Wittij | Consulting

IATI System Design

v3

iati International Aid Transparency Initiative

Approach

- Reviewed Strategy, Prior Assessments, and 20 existing mission and technical background documents
- Interviewed 40 IATI stakeholders* across publisher, data user, and developer communities
- Diagrammed 19 detailed "as is" architectures for 6 core systems: Registry, Web Site (incl SSOT), New Datastore, New Validator, Stats/Dashboard, d-portal
- Analyzed findings and drafted future state views for discussion
- Presented draft proposals to community participants, facilitated 3 workshops, processed discussions, and updated proposed design

* See Appendix for details.

Headlines

- Historically, focus has been on the standard and the technical community, but this is shifting to **ensuring that a full set of IATI capabilities is in place** to support the use of humanitarian and development data.
- To support the strategy and the community, IATI Architecture should be **transitioned from a siloed set of applications to a single core architecture** providing critical IATI capabilities necessary to gain insights: publishing, validating, storing, and providing quality humanitarian and development transparency data.
- This core architecture should be **production supported to a defined service level** and provide both an **excellent end user experience** and a **stable foundation for the developer community** (to enable them to do the same).
- Poor data quality inhibits use of the IATI data. Some of this can be addressed with **a technical validation and data architecture approach** but this will also require a **process focus and potential changes to the standard**.
- An **enhanced data portal** needs to be provided that provides easy access to reporting, visualization, and analytics of humanitarian and development data **based on a solid quality IATI data foundation**.
- There are multiple **options for enhancing publishing for smaller organizations**; a direction should be selected to shore up this focus area.

Interview Themes

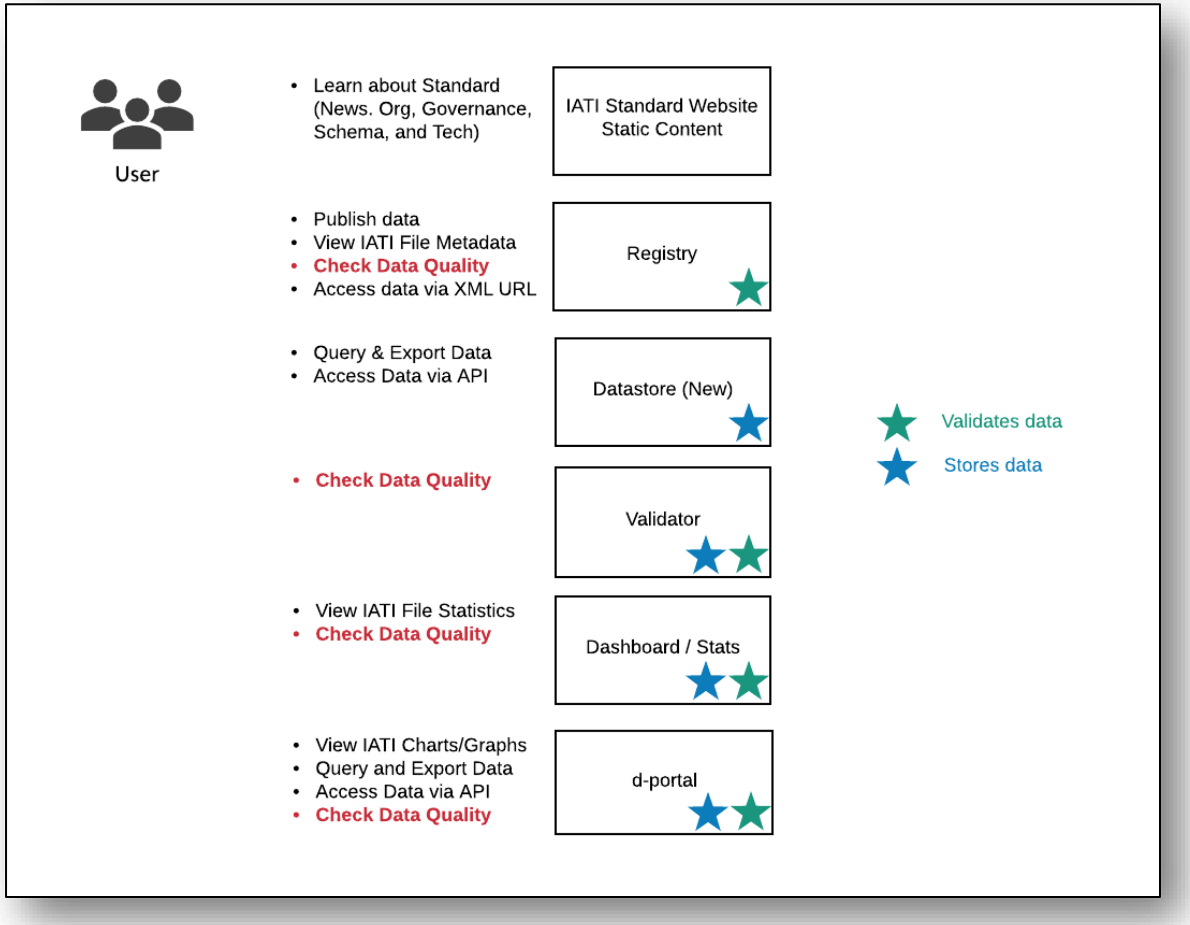
All the conversations went into the recommendations, but these were repeated themes.

Tension between wanting more data and better data (less rigid schema vs. more rigid schema).	Gap in strong technical architecture direction (roadmap and leadership) from IATI.	Lack of trust in IATI tools (support, availability) and in data (different from different sources). *Beta | Alpha*	Insufficient attention to ongoing care and feeding of assets once developed.
Challenge of aggregating IATI data due to the expanse and flexibility of the standard.	**IATI XML vs. Humanitarian and Development Information**	**IATI Passion** Reinforcement for the community of people that advocate strongly for the standard.	Concern around transparency of technical decisions and follow through on technical stocktakes.
Difficulty for country users (and others) who are not expert in the IATI standard to acquire useful information. *Complexity | Quality*	Critical role a data portal plays in the IATI tools landscape.	Too many places to look for functionality within the IATI tools (saving links to find things), yet no single place to go for some things (how does my data look?) *Tools by and for Developers*	Shift from "if you build it, they will come" focus on standard alone to a recognition that IATI needs to ensure availability of tools.

Current State - Disconnected

Multiple applications for users to perform similar activities creates user confusion. Overlap in internal capabilities of applications complicates the architecture.

User Actions	AIDStream*	Registry	Datastore	Validator	Dashboard/Stats	d-portal
Setup User & Org	🟦	🟦				
Publish Files	🟦	🟦				
Check Data Quality	🟦	🟦		🟦	🟦	🟦
Get/Export Data		🟦	🟦			🟦
Run Queries			🟦			🟦
System Processing						
Gets Data From Source		🟩	🟩		🟩	🟩
Validate Data	🟩	🟩		🟩	🟩	🟩
Stores IATI XML Files			🟩	🟩	🟩	🟩
Stores IATI Metatata	🟩	🟩		🟩		
Loads IATI Database			🟩	🟩	░	🟩
Provides Data API			🟩			🟩



User

- Learn about Standard (News. Org, Governance, Schema, and Tech)

IATI Standard Website Static Content

- Publish data
- View IATI File Metadata
- **Check Data Quality**
- Access data via XML URL

Registry ★

- Query & Export Data
- Access Data via API

Datastore (New) ★

- **Check Data Quality**

Validator ★★

- View IATI File Statistics
- **Check Data Quality**

Dashboard / Stats ★★

- View IATI Charts/Graphs
- Query and Export Data
- Access Data via API
- **Check Data Quality**

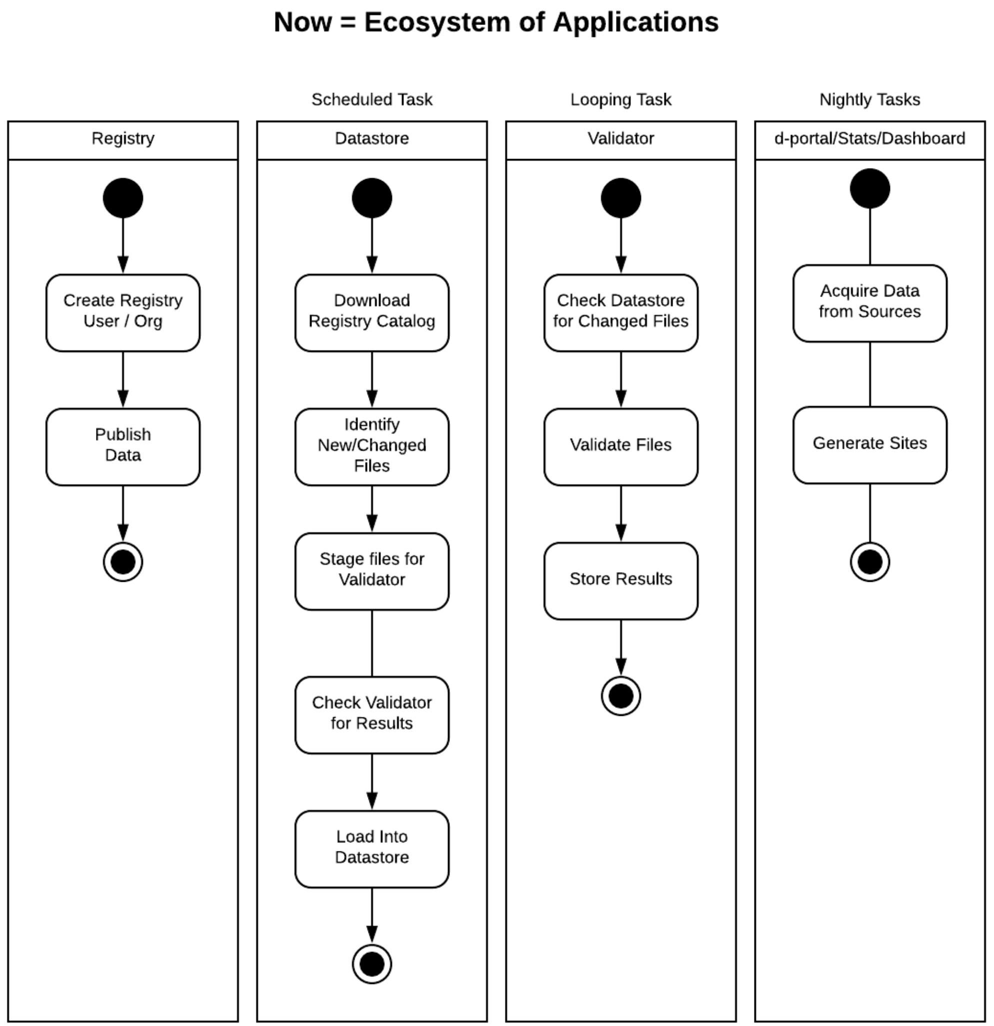
d-portal ★★

★ Validates data

★ Stores data

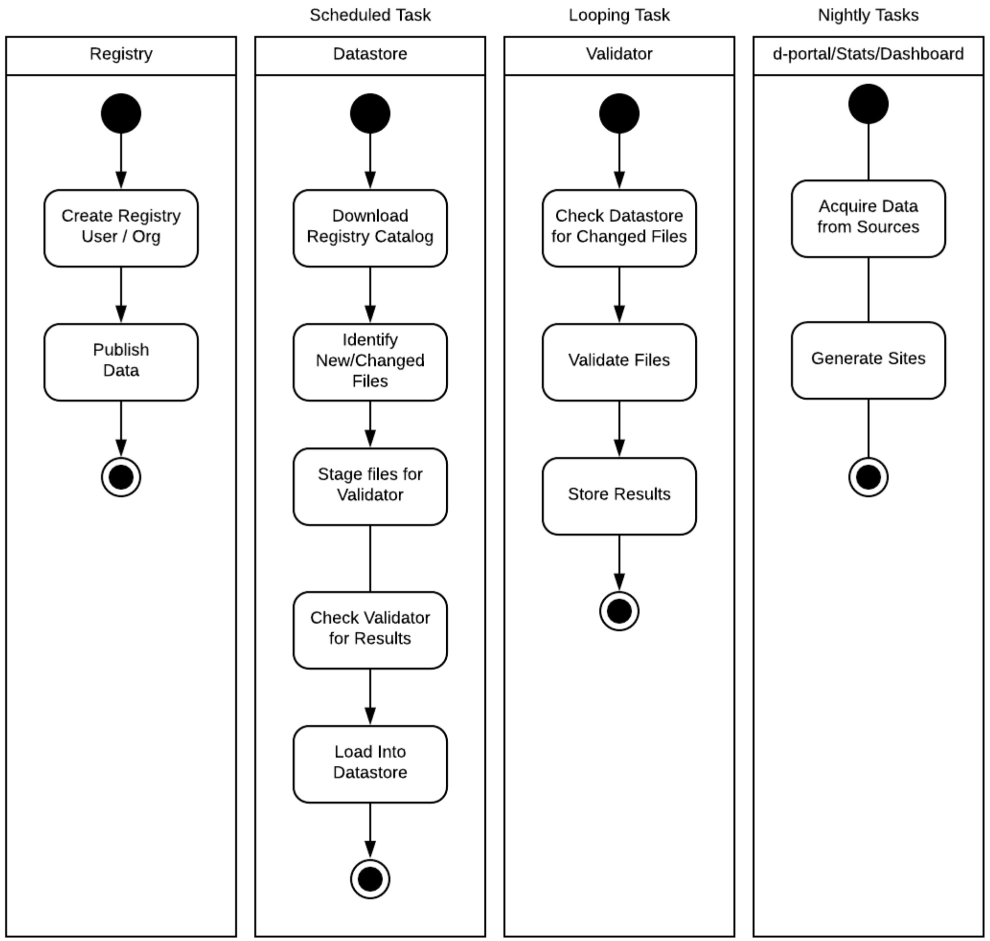
* AIDStream is not part of core IATI architecture, but a highly depended upon publishing platform for small publishers.

Integrated Architecture Concept

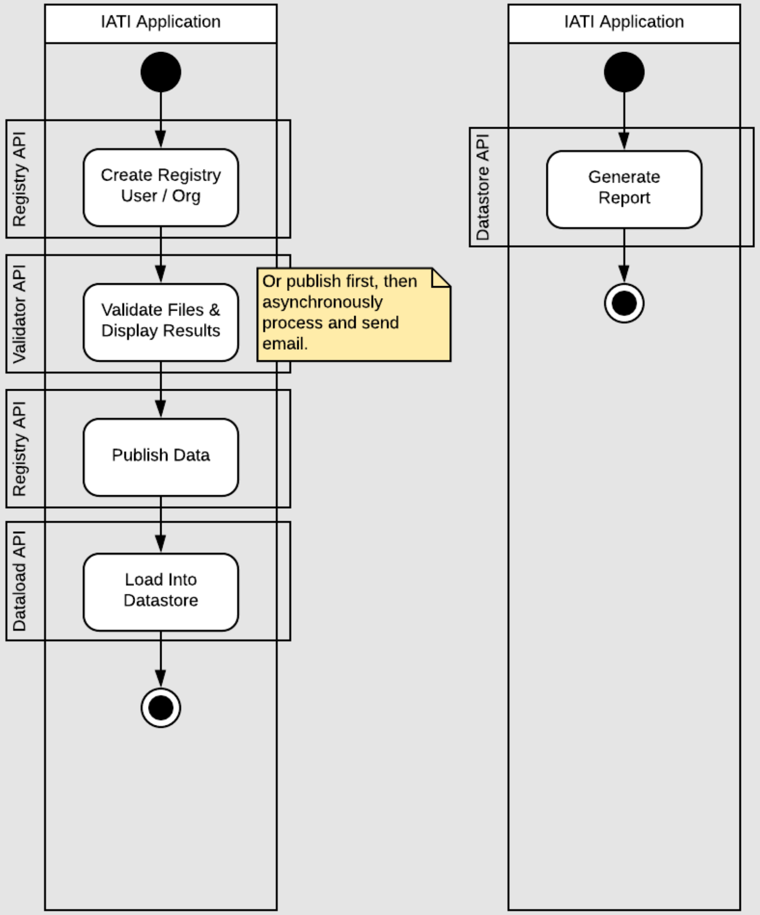

Now = Ecosystem of Applications
Scheduled Task
Looping Task
Nightly Tasks
Registry
Create Registry User / Org
Publish Data
Datastore
Download Registry Catalog
Identify New/Changed Files
Stage files for Validator
Check Validator for Results
Load Into Datastore
Validator
Check Datastore for Changed Files
Validate Files
Store Results
d-portal/Stats/Dashboard
Acquire Data from Sources
Generate Sites

Integrated Architecture Concept

Now = Ecosystem of Applications

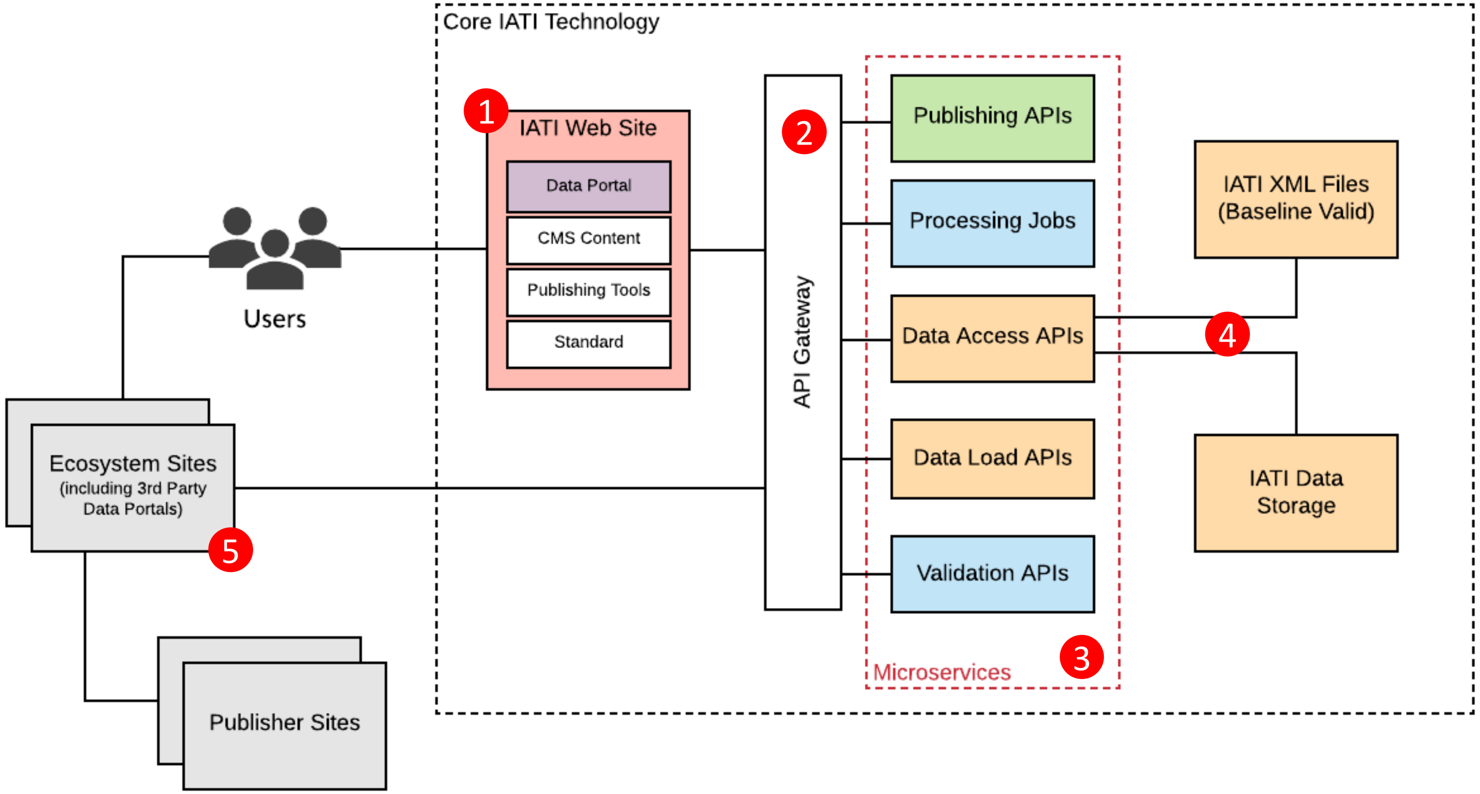

Scheduled Task
Looping Task
Nightly Tasks
Registry
Create Registry User / Org
Publish Data
Datastore
Download Registry Catalog
Identify New/Changed Files
Stage files for Validator
Check Validator for Results
Load Into Datastore
Validator
Check Datastore for Changed Files
Validate Files
Store Results
d-portal/Stats/Dashboard
Acquire Data from Sources
Generate Sites

Proposed Future State = Integrated Architecture



Or publish first, then asynchronously process and send email.

Integrated Architecture



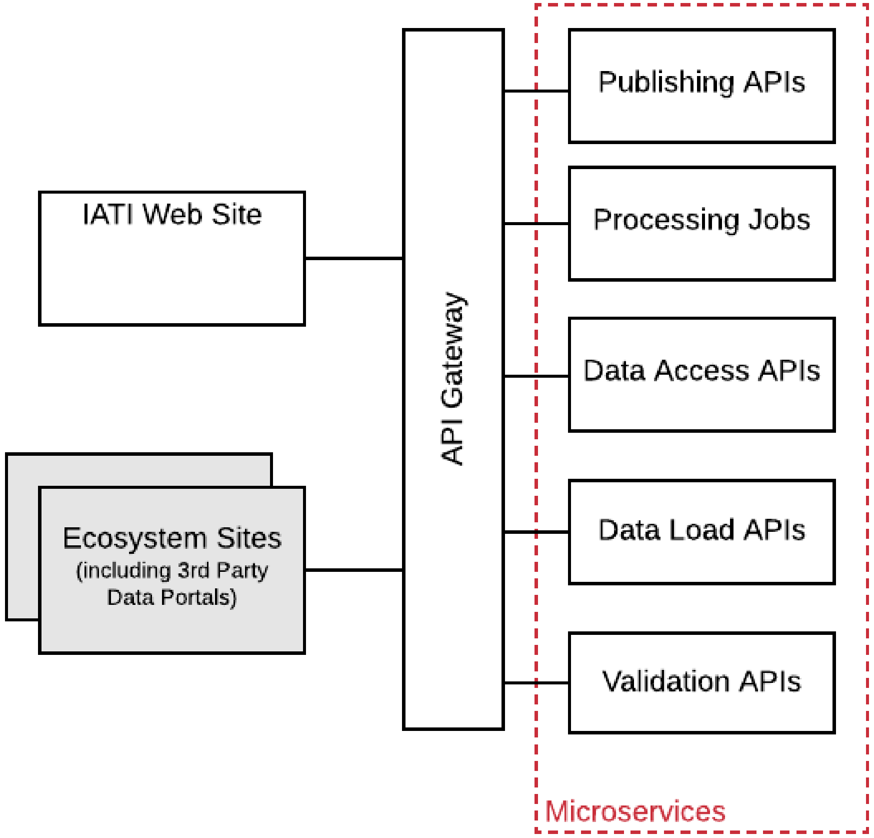
External	Web Site
Registry	d-portal
Validator	Stats
DataStore	Dashboard

Proposal.

1. Unified user experience prioritizing data portal and driven by persona and user journey analysis, with singular information architecture and design. Integrate:
    1. Improved Data Portal
    2. Enhanced Publishing Capabilities (see publishing slide)
2. API Gateway to abstract and manage API's across platforms, enable integrated architecture, and understand service usage. *Enables roadmap to unified architecture.*
3. IATI capabilities implemented as independently deployable microservices.
4. Expand IATI data services in both directions to include published XML files and curated reporting data (see data slide)
5. Continue to support an external ecosystem of sites and applications that integrate seamlessly with the IATI Core Technology.
    1. Document and support clear Software Development Kit (SDK) for extension points for the IATI technical community.
    2. Design data portal code such that framework (SDK) is available for publishers (or anyone) to create a custom data portal.

- Leverage existing platforms and migrate to unified architecture over time.

Microservices Architecture



Proposal.

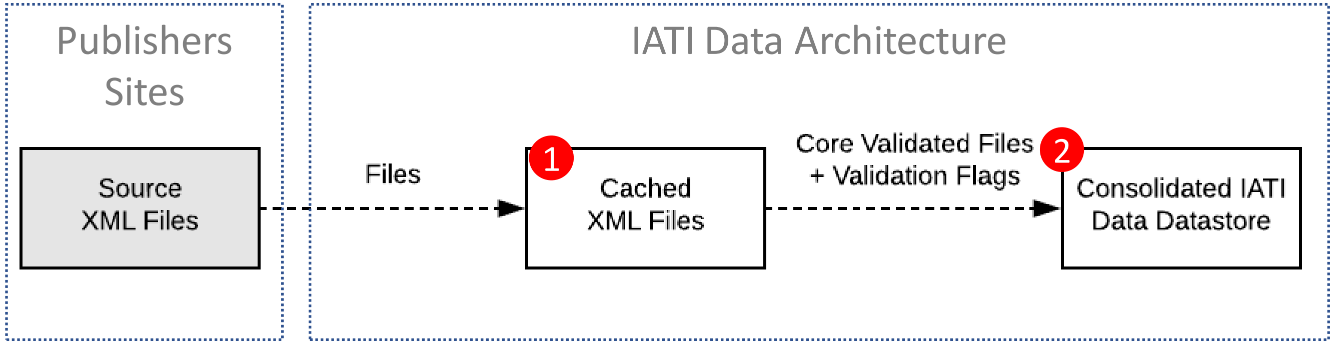
Initially, current APIs are publicly published through the API Gateway. Over time additional APIs are added per this proposed architecture and driven by User Personas and Stories, just like the unified User Interface. These microservices are loosely coupled, independent microservices.

1. **Publishing APIs.** Includes registry APIs, both current and proposed, as well as newer enhanced publishing APIs that enable start to finish publishing, as well as real-time fixes to any data stored or processed by the IATI system.
2. **Processing Jobs.** Any automated tasks related to ongoing caring and feeding of the core. *These would not be publicly available.*
3. **Data Access APIs.** Provide access to all levels of the data proposed by the architecture, from the current datastore services to the XML files and processed data (like stats).
4. **Data Load APIs.** Services for loading data into the data store. *These would not be publicly available but used by internal processes that need to update the datastore. External data loading would be done via Publishing APIs (which would need to authenticate publishers).*
5. **Validation APIs.** Services to apply validation sets to data, per the validation recommendation and based on the existing validator.

This is not intended to be a comprehensive list. Any features added to the IATI system should be implemented using this microservices approach.

Internal access to data would also always be through APIs to abstract access to the raw data repositories and keep them loosely coupled.

Proposed Data Architecture

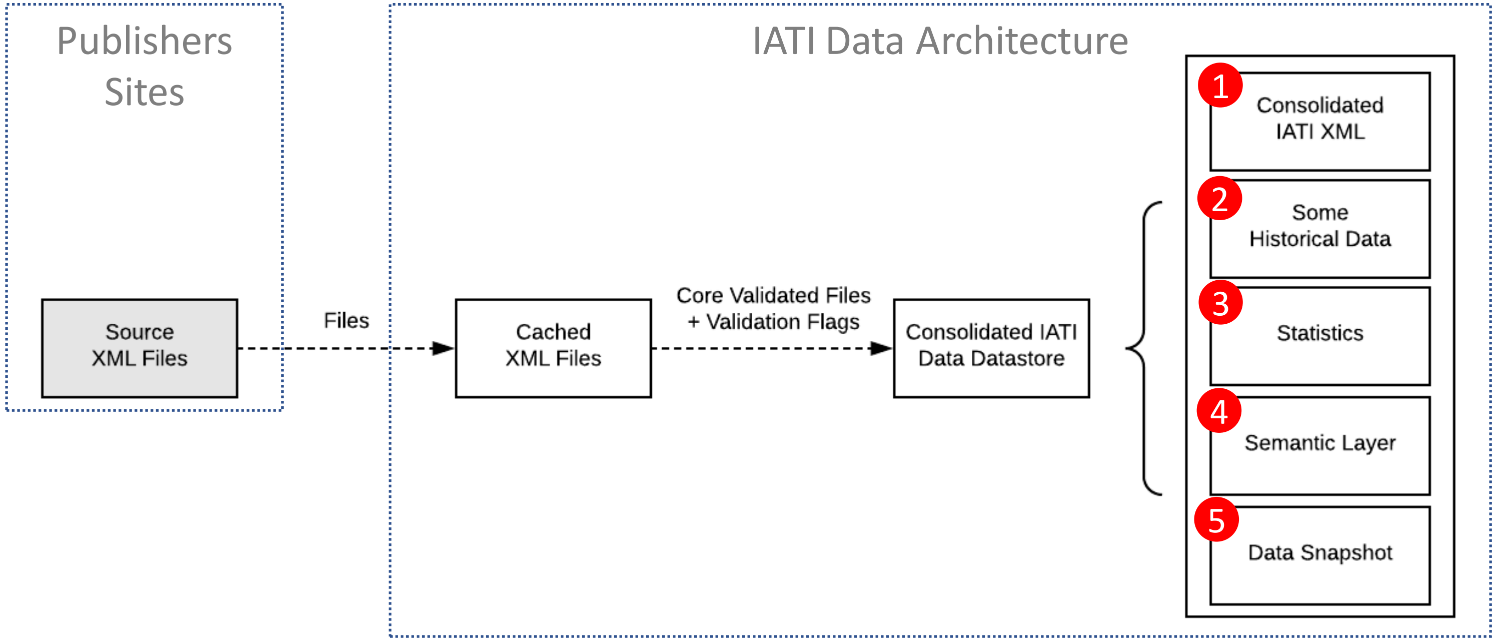


Proposal.

1. IATI stores copy of raw, published XML and provides access point. While a shift from historical direction of referring users to publishers, insulates consumers from publisher site issues. Publishers still own the data; this just provides access to the data via IATI data architecture before it is loaded into the Datastore, where it is possible not all file pass baseline validate and are loaded.
2. Files that meet baseline core validation (see validation slide) are loaded into IATI consolidated dataset (which is what new Datastore already does).

Proposed Data Architecture

Datastore Zoom In



Proposal.

1. Consolidated XML Data. Combined set of all IATI data submitted by publishers; searchable and filterable. Stored in a data model based on the IATI XML schema with no transformations applied to the data.

2. Some Historical Data. Specific data identified as being important for trending is copied periodically to history tables.

3. Statistics. Precalculated statistics, trend data, etc. Like current statistics calculated today.

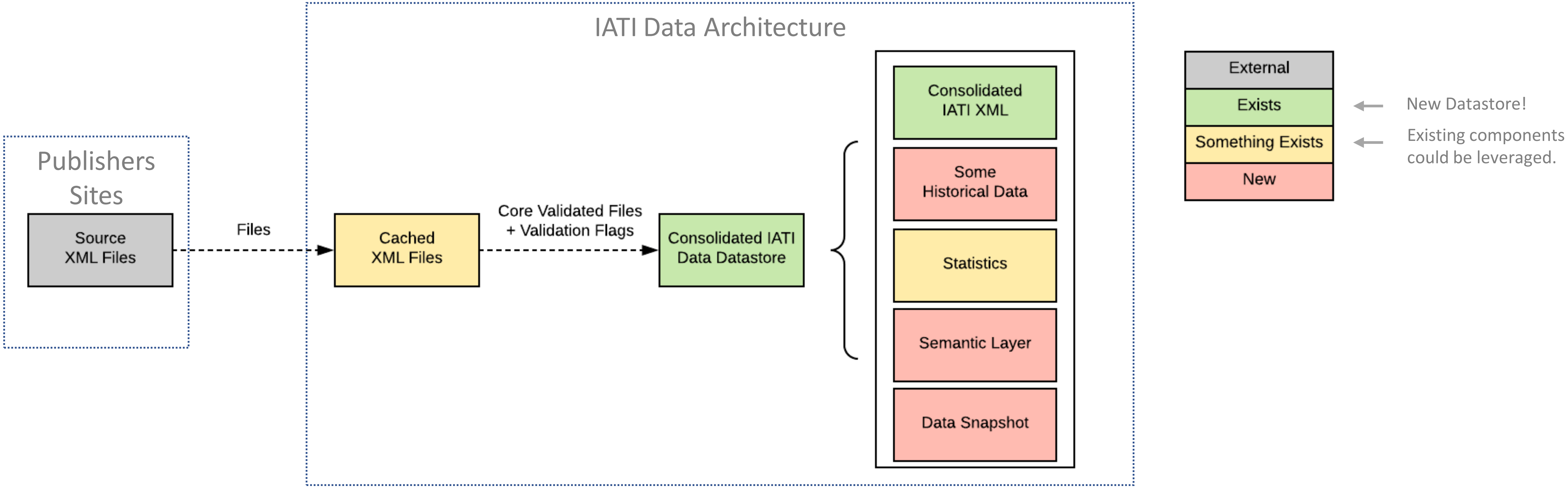
4. Semantic Layer. New curated data set that provides an "opinionated" version of the data that is easier to use than users interested in humanitarian and development data, but not in IATI XML. Level of transformations applied to be determined through user persona and story analysis, but could include transformations like: field format and selection, calculating multiple columns into a single column, mapping taxonomies, using understandable terms vs. codes, currency conversions, etc. Would be used for many (but probably not all) views on data portal.

5. Data Snapshot. If needed to support consistency in data portal views, a rolling, time specific snapshot (eg. Daily) could be made available as well to provide an unchanging dataset for analysis during the snapshot period.

Proposed Data Architecture

Leveraging existing assets

Proposal.
Leverage what has already been created.

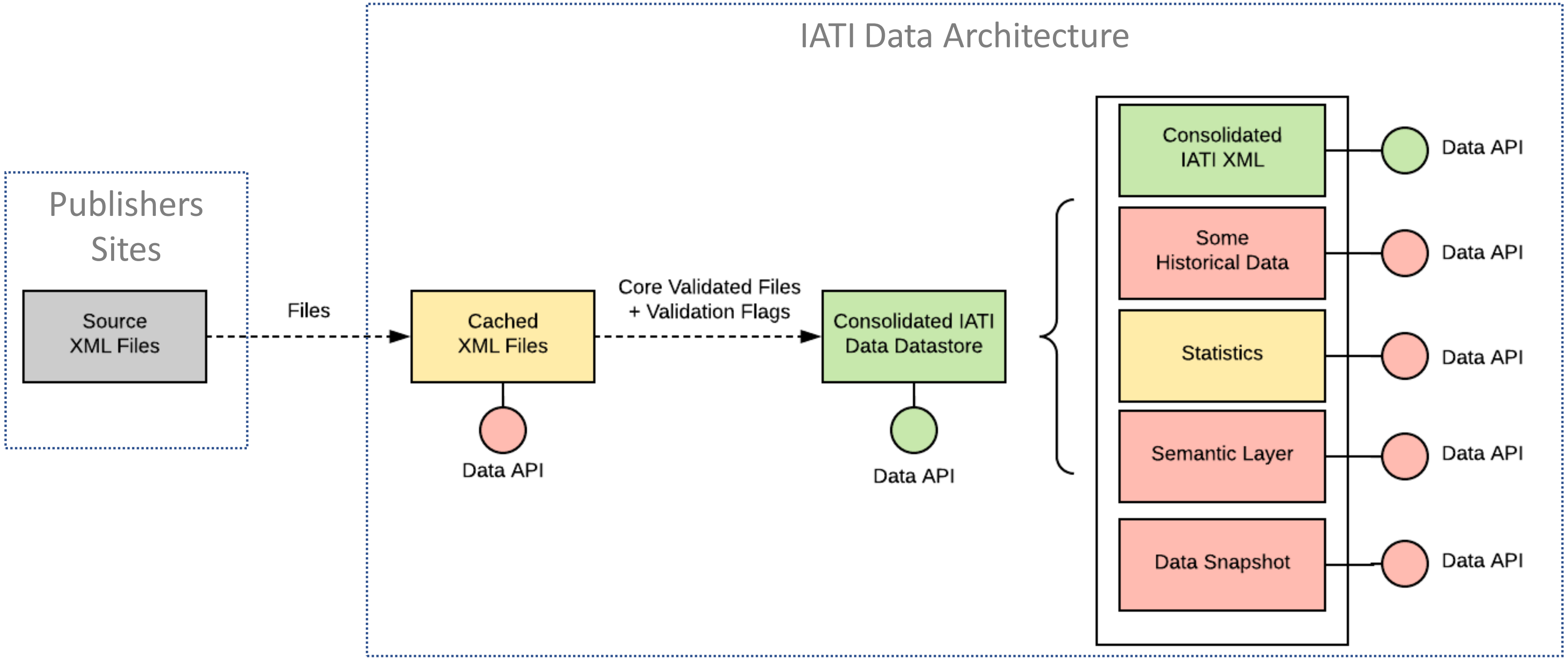


New Datastore!

Existing components could be leveraged.

Proposed Data Architecture

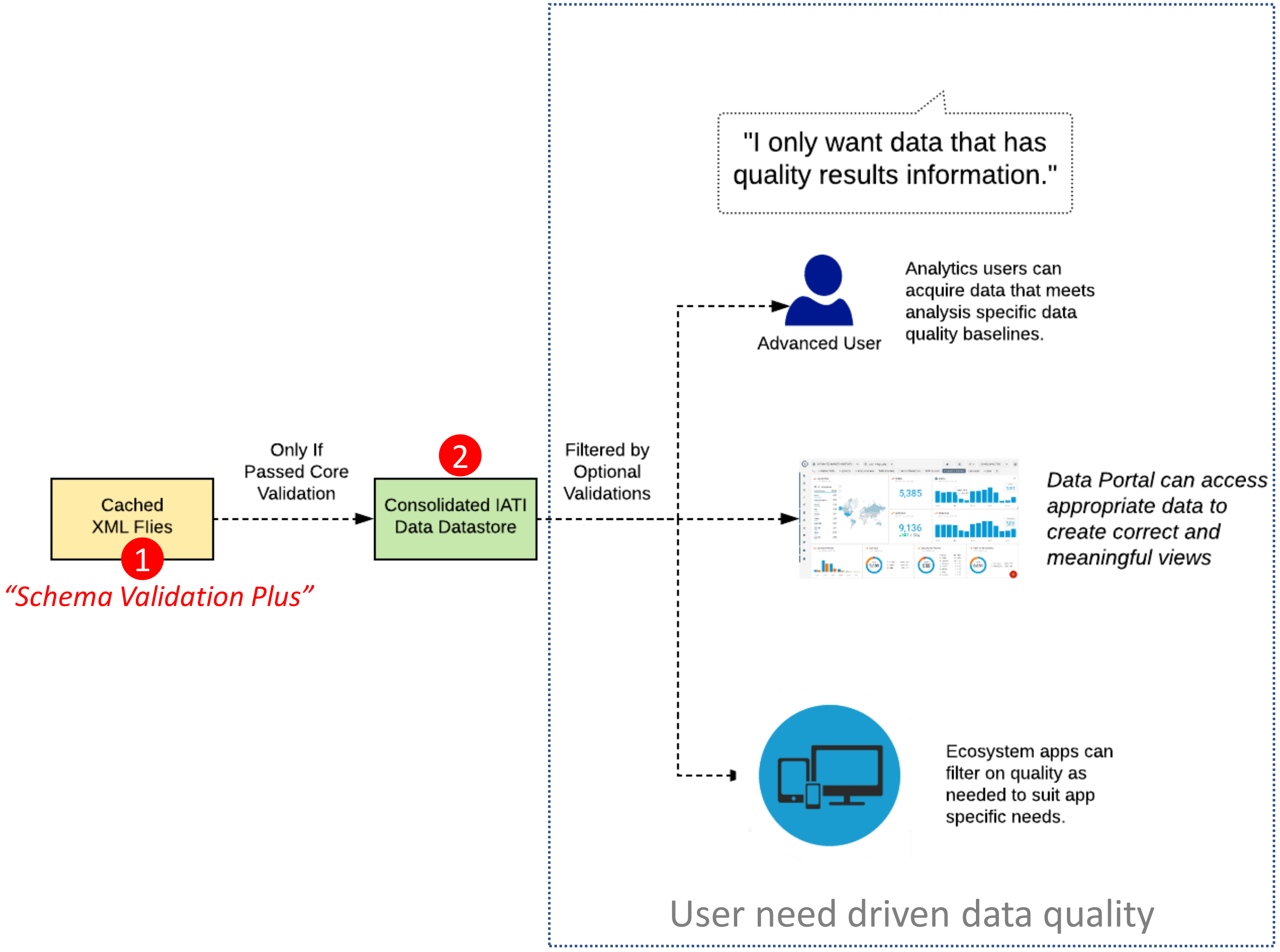
API Enabled



Proposal.

All layers of the data architecture are API enabled, so external ecosystem applications can access the data at any level they need.

Proposed Validation Architecture



Cached XML Flies

1

“Schema Validation Plus”

Only If Passed Core Validation

2

Consolidated IATI Data Datastore

Filtered by Optional Validations

"I only want data that has quality results information."

Advanced User

Analytics users can acquire data that meets analysis specific data quality baselines.

Data Portal can access appropriate data to create correct and meaningful views

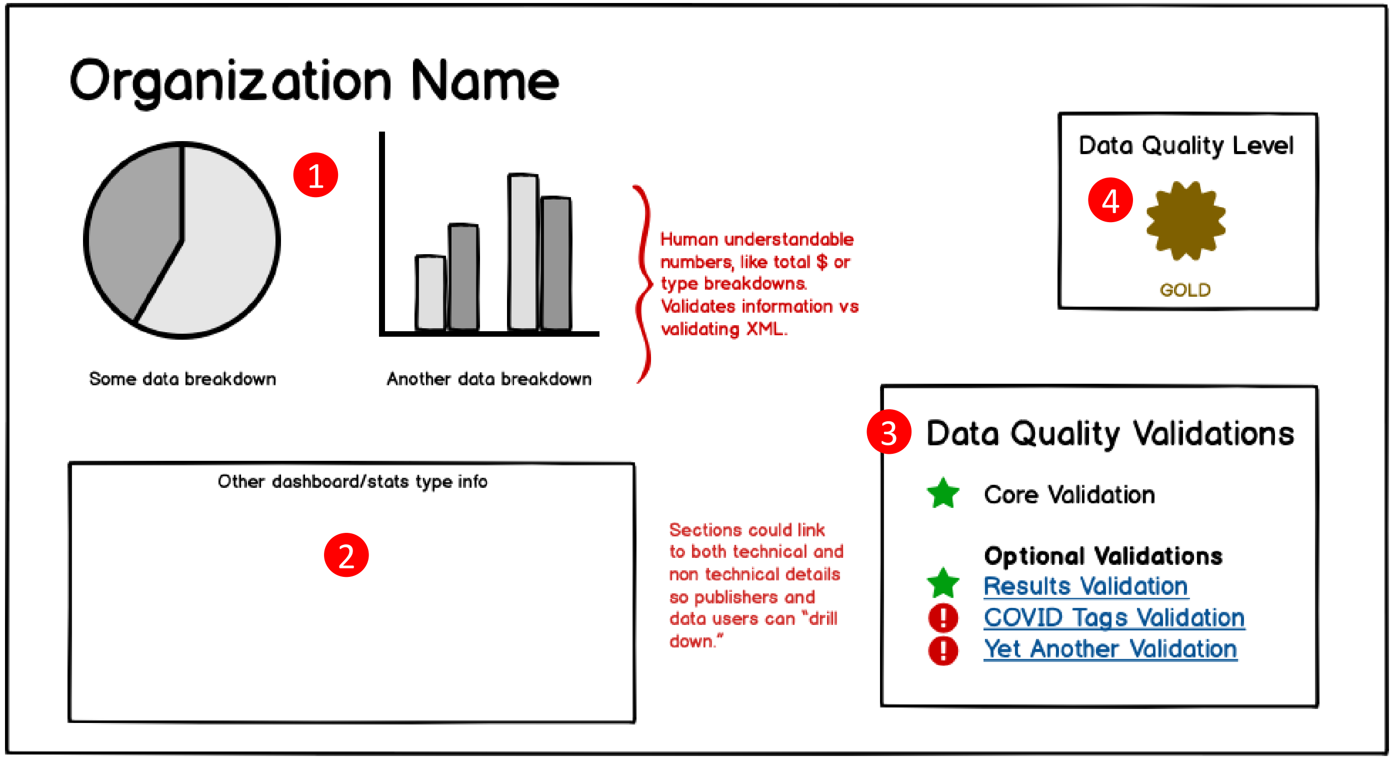
Ecosystem apps can filter on quality as needed to suit app specific needs.

User need driven data quality

Proposal.
Introduce optional validation levels to support both quality and flexibility in the data published:

1. Tighten the standard for a core IATI validation and **do not load** data that does not meet this baseline into the aggregated dataset. It would still be available in raw XML form. *This may be more rigorous than the current XML schema validation alone.*
2. Run additional optional validations based on context. The pass/fail result would be stored with the data, so users could access quality data based on their data needs. *Could validate anything for which rules can be applied to the data, like “are results published,” or “has sufficient history been provided.”*

Preview Page Concept



Organization Name

Human understandable numbers, like total $ or type breakdowns. Validates information vs validating XML.

Sections could link to both technical and non technical details so publishers and data users can "drill down."

Mockup is solely to illustrate a concept. Actual screen content and information would need to be designed as part of implementation.

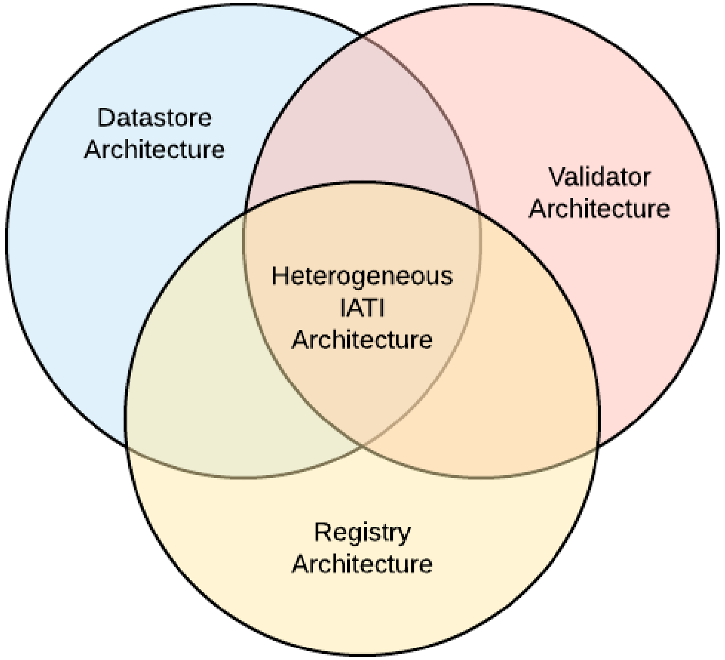
Proposal.

Enrich and consolidate numerous user interfaces for checking state of IATI data to a single Preview page:

1. Provide data summary to enable user with zero knowledge of IATI Standard to assess immediately if the data looks correct. Assists with validation of data content vs. technical data quality.
2. Integrate most useful dashboard/stats info for validating data and allow "click through" to details.
3. Summarize validation results using optional validations, which enables both quality and flexibility in the data:
4. Optionally, roll validations up into a quality score.

Consolidate Technical Architectures

Best practices at the component level, makes best practices at the IATI level more difficult (frameworks, architecture, development languages).



Proposal.

1. Converge on homogeneous architecture by creating a core set of non-functional requirements (NFR).
2. Does not need to be "big bang," can use strategy of encapsulating components to work in the target first, then converge on consolidated architecture using NFRs as other work occurs.
3. *It is always okay to decide that a component requires a different internal architecture; but you never want it to be different by accident.*

- **Functional** requirements are **what** a solution needs to do to support the user needs.
- **Non-Functional Requirements (NFRs)** are **how** a solution needs to be <u>developed</u>, <u>installed</u>, or <u>operated</u> to work optimally in the target environment.

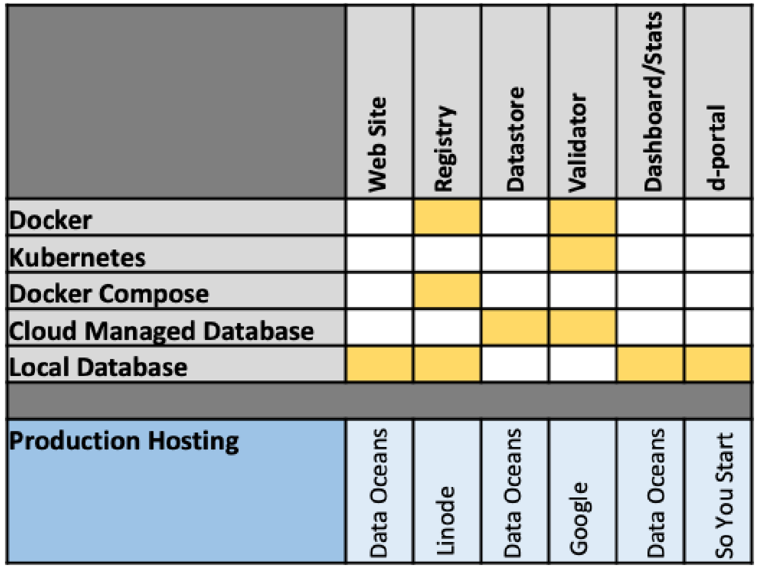
Wikipedia has a huge list of NFR categories if you want to see examples, but they are things like: Reliability, Availability, Supportability.

The ISO SQuaRE model provides a good taxonomy for ensuring coverage.

Consolidate Deployment Architectures

Similarly, a variety of deployment approaches and production support operations are in place across the footprint.

Production Environments



	Web Site	Registry	Datastore	Validator	Dashboard/Stats	d-portal
Docker		🟨		🟨		
Kubernetes				🟨		
Docker Compose		🟨				
Cloud Managed Database			🟨	🟨		
Local Database	🟨	🟨			🟨	🟨
Production Hosting	Data Oceans	Linode	Data Oceans	Google	Data Oceans	So You Start

Proposal.

1. Consolidate hosting to single provider and consistent deployment architecture.
    1. Ensure architecture is designed to meet defined availability, fault-tolerance, and recovery objectives.
    2. Evaluate single production support methodology with defined service level agreements.
2. Attempt to use deployment approaches that support elastic scaling and migrate toward an auto-scaling environment.
3. Look for opportunities to automate production deployment and support tasks.

This allows optimization of the production environment to best meet the computing resource needs of the systems.

Small Publishing Platform

Challenge.
An easy method to publish to IATI for smaller organizations without the resources to create something more sophisticated has become a critical need for IATI publishing currently filled predominately by a single vendor. Given its importance, go forward options should be evaluated.

Status Quo Poses Some Risks

1. A vendor (Young Innovations) supports the predominant platform in the marketplace as a goodwill and brand recognition activity. A pay model has recently been introduced, but there is risk that a strategy shift on the vendors part could leave the platform unsupported.
2. IATI has little control over how the platform operates and therefore limited ability to influence what data is published for a large body of small publishers.

Proposal.
Options Analysis should be performed to select the best go forward strategy*.

Some potential options to consider:

1. Enable a more full-featured publishing API to potentially encourage more market participation. (A good idea regardless of what other direction is taken and could be used by AIDStream and other publishing tools.)
2. Enhance partnership in AIDStream. This could take the form of investment or of partnering to support the underlying Open Source platform while still enabling AIDStream to have a premium add-on model for that platform. Investment could drive features necessary to enhance platform in line with the finding and proposal of this presentation.
3. Implement a new publishing UI in the integrated web site with workflows to service both large and small publishers.

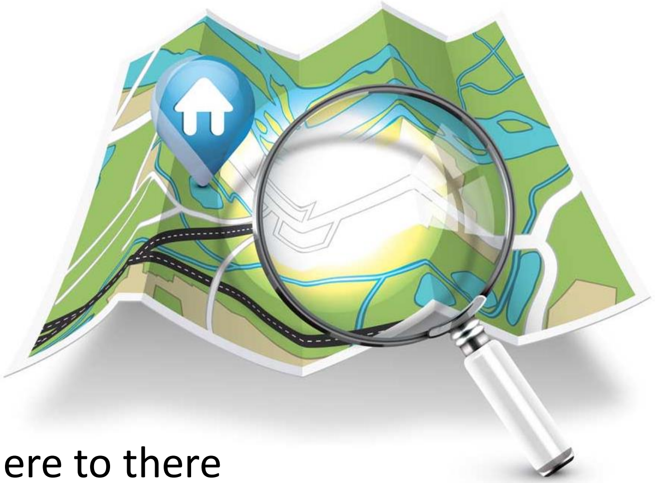
* See “Open Architecture Decisions” in Workshop Outputs section.

General Recommendations

- Continue establishing roles and role priorities focused on:
    - Data quality
    - Supporting and promoting use of the data by people interested in humanitarian and development information and not IATI XML
- Define, publish, and maintain :
    - A public product roadmap
    - A public technology architecture roadmap
- Clarify and strengthen architecture leadership role to work with the product owner, set technical direction for platform, and lead collaborations with technical community.
- Host marketplace for IATI tools and extensions.

SIDEBAR: What is a Roadmap?

- A **product roadmap** is a high-level visual summary that maps out the vision and direction of your **product** offering over time.
  - The publicly published version is typically high-level features
  - The timelines are more accurate in the near term (~1yr), and becomes more aspirational the further it goes into the future
  - It summarizes in what features investments (people and funds) will be made and in what order



- A **technical architecture roadmap** provides a summary for the non-functional aspects of the product, and includes:
  - An architecture diagram of the aspirational vision for the architecture
  - One or more transitional architecture diagrams plotting the path from here to there
  - A timeline (with same accuracy as above) for what technical capabilities will be invested in and when
  - It can also include technical guardrails to guide the implementation like technology principles, standards, and NFRs.

Some Process Recommendations

- Document and publish processes related to product management, architecture design, development, and user acceptance testing.
- Enhance Terms of Reference (TOR)
  - Specify the “how” and not just the “what” with Non-Functional Requirements
  - Include explicit user acceptance testing for both functional and non-functional (like performance) aspects of the solution.
  - If distributed production support continues, ensure production support Support Level Agreements are in TOR, if production support is consolidated, ensure production handoff needs are covered.
  - Don’t limit market for outsourcing by requiring IATI expertise unless it truly is critical for a given TOR. Systems are systems and data is data. In some areas you will require XML expertise, but not even that in all areas.

Priorities

With workshop vote counts. Attendees had 3 votes each. 1 attendee abstained on 2 votes.

7 Publishing User Interface (depends on decision)	4 Validation Sets & Data-loading Changes	1 IATI XML File Storage & Access	Consolidate Hosting[4]
7 API Gateway / Service Catalog[2]	4 Centralized Production Support with SLAs	1 Tighten the Standard (Standard +)	Migrate Stats/ Dashboard to Datastore and Portal
7 Define and Publish a Roadmap	4 Rearchitect for Stability and Fault Tolerance	1 Storing History	New User Experience[1]
6 Data Portal[3]	1 New "Preview Page"	Normalize Application Deployment Architectures[4]	

1. New User Experience is foundational and includes information architecture, design language, technical design, and migration of all existing functionality that survives the re-design. Other modules will be added as they are migrated to new architecture.

2. All changes going forward should include work to fit component into target architecture. Some foundational elements (API Gateway) should be part of whatever the top priority is and include publishing service interfaces "as is" through the gateway.

3. Data Portal is particularly large with significant design work (functional requirements and design, information needs, data model design, datastore implementation, and user experience)

4. Or… keep each as is until some other change opens it up for surgery.

Appendix – Stakeholders

Interviews

Internal Stakeholders	External Stakeholders	Other IATI experts – working with other data standards	Tool Providers
IATI Secretariat Annelise Parr Anna Witson Sarah McDuff Wendy Thomas Kate Hughes Petya Kangalova Alex Miller Alex Lydiate John Askew Rohini Simnodyal Bill Anderson*	Rory Scott (DFID) John Adams (DFID) Leo Stolk (Oxfam Novib) Nick Imboden (OCHA FTS) Sean Foo (OCHA FTS) Andie Vaughn (USAID) Ryan Boles (USAID) Sarah Scholz (USAID) Darren Enterline (USAID)	David Megginson (Humanitarian Data Exchange) Wendy Rogers (DI/Grand Bargain) Mark Brough (DI/Grand Bargain)	Siem Vassen (Zimmerman) Lu Min Han (Zimmerman) Rolf Kleef (D4D – Validator) Swaroop S Bhat (Derilinx) Pierre Baviera (Derilinx) Kriss Blank (Wet Genes) Shi Blank (Wet Genes) Steven Flower (Open Data Services) Anthony Gonzalez (Akvo) Geert Soet (Akvo) Abdoulaye Semdé (Akvo) Emeline Bereziat (Akvo) Bibhusan Bista (Young Innovations) Anjesh Tuladhar (Young Innovations) Reid Porter (DevResults) Matt Geddes (Somalia AIMS) Anders Hodstee (Myanmar AIMS) Joshua Powell (Development Gateway)

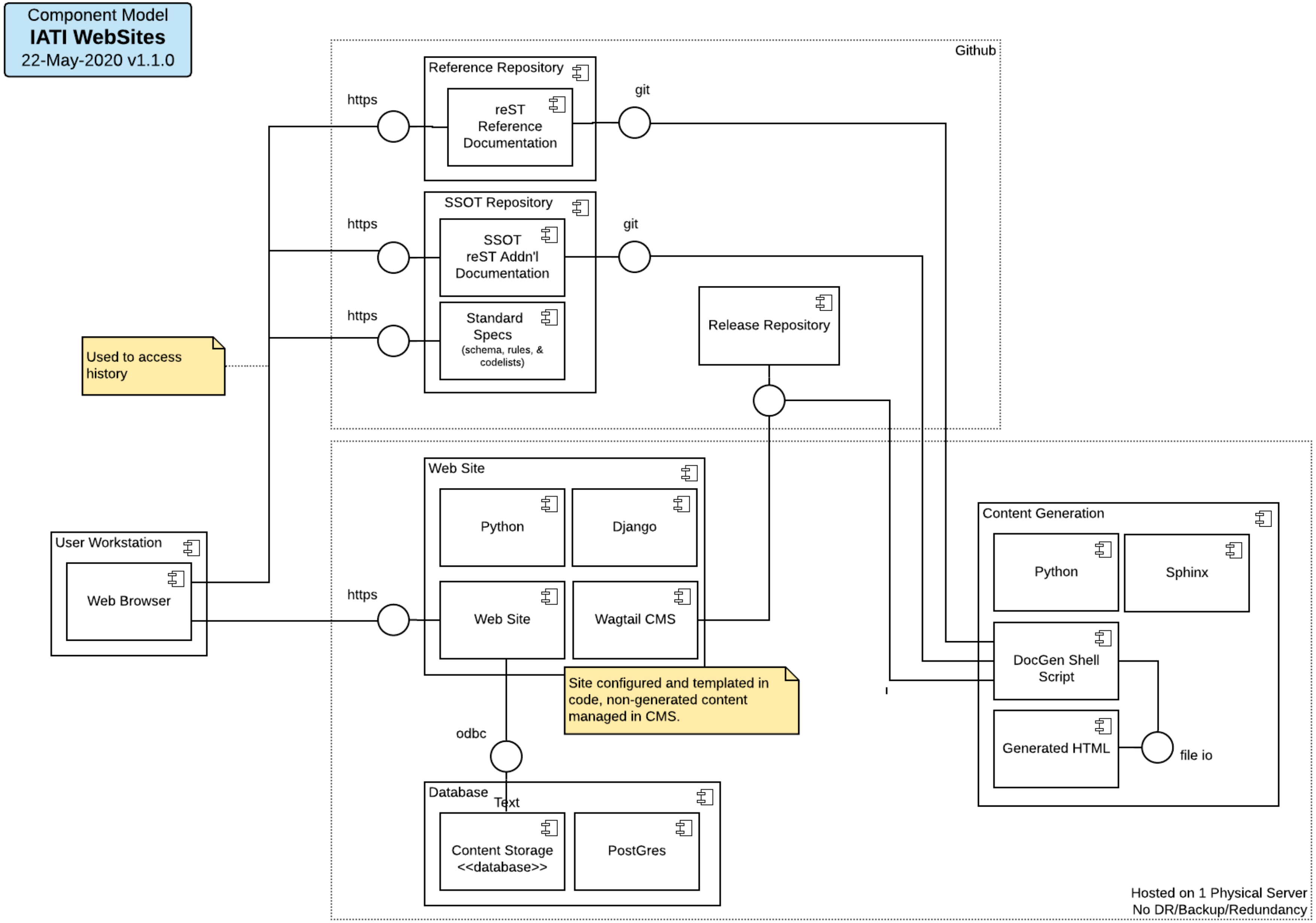
* Former

Workshops (Invitees listed, Attendees **bolded**)

Supplier Group	Group A	Group B
Siem Vassen, Pierre Baviera, Kris and Shi Blank, Bibusan Bista, Anders Hofstee, Steven Flower, Taryn Davies, Rolf Kleef, Reid Porter, IATI Secretariat **Wendy Thomas, Annelise Parr, Kate Hughes, Sarah McDuff** Facilitator **Dan Hughes**	**Mark Brough, Ole Jabcob Hjolland, Leo Stolk, David Megginson, Gary Forster, Michelle Levesque, Henry Asor,** Nick Imboden, Kalilu Totangi, Innocent Mugabe IATI Secretariat **Annelise Parr, Anna Whitson, Alex Lydiate, Kate Hughes, Sarah McDuff** Facilitator **Dan Hughes**	**Charlie Martial, Darren Enterline, Herman Van Loon, Edward Chionia, Rory Scott, Matt Geddes, Abdul Riza, Sarah Johns, Melinda Cuzner, Sohir Debbiche,** Tim Davis IATI Secretariat **Annelise Parr, Alex Miller, Kate Hughes, Petya Kangalova** Facilitator **Dan Hughes**

Appendix – Current State Diagrams

WebSites



International Aid Transparency Initiative

WITTIJ CONSULTING

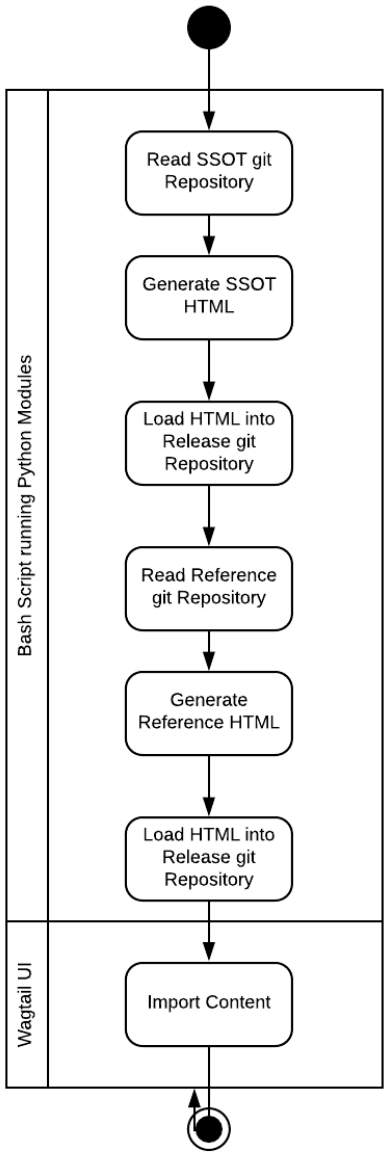
WebSites

Activity Diagrams
Web Site
22-May-2020 v1.1.0

Switch to Wagtail import approach; is currently awiting Community approval. Until then, Generated HTML is hosted on the same server using Apache httpd.

Content Generation

Run "on demand" when content changes


Bash Script running Python Modules
Read SSOT git Repository
Generate SSOT HTML
Load HTML into Release git Repository
Read Reference git Repository
Generate Reference HTML
Load HTML into Release git Repository
Wagtail UI
Import Content

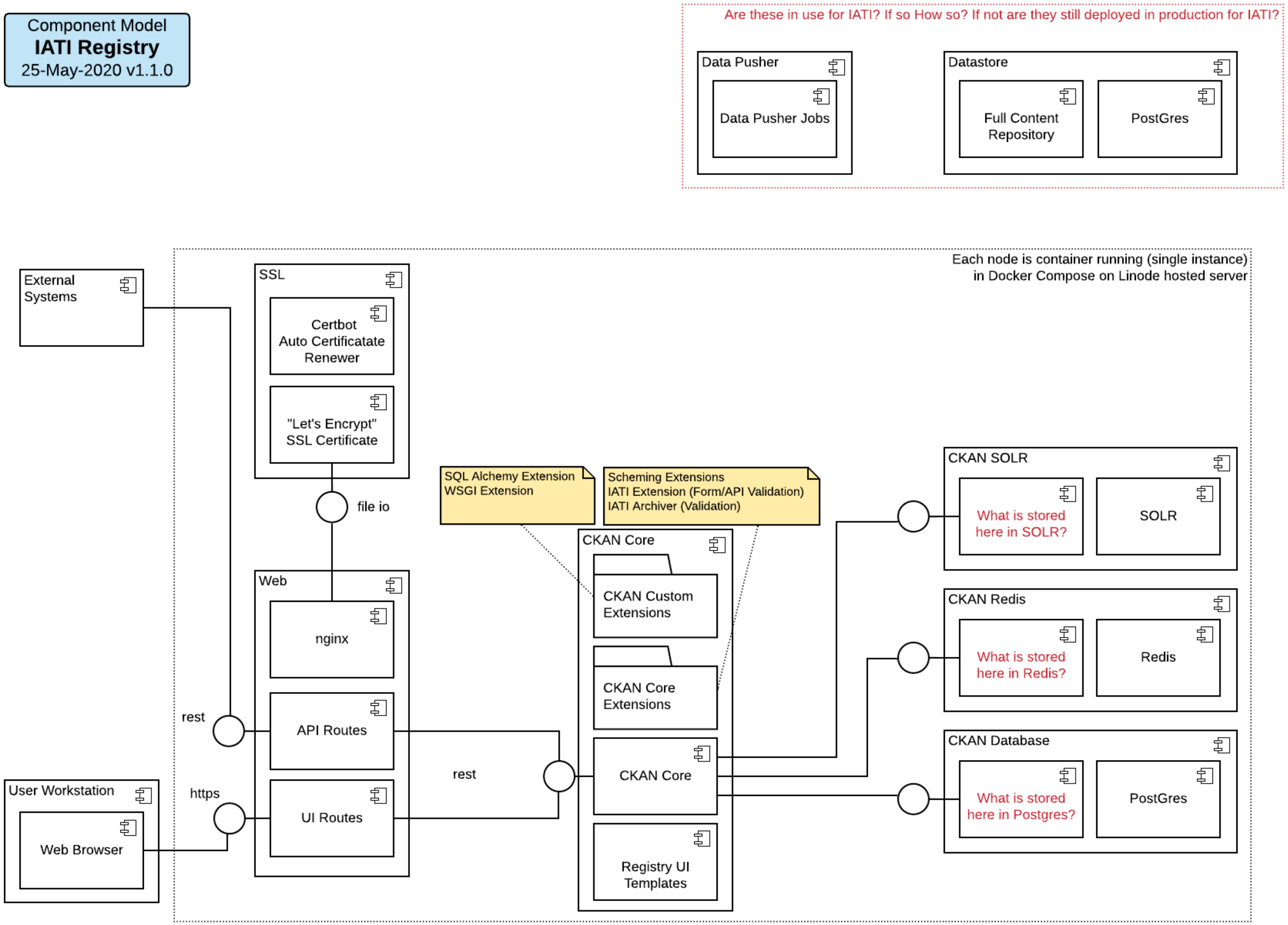
Can import SSOT and Reference content independently.

iati International Aid Transparency Initiative

W WITTIJ CONSULTING

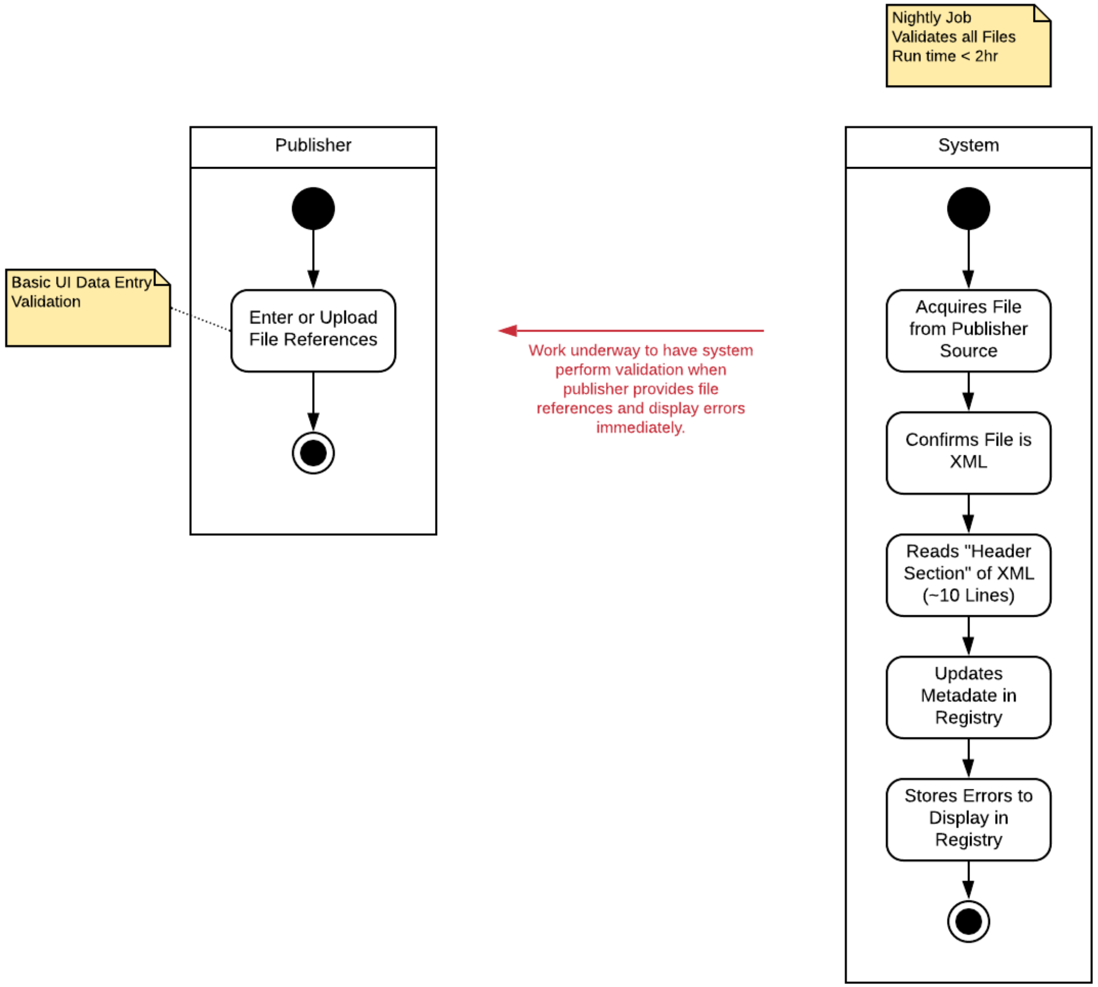
Registry

Component Model
IATI Registry
25-May-2020 v1.1.0


Are these in use for IATI? If so How so? If not are they still deployed in production for IATI?
Data Pusher
Data Pusher Jobs
Datastore
Full Content Repository
PostGres
Each node is container running (single instance) in Docker Compose on Linode hosted server
External Systems
SSL
Certbot Auto Certificatate Renewer
"Let's Encrypt" SSL Certificate
file io
Web
nginx
rest
API Routes
https
UI Routes
User Workstation
Web Browser
rest
SQL Alchemy Extension
WSGI Extension
Scheming Extensions
IATI Extension (Form/API Validation)
IATI Archiver (Validation)
CKAN Core
CKAN Custom Extensions
CKAN Core Extensions
CKAN Core
Registry UI Templates
CKAN SOLR
What is stored here in SOLR?
SOLR
CKAN Redis
What is stored here in Redis?
Redis
CKAN Database
What is stored here in Postgres?
PostGres

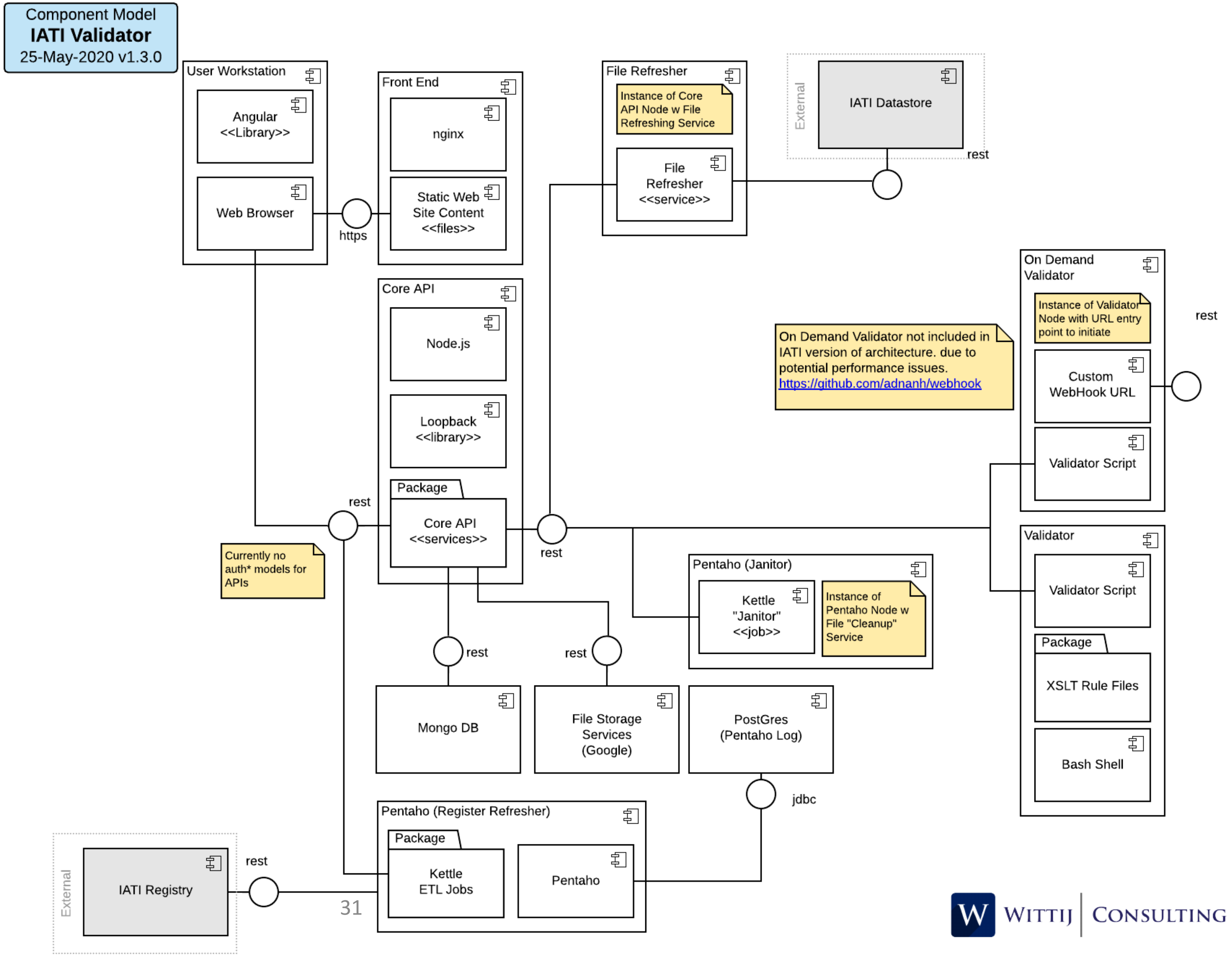
Registry

Activity Diagrams
IATI Registry
25-May-2020 v1.0.0


Nightly Job
Validates all Files
Run time < 2hr
Publisher
Basic UI Data Entry Validation
Enter or Upload File References
System
Acquires File from Publisher Source
Confirms File is XML
Reads "Header Section" of XML (~10 Lines)
Updates Metadate in Registry
Stores Errors to Display in Registry

Work underway to have system perform validation when publisher provides file references and display errors immediately.

Validator



Component Model
IATI Validator
25-May-2020 v1.3.0

Instance of Core API Node w File Refreshing Service

On Demand Validator not included in IATI version of architecture. due to potential performance issues. https://github.com/adnanh/webhook

Instance of Validator Node with URL entry point to initiate

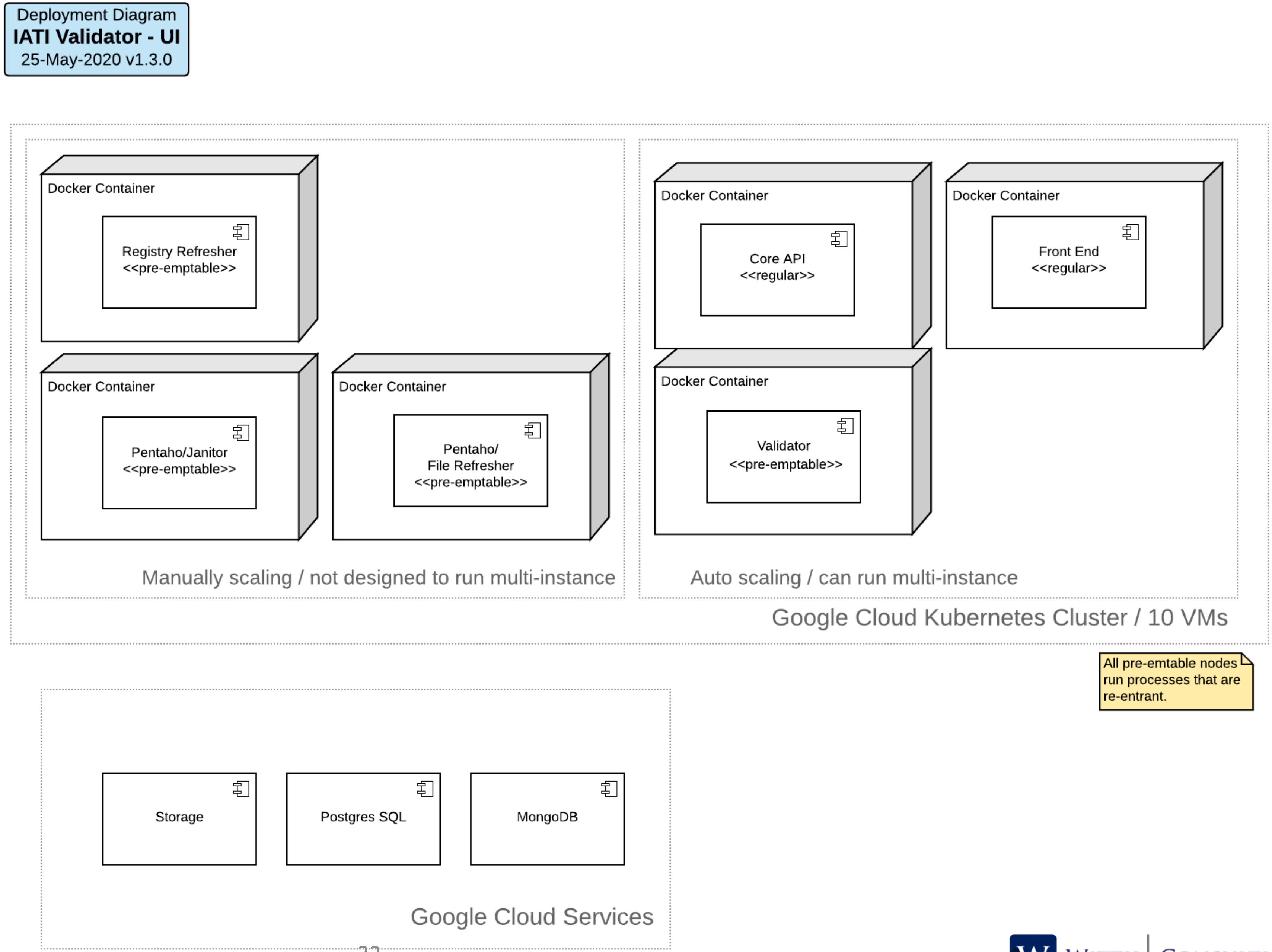
Currently no auth* models for APIs

Instance of Pentaho Node w File "Cleanup" Service

iati International Aid Transparency Initiative

W WITTIJ | CONSULTING

Validator



Deployment Diagram
IATI Validator - UI
25-May-2020 v1.3.0

Docker Container
Registry Refresher
<<pre-emptable>>

Docker Container
Pentaho/Janitor
<<pre-emptable>>

Docker Container
Pentaho/
File Refresher
<<pre-emptable>>

Manually scaling / not designed to run multi-instance

Docker Container
Core API
<<regular>>

Docker Container
Front End
<<regular>>

Docker Container
Validator
<<pre-emptable>>

Auto scaling / can run multi-instance

Google Cloud Kubernetes Cluster / 10 VMs

All pre-emtable nodes run processes that are re-entrant.

Storage

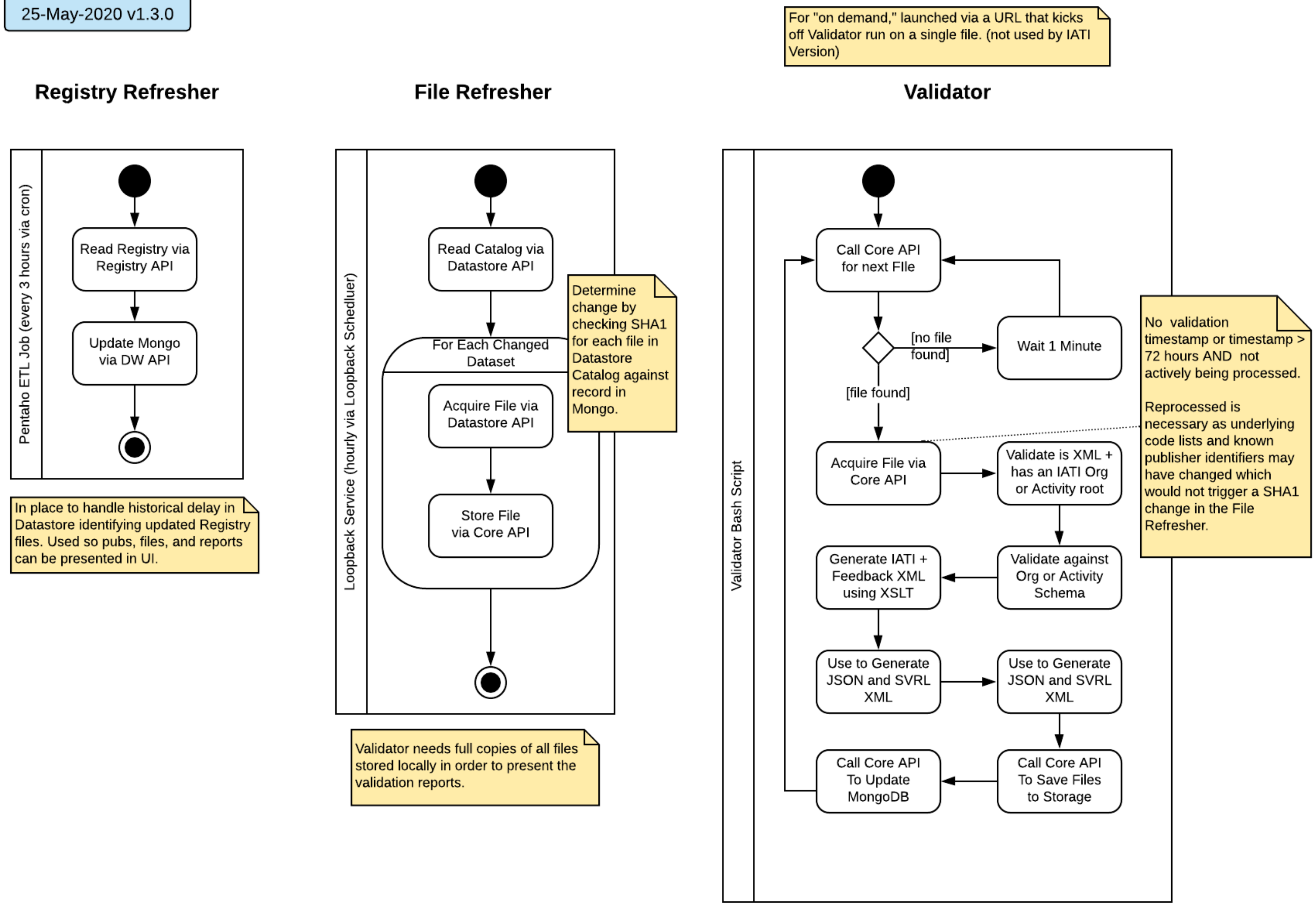
Postgres SQL

MongoDB

Google Cloud Services

Validator

Activity Diagrams
IATI Validator
25-May-2020 v1.3.0



Registry Refresher

Pentaho ETL Job (every 3 hours via cron)

Read Registry via Registry API

Update Mongo via DW API

In place to handle historical delay in Datastore identifying updated Registry files. Used so pubs, files, and reports can be presented in UI.

File Refresher

Loopback Service (hourly via Loopback Schedluer)

Read Catalog via Datastore API

For Each Changed Dataset

Acquire File via Datastore API

Store File via Core API

Determine change by checking SHA1 for each file in Datastore Catalog against record in Mongo.

Validator needs full copies of all files stored locally in order to present the validation reports.

Validator

For "on demand," launched via a URL that kicks off Validator run on a single file. (not used by IATI Version)

Validator Bash Script

Call Core API for next FIle

[no file found]

Wait 1 Minute

[file found]

Acquire File via Core API

Validate is XML + has an IATI Org or Activity root

Validate against Org or Activity Schema

Generate IATI + Feedback XML using XSLT

Use to Generate JSON and SVRL XML

Use to Generate JSON and SVRL XML

Call Core API To Save Files to Storage

Call Core API To Update MongoDB

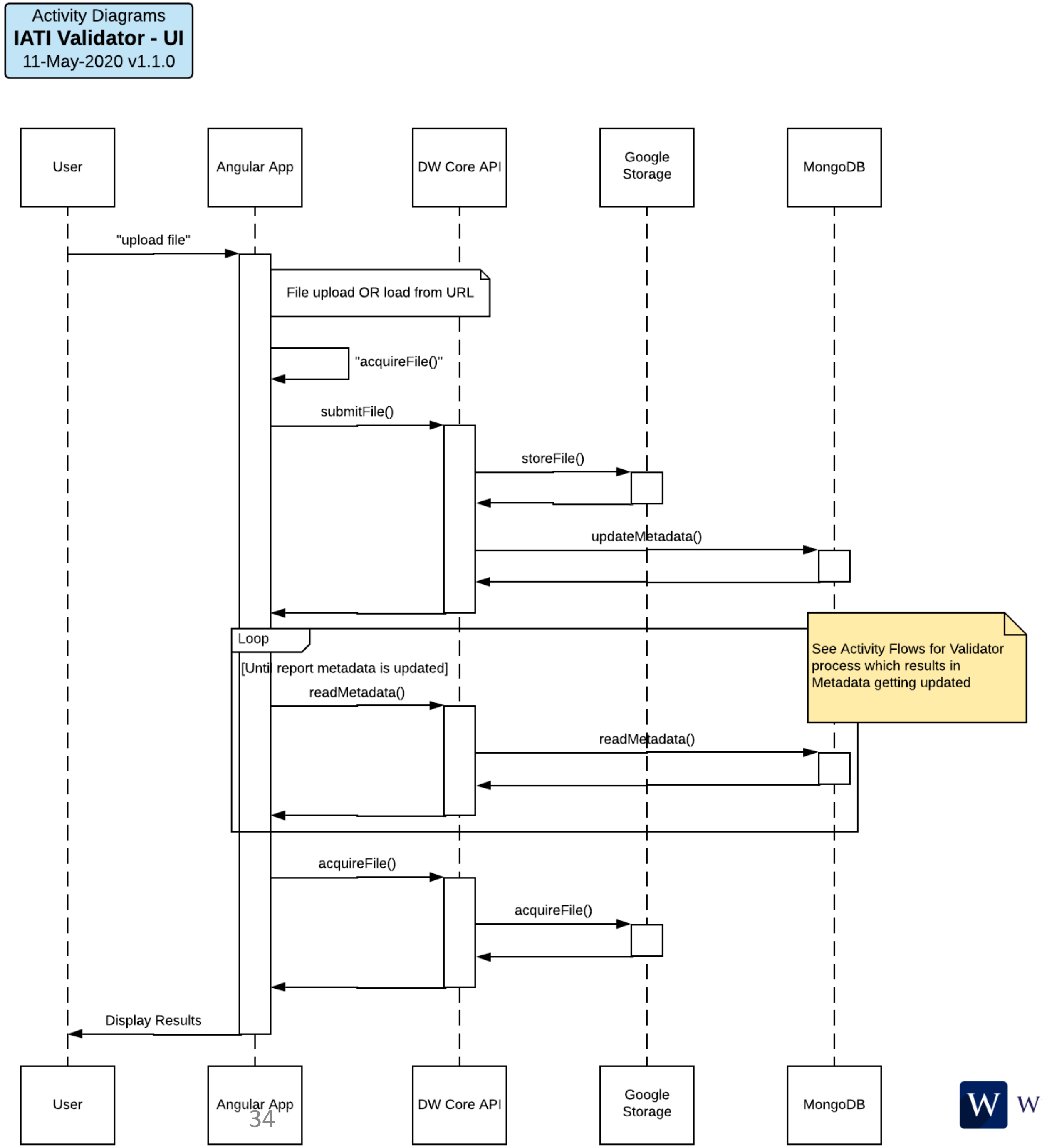
No validation timestamp or timestamp > 72 hours AND not actively being processed.

Reprocessed is necessary as underlying code lists and known publisher identifiers may have changed which would not trigger a SHA1 change in the File Refresher.

iati International Aid Transparency Initiative

W WITTIJ CONSULTING

Validator



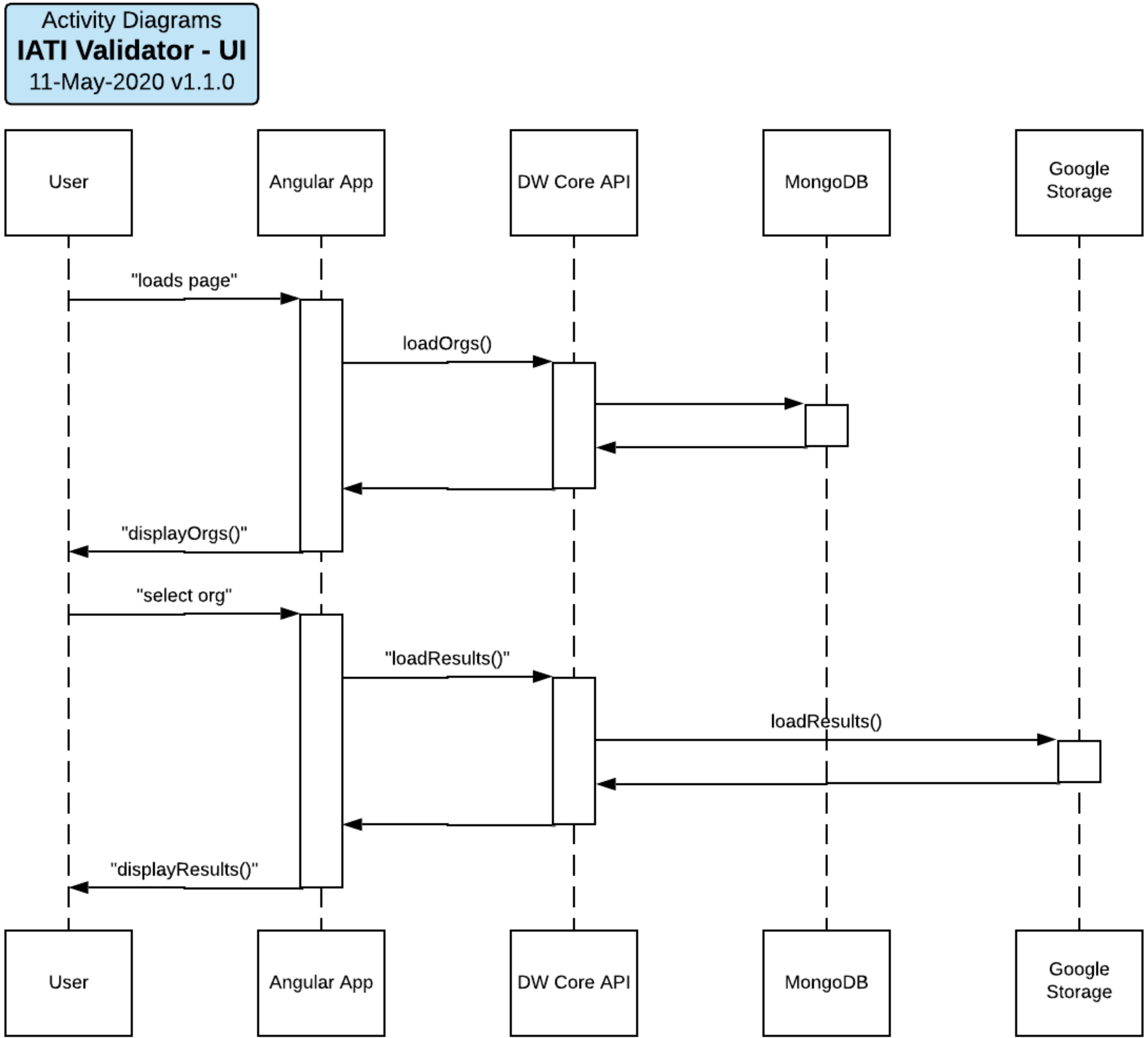
Activity Diagrams
IATI Validator - UI
11-May-2020 v1.1.0

See Activity Flows for Validator process which results in Metadata getting updated

International Aid Transparency Initiative

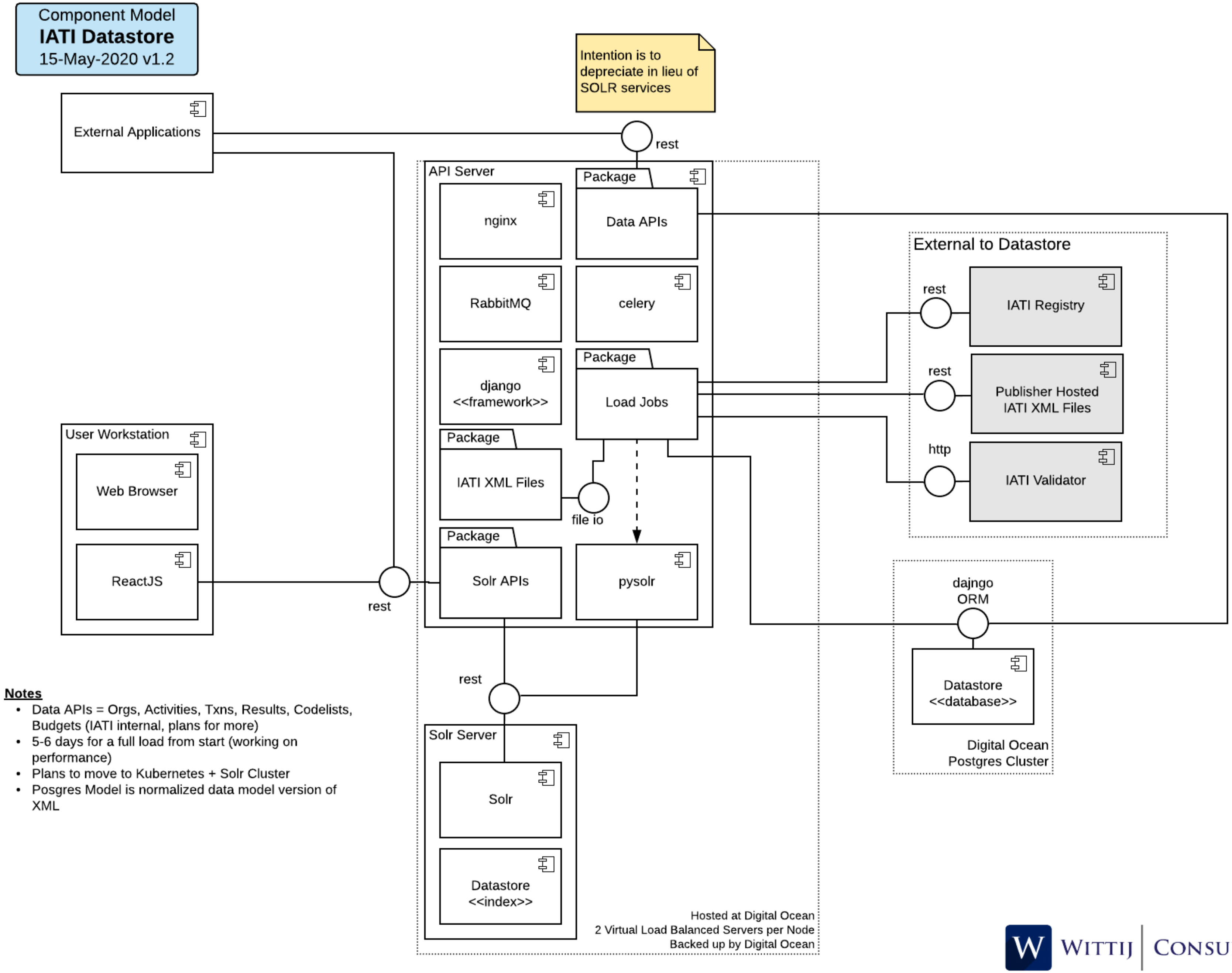
Wittij | Consulting

Validator



Activity Diagrams
IATI Validator - UI
11-May-2020 v1.1.0

Data Store



Notes

- Data APIs = Orgs, Activities, Txns, Results, Codelists, Budgets (IATI internal, plans for more)
- 5-6 days for a full load from start (working on performance)
- Plans to move to Kubernetes + Solr Cluster
- Posgres Model is normalized data model version of XML

Hosted at Digital Ocean
2 Virtual Load Balanced Servers per Node
Backed up by Digital Ocean

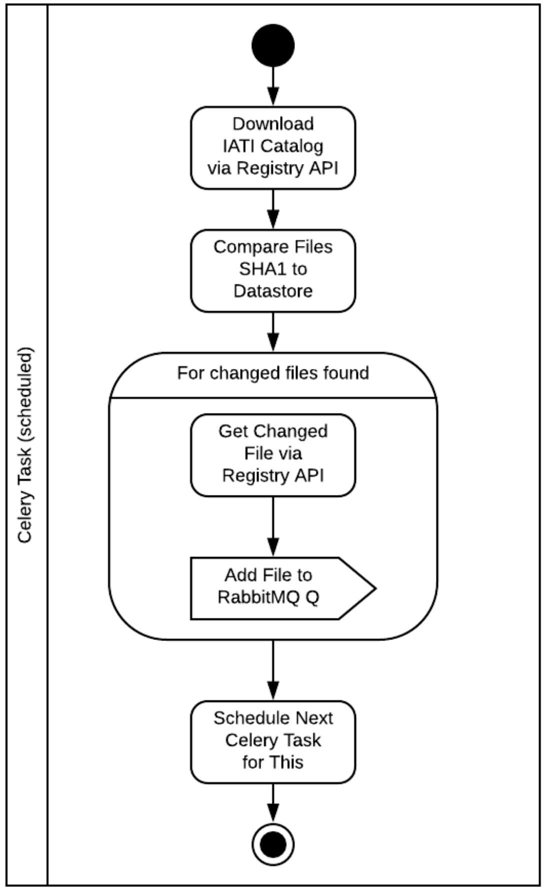
iati International Aid Transparency Initiative

W WITTIJ CONSULTING

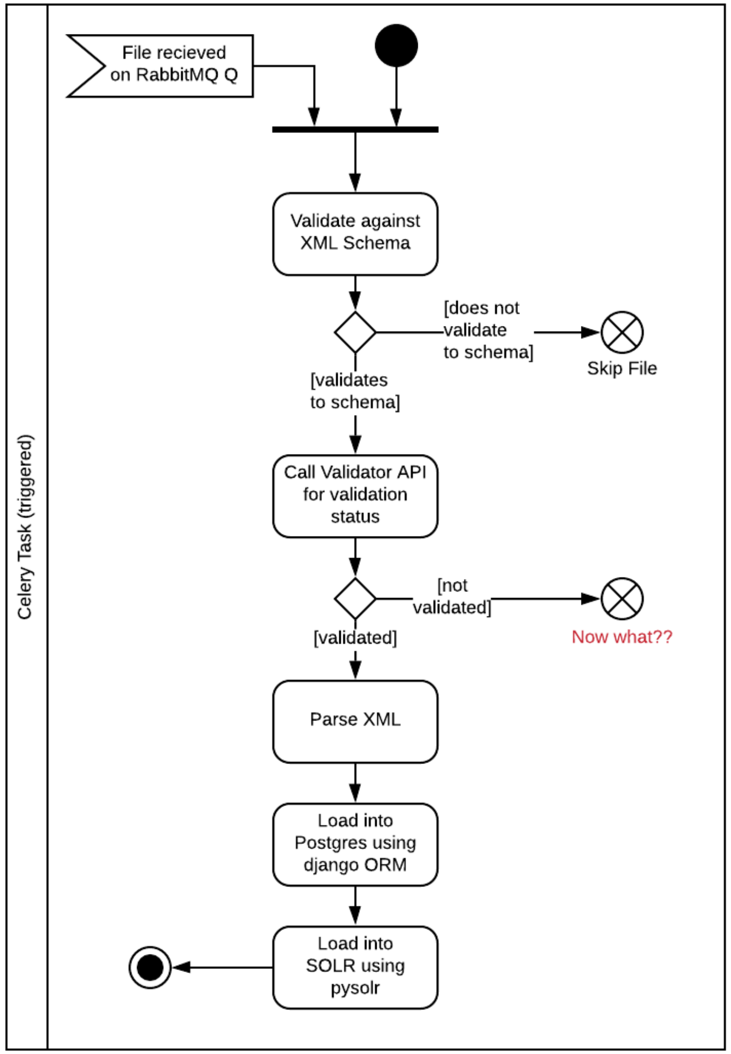
Data Store

Activity Diagrams
IATI Datastore
15-May-2020 v1.2.0

Acquire Registry Files


Celery Task (scheduled)
Download IATI Catalog via Registry API
Compare Files SHA1 to Datastore
For changed files found
Get Changed File via Registry API
Add File to RabbitMQ Q
Schedule Next Celery Task for This

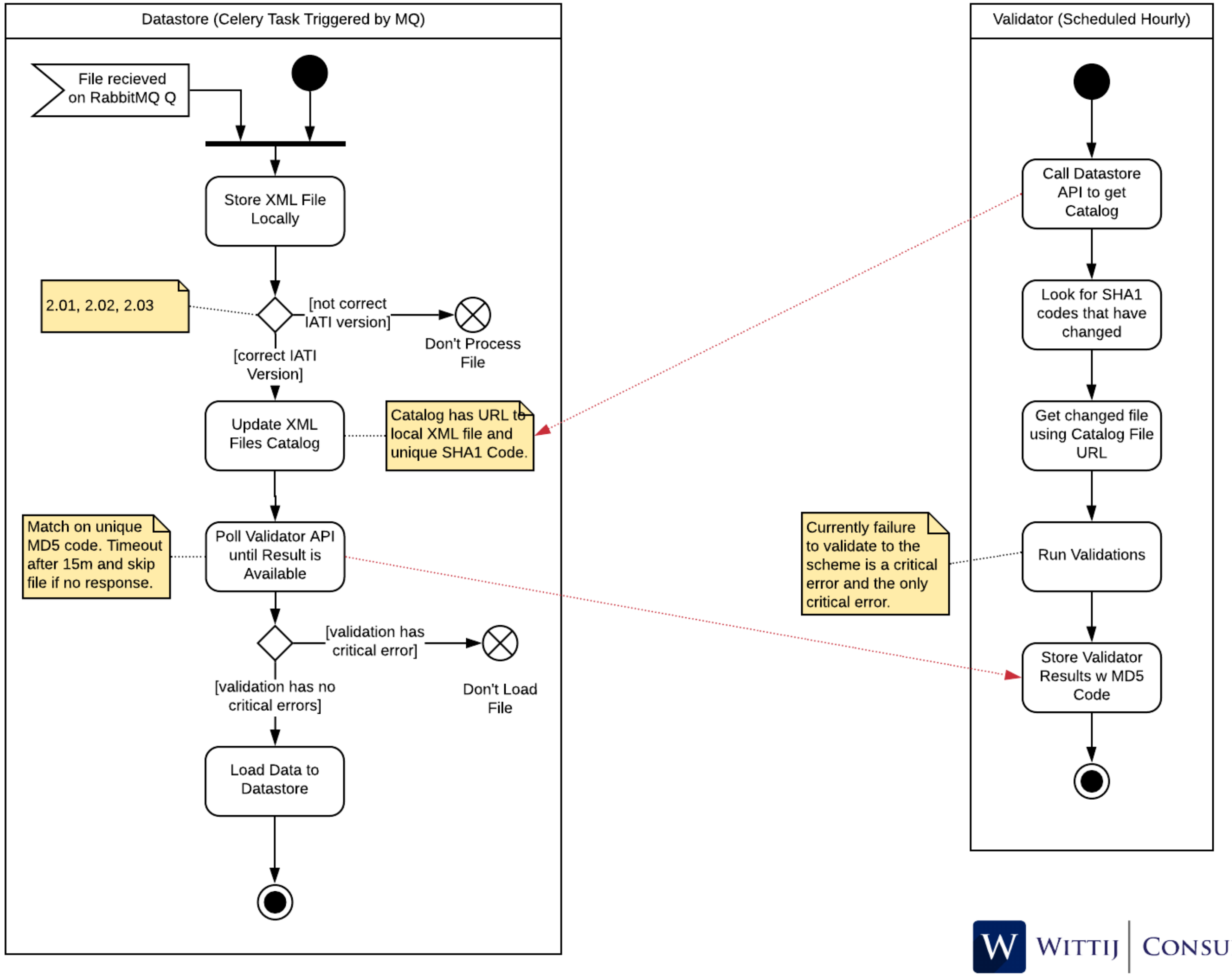
Ignore This One For Now


Celery Task (triggered)
File recieved on RabbitMQ Q
Validate against XML Schema
[does not validate to schema]
Skip File
[validates to schema]
Call Validator API for validation status
[not validated]
Now what??
[validated]
Parse XML
Load into Postgres using django ORM
Load into SOLR using pysolr

Data Store

Activity Diagrams
IATI Datastore
27-May-2020 v1.4.0

These just focus on understand the interface. See activity diagrams for each component for details.



Datastore (Celery Task Triggered by MQ)

File recieved on RabbitMQ Q

Store XML File Locally

2.01, 2.02, 2.03

[not correct IATI version]

Don't Process File

[correct IATI Version]

Update XML Files Catalog

Catalog has URL to local XML file and unique SHA1 Code.

Match on unique MD5 code. Timeout after 15m and skip file if no response.

Poll Validator API until Result is Available

[validation has critical error]

Don't Load File

[validation has no critical errors]

Load Data to Datastore

Validator (Scheduled Hourly)

Call Datastore API to get Catalog

Look for SHA1 codes that have changed

Get changed file using Catalog File URL

Currently failure to validate to the scheme is a critical error and the only critical error.

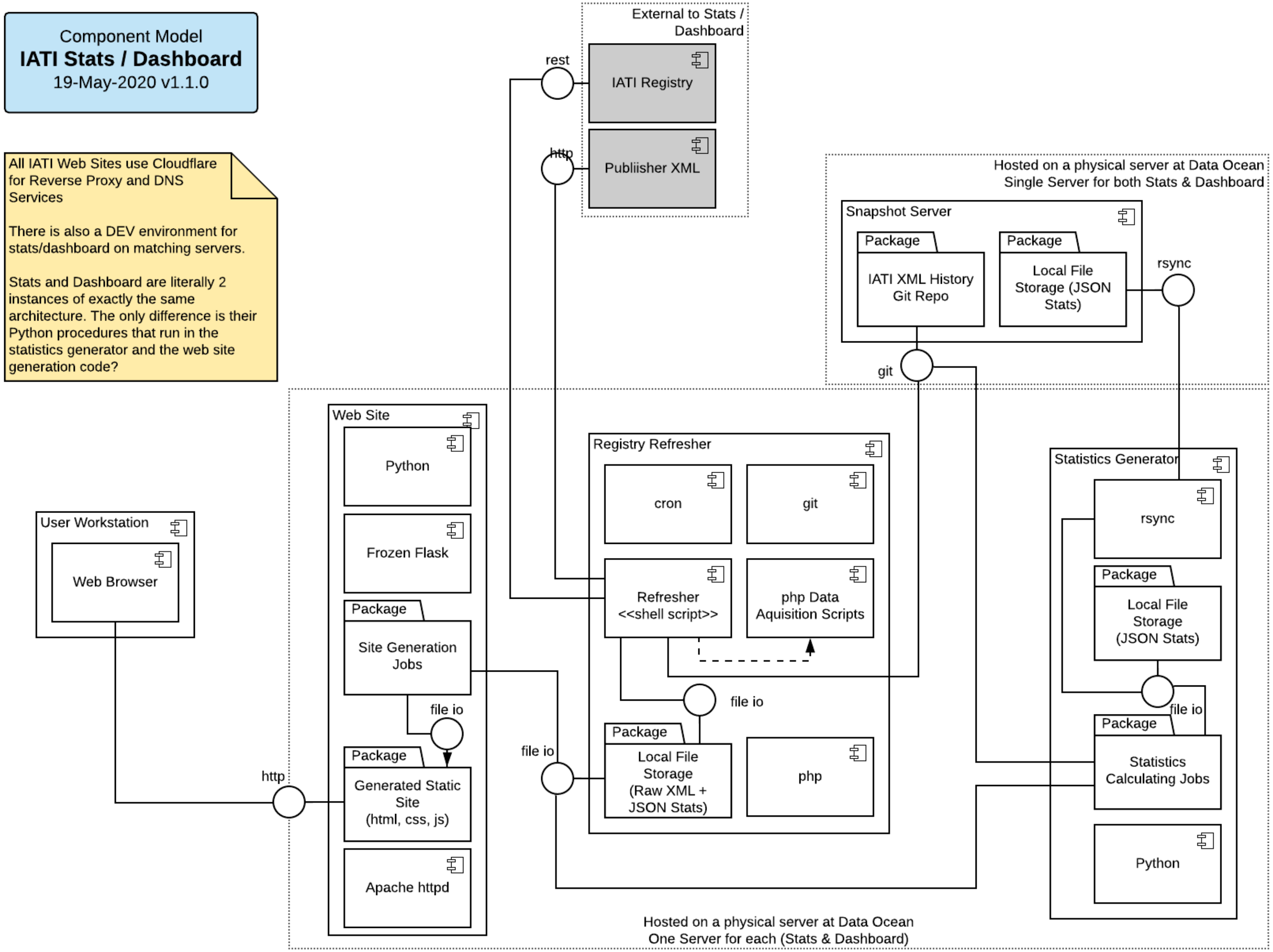
Run Validations

Store Validator Results w MD5 Code

iati International Aid Transparency Initiative

W WITTIJ | CONSULTING

Stats and Dashboard



Component Model
IATI Stats / Dashboard
19-May-2020 v1.1.0

All IATI Web Sites use Cloudflare for Reverse Proxy and DNS Services

There is also a DEV environment for stats/dashboard on matching servers.

Stats and Dashboard are literally 2 instances of exactly the same architecture. The only difference is their Python procedures that run in the statistics generator and the web site generation code?

External to Stats / Dashboard
rest
IATI Registry
http
Publiisher XML
Hosted on a physical server at Data Ocean
Single Server for both Stats & Dashboard
Snapshot Server
Package
IATI XML History Git Repo
Package
Local File Storage (JSON Stats)
rsync
git
Web Site
Python
Frozen Flask
Package
Site Generation Jobs
file io
Package
Generated Static Site (html, css, js)
Apache httpd
User Workstation
Web Browser
http
Registry Refresher
cron
git
Refresher <<shell script>>
php Data Aquisition Scripts
file io
Package
Local File Storage (Raw XML + JSON Stats)
php
file io
Statistics Generator
rsync
Package
Local File Storage (JSON Stats)
file io
Package
Statistics Calculating Jobs
Python
Hosted on a physical server at Data Ocean
One Server for each (Stats & Dashboard)

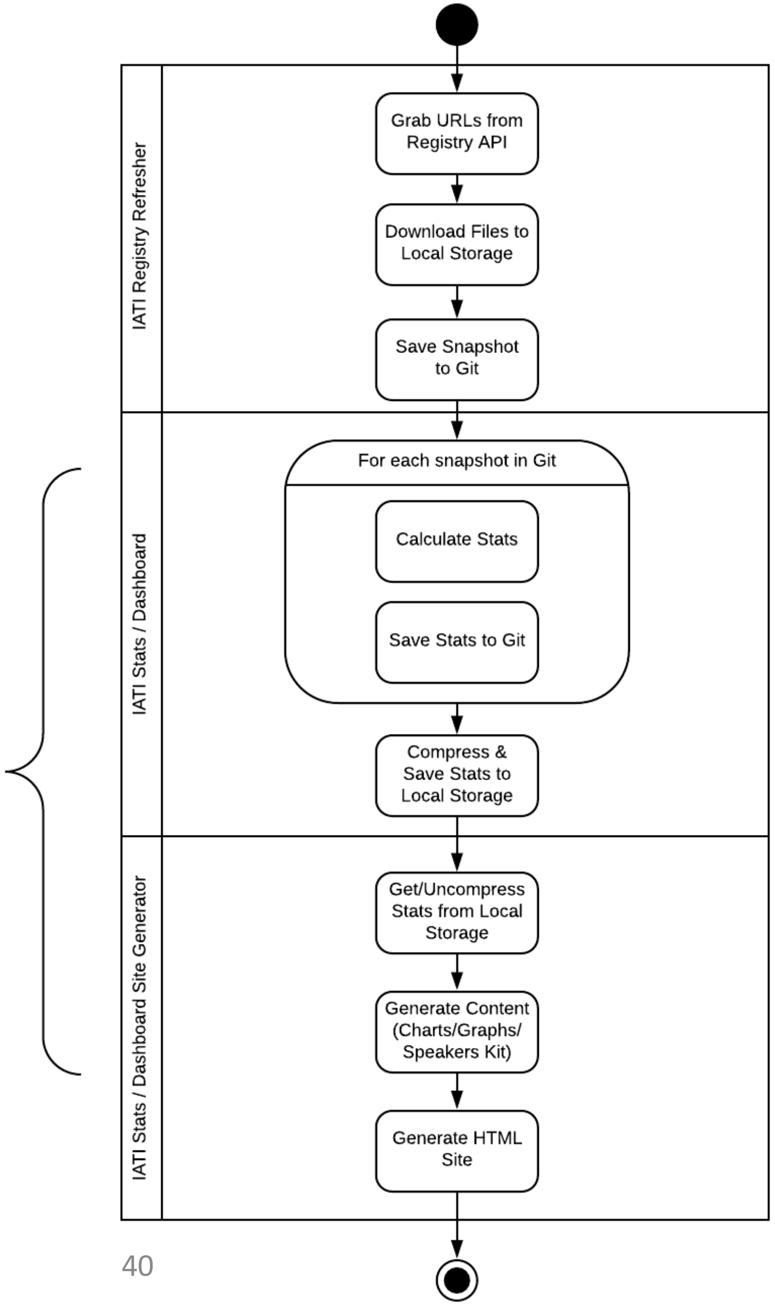
iati International Aid Transparency Initiative

W WITTIJ CONSULTING

Stats and Dashboard

Activity Diagrams
IATI Stats / Dashboard
19-May-2020 v1.1.0

Same process, but seperate code bases for Stats and Dashboard (were originally clones)


IATI Registry Refresher
Grab URLs from Registry API
Download Files to Local Storage
Save Snapshot to Git
IATI Stats / Dashboard
For each snapshot in Git
Calculate Stats
Save Stats to Git
Compress & Save Stats to Local Storage
IATI Stats / Dashboard Site Generator
Get/Uncompress Stats from Local Storage
Generate Content (Charts/Graphs/ Speakers Kit)
Generate HTML Site

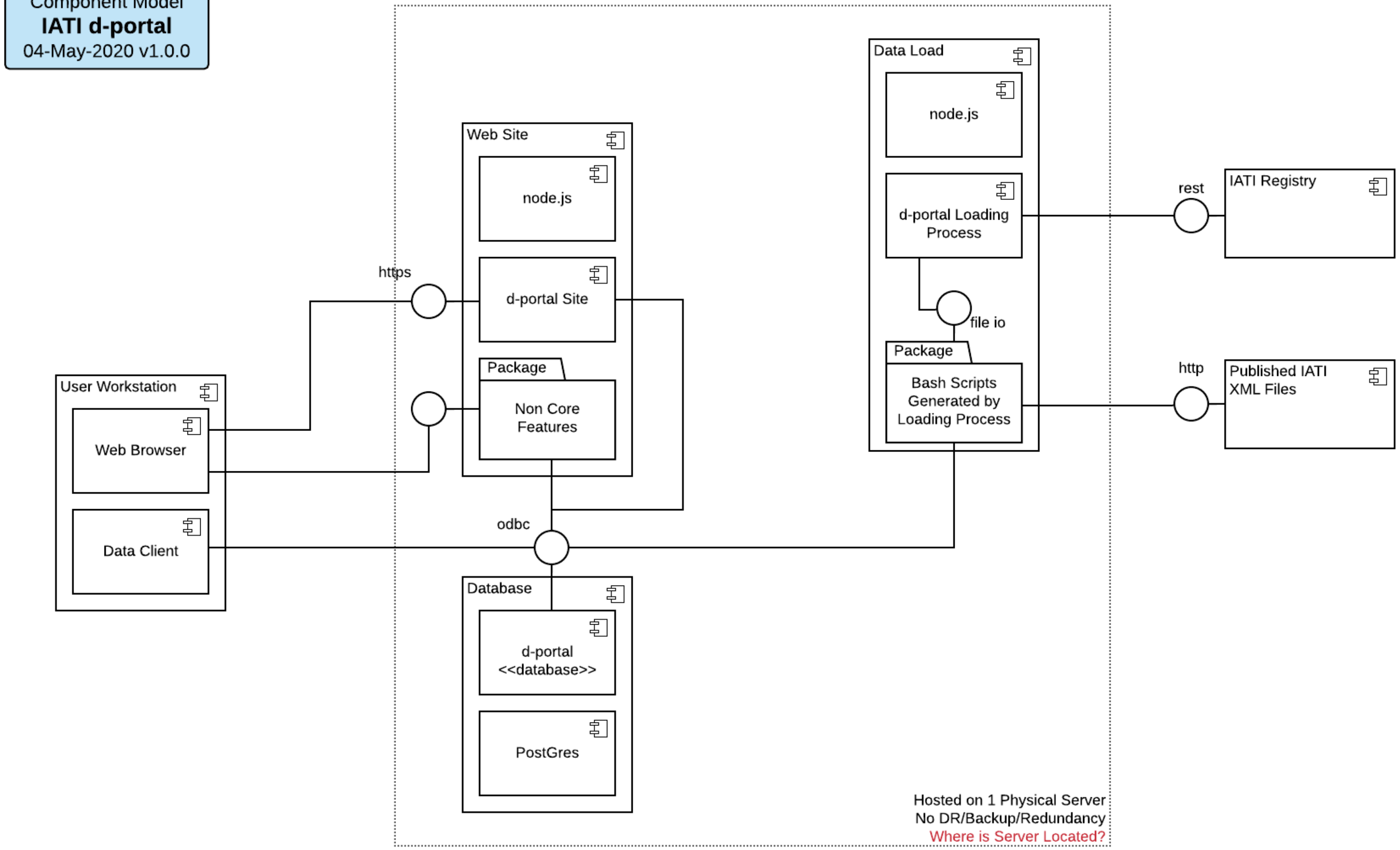
./validate.sh runs at end of Registry Refresher, but fails due to GIST endpoint that no longer works.

d-portal

Component Model

IATI d-portal

04-May-2020 v1.0.0



Hosted on 1 Physical Server
No DR/Backup/Redundancy
Where is Server Located?

Non-Core Features

- **Generator** provides embedable version of d-portal charts
- **Dash** shows status of data import for a specific publisher
- **xPath** provides stats on the use of elements within the schema (since dPortal implementation)
- **dQuery** is a read only query interface.
- **Preview** displays a summary of an uploaded IATI to review before publishing to Registry.
- **dflat** translats XML to/from CSV or JSON

Notes

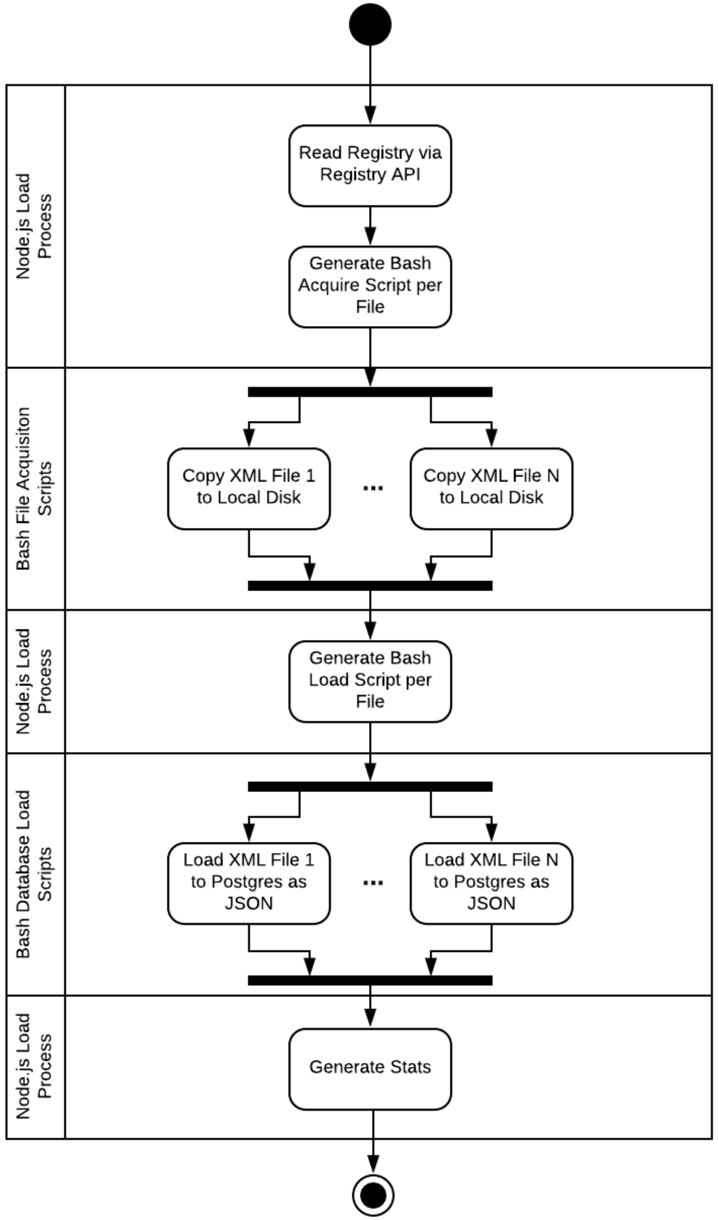
- No authentication; data is 100% reloaded nightly (30m)
- d-portal Site node.js generated HTML + CSS + JS

iati International Aid Transparency Initiative

W WITTIJ | CONSULTING

d-portal

Activity Diagrams
IATI d-portal
04-May-2020 v1.0.0



Node.js Load Process

Read Registry via Registry API

Generate Bash Acquire Script per File

Bash File Acquisiton Scripts

Copy XML File 1 to Local Disk

...

Copy XML File N to Local Disk

Node.js Load Process

Generate Bash Load Script per File

Bash Database Load Scripts

Load XML File 1 to Postgres as JSON

...

Load XML File N to Postgres as JSON

Node.js Load Process

Generate Stats

cUrl used to access XML at publishers site and make local copy.

Loaded in parallel, but launched in random order so different publisher's files fail in the case of database contention over duplicated activity Ids. That way if one fails one night, it loads the next.

iati International Aid Transparency Initiative

W WITTIJ | CONSULTING