NIST Reusable Analytics Service Framework
Volume 1, Overview

Gregor von Laszewski
Wo Chang
Russell C. Reinsch
Geoffrey C. Fox

March 8, 2022

Edit ⇒ section-abstract.tex

Abstract

This document summarizes the NIST Analytics Service Framework that targets to analytics functionalities to be hosted on computational resources including Clouds, Containers, and High Performance Computing (HPC). Although we use the RE presentational State Transfer (REST) to formulate some details of the architecture, it is independent from REST and can be formulated in other frameworks. While using REST we use a familiar pattern for architect, implementers, and strategists. Due to the many frameworks, programming languages and services supporting REST the architecture can easily be enhanced and implemented with various technical solutions. The analytics framework also targets big data. Big data is a term used to describe extensive datasets, primarily in the characteristics of volume, variety, velocity, and veracity. While opportunities exist with Big Data analytics, the data characteristics can overwhelm traditional technical approaches, and the growth of data is outpacing scientific and technological advances in data analytics. To advance progress in Big Data analytics, the NIST Big Data Public Working Group (NBD-PWG) is working to develop consensus on important fundamental concepts related to Big Data. The results are reported in the NIST Big Data Interoperability Framework (NBDIF) series of volumes.

Key words

Adoption; barriers; implementation; interfaces; market maturity; organizational maturity; project maturity; system modernization.

Contents

List of Tables

List of Figures

Edit ⇒ section-production.tex

Report Production

This document is conducted as part of the NIST OUR WORKING GROUP. The working group holds public meetings regularly Tuesdays at 1 pm. The working group is public, and anyone can join that is interested in contributing to this document and bringing it to completion. We will be using GitHub to coordinate this work. Although the work is done in a public working group, we had one member of the group asking to keep the document private. As such, you need to join our meeting group first before we grant you access to this document. You can join the working group by contacting Wo Chang at wchang@nist.gov.

The production of this document is conducted to address the following needs

1. a one-page executive summary,
2. a detailed specification,
3. use cases that support this document that may be hosted in separate documents. Such documents could follow the template as provided at [1].

We are using the following tools to manage the completion of the document:

- an online document → we work on in Overleaf. Oevrleaf sinks to GitHub are done however only on a best effort basis. SO before you start working, make sure you pull in overleaf from GitHub.
- a task list → in GitHub,
- a versioned document source → in GitHub,
- a slack channel → to increas the communication outside of the Working group meetings and to coordinate task.
- a regularly updated PDF document with editing links pointing to the GItHub versioned documents. Please note that for the most recent document you will have to use GitHub or Overleaf.

Once you have joined the Working group, you can get access to the document directories by contacting laszewski@gmail.com.

> *Please note that we stopped using Word for the document management contributors consistently used different templates and formatting rules resulting in unsustainable multi hourlong cleanups every week instead of being able to focus on content. For this reason, we will not switch back to Word. However, a section could be written in markdown, which can easily be integrated into the document. Formatting should be kept at a minimum.*

Annotations

It is easy to anotate content in the document we use the following, while the person specified is supposed to fix it:

Gregor: The issue we observe

```
\TODO{Gregor}{The issue we observe}
```

If a person name is not specified anyone is encouraged to fix it. Todos will beintgrated in a list of todos that are included at the end of the document in an appendix

We also use comments that are not included in the todolist, but may serve as temprary visual aides to alert, highlight or anotate text. This includes

`\alert{Please address this}`	Please address this
`\highlight{a highight}`	a highight
`\add{add this text}`	add this text
`\remove{remove this text}`	~~remove this text~~
`\replace{old text}{with new text}`	~~old text~~ with new text

Edit ⇒ section-summary

1 Executive Summary

: TBD

Edit ⇒ section-introduction.tex

2 Introduction

2.1 Background

With Big Data's compound annual growth rate at 61 percent and its ever-increasing deluge of information in the mainstream, the collective sum of world data will grow from 33 zettabytes (ZB, 1021) in 2018 to 175 ZB by 2025 . The presence of such a rich source of information requires a massive analysis that can effectively bring about much insight and knowledge discovery. While previous work focused on developing a Big Data Reference Architecture. This work specifically focused on the definition of ***Analytics Services***.

: likely outdated by now. find out current trends and numbers

We leverage activities conducted previously as part of the NIST Big Data Reference Architecture (NBD-RA) and NBD-RA Interfaces. However, the work here targets explicitly ***Big Data Analytics*** while integrationg legacy analytics with machine and deep learning analytics. We will focus on a service oriented framework.

2.2 Scope and Objectives

NBD-PWG[1] is exploring how to extend NBDIF [2] for packaging scalable analytics as services to meet the challenges of today's information analytics. These services are intended to be reusable, deployable, and operational for Big Data, High Performance Computing, AI machine learning (ML), and deep learning (DL) applications, regardless of the underlying computing environment.

This document explores key focus areas and document level of interest from industry, government, and academia in extending the NBDIF to develop scalable analytics as services that are reusable, deployable, and operational, regardless of the underlying computing environment.

: introduce in the background section. THe background section has been integrated here. check valitity and fix somehow.

The document is organized as follows and motivated by the tasks identified in the previous section. Each section is augmented with the key area it contributes to.

- Section 3: Definitions and Concepts: Develop a brief list of definitions that can be used to improve communication between different interdisciplinary groups while allowing them to use the same language.
- Section 4: Use Cases for Analytic Services: A compilation and organization of use cases focusing on analytic services including traditional statistical, AI/ML/DL, and emerging analytics application domains. It will help identifying the meta- and technical requirements.
- Section 5: Defining and Finding Reusable Analytics Services. This section will include the definition and conceptual architecture of reusable analytics services. This includes the following concerns that are organized as subsections.

1

2

  - Section 5.1: Adaptation of the FAIR principle to supprt an Analytics Service FAIR Principle.
  - Section 5.2: Service Catalog: To communicate the existence of the services to others service registries can be used.
  - Section 5.3: Service Registry: To communicate the the features of the services to others service registries can be used.

- Section 6 Service Analytics Federation: To leverage multiple existing services federated services can be used to integarte them.

- Section 7: Data in Reusable Analytics Services: Here we explore a number of important issues related to data that is used by the analytics services.

- Section 8: Package Analytic Algorithms as Service Payloads: Here we explore how to package analytic algorithms with well-defined input and output parameters as service payloads that can be reusable, deployable, and operational across multi-cores, CPUs, and GPU computing platforms.

- Section 9: Analytics Service Interfaces and Encapsulation: Here we explore a minimal set of services and their interfaces to be used as part of a generalized analytics framework. It includes to encapsulate the service payload with well-defined format, interface, and end-to-end access control for open and secure computing environment.

- Section 10: Resource Management: Here we investigate and define a minimal set of resource management services and interfaces for application orchestration and workflow between processes.

- Section 11: Security Considerations in Reusable Analytics Services: Here we explore a number of important security considerations related to reusable analytics services.

- Section 12: Outreach Activity: In our outreach activity we investigate the inclusion and collaboration with other interested parties.

Edit ⇒ section-definitions.tex

3 Definitions and Concepts

In this section we provide a list of definitions and doncepts that help communication between different interdisciplinary groups while allowing them to use the same language.

A comprehensive glossary (Appendix A) is provided in the appendix.

Analytics Service

Analytics Catalog

Analytics Registry

Analytics Workflow

Payload

Analytics: The systematic analysis of data, to uncover patterns and relationships between data, historical trends, and attempts at predictions of future behaviors and events.

Analytics management: A sub function within the [metadata] registry. Analytics services azure cognitive, google analytics, aws [dozens], watson analytics... in contrast to ML frameworks like tensorflow, pytorch, caffe2, and in contrast to Programming libraries like python, scikit, shiny, or R Studio [??]

Analytics Workflow: The sequence of processes or tasks part of the analysis

Edit ⇒ section-usecases.tex

4 Use Cases for Analytics Services

4.1 Use Case Development Process

To specify use cases for our analytics framework, we encourage contributors to contact us and provide us with their high-level descriptions of their use cases. The use cases should be focusing on highlighting one or multiple aspects of the features related to analytics frameworks. While inspecting the various features we intend to collect and analyze them in various contexts that are relevant for analytics users. The lessons learned from this analysis are to be integrated into this document in order to formulate a comprehensive vendor neutral analytics framework. Use cases can be formulated in various format but should include diagrams that make them easy to comprehend as well as allowing the reader to extract the specific analytical aspects. Such diagrams can include functional cross diagrams, process diagrams and others. Use cases should especially address the use of metadata describing the functional and the data related properties. This includes metadata related to time, space, exchange/protocols, privacy, and security related aspects. A functional description of the use case is to be included as a subsection called Functionalities and Activities. This section is mirroring our experience with documenting use cases as part of the Big Data Application Provider of NBDIF Reference Architecture. Hence, we assume the following draft form for a use case:

Title: Title of the use case

Contributor: The list co contributors

Description: One to two sentences about major functionalities and activities with respect to the sample cross-functional diagram

Cross-Functional Diagram: Inclusion of a cross functional Diagram, alternatively other diagrams could be chosen.

Functionality Activities:
1. Activity #1 – description...
2. ...
3. Activity #n – description...
4. Use Case Summaries

: introduce in the background section. THe background section has been integrated here. check valitity and fix somehow.

Next, we will list use case summaries and if available point to specific publications on the NBDIF Web page that include more details. The expectation of this section is to

- Provide an overview of use cases that motivate this document
- It will summarize requirements that we obtain from these use cases that influence how we proceed.

As a result, we identify how they fit into the workflow of data analytics. This includes the description of a subset of functionality that is used in general by data analytics. In particular, it described the relationship between input and output of data analytics components and interfaces. The use case summaries are expected to be available through the BigDataWG Web page and includes currently the following examples:

1. M0701 – Use case template [2]
2. M0702 - Numeric weather prediction [3]
3. M0703 - HVAC Heat ventilation and air conditioning [4]

Edit ⇒ usecase/weather.tex

4.2 Use Case: Numeric Weather Prediction

Background. Large amounts of weather data ar e produced continually and stored in many different databases. Accurate weather predictions require large amount s of processing power to accurately simulate conditions worldwide at a high resolution and fre quent intervals. One of the most computationally consuming parts of a reliable weather model is th e microphysics scheme. The current microphysics scheme, Weather Research and Forecasting (WRF) Single Moment 6-class Microphysics (WSM6), simulates the processes in the atmosphere that leadto the formation and precipitation of rain, snow, and graupel and requires complex floating-point operations needing to be performed on vast amounts of data for accurate simulations. As computer performanceimproves, so does the Numerical Weather Prediction (NWP) models' resolution and accuracy. However, there is still much progress to be made, as simulation accuracy still falls off sign ificantly for predictions more than 36 hours in advance. Figure 4.2 shows the general WRF modeling system flow chat.

russell: note that the 'string' reusable, deployable, and operational was also used in the previous ppg.

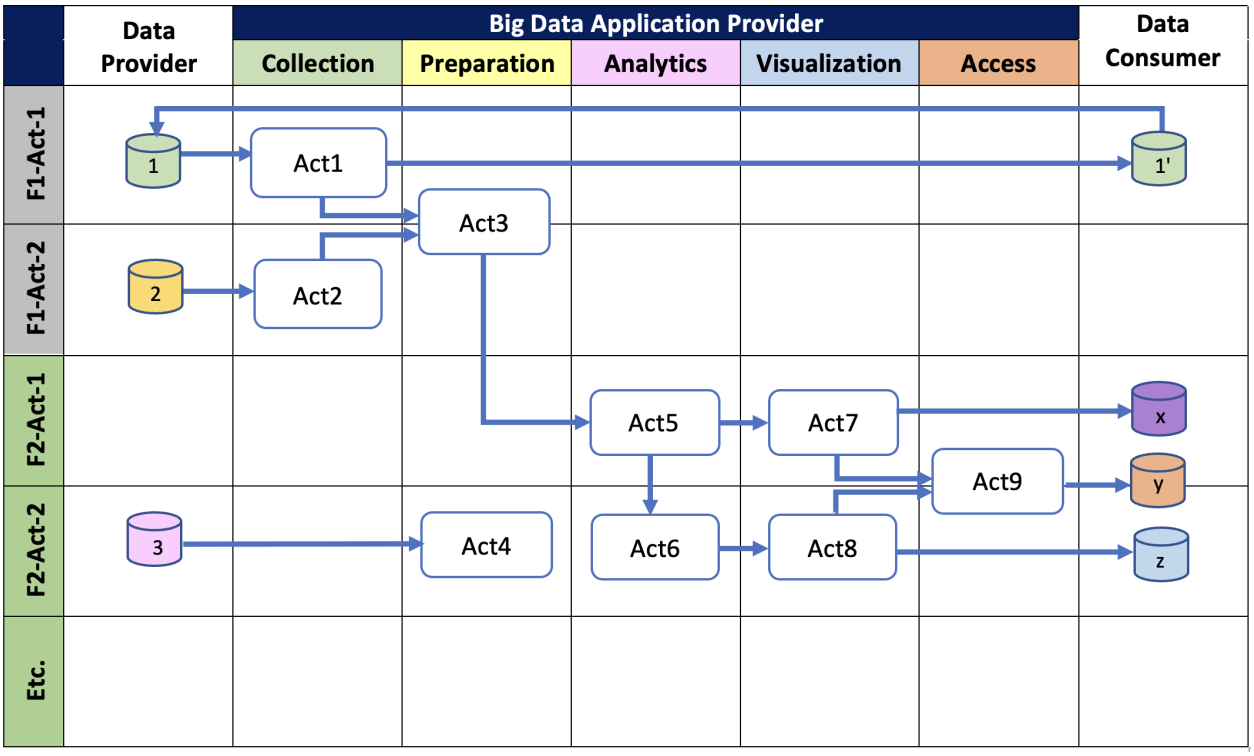

Figure 1: Cross-functional Diagram

Functionalities and Activities (based on Big Data Application Provider of NBDIF Ref. Architecture). In this case study, we only focus on two main functionalities, namely WPS and WRF, and their activities. Figure 4.2 shows the cross-functional diagram for their actions.
WPS Activities:

1. geogrid – defines simulation domains and interpolate various terrestrial data sets to the model grids. Input data available at [1].

2. ungrib – extracts needed meteorological data and packs it into an intermediate file format. Input data available at [2]

3. medgrid – prepares horizontally interpolate the meteorological data onto the model domain. Input data from the output of geogrid and ungrib.

WRF Activities:

1. real – prepares vertically interpolates the output from metgrid, and creates a boundary and initial condition files with some consistency checks.

2. wrf – generates a model forecast.

Datasets.

1. WRF Users Page, WPS V4 Geographical Static Data Downloads Page [5]

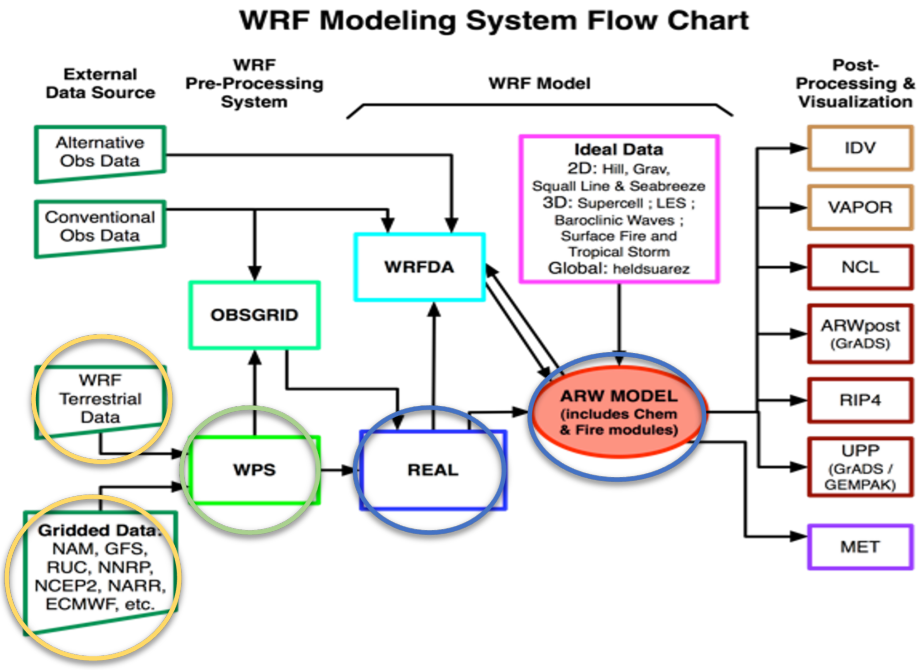

Figure 1: WRF Modeling System Flow Chart with Various Configuration.

Figure 2: WRF Modeling System Flow Chart with Various Configuration.

2. NCEP FNL Operational Model Global Tropospheric Analyses, continuing from July 1999 [6]

Edit ⇒ usecase/hvac.tex

: remove caption from within image 4.2

4.3 Use Case: HVAC Recommendation

Background. Continuous streaming data is produced by heat ventilation and air conditioning (HVAC) systems every day from the residential houses. This data is stored in a databased on the cloud as it arrives. The data is used to calculate what should be the next HVAC set point in the house with respect to user preferences. Periodic recommendations considering environmental parameters, user comfort level and past user preferences require advanced machine learning algorithm called reinforcement learning this sentence needs grammar edit. Accurate recommendations can save energy and reduce cost. This functionality has three parts Environmental Forecasting (EF), Learning from the past, (LP), and Set-Point Recommendation (SPR). EF calculates weather temperature and price predictions. LP learns from the behavior in the past. SPR model calculates next set-point based on past experience and EF predictions. Figure 4.3 shows the general modeling system flow chat.

Functionalities and Activities (based on Big Data Application Provider of NBDIF Ref. Architecture).

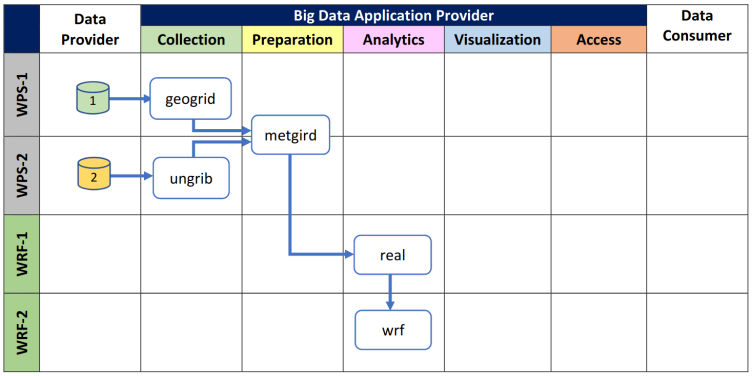

Figure 3: Cross-Functional Diagram Numerical Weather Prediction.

In this case study, we only focus on three main functionalities, namely EF, LP and SPR, and their activities. Figure **??** shows the cross-functional diagram for their actions.

: move to authors: Olivera Kotevska, Research Scientist, Oak Ridge National Laboratory, U.S.A.

EF Activities:

1. weatherD – Collects current weather temperature and predicted temperature for timestamp X.
2. priceD – Collects current electricity price and predicted price for timestamp X.
3. pred – Extract needed data fields and packs it into an intermediate file format. Input data from the output of weatherD and priceD.

LP Activities:

1. hist – Prepares history data points and creates initial condition weights.
2. reward – Generates reward based on the current weatherD and priceD.
3. learn – Collects data from current weatherD, priced, reward.

SPR Activities:

1. rules – Creates rules based on user preferences and conversion preferences.
2. rlmodel – Interpolates the output from learn, rules and generates set point recommendation

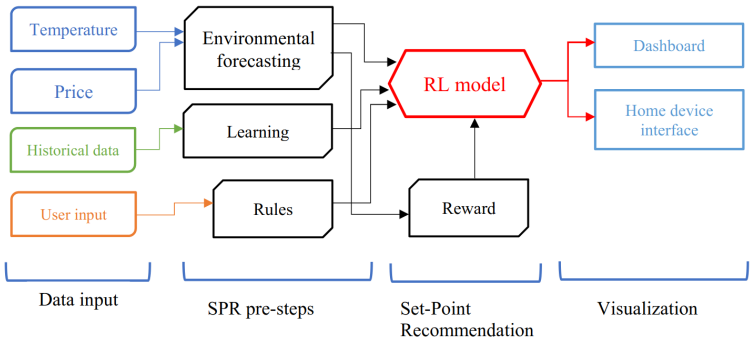

Figure 4: HVAC general modeling system flow chat.

Edit ⇒ section-defining.tex

5 Defining and Finding Reusable Analytics Services

Defining and Finding Reusable Analytics Services. This section will include the definition and conceptual architecture of reusable analytics services. This includes the following concerns that are organized as subsections.

Edit ⇒ section-fair.tex

5.1 AS-FAIR-DO: Analytics Service FAIR Principle

To project easy reusability, we strive towards the implementation of the AS-FAIR-DO principle for analytics services. The FAIR principle is typically applied to data and as such we can apply it the metadata associated with analytics services. The FAIR principal addresses who to be findable, be accessible, be interoperable, and be reusable. In Figure 6 we explicitly augmented the general FAIR principle with terminology so it can apply to analytics services. The augmentations are colored in red.

To be Findable:
F1 analytics services metadata are assigned a globally unique and persistent identifier
F2 analytics services data are described with rich metadata (defined by R1)
F3 analytics services metadata clearly and explicitly include the identifier of the data related to the analytics services it describes
F4 analytics services metadata are registered or indexed in a searchable resource
To be Accessible:
A.1 analytics services metadata are retrievable by their identifier using a standardized communications protocol
A1.1 analytics services the protocol is open, free, and universally implementable
A1.2 the analytics services protocol allows for an authentication and authorization procedure, where necessary
A.2 metadata are accessible, even when the data are no longer available
To be Interoperable:
I1. analytics services metadata use a formal, accessible, shared, and broadly applicable language for knowledge representation.
I2. analytics services metadata use vocabularies that follow FAIR principles
I3. analytics services metadata include qualified references to other metadata
To be Reusable:
R1. analytics services metadata are richly described with a plurality of accurate and relevant attributes
R1.1 (meta)data are released with a clear and accessible data usage license
R1.2 (meta)data are associated with detailed provenance
R1.3 (meta)data meet domain-relevant community standards
To be Deployable:
D.1 analytics services metadata ...
To be Operational:
O.1 analytics services metadata ...

Figure 6: Fair guiding principles adapted to analytics services: Analytics Services - FAIR - Deployable and Operational (AS-FAIR-DO).

Edit ⇒ section-catalog.tex

5.2 Analytics Service Catalogue

Motivation. Cloud providers offered a considerable set of analytics services to their customers. There are many analytics services available. A user needs to be able to quickly obtain an overview of such available services. This helps identifying further actions in order to evaluate them and identify if further investigation is justified. The catalouge contains enough details to locate the service and evaluate if it is useful. However, it may not provide technical details which are captured by a service registry instead.

Access Requirements. The catalogue may be public or may be restricted while authorized entities may access it. As analytics services may evolve over time, time dependent versioned descriptions of the services must be able to be included. An organizational entity may manage their own catalogues. It is desirable to have the catalogues be uniform, so that they can be combined into a larger catalogue combining entries of multiple organizations.

: 8.2.4.2 and 8.2.4.4 and 8.2.6.2 are all labelled Access Requirements. perhaps we should be more specific

Federation. The offerings are typically limited to a particular vendor. Users can benefit from a federates service catalogue to search and explore for needed services by the user. In contrast to a registry a catalogue may not include all technical details but could in contrast include services that lack such details and thus can be the basis of an exploratory process. A Federated analytics service repository is planned to be hosted on GitHub (LINK TBD) The catalogue contains the following attributes, many of which are also used in an analytics service registry.

The catalogue is organized as a list of entries, where each entry contains a number of attributes. These attributes may be required or optional. We list in Table 1.

Edit ⇒ section-registry.tex

5.3 Analytics Service Registry

Motivation. The goal of a federated analytics service registry is to establish federated registries to locate and consume analytics services with persistent identifiers across organizations.

A service registry can serve as a public, private, or federated registry. The first two properties define if the registry is public or private. In case of a private registry proper security measures need to be taken into account to govern access. Our framework does not make any recommendations about the security framework chosen and it is up to the implementer to specify it. In case of a federated registry, more than one registry can be joined, to provide the user the impression of a single registry.

Within the analytics services we distinguish two classes. The first class are instantiated (running) services that are offered by a service provider and allow direct reuse. The second class are library providers that distribute analytics activity not as an instantiated service, but as a source code library which can be deployed as a service.

: use case

A user wants to find an analytics service and needs to identify candidate services based on their descriptions and features. A user wants to find services quickly and therefor expects modern keyword search and taxonomy, faceted search, query functionalities; as well as descriptions that facilitate location and identification of relevant and appropriate analytics services, from the registry.

Table 1: Catalouge attributes

Name	Description	Required
ID	UUID, globally unique	✓
Name	Name of the service	✓
Title	Human readable title	✓
Public	True if Public (needs use case to delineate what pub private means)	✓
Description	Human readable short Description of the Service	✓
Version	The version number or tag of the service	✓
License	The license description	✓
Microservice	✓/No/Mixed	✓
Protocol	REST	✓
Owner	Name of the distributing entity, organization or individual. It could be a vendor.	✓
Modified	Modification Timestamp (when first same as created)	✓
Created	Date on which the entry was first created	✓
Documentation	Link to a URL with detailed description of the service	O
Source	Link to the source code if available	O
Tag(s)	Human readable common tags that are used to identify the service that are associated with the service	O
Category(s)	A category that this service belongs to (NLB, Finance, ...)	O
Specification/ Schema	Pointer to where schema is located	O
Additional metadata	Pointer to where additional is located including the one here.	O
Endpoint	The endpoint of the service	O
SLA/Cost		O
Authors	contact details of the people or organization responsible for the service (freeform string)	O
Data	Description on how data is managed	O

✓= Required; O = Optional

The registry contains enough details to not only locate the service, but also how to use it.

Access Requirements.

: possibly change section to privacy requirements so that LoA and Authentication can be moved to separate section ?

Public Analytics Service Registry. Public analytics discovery services are intended to allow users to find publicly hosted services. The information provided includes the provider, [x], and [y], and / thus reduce users' efforts in locating relevant services.

Levels of Assurance (LoA) in User Identity Most readers should be familiar with functionality to *sign in with ORCHID, or Facebook* or something known to the user. In general identity management scenarios, this provision enables what is referred to as *guest identities*, which is useful for many users who are interested in invoking low level activities or less sensitive operations. With respect to federated service authentication and authorization, OIDC guest identities meet a low level of assurance. In contrast, users with higher LoAs are afforded permissions to perform to privileged activities or gain access to more sensitive xyz.

Multi factor Authentication in User Identity A means for authenticating users via two or more types of authentication. An MFA instrument can elevate a user's level of assurance profile. RAF and IGTF are examples of such assurance framework standards. OpenID Connect, SAML, and X.509 are examples of services that expose interfaces for multiple authentication.

Private Analytics Service Registry. Analytics Services stored in private registries are only available to aut that advertise specialized services to its user community. In contrast to a public analytics registry, access controls in private registries are more restricted. In addition, different group privileges may restrict the visible analytics service to the user. See related sub sections on user identity and levels of user priveledge ...

Federated Analytics Service Registry. A user wants to make selection decisions regarding which service to use. Analytics service brokers / providers therefore offer a federated analytics service in which multiple services from multiple providers are included. Rather than having to visit multiple, separate providers' registries, the user can visit the federated registry of the analytics broker to lookup all potentially suitable services, via a single interface / browser. It may be expected that federated registries abstract the technical effort that casual users would experience during location and inspection of published analytics services. Underlying analytics service registry technologies leverage cross - organization persistent identifiers, enhanced with information that the original service provider may not have available, and xyz. such "enrichment" may could include for example, cost comparisons, or (some type of) ratings from its user community.

Enhanced Analytics Service Registry. Both public and private registries my need to be enhanced by providing detailed information so the user has a better understanding of the offering and allows comparison to similar artefacts maintained and published in the registry. Information details may include for example, benchmark information, service level agreements, or cost measures such as carbon cost, or technical limitations such as storage access and availability for big data.

Registry Namespace. To allow uniform integration of entries into a unified namespace, URLs are used to distinguish the services. This includes two different entities. Firstly, an entity that defines the code base of a service. Such a code base could be for example hosted on publicly accessible code repositories. Secondly, the namespace could include instantiated analytics service endpoints that define a running instance of an analytics service.

The attributes are listed in Table 2. Some attributes may be optional and may be dependent on if they are deployed services, or contain a library that may be deployed.

Benefits of a federated analytics service registry A service registry can publicize and improve end user access to data from different sources, by overcoming some of the challenges inherent in describing and surfacing document content and format. Publication, and discovery of information resources are enriched with metadata enabling the findability and reusability of a service supporting the FAIR principle. While describing the interfaces and allow for the instantiation or the reuse of already instantiated services we address the accessibility and interoperability. With respect to analytics as a service, end users should be able to find various analytic services and similar services without having to individually search multiple 'locations' or databases, each built to operate on its own, unique storage and retrieval constructs. Through these descriptions automated service integration can be provisioned while targeting not only the functionality involved, but also

allowing service level considerations to be addressed. Furthermore, such analytics services could provide significant security implications such as the protection of a database while only exposing a subset of approved analytics functions that are executed on the data sets. This includes partial and controlled sharing of data mashup that can be made available to the community and registered to make reuse easier without everyone having to replicate the service.

Table 2: Registry attributes

Name	Description	Service provider	Library provider
ID	UUID, globally unique	✓	✓
Name	Name of the service	✓	✓
Title	Human readable title	✓	✓
Public	True if Public (needs use case to delineate what pub private means)	✓	✓
Description	Human readable short Description of the Service	✓	✓
Endpoint	The endpoint of the service	✓	N/A
List of Input Parameters	A list of parameters to the service. The parameters have each the form of name, function, type, value, access. The type indicates the data type. The access indicates if the parameter is a data stream, database, single value/function, event. The function responds to a different function in case multiple are provided by the service.	✓	✓
List of Output Parameters style (event, stream, data) value timestamp	List of responses cast by the service. The responses have the form of function, name, type, value, access, timestamp. The type indicates the data type. The access indicates if the parameter is a data stream, database, single value/function, event. The function responds to a different function in case multiple are provided by the service.	✓	✓
Version	The version number or tag of the service	✓	✓
License	The license description	✓	✓
Protocol	REST	✓	✓
Modified	Modification Timestamp	✓	✓
Owner	Name of the distributing entity, organization or individual. It could be a vendor.	✓	O
Author	Contact details of the people or organization responsible for the service	O	✓
Tags	Human readable common tags that are used to identify the service that are associated with the service	O	O
Categories	A category that this service belongs to (NLB, Finance, ...)	O	O
Created	date and time on which the analytics service was instantiated or created instantiated	✓	✓
Heartbeat	State and timestamp of the last check when the service was active	O	N/A
Documentation	Link to a URL with detailed description of the service Source Link to the source code if available	O	O
Specification	Pointer to where specification schema is located	O	O
AdditionalMetadata	Pointer to where additional is located including the one here.	O	O
SLA	Serves level agreement including cost	O	O
CachingInterval	If a service is accessed a lot, the caching interval can be used to put a limitation on the Response with an LRU cache	O	N/A
DataIntegration	In case of big data the data cannot be provided as a parameter to the analysis function. Instead, we need to provide the data as endpoint. However, often tata may need to be uploaded or can be downloaded. In this case this field provides the upload and download endpoints and the protocol to access the data	O	O

✓= Required; O = Optional

Edit ⇒ section-federation.tex

6 Service Federation

This section discusses aspect of federated registries to locate and consume analytics services with persistent identifiers across organizations.

This is not the term is at this time in the document not properly used.

We use so far

(1) federation of catalog and registry

(2) federation of services stored in the registry and catalog

(3) federaion of services through high level services delegatiing to other services.

We will clarify this and appropriately address.

Edit ⇒ section-data

7 Data

Edit ⇒ section-package.tex

8 Package Analytic Algorithms as Service Payloads

Here we explore how to package analytic algorithms with well-defined input and output parameters as service payloads that can be reusable, deployable, and operational across multi-cores, CPUs, and GPU computing platforms.

Edit ⇒ section-interfaces.tex

9 Analytics Interfaces

10 Resource Management

Here we investigate and define a minimal set of resource management services and interfaces for application orchestration and workflow between processes.

Edit ⇒ section-security.tex

11 Security

11.1 Artifacts

function data logs / audit

11.2 Privacy

privacy input output function
asynchronous events, how does privacy apply batch functions streaming functions
data

11.3 Federation

NIST document on federation

11.4 Authentication

11.5 Authorization

11.6 Potential role of blockchain

Edit ⇒ section-outreach.tex

12 Outreach

TBD

References

References

[1] NIST Home page, Web Page.

[2] Chang W (2020) Case study title template. V1.

[3] Wo Chang DK (2020) Case study: Numeric weather prediction. V1 Available at https://bigdatawg.nist.gov/_uploadfiles/M0702_v1_2020102002.pdf.

[4] Kotevska O (2020) Case study title: Hvac recommendation. V1.

[5] NCAR (2022) Wrf users page, wps v4 geographical static data downloads page. [Online; accessed 7. Mar. 2022] Available at https://www2.mmm.ucar.edu/wrf/users/download/get_sources_wps_geog.html.

[6] National Centers for Environmental Prediction, National Weather Service, NOAA, US Department of Commerce (2000) Ncep fnl operational model global tropospheric analyses, continuing from july 1999. Available at https://doi.org/10.5065/D6M043C6.

Edit ⇒ section-glossary.tex

A Glossary

THis Glossary provides terms that aur used in this document. In addition we have provided a definitions in Section 5 to focus on the details of some terms and terminology used in this document specifically focussing on Analytics Servoces.

AAI: Authentication and Authorization Infrastructure. Facilitates single, virtualized identities (issued by the *user's home organization.*)

AARC:

ABAC: attribute based access control

ACL:

ACID: Atomicity, Consistency, Isolation, Durability.

Analytics: The systematic analysis of data, to uncover patterns and relationships between data, historical trends, and attempts at predictions of future behaviors and events.

Russell: The Authentication and Authorisation for Research and Collaboration project. I will write a descriptive sentence for it that you can add later. Also, below, ACL: access control list? I have a bunch other acronyms with descriptions that I can just send you as a list that you can choose from and add if you wish. Under Iaas, p is probably

Analytics management: A sub function within the [metadata] registry.

Analytics services azure cognitive, google analytics, aws [dozens], watson analytics... in contrast to ML frameworks like tensorflow, pytorch, caffe2, and in contrast to Programming libraries like python, scikit, shiny, or R Studio [??]

Analytics Workflow: The sequence of processes or tasks part of the analysis

API: Application Programming Interface

ASCII: American Standard Code for Information Interchange

BASE: Basically Available, Soft state, Eventual consistency Classification scheme per 11179, a container of the classifiers for all kinds of administered items including common data elements [CDE]s.

CIA: Confidentiality, Integrity, and Availability.

CLI: Command Line Interface.

Consumer: Ametadata consumer, per IHE, is responsible for the import of metadata created by the source. In the context of section A.3,

Container: See http://csrc.nist.gov/publications/drafts/800-180/sp800-180_draft.pdf Cloud Computing The practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer. See http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf.

CDE: Common data element = smallest meaningful data container in a given context.

DDC: data dictionary component. library of data elements that are used to establish common understanding of the meaning of coding systems.

Data element: Describes [or defines] the logical unit of data. Per 11179, the element refers to the structure of the data, distinct from a data instance.

Data element concept: the combination of an object class, and a related property.

DEX: Data element exchange = interoperability profile. Enables retrieval of extraction specifications for data elements which are defined in particular domains. "Options" including the cross enterprise doc sharing [XDS] doc type binding option and the cross community access [XCA] doc type binding option, extend basic DEX functionality, addressing interoperability with Secondary Data Usage[s]. Allowing secondary users to know if and where [data] is available when it is organized as a doc sharing environment, I.e. XDS, MPQ, XCA.

DevOps: A clipped compound [?] [portmanteau?] of software DEVelopment and information technology OPerationS improve

Deployment: The action of installing software on resources

DMTF: Distributed Management Task Force. A standards organization.

Extraction Specification: a map of data locations used as a guide for extracting data. SPARQL, SQL, and XPath scripts, aka mapping scripts, are examples of specifications for locating a data element in a particular content model.

FIM: federated identity management. A core component of AAI.

Federated database system: two definitions: 1. a system that maps multiple autonomous database systems using a combining scheme where one DB interface is provided for local / owner access to data, and another simpler interface is provided for guest access to non owner data. 2. a DBMS which is an element of a federated group, that allows members belonging to the same federated group, to access data residing in the DBMS.

HTTP: HyperText Transfer Protocol HTTPS HTTP Secure

Hybrid Cloud: See http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf.

IaaS: Infrastructure as a Service SaaS Software as a Service Implementation.

IGTF: Interoperable Global Trust Federation.

ITL: Information Technology Laboratory metadata data employed to annotate other data with descriptive information. russel: TBD

LDAP: Lightweight Directory Access Protocol. A directory/registry standard.

Metadata generator: A sub function within the repository Metadata Registry [MDR] a database that manages the semantics of data elements, and this case, provides discovery and analytics management services.

MRR: Metadata registry / repository = specialized DB of metadata which describe data constructs.

Microservice: Architecture Is an approach to build applications based on many smaller modular services. Each module supports a specific goal and uses a simple, well-defined interface to communicate with other sets of services.

NBDIF : TBD

NBD-PWG: NIST Big Data Public Working Group.

NBDRA: NIST Big Data Reference Architecture.

NBDRAI: NIST Big Data Reference Architecture Interface.

NIEM: National information exchange model = government wide standards based approach to exchanging information in the US.

NIST: National Institute of Standards and Technology.

OGF: Open Grid Forum.

OS: Operating System.

P2P: Peer to Peer.

PKI: Public Key Infrastructure. a security related certificate aka X.509.

Proxy: : TBD

Registry: : TBD

Registry, federated: : TBD

REST: REpresentational State Transfer Retrieval a transaction where a system returns a selection I.e. a list of data elements from a database, or in the scope of this document, a list of elements in a metadata registry.

SAML: Security assertion markup language. a security standard; web browser service that defines "syntax and semantics to exchange auth and auth data between security domains." Not compatible with other authentication protocols such as Secure socket?, OIDC, etc.

Serverless Computing: Serverless computing specifies the paradigm of function as a service (FaaS). It is a cloud computing code execution model in which a cloud provider manages the function deployment and utilization while clients can utilize them. The charge model is based on execution of the function rather than the cost to manage and host the VM or container.

Services: : TBD

Service registry: in the context of an SOA architecture, this registry is a network based directory that contains available services.

Software stack: A set of programs and services that are installed on a resource to support applications. Value domain the description of a permissible set of values for the property of a data element definition.

XACML eXtensible Access Control Markup Language. a security related standard developed by OASIS, circa 2005.

B Changelog

1. add appendix
2. add registry
3. add gitignore
4. split up in sections
5. add makefile
6. * removed duplication of wrf, * added file macro
7. russels changes
8. update russell.md: example text changes
9. initial russell overleaf commit check for russel.md
10. removed the use case line in ali.md
11. setup
12. add alis document
13. update use case images
14. add tree

15. Merge overleaf-2022-01-11-1526 into main
16. Updates from Overleaf
17. Delete realtime.md
18. add wrf
19. real time analysis
20. Merge overleaf-2021-12-14-1748 into main
21. Updates from Overleaf
22. Update README.md
23. Create README.md
24. Initial Overleaf Import

Todo list