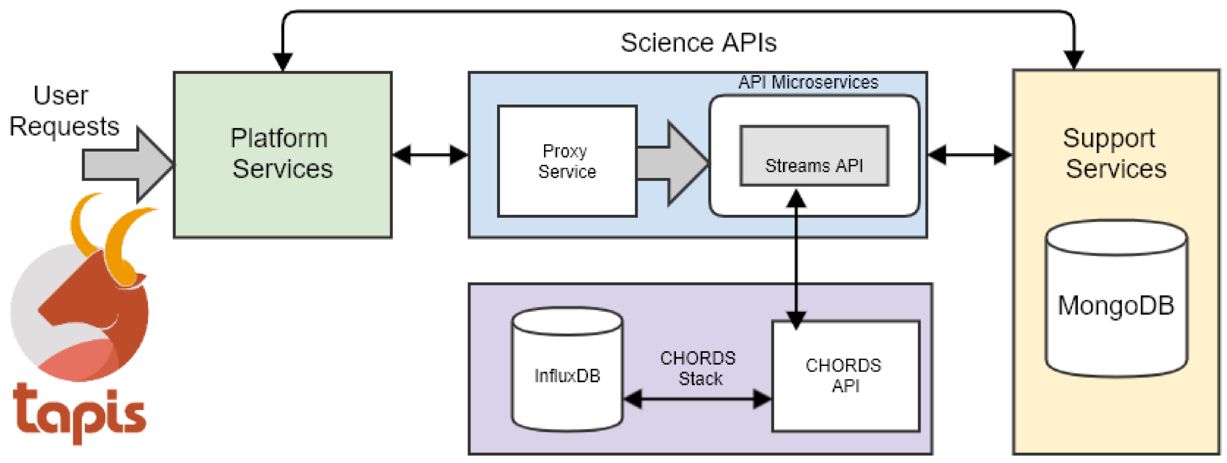


Fig. 2. Tapis Streams API diagram - Tapis platform services that process all user requests in green, Science APIs in blue with the Streams API service in gray and the Support services in yellow that contain the MongoDB for persisting Stream API metadata mappings to CHORDS. CHORDS service stack is in purple with the API and influxDB that Tapis integrates via the Streams API to offer streaming data support within Tapis.

TABLE I
STREAMS API ENDPOINTS

Name	Method & Endpoint	Description
Site	GET, POST streams/v2/sites	Geographical location which may host one or more instruments
Instrument	GET, POST streams/v2/instruments	A source of related measurements
Variable	GET, POST streams/v2/variables	A particular measurement made by an instrument
Measurement	GET, POST streams/v2/measurements	A single observation of a variable

to be local, and a CHORDS service stack (CHORDS web application, MYSQL, INFLUXDB, NGINX). The UH and TACC implementation have deployed the CHORDS stack and Streams API as docker containers. The UH and TACC implementations have the CHORDS services configured as only accessible to the Streams API service to protect the data, this allows security to remain at the Tapis level when the Streams API service is configured as part of the TAPIS API management system, which sits in front of all the Tapis APIs.

V. PERFORMANCE BENCHMARKING: STREAMS API

To demonstrate the feasibility of using the Streams API for handling and querying time-series data, we measured the performance metrics, *accuracy* and *latency* for data ingestion and fetch using Streams API. We compare it with the same metrics obtained with an out-of-box installation of the CHORDS server. For our experiments, we stood up a CHORDS server on a virtual machine with 6 cores and 4GB of memory, running CentOS. The CHORDS web-application, along with other components from CHORDS stack like, Grafana (graphical dashboard for data visualization),InfluxDB, MySQL and Kapacitor run inside separate Docker containers on the VM. We chose to install CHORDS version 0.9.7.1, available at the time of the experiments. Administrative account, user roles and an API security key settings are configured from the CHORDS portal. First, we derive the performance data for CHORDS server. HTTP POST and GET requests to create and fetch measurements for a specific instrument id are directly

TABLE II
PERFORMANCE BENCHMARK: DATA INGESTION

Service	Number of POST requests per test run	Data Accuracy (%)	Avg. request response time (seconds)
CHORDS	100	100	0.11
CHORDS	200	100	0.11
CHORDS	500	100	0.12
Streams	100	100	0.65
Streams	200	100	0.6
Streams	500	100	0.6

sent to the CHORDS server. Eight measurements including, wind direction, wind speed, wind max, temperature, humidity, pressure, rain total and battery are created with every request and data is stored in the CHORDS database with date and time annotations. Data accuracy and average response times for data ingestion for 100, 200 and 500 sequential HTTP POST requests is as shown in Table II. Table III provides average response time to fetch about 16000 measurements with every GET request.

Next, to ensure that adding an additional wrapper to the CHORDS data object is performant and accurate, we conducted performance benchmark tests on Streams API while doing data ingestion and fetch operations. HTTP POST and GET requests to create and fetch the measurements were made using the streams/v2/measurements end point along with the associated instrument uuid (stored in the Tapis Metadata).

TABLE III
PERFORMANCE BENCHMARK:DATA FETCH

Number of GET requests per test run	**Avg. request response time CHORDS (seconds)**	**Avg. request response time STREAMS(seconds)**
100	0.61	0.98
200	0.67	1.0
500	0.64	0.91

Requests included an authorization header with a Tapis API token for the user from TACC tenant. Data accuracy and average request response time for 100, 200 and 500 HTTP POST and GET requests using the Streams API is provided in the Table II and Table III respectively.

The comparison of the Stream API performance to the base CHORDS service appears to be consistent when handling many insertions and fetch. The average CHORDS measurement insertion request processed in 0.12 seconds for 500 serial insertions submitting as fast as the script could execute curl calls asynchronously (Table II). The Streams API averaged 0.48 seconds longer than the base CHORDS service under the same conditions (Table II). A request to the Tapis metadata service on average requires 0.42 seconds, which accounts for the extra authentication and permissions overhead validation in the Streams API. Similarly, the fetch request with CHORDS were processed in 0.64s for 500 serial requests, whereas the Streams API averaged 0.27 seconds longer (Table III). Handling parallel request is standard for web services so the additional overhead in the Streams API is not seen as detrimental at this early junction. The Ike Wai and DataX gateways use cases both have sensors that submit at a rate of 1 measurement per second or less, so the current proof of concept solution is performing within those parameters and has revealed no issues.

VI. FUTURE WORK

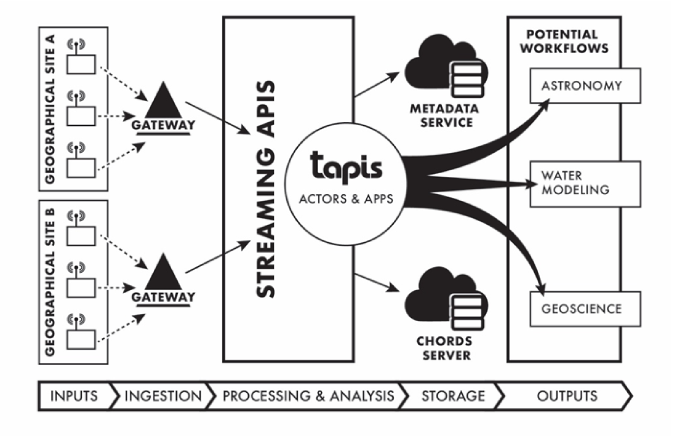


Fig. 3. Tapis workflow diagram illustrating support for actor and application integration streaming data workflows.

The current Stream API release supports basic support for streaming data storage, retrieval and querying, however there are additional opportunities that are available with Tapis. One future feature will be the ability to register an Abaco [9], [10] application to preform automated processing upon data ingestions. This feature would open up a host of possibilities for supporting automated monitoring, pre-processing, computational workflows and machine learning opportunities (Fig 3).

VII. CONCLUSION

The new Tapis Streams API that integrates CHORDS services in a functional infrastructure solution that successfully supports streaming data for research currently supporting two production science gateways.

ACKNOWLEDGMENT

This work was made possible by grant funding from the following: Project Tapis: Next Generation Software for Distributed Research NSF CSSI #1931439, #1931575, RII Track 1: Ike Wai Securing Hawaii's Water Future NSF OIA #1557349, Collaborative Research: SS2-SSI: The Agave Platform: An Open Science-As-A-Service Cloud Platform for Reproducible Science NSF ACI #1450413, and EarthCube RCN IS-GEO: Intelligent Systems Research to Support Geosciences; funded by the NSF ICER #1632211

REFERENCES

[1] D.G. Tarboton et al. 2009. "Development of a community hydrologic information system," 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation (2009), 988994.
[2] S.J.K. Mason et al. 2014. "A centralized tool for managing, archiving, and serving point-in-time data in ecological research laboratories," Environmental Modeling & Software. 51, (Jan. 2014), 5969.
[3] Daniels, M. D., Kerkez, B., Chandrasekar, V., Graves, S., Stamps, D. S., Martin, C., Botnick, A., Gooch, R., Bartos, M., Jones, J., Keiser, K. (2016). Cloud-Hosted Real-time Data Services for the Geosciences (CHORDS) software (Version 0.9). UCAR/NCAR - Earth Observing Laboratory. https://doi.org/10.5065/d6v1236q
[4] Dooley, R. et al. 2012. "Software-as-a-service: the iPlant foundation API," 5th IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS) (2012).
[5] Dooley, R; Fonner, J; Jacobs, G.A; Brandt, S. (2018), "The Agave Platform: An Open Science-As-A-Service Cloud Platform for Reproducible Science," figshare. Poster. https://doi.org/10.6084/m9.figshare.4675765.v2
[6] Tapis-CHORDS API source code https://github.com/UH-CI/tapis_chords_api
[7] Cleveland, S.B.; Geis, J.; Jacobs G.A; 2018. "The 'Ike Wai Gateway-A Science Gateway For The Water Future of Hawai'i","Proceedings of Science Gateways 2018, Austin TX, USA September 2018 (SGC18) DOI:https://doi.org/10.6084/m9.figshare.7152464.v2
[8] PT2050 https://bridgingbarriers.utexas.edu/planet-texas-2050/
[9] Stubbs J., Vaughn M., Looney J.," Rapid Development of Scalable,Distributed Computation with Abaco," Proceedings of the 10th International Workshop on Science Gateway, 2018.
[10] Arora R., Dooley R., Looney J., Poindexter M., Stubbs J.,"Design and Architecture of a Gateway for Supporting Both Batch and Interactive Computing Modes on Supercomputers," Gateways 2018