



Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

What's new in Apache Tika 2.0 – we mean it this time!

Tim Allison, Ph.D.

Data Scientist/Relevance Engineer

Artificial Intelligence, Analytics and Innovative
Development Organization(1740)

ITSD

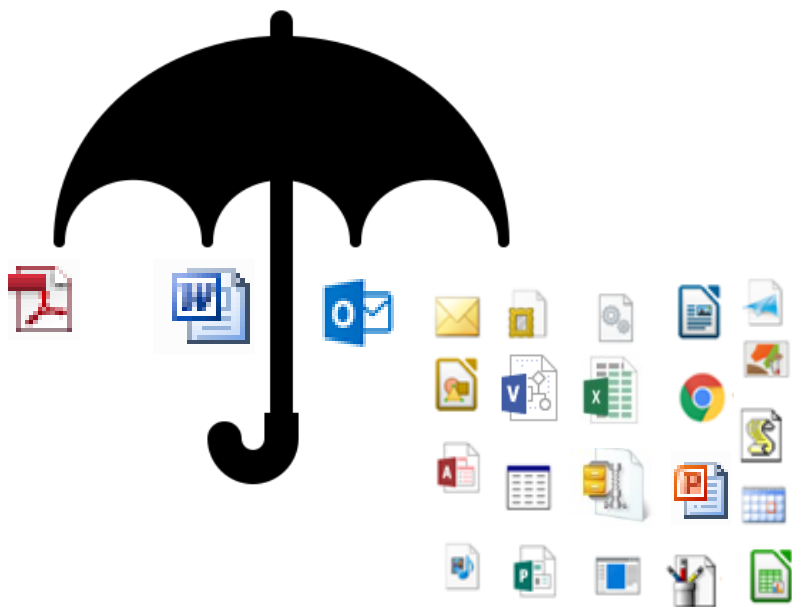
The research was carried out at the NASA (National Aeronautics and Space Administration) Jet Propulsion Laboratory, California Institute of Technology under a contract with the Defense Advanced Research Projects Agency (DARPA) SafeDocs program. © 2021 California Institute of Technology. Government sponsorship acknowledged.



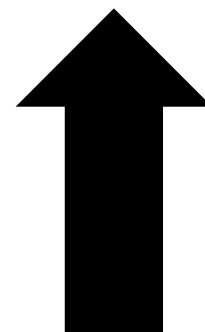
Jet Propulsion Laboratory
California Institute of Technology

Apache Tika, an overview

Framework for file type detection, parsing and uniform output for ~75 parsers, ~100+ formats



text and metadata



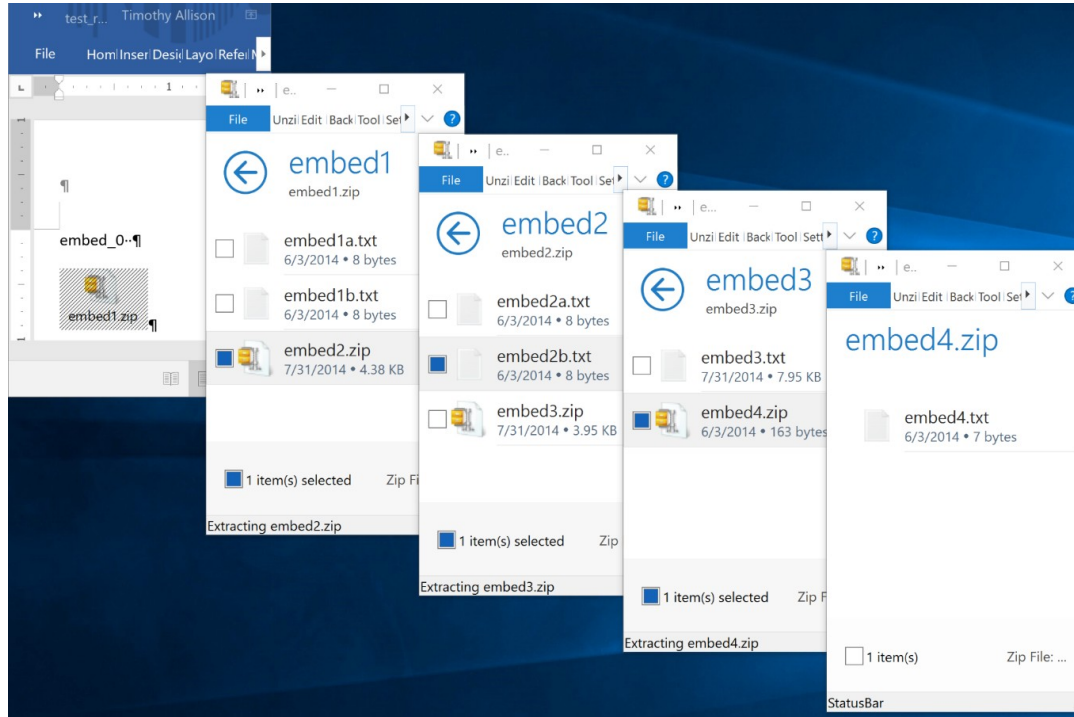
bytes

<https://tika.apache.org/>

Apache Tika, features

- Easy to add new file types for detection
- Easy to add new parsers
- Works recursively with embedded files/attachments
- Integration with tesseract-ocr

Embedded files/attachments



Outline

- Status
- Breaking changes
- Maven modularization
- Motivations for tika-pipes
 - Crashing JVMs at scale
 - Brief history of tika-server
- tika-pipes

Status

- Tika 2.0.0-ALPHA released January 2021.
- Tika 2.0.0-BETA released May 2021.
- Welcome to Nicholas DiPiazza as a committer/PMC!

Tika 2.0.0 – Breaking Changes

- PDFParser will call tesseract in “auto” mode if installed by default (like all the other parsers)
- Parsers Maven modularized and moved around
- tika-server operates in –spawnChild mode by default
- Metadata – streamlined with preference for standards, e.g. Dublin Core

Maven Modularize All the Things

tika-parsers

- tika-parsers-standard
- tika-parsers-extended
 - May require heavy dependencies, parsers that require external resources (network calls), parsers that require native libs (e.g. sqlite3)
- tika-parsers-advanced
 - Super heavy dependencies (dl4j)
 - Heavy processing beyond “text from bytes” (image recognition/NLP)



tika-parsers-standard and tika-parsers-extended

- ▼ **tika-parsers-classic**
 - ▼ **tika-parsers-classic-modules**
 - > tika-parser-apple-module
 - > tika-parser-audiovideo-module
 - > tika-parser-cad-module
 - > tika-parser-code-module
 - > tika-parser-crypto-module
 - > tika-parser-digest-commons
 - > tika-parser-font-module
 - > tika-parser-html-commons
 - > tika-parser-html-module
 - > tika-parser-image-module
 - > tika-parser-jdbc-commons
 - > tika-parser-mail-commons
 - > tika-parser-mail-module
 - > tika-parser-microsoft-module
 - > tika-parser-miscoffice-module
 - > tika-parser-news-module
 - > tika-parser-ocr-module
 - > tika-parser-pdf-module
 - > tika-parser-pkg-module
 - > tika-parser-text-module
 - > tika-parser-xml-module
 - > tika-parser-xmp-commons
 - > tika-parser-zip-commons

- ▼ **tika-parsers-extended**
 - > **tika-parser-scientific-module**
 - > **tika-parser-sqlite3-module**

- ▼ **tika-parser-scientific-module**
 - ▼ src
 - ▼ main
 - ▼ java
 - > org.apache.tika.parser.envi
 - > org.apache.tika.parser.gdal
 - > org.apache.tika.parser.geoinfo
 - > org.apache.tika.parser.grib
 - > org.apache.tika.parser.hdf
 - > org.apache.tika.parser.isatab
 - > org.apache.tika.parser.netcdf

tika-parsers-advanced

- ▼ tika-parsers-advanced
 - > tika-age-recogniser
 - > tika-dl
 - > tika-parser-advancedmedia-module
 - > tika-parser-nlp-module

ASR/STT! Thanks to TIKA-94 and Lewis McGibbney and team!

- ▼ tika-parser-advancedmedia-module
 - ▼ src
 - ▼ main
 - ▼ java
 - > org.apache.tika.parser.captioning
 - > org.apache.tika.parser.captioning.tf
 - > org.apache.tika.parser.pot
 - > org.apache.tika.parser.recognition
 - > org.apache.tika.parser.recognition.tf







tika-app and parser modules

- Same as before but with only tika-parsers-standard included
- Users must add tika-parsers-extended submodules for scientific format parsing and/or sqlite3 parsing





tika-server and parser modules

- tika-server-core
 - Basic server with *no* parsers
- tika-server-standard
 - Server with standard parser
 - Users must add tika-parsers-extended submodules for scientific format parsing and/or sqlite3 parsing

langdetect modularized

- ▼  **tika-langdetect**
 - >  **tika-langdetect-commons**
 - >  **tika-langdetect-lingo24**
 - >  **tika-langdetect-mitll-text**
 - >  **tika-langdetect-opennlp**
 - >  **tika-langdetect-optimize**

tika-server modularized

- ∨  **tika-server**
 - >  **tika-server-classic**
 - >  **tika-server-client**
 - ∨  **tika-server-core**

tika-eval modularized

- ✓  **tika-eval**
 - >  **tika-eval-app**
 - >  **tika-eval-core**
 pom.xml

Drop `tika-eval-core.jar` on your classpath with `tika-server` and you'll get `tika-eval` stats!

	Tika 1.14	Tika 1.15-SNAPSHOT
Unique Tokens	786	156
Total Tokens	1603	272
LangId	zh-ch	de
Common Words	0	116
Alphabetic Tokens	1603	250
Top N Tokens	搯敌 : 18 獠档 : 14 略獠 : 14 m: 11 柿湊 : 11 瑤搯 : 11 畚柿 : 11 档湊 : 10 搯敦 : 9 敌湊 : 9	die: 11 und: 8 von: 8 deutschen: 7 deutsche: 6 1: 5 das: 5 der: 5 finanzministerium: 5 oder: 5
OOV%	$1 - (0/1603) = 100\%$	$1 - (116/250) = 54\%$

tika-eval's Out-of-vocabulary (OOV) in action

Text As Stored in File

```
!"#$%& (') *,+-. ' / 0 1,23 *. 457698;::;<>=75?&@78;ACB  
D(B7E;FHGJICBK5MLNBKOPBKF;B DJD Q R S.TVU9WNXMY[Z\T]^W_S  
`badc 5KICedFgfh5 cji ;;edF;A^5KEk<>ImIn;e[<>EnloedACICe a  
lo<p57Eg5Kqsr;E;<jloe[E 8;O 6hedA5Kq adc 57ItedFk;:;B c qslCf;B a
```

Text from Tesseract OCR

Constrained Least Squares Linear Spectral Unmixture by the Hybrid Steepest Descent Method

Nobuhiko Ogura' and Isao Yamada"

1 Introduction

A closed polyhedron is the intersection of finite number of closed half spaces, i.e., the set of points satisfying finite number of linear inequalities, and is widely used as a constraint in various applications, for example specifications or constraints in signal processing or estimation problems, resource restrictions in financial applications and feasible sets of

Crashing JVMs at Scale*

*Credit: Nick Burch

When bad things happen...

- Out of memory errors
- Infinite loops
- Slow-building memory leaks
- Crashes (System.exit) or OS oom-killer
- Malicious code
- Runaway forked processes

Tika is not alone!



<https://twitter.com/dinodaizovi/status/1389620920789131266>

Parsing is dangerous!

- Change the mindset:

Parsing is dangerous and needs to be done carefully

We need to take errors seriously



“High-Assurance Input Validation: Locking the Front Door.”
Kathleen Fisher’s Keynote. LangSec 2021 Workshop. IEEE Security and Privacy.

https://langsec.org/spw21/slides/Fisher_LangSec21.pdf

Really, quite dangerous!

- 80% of CVEs are in code that handles input data
- 2020 CWE Top 25 most dangerous software weaknesses
 - #1 Improper neutralization of input during web page generation
 - #3 improper input validation
- 1000 parser bugs in Mozilla

“High-Assurance Input Validation: Locking the Front Door.” Kathleen Fisher’s Keynote.
LangSec 2021 Workshop. IEEE Security and Privacy.
https://langsec.org/spw21/slides/Fisher_LangSec21.pdf

Font parsers as components of exploit chains

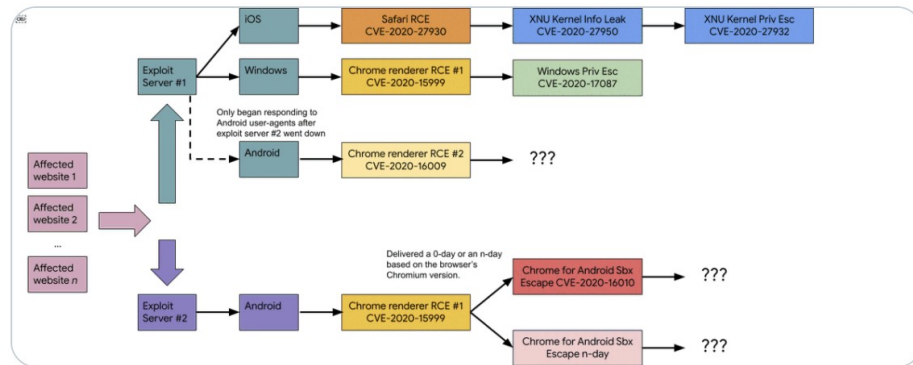


Catalin Cimpanu
@campuscodi



NEW: Google has published additional details about a mysterious threat actor that has deployed 11 different zero-days last year as part of a hacking campaign that targeted Android, iOS, and Windows users

therecord.media/a-mysterious-h...



Four of these are font parsers!

CVE-2020-0938

CVE-2020-1020

CVE-2020-15999

CVE-2020-27930

- <https://twitter.com/campuscodi/status/1372628034679930886>

- <https://therecord.media/a-mysterious-hacking-group-used-11-different-zero-days-in-2020>

3:16 PM · Mar 18, 2021 · Twitter Web App

“Rogue” Documents

How Well Do You Understand Your Content Processing Pipeline?



By Martin White | *May 12, 2021*

 Follow

3,293 followers

Building the Search Index

Building the initial index is no time to be around a search team. Not so very long ago it might have taken a couple of weeks to index a reasonable-sized document collection. Halfway through, **a rogue document might bring the process to a halt** or a post-index check might show that some stop words had not been correctly recognized. Issues such as these might require starting all over again.

<https://www.cmswire.com/knowledge-findability/how-well-do-you-understand-your-content-processing-pipeline/>

TIKA-1132



Tika / TIKA-1132

Parsing some XLS documents hangs entire JVM, requires kill -9

▼ Details

Type:	 Bug	Status:	CLOSED
Priority:	 Major	Resolution:	Fixed
Affects Version/s:	1.2, 1.3	Fix Version/s:	1.5
Component/s:	parser		
Labels:	None		
Environment:	› Linux Suse: java version "1.7.0" Java(TM) SE Runtime Environment (build 1.7.0-b147) Java ...		

▼ Description

Some XLS documents hang the entire JVM. A control-C or regular kill won't stop the JVM, a kill -9 is required.

We're running within an email server application parsing documents to extract text of all attachments. When we hit a message with the affected attachment the entire JVM hangs and we mark the message to skip extracting the text from the affected message the next attempt. Unfortunately, it kills all email processing on the server until the internal watchdogs kill -9 the application.

Are Infinite Loops Really that Bad?



<https://metro.co.uk/2008/05/15/struggling-polar-bears-put-on-endangered-list-137306/>



<https://www.scienceabc.com/eyeopeners/would-oceans-become-less-salty-if-all-the-polar-ice-caps-melted.html>

Steps we've taken on Tika to mitigate catastrophe

- Reporting: <https://tika.apache.org/security.html>
- Code reviews/forks of external dependencies
- File format aware fuzzing module (just started)
- Regression corpora (2TB/~2 million files)
 - <https://corpora.tika.apache.org/base/docs/>
- MockParser for testing your pipeline's robustness (next slide)

MockParser: Mock a misbehaving parser

Add tika-core's test-jar to your path and specify mayhem

```
<dependency>
  <groupId>${project.groupId}</groupId>
  <artifactId>tika-core</artifactId>
  <version>${project.version}</version>
  <type>test-jar</type>
  <scope>test</scope>
</dependency>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<mock>
  <metadata action="add" name="author">Nikolai Lobachevsky</metadata>
  <write element="p">some content</write>
  <print_out>writing to System.out</print_out>
  <print_err>writing to System.err</print_err>
  <hang millis="100" heavy="true" pulse_millis="10" interruptible="true" />
  <throw class="java.io.IOException">not another IOException</throw>
  <oom/>
  <system_exit />
</mock>
```



Now with
FakeLoad

<https://github.com/msigwart/fakeload>

Some options

- ForkParser
- tika-batch (-i -o options in tika-app)
- tika-server `_(ツ)_/`
- In 2.x: PipesParser, `/pipes` and `/async` handler
- See:
<https://wiki.apache.org/confluence/display/TIKA/The+Robustness+of+Apache+Tika>

Overall goal: Boring!

DATA / DEVELOPMENT / SPONSORED / CONTRIBUTED

It's Not Real Engineering Until It's Boring (to Outsiders)

13 May 2021 6:34am, by [Ted Dunning](#)



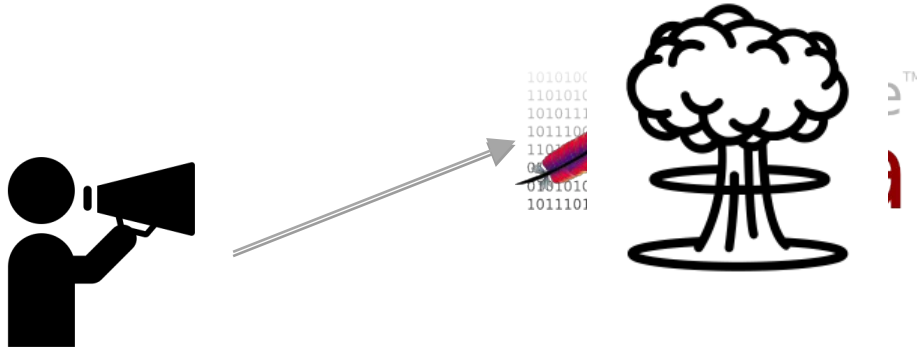
Ted Dunning

That is to say, this technology is mostly amazing because it can be boringly, invisibly, incredibly reliable. Boring is good if you want to get on with your life by using technology rather than inventing it.

<https://thenewstack.io/its-not-real-engineering-until-its-boring-to-outsiders/>

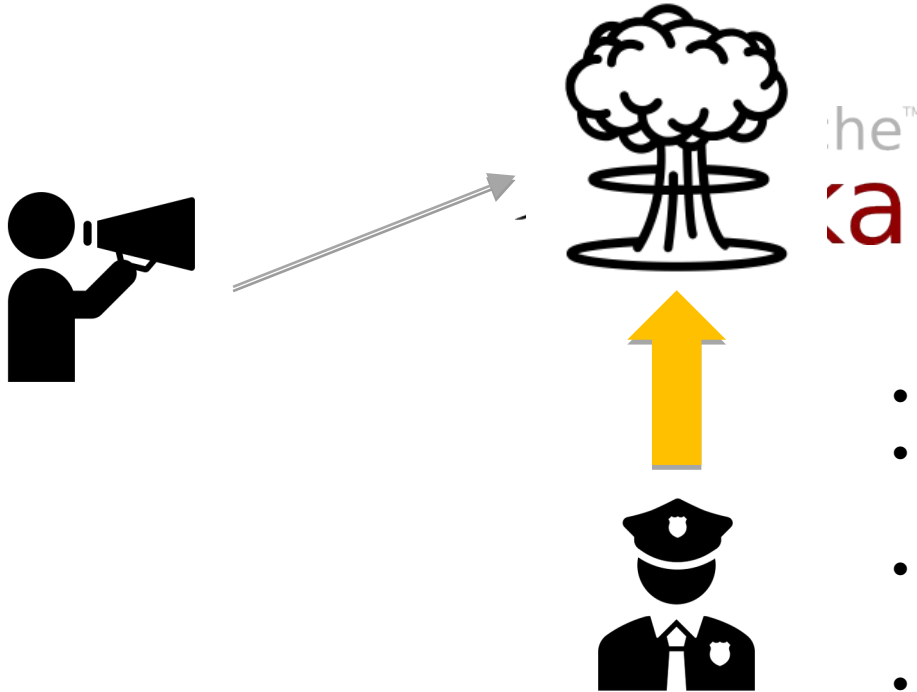
Evolution of tika-server

In the beginning



- Noticeable
 - Crashes
- Hidden
 - Infinite loops
 - Slow building memory leaks
 - Out of Memory errors

--spawnChild (default in Tika 2.x)



- Crashes (restart)
- Infinite loops (timeout and restart)
- Out of Memory errors (restart)
- Slow building memory leaks (restart after X files)

tika-pipes module

Goals

- Isolate parsing into its own process
- Keep iterator/command module in separate process
- Keep parsing out of indexing processes
- Allow for robust timeouts
- Allow for long, long parse times, e.g. OCR on 100 page PDF

Fetchers/emitters

- Configured in tika-config.xml file
- Fetcher – grab an inputstream and metadata from source, e.g. s3://mybucket/path/to/my_file.pdf
- Emitter – after parsing the file, forward it to, e.g. Solr or Elasticsearch

- Isolate iterator/file selector from parsing; isolate parsing from indexer.
- Client only sends ids (not data)
- Potentially put Tika processors closer to the data
- Skip return trip

<https://issues.apache.org/jira/browse/TIKA-3226>

Fetchers

File system, S3, http...tbd: samba, jdbc?

```
<fetchers>
  <fetcher class="org.apache.tika.pipes.fetcher.fs.FileSystemFetcher">
    <params>
      <name>fsf</name>
      <basePath>/path/to/docs</basePath>
    </params>
  </fetcher>
</fetchers>
```

Parse -> extract

```
[
  {
    "dc:title": "My Title",
    "dc:creator": ["creator A", "creator B"],
    "Content-Length": "1000",
    "X-TIKA:content": "this is the actual content of the file"
  }
]
```

Content may be xhtml or plain text

Metadata Filters

```
<metadataFilters>
  <metadataFilter class="org.apache.tika.metadata.filter.FieldNameMappingFilter">
    <params>
      <excludeUnmapped>true</excludeUnmapped>
      <mappings>
        <mapping from="X-TIKA:content" to="content"/>
        <mapping from="Content-Length" to="length"/>
        <mapping from="dc:creator" to="creators"/>
        <mapping from="dc:title" to="title"/>
        <mapping from="Content-Type" to="mime"/>
        <mapping from="X-TIKA:EXCEPTION:container_exception" to="tika-exception"/>
      </mappings>
    </params>
  </metadataFilter>
</metadataFilters>
```

Emitters

After the parse, send the data to...S3, Solr...

```
<emitters>
  <emitter class="org.apache.tika.pipes.emitter.solr.SolrEmitter">
    <params>
      <name>solr1</name>
      <url>http://localhost:8983/solr/tika-integration-example</url>
      <attachmentStrategy>concatenate-content</attachmentStrategy>
      <contentField>content</contentField>
      <commitWithin>10</commitWithin>
    </params>
  </emitter>
  <emitter class="org.apache.tika.pipes.emitter.fs.FileSystemEmitter">
    <params>
      <name>fse</name>
      <basePath>/path/to/extracts</basePath>
    </params>
  </emitter>
</emitters>
```


Example FetchEmitTuple

```
{
  "fetcher": "my-fetcher",
  "fetchKey": "some/path/to/my/file/some_file_or_another.pdf",
  "emitter": "my-emitter",
  "emitKey": "some/path/to/my/file/some_file_or_another.pdf",
  "metadata": {
    "k1": "v1",
    "k2": ["v2", "v3"]
  },
  "onParseException": "emit"
}
```

New endpoints in tika-server: /pipes, /async

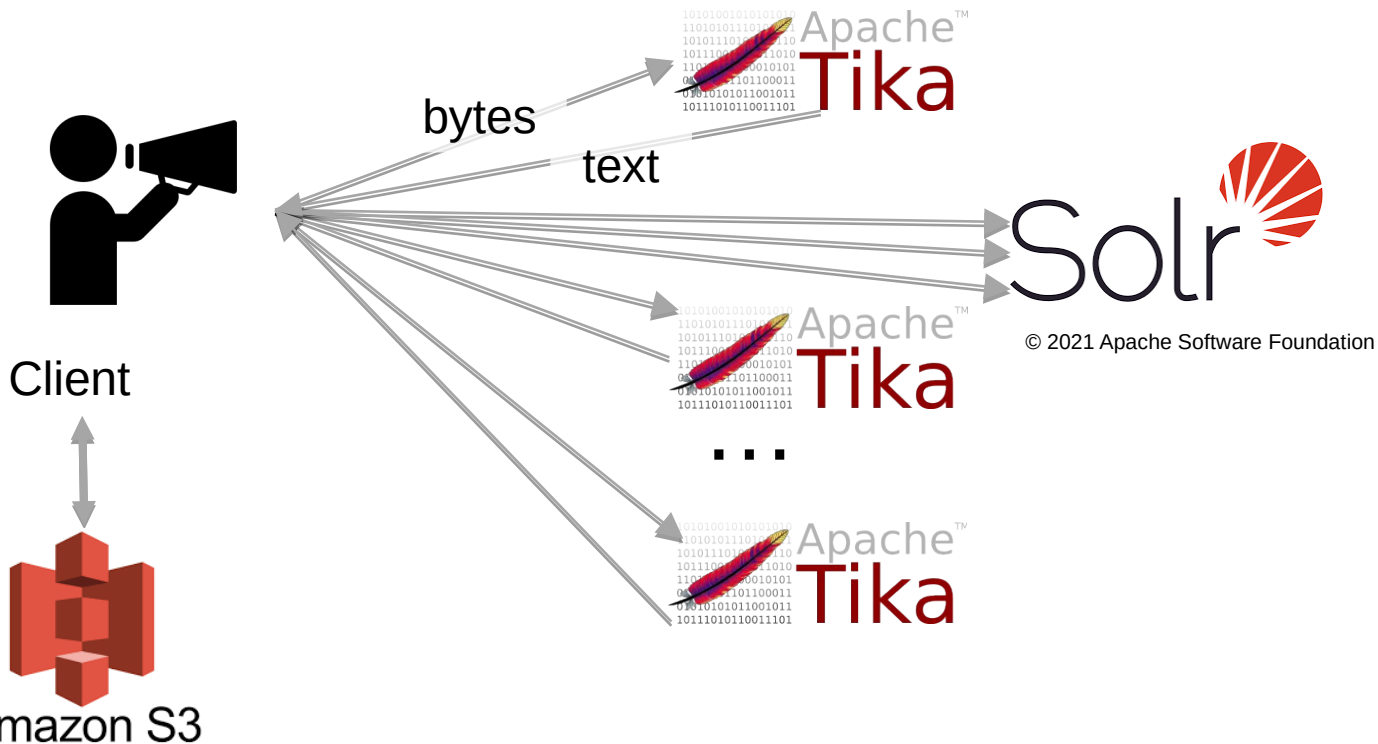
- /pipes is /rmeta but fetches, emits and returns status value
- /async is /pipes, but it adds a list of FetchEmitTuples to a queue

Fetchers/Emitters

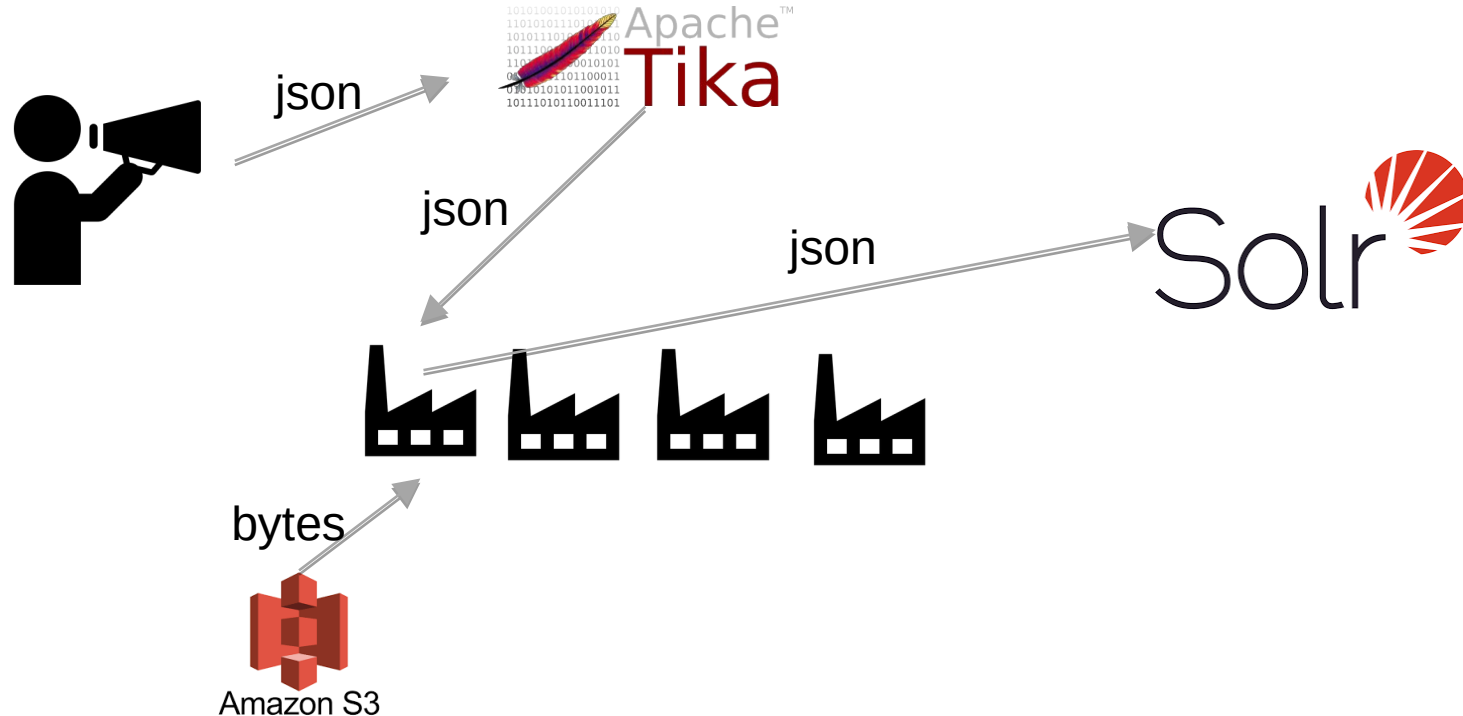
- Fetchers
 - FileShare
 - s3
 - URL
 - JDBC
 - Solr
- Emitters
 - FileShare
 - s3
 - Solr
 - OpenSearch (planned)

Current state

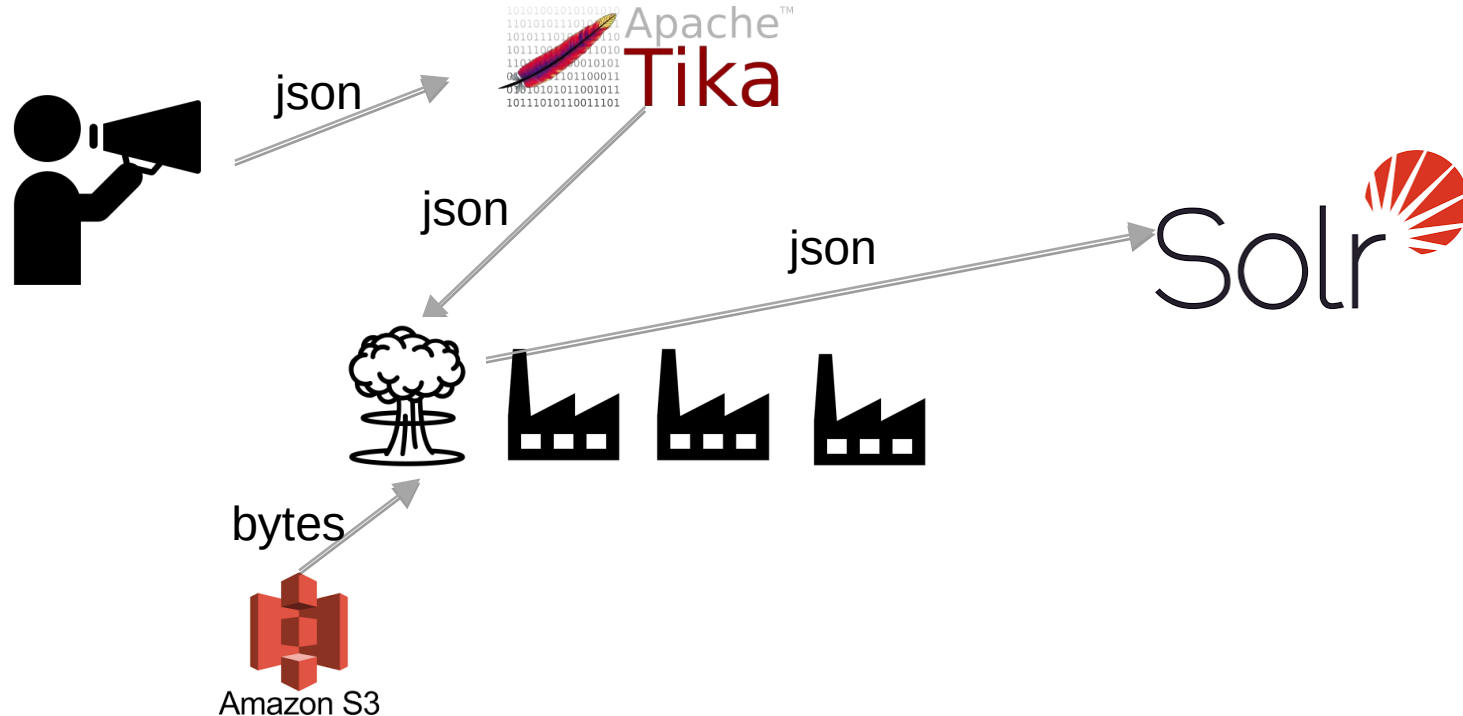
An example with Amazon S3 -> Apache Solr



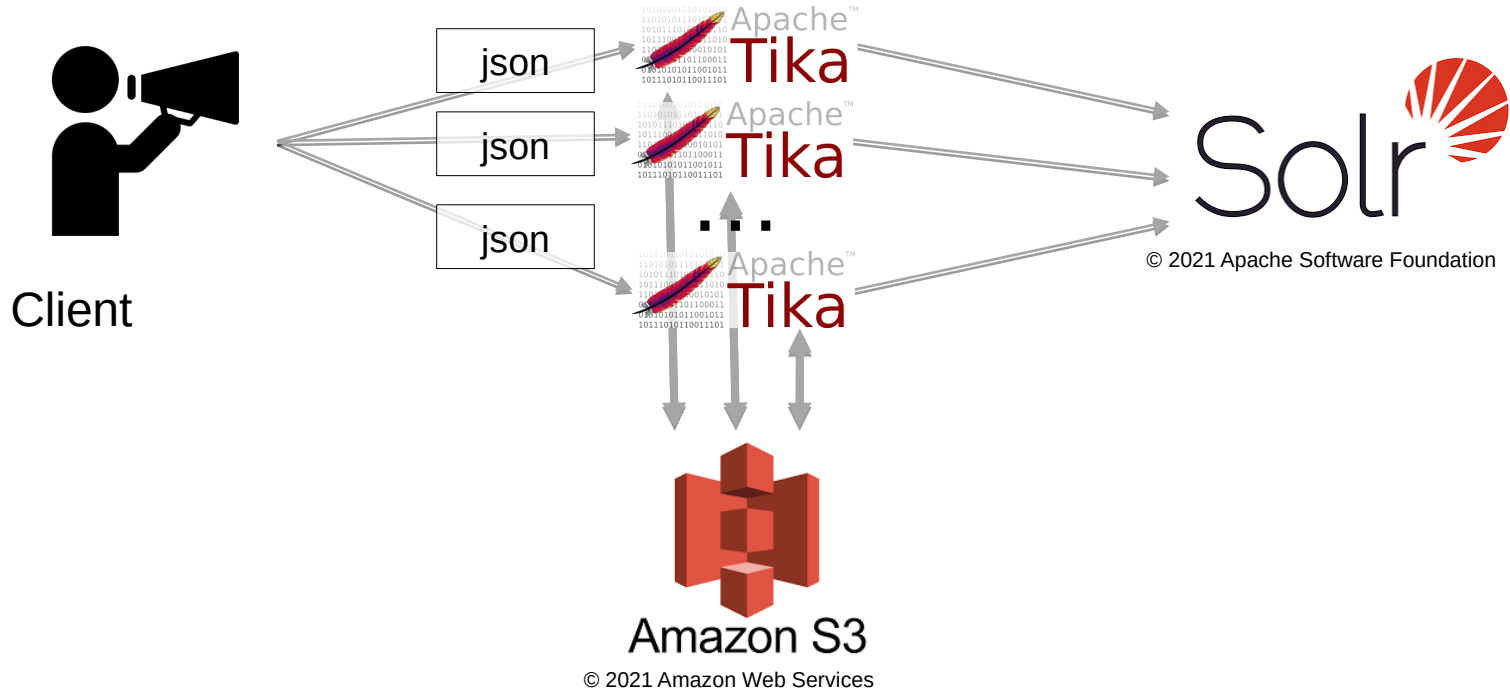
Fetchers/emitters: Process isolation per parse



Fetchers/emitters: Process isolation per parse



Fetchers/emitters: Scaling across a cluster



AsyncClient and FetchIterators

```
<!-- emitter -->  
<pipesIterator class="org.apache.tika.pipes.pipesiterator.FileSystemPipesIterator">  
  <params>  
    <basePath>/path/to/docs</basePath>  
    <fetcherName>fsf</fetcherName>  
    <emitterName>solr1</emitterName>  
  </params>  
</pipesIterator>
```


Async next steps...

- tika-batch will be replaced with the AsyncProcessor
- More tests
- Documentation

TODO

- More integration tests, esp. w Dockerized/Mocked S3, Solr, Opensearch
- Figure out a way to package jars for tika-server
- Documentation

Please join the fun!

- <https://tika.apache.org/>
- <https://cwiki.apache.org/confluence/display/TIKA/TikaEval>
- corpora-dev@tika.apache.org
- <https://issues.apache.org/jira/projects/TIKA>

- @ApacheTika

Questions?

timothy.b.allison@jpl.nasa.gov
@_tallison



Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov