

# Versus Framework

Luigi Marini

December 13<sup>th</sup>, 2013



National Center for Supercomputing Applications  
University of Illinois at Urbana-Champaign

# Content-Based Comparison

- Goal: Comparing digital data
- Given two or more digital objects establish their proximity
- Arbitrary?
- Not really, comparing two files, videos, documents, etc. has many applications, for example:

# Some Applications

- Information loss
  - Information loss when applying file format conversion
  - Polyglot
- Content-based retrieval
  - Given a multimedia file (image) find the closest ones in a large collection
- Find duplicates
  - Across formats

# Census Information Retrieval

Query:

Illinois

Collection:

Missouri

Penn

Illinois

None

Kansas



# History



- Funding by National Archives and Records Administration (NARA)
- Research and development started in 2010
- Originally focused on pairwise comparison
- Adding support for the creation of indexes over past two years
- Current version is 0.6
  - Usable but still in flux
  - Particular important with APIs
  - Lots of exploratory work over the years

# Two Main Components

## Core

- A set of Java interfaces
- Multithreaded Execution Engine
- Registry to register and query for methods

## Web Service

- HTTP API wrapping Core
- Master/slave architecture

# Several Clients

- Command Line Interface
- Web Application
- Desktop App
- Medici 2

```
Terminal — bash — 115x11
faye:bin lmarini$ ./versus-cli.sh file1.tiff file2.tiff edu.illinois.ncsa.versus.adapter.impl.BytesAdapter edu.illinois.ncsa.versus.extract.impl.MD5Extractor edu.illinois.ncsa.versus.measure.impl.MD5DistanceMeasure
DEBUG [main] (ComputeThread.java:67) - Selected adapter is edu.illinois.ncsa.versus.adapter.impl.BytesAdapter
DEBUG [main] (ComputeThread.java:70) - Selected extractor is edu.illinois.ncsa.versus.extract.impl.MD5Extractor
DEBUG [main] (ComputeThread.java:72) - Selected measure is edu.illinois.ncsa.versus.measure.impl.MD5DistanceMeasure
DEBUG [main] (ExecutionEngine.java:58) - Job submitted
DEBUG [pool-1-thread-1] (ComputeThread.java:130) - Compared file1.tiff with file2.tiff = 0.0
Comparison's result: 0.0
faye:bin lmarini$
```

The image shows two windows side-by-side. The left window is the Versus desktop application. It has a 'File' pane on the left with a list of image files: FernColIMAC.bmp, FernColIMAC.gif, FernColIMAC.jpg, FernColIMAC.png, and FernColIMAC.tif. The main area is titled 'Select Data Representation, Feature Extractor, Similarity Measure'. It has three dropdown menus: 'Data Representation' set to 'Image Object', 'Feature Extractor' set to 'Pixels to Array', and 'Similarity Measure' set to 'Euclidean Distance'. Below these is a 'Compute' button. At the bottom, there is a table titled 'Euclidean Distance' with columns for file names and similarity scores.

	FernColIMAC.b...	FernColIMAC.gif	FernCol
FernColIMAC.bmp	0.0		
FernColIMAC.gif	309.80245570...	0.0	
FernColIMAC.jpg	66.980320717...	317.27164117...	0.0
FernColIMAC.png	0.0	309.80245570...	66.9803
FernColIMAC.tif	0.0	309.80245570...	66.9803

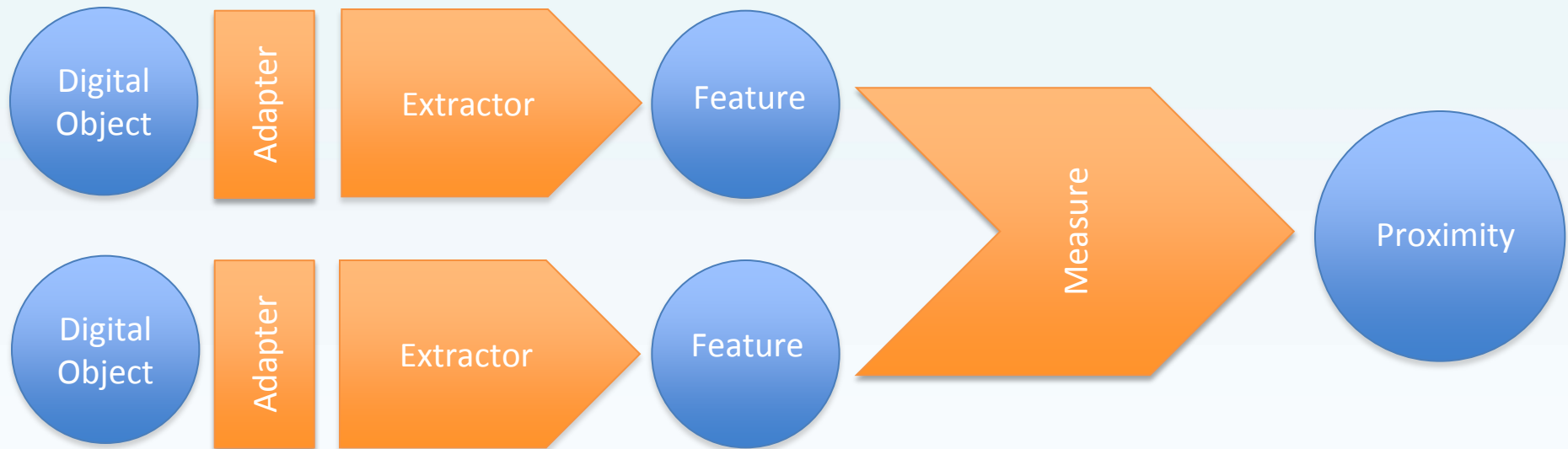
The right window is a Mozilla Firefox browser displaying the Versus web application. The address bar shows '129.6.18.151/versus-web/#'. The page has a navigation bar with 'Versus', 'Workflow', 'Data', 'Collections', and 'Upload'. Below this are tabs for 'Selected Data', 'Compare', and 'View Results'. The main content area is divided into three columns: 'Adapters', 'Extractors', and 'Measures'. Under 'Adapters', 'Image Object' is selected. Under 'Extractors', 'Pixels to Array' is selected. Under 'Measures', 'Euclidean Distance' is selected. At the bottom, there is a breadcrumb trail: 'Image Object --> Pixels to Array --> Euclidean Distance' and a 'Launch' button.

# Why would one use Versus instead of writing specific implementations as need be?

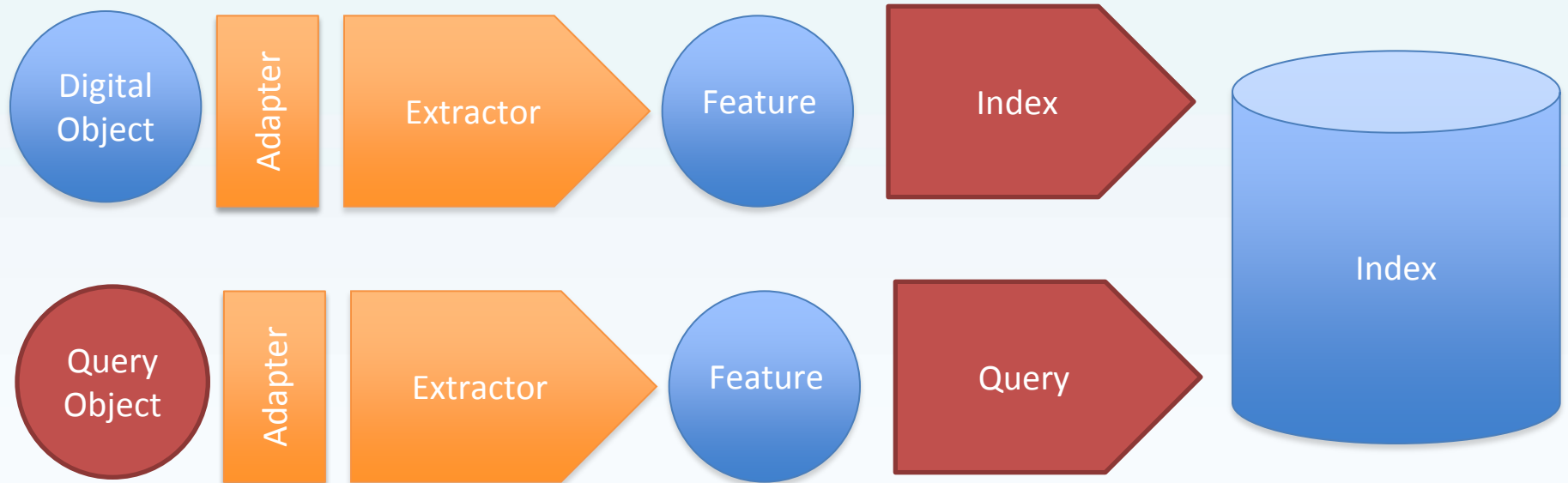
- **Reuse** existing methods
- **Share** methods with community
- **Organize** code in clear components
- **Leverage** execution environment and service infrastructure



# Pairwise Comparison API



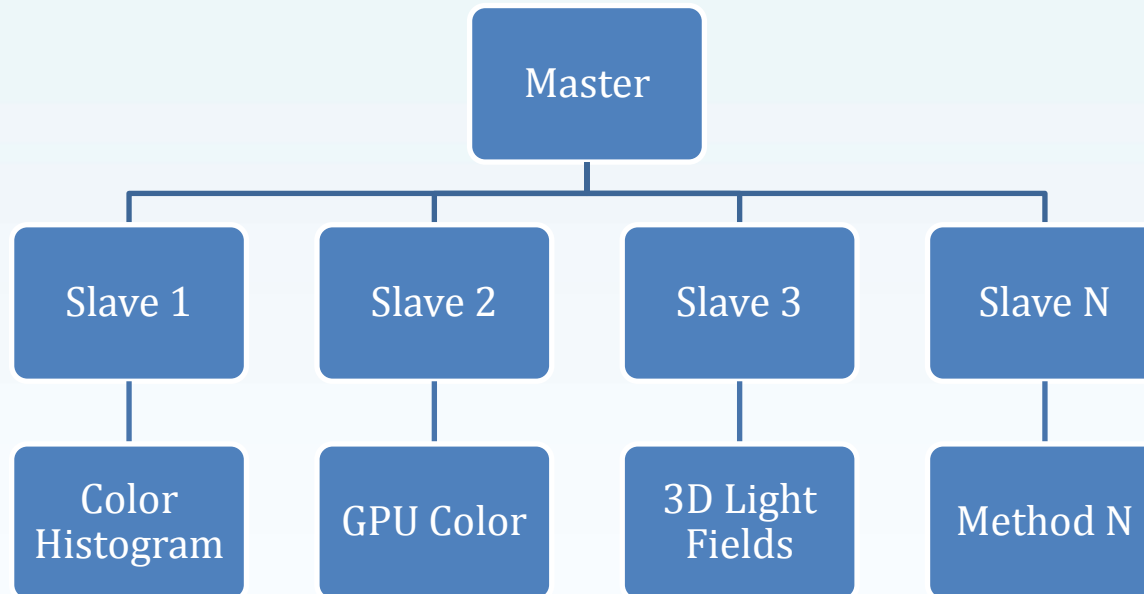
# Indexing API



# Service Demo

- <http://isda.ncsa.illinois.edu/documentation/versus/tutorial>

# Master/Slave



# Demo: Master/Slave configuration

- Starting and stopping slaves

# Storage

- Different implementations of persistence layer available
  - In memory
  - File system
  - Mysql
  - MongoDB

# Adding Implementations

- Write Java Code
- Can execute arbitrary code using
  - `Runtime.getRuntime().exec(args)`
  - JNI
- Register them using Java services
  - Add fully qualified class name to respective service file
  - For example add
    - `edu.illinois.ncsa.versus.extract.impl.RGBHistogramExtractor`
    - To
    - `/META-INF/services/edu.illinois.ncsa.versus.extract.Extractor`
    - Restart

# Demo: Deploying new methods



# Demo: Medici as a client

# Future Work

- Store intermediate data structures to disk
  - Caching between overlapping comparisons
- Service Reliability
  - Recovering if a node goes down
- Split steps across nodes
  - Ability to execute extractions and calculate measures on different nodes