What's new with Apache Tika?

APACHE: BIG_DATA EUROPE

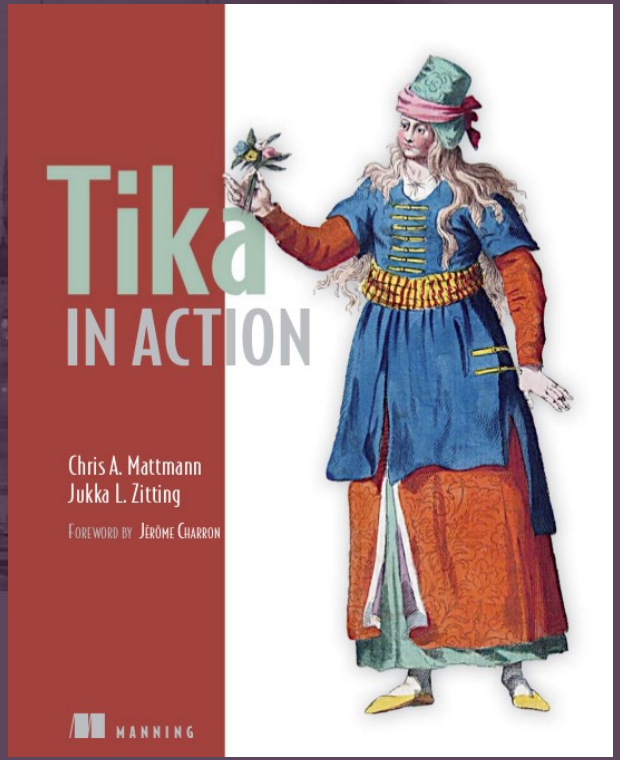
What's New with Apache Tika?

Nick Burch @Gagravarr
CTO, Quanticate

Tika, in a nutshell

“small, yellow and leech-like, and probably the oddest thing in the Universe”

- Like a Babel Fish for content!
- Helps you work out what sort of thing your content (1s & 0s) is
- Helps you extract the metadata from it, in a consistent way
- Lets you get a plain text version of your content, eg for full text indexing
- Provides a rich (XHTML) version too



A bit of history

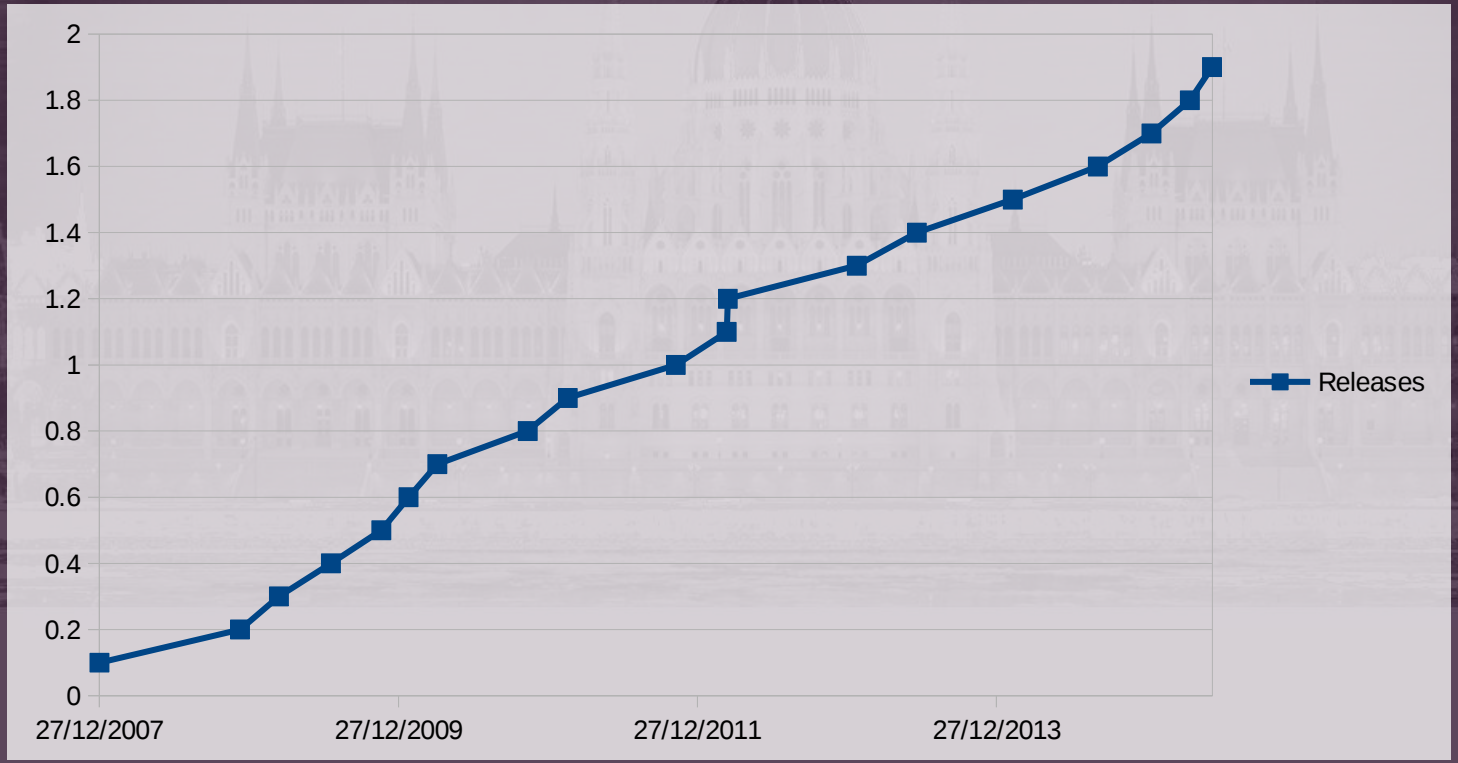
Before Tika

- In the early 2000s, everyone was building a search engine / search system for their CMS / web spider / etc
- Lucene mailing list and wiki had lots of code snippets for using libraries to extract text
- Lots of bugs, people using old versions, people missing out on useful formats, confusion abounded
- Handful of commercial libraries, generally expensive and aimed at large companies and/or computer forensics
- Everyone was re-inventing the wheel, and doing it badly....

Tika's History (in brief)

- The idea from Tika first came from the Apache Nutch project, who wanted to get useful things out of all the content they were spidering and indexing
- The Apache Lucene project (which Nutch used) were also interested, as lots of people there had the same problems
- Ideas and discussions started in 2006
- Project founded in 2007, in the Apache Incubator
- Initial contributions from Nutch, Lucene and Lius
- Graduated in 2008, v1.0 in 2011

Tika Releases



A (brief) introduction to Tika

(Some) Supported Formats

- HTML, XHTML, XML
- Microsoft Office – Word, Excel, PowerPoint, Works, Publisher, Visio – Binary and OOXML formats
- OpenDocument (OpenOffice)
- iWorks – Keynote, Pages, Numbers
- PDF, RTF, Plain Text, CHM Help
- Compression / Archive – Zip, Tar, Ar, 7z, bz2, gz etc
- Atom, RSS, ePub Lots of Scientific formats
- Audio – MP3, MP4, Vorbis, Opus, Speex, MIDI, Wav
- Image – JPEG, TIFF, PNG, BMP, GIF, ICO

Detection

- Work out what kind of file something is
- Based on a mixture of things
  - Filename
  - Mime magic (first few hundred bytes)
  - Dedicated code (eg containers)
  - Some combination of all of these
- Can be used as a standalone – what is this thing?
- Can be combined with parsers – figure out what this is, then find a parser to work on it

Metadata

- Describes a file
  - eg Title, Author, Creation Date, Location
- Tika provides a way to extract this (where present)
- However, each file format tends to have its own kind of metadata, which can vary a lot
  - eg Author, Creator, Created By, First Author, Creator[0]
- Tika tries to map file format specific metadata onto common, consistent metadata keys
- “Give me the thing that closest represents what Dublin Core defines as Creator”

Plain Text

- Most file formats include at least some text
- For a plain text file, that's everything in it!
- For others, it's only part
- Lots of libraries out there which can extract text, but how you call them varies a lot
- Tika wraps all that up for you, and gives consistentency

- Plain Text is ideal for things like Full Text Indexing, eg to feed into SOLR, Lucene or ElasticSearch

XHTML

- Structured Text extraction
- Outputs SAX events for the tags and text of a file
- This is actually the Tika default, Plain Text is implemented by only catching the Text parts of the SAX output
- Isn't supposed to be the “exact representation”
- Aims to give meaningful, semantic but simple output
- Can be used for basic previews
- Can be used to filter, eg ignore header + footer then give remainder as plain text

What's New?

APACHE:
BIG_DATA
EUROPE

Formats and Parsers

Supported Formats

- HTML
- XML
- Microsoft Office
  - Word
  - PowerPoint
  - Excel (2,3,4,5,97+)
  - Visio
  - Outlook

Supported Formats

- Open Document Format (ODF)
- iWorks
- PDF
- ePUB
- RTF
- Tar, RAR, AR, CPIO, Zip, 7Zip, Gzip, BZip2, XZ and Pack200
- Plain Text
- RSS and Atom

Supported Formats

- IPTC ANPA Newswire
- CHM Help
- Wav, MIDI
- MP3, MP4 Audio
- Ogg Vorbis, Speex, FLAC, Opus
- PNG, JPG, BMP, TIFF, BPG
- FLV, MP4 Video
- Java classes

Supported Formats

- Source Code
- Mbox, RFC822, Outlook PST, Outlook MSG, TNEF
- DWG CAD
- DIF, GDAL, ISO-19139, Grib, HDF, ISA-Tab, NetCDF, Matlab
- Executables (Windows, Linux, Mac)
- Pkcs7
- SQLite
- Microsoft Access

APACHE:
BIG_DATA
EUROPE

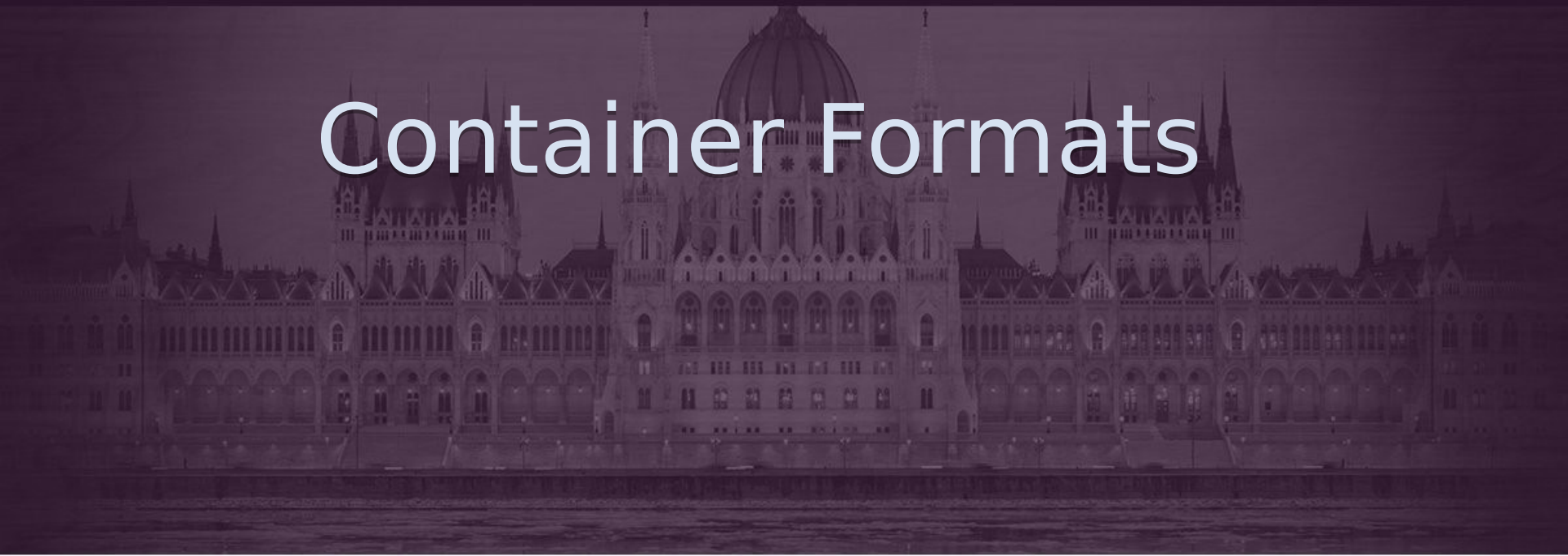
OCR

OCR

- What if you don't have a text file, but instead a photo of some text? Or a scan of some text?
- OCR (Optical Character Recognition) to the rescue!

- Tesseract is an Open Source OCR tool
- Tika has a parser which'll call out to Tesseract for suitable images found
- Tesseract is found and used if on the path
- Explicit path can be given, or can be disabled

APACHE:
BIG_DATA
EUROPE



Container Formats

APACHE:
BIG_DATA
EUROPE

Databases

Databases

- A surprising number of Database and “database” systems have a single-file mode
- If there's a single file, and a suitable library or program, then Tika can get the data out!
- Main ones so far are MS Access & SQLite

- How best to represent the contents in XHTML?
- One HTML table per Database Table best we have, so far!

APACHE:
BIG_DATA
EUROPE

Tika Config XML

Tika Config XML

- Using Config, you can specify what Parsers, Detectors, Translator, Service Loader and Mime Types to use
- You can do it explicitly
- You can do it implicitly (with defaults)
- You can do “default except”

- Tools available to dump out a running config as XML
- Use the Tika App to see what you have

Tika Config XML example

```
<?xml version="1.0" encoding="UTF-8"?>
<properties>
  <parsers>
    <parser class="org.apache.tika.parser.DefaultParser">
      <mime-exclude>image/jpeg</mime-exclude>
      <mime-exclude>application/pdf</mime-exclude>
      <parser-exclude
class="org.apache.tika.parser.executable.ExecutableParser"/>
    </parser>
    <parser class="org.apache.tika.parser.EmptyParser">
      <mime>application/pdf</mime>
    </parser>
  </parsers>
</properties>
```

APACHE:
BIG_DATA
EUROPE

Embedded Resources

APACHE:
BIG_DATA
EUROPE

Tika App

APACHE:
BIG_DATA
EUROPE

Tika Server

APACHE:
BIG_DATA
EUROPE

OSGi

Tika Batch

APACHE:
BIG_DATA
EUROPE

Apache cTAKES

APACHE:
BIG_DATA
EUROPE

Troubleshooting

Troubleshooting

- http://wiki.apache.org/tika/Troubleshooting%20Tika

What's Coming Soon?

APACHE:
BIG_DATA
EUROPE

Apache Tika 1.11

Tika 1.11

- Library upgrades for bug fixes (POI, PDFBox etc)
- Tika Config XML enhancements
- Tika Config XML output / dumping
- Apache Commons IO used more widely
- GROBID

- Hopefully due in a few weeks!

APACHE:
BIG_DATA
EUROPE

Apache Tika 1.12+

Tika 1.12+

- Commons IO in Core? TBD
- Java 7 Paths – where java.io.File used
- More NLP enhancement / augmentation
- Metadata aliasing

- Plus preparations for Tika 2

Tika 2.0

Why no Tika v2 yet?

- Apache Tika 0.1 – December 2007
- Apache Tika 1.0 – November 2011
- Shouldn't we have had a v2 by now?

- Discussions started several years ago, on the list
- Plans for what we need on the wiki for ~1 year

- Largely though, every time someone came up with a breaking feature for 2.0, a compatible way to do it was found!

Deprecated Parts

- Various parts of Tika have been deprecated over the years
- All of those will go!

- Main ones that might bite you:
  - Parser parse with no ParseContext
  - Old style Metadata keys

Metadata Storage

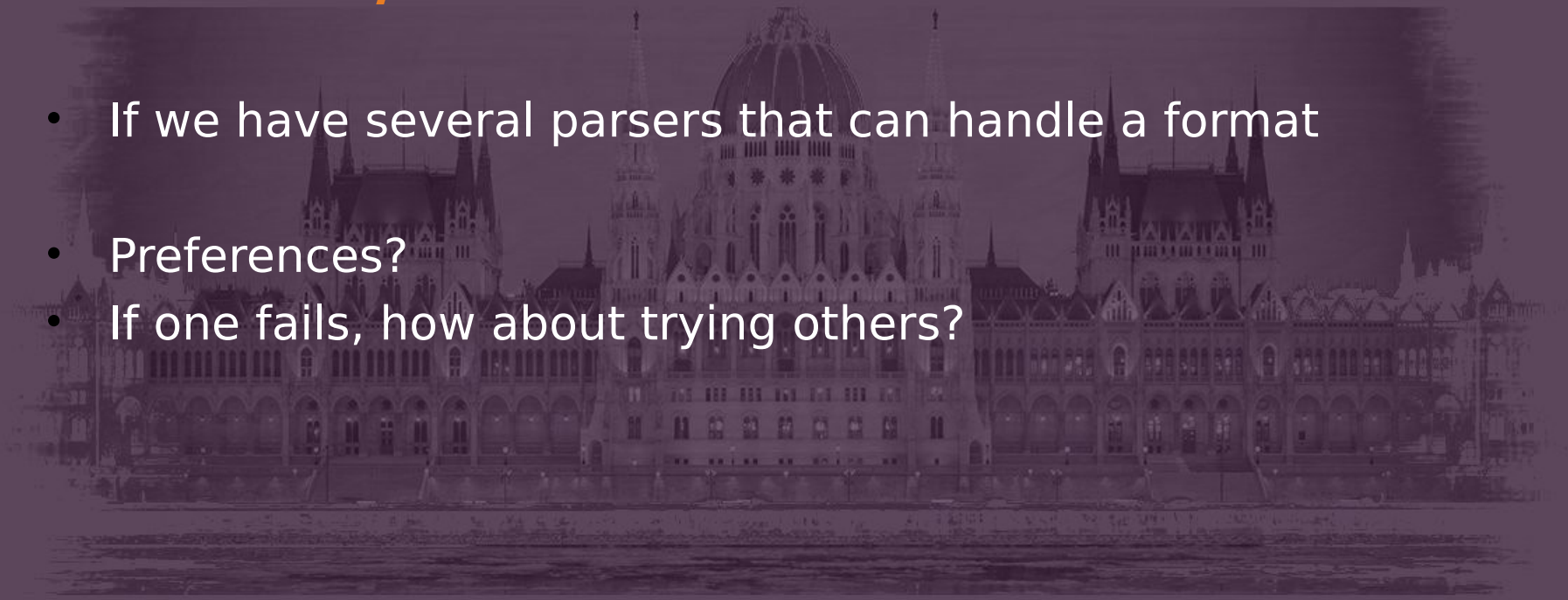
- Currently, Metadata in Tika is String Key/Value Lists
- Many Metadata types have Properties, which provide typing, conversions, sanity checks etc
- But all still stored as String Key + Value(s)

- Some people think we need a richer storage model
- Others want to keep it simple!
- JSON, XML DOM, XMP being debated
- Richer string keys also proposed

Java Packaging of Tika

- Maven Packages of Tika are
  - Tika Core
  - Tika Parsers
  - Tika Bundle
  - Tika XMP
  - Tika Java 7
- For just some parsers, you need to exclude maven dependencies
- Should we have “Tika Parser PDF”, “Tika Parsers ODF” etc?

Fallback/Preference Parsers



- If we have several parsers that can handle a format
- Preferences?
- If one fails, how about trying others?

Multiple Parsers

- If we have several parsers that can handle a format

- What about running all of them?
- eg extract image metadata
- then OCR it
- then try a second parser for more metadata

Parser Discovery/Loading?

- Currently, Tika uses a Service Loader mechanism to find and load available Parsers (and Detectors+Translators)
- This allows you to drop a new Tika parser jar onto the classpath, and have it automatically used
- Also allows you to miss one or two jars out, and not get any content back with no warnings / errors...
- You can set the Service Loader to Warn, or even Error
- But most people don't, and it bites them!
- Change the default in 2? Or change entirely how we do it?

What we still need help with...

Content Handler Reset/Add

- Tika uses the SAX Content Handler interface for supplying plain text
- Streaming, write once

- How does that work with multiple parsers?

Content Enhancement

- How can we post-process the content to “enhance” it in various ways?
- For example, how can we mark up parts of speach?
- Pull out information into the Metadata?
- Translate it, retaining the original positions?
- For just some formats, or for all?
- For just some documents in some formats?
- While still keeping the Streaming SAX-like contract?

Metadata Standards

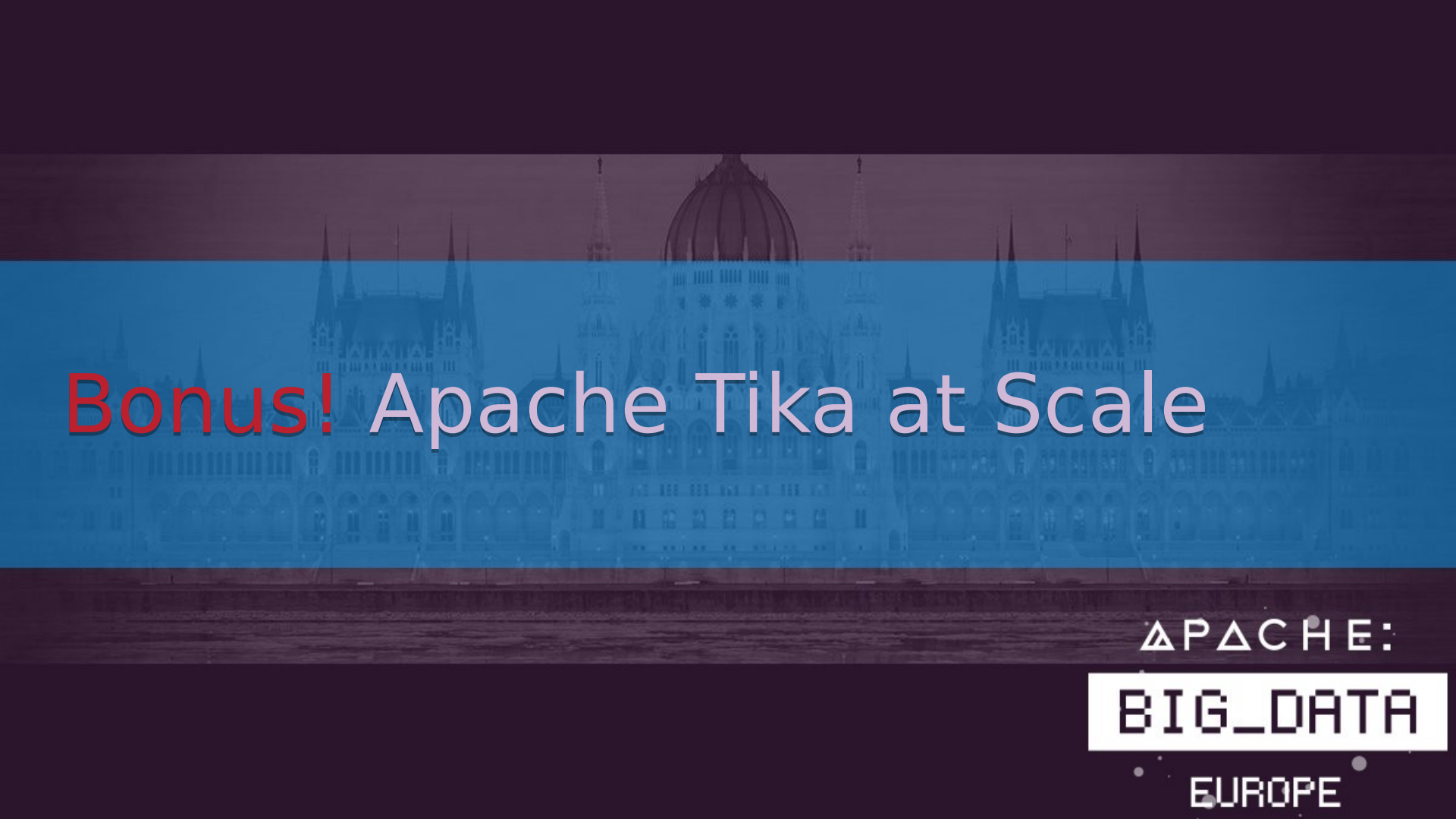
- Currently, Tika works hard to map file-format-specific metadata onto general metadata standards
- Means you don't have to know each standard in depth, can just say “give me the closest to dc:subject you have, no matter what file format or library it comes from”

- What about non-File-format metadata, such as content metadata (Table of Contents, Author information etc)?
- What about combining things?

Richer Metadata

- See Metadata Storage slides!

Bonus! Apache Tika at Scale



Lots of Data is Junk

- At scale, you're going to hit lots of edge cases
- At scale, you're going to come across lots of junk or corrupted documents
- 1% of a lot is still a lot...
- 1% of the internet is a huge amount!
- Bound to find files which are unusual or corrupted enough to be mis-identified
- You need to plan for failures!

Unusual Types

- If you're working on a big data scale, you're bound to come across lots of valid but unusual + unknown files
- You're never going to be able to add support for all of them!
- May be worth adding support for the more common “uncommon” unsupported types
- Which means you'll need to track something about the files you couldn't understand
- If Tika knows the mimetype but has no parser, just log the mimetype
- If mimetype unknown, maybe log first few bytes

Failure at Scale

- Tika will sometimes mis-identify something, so sometimes the wrong parser will run and object
- Some files will cause parsers or their underlying libraries to do something silly, such as use lots of memory or get into loops with lots to do
- Some files will cause parsers or their underlying libraries to OOM, or infinite loop, or something else bad
- If a file fails once, will probably fail again, so blindly just re-running that task again won't help

Failure at Scale, continued

- You'll need approaches that plan for failure
- Consider what will happen if a file locks up your JVM, or kills it with an OOM
- Forked Parser may be worth using
- Running a separate Tika Server could be good
- Depending on work needed, could have a smaller pool of Tika Server instances for big data code to call
- Think about failure modes, then think about retries (or not)
- Track common problems, report and fix them!

Bonus! Tika Batch, Eval & Hadoop

Tika Batch – TIKA-1330

- Aiming to provide a robust Tika wrapper, that handles OOMs, permanent hangs, out of file handles etc
- Should be able to use Tika Batch to run Tika against a wide range of documents, getting either content or an error
- First focus was on the Tika App, with adisk-to-disk wrapper
- Now looking at the Tika Server, to have it log errors, provide a watchdog to restart after serious errors etc
- Once that's all baked in, refactor and fully-hadoop!
- Accept there will always be errors! Work with that

Tika Batch Hadoop

- Now we have the basic Tika Batch working – Hadoop it!
- Aiming to provide a full Hadoop Tika Batch implementation
- Will process a large collection of files, providing either Metadata+Content, or a detailed error of failure
- Failure could be machine/enivornment, so probably need to retry a failure incase it isn't a Tika issue!
- Will be partly inspired by the work Apache Nutch does
-
- Tika will “eat our own dogfood” with this, using it to test for regressions / improvements between versions

Tika Eval – TIKA-1332

- Building on top of Tika Batch, to work out how well / badly a version of Tika does on a large collection of documents
- Provide comparible profiling of a run on a corpus
- Number of different file types found, number of exceptions, exceptions by type and file type, attachements etc
- Also provide information on langauge stats, and junk text
- Identify file types to look at supporting
- Identify file types / exceptions which have regressed
- Identify exceptions / problems to try to fix
- Identify things for manual review, eg TIKA-1442 PDFBox bug

Batch+Eval+Public Datasets

- When looking at a new feature, or looking to upgrade a dependency, we want to know if we have broken anything
- Unit tests provide a good first-pass, but only so many files
- Running against a very large dataset and comparing before/after the best way to handle it

- Initially piloting + developing against the Govdocs1 corpus http://digitalcorpora.org/corpora/govdocs
- Using donated hosting from Rackspace for trying this
- Need newer + more varied corpuses as well! Know of any?


Any Questions?
APACHE:
BIG_DATA
EUROPE

APACHE:
BIG_DATA
EUROPE

Nick Burch

@Gagravarr

nick@apache.org