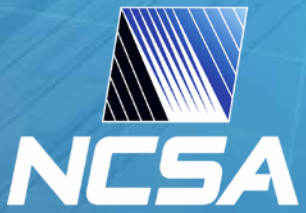
Versus Framework

Luigi Marini

December 13th, 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:

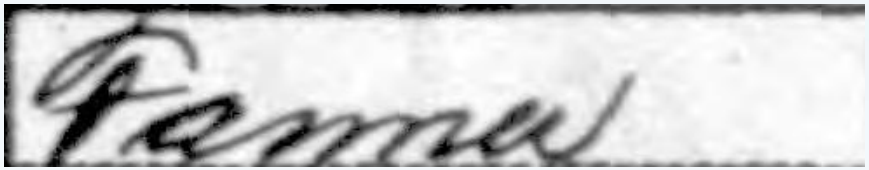
Some Applications

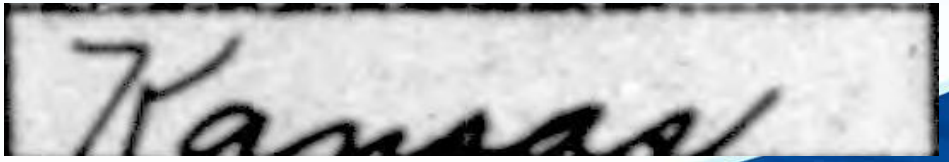
- Information loss
  - Information loss when applying file format conversion
  - Polyglot
- Content-based retrieval
  - Given a multimedia file (image) find the closest ones in a large collection
- Find duplicates
  - Across formats

Census Information Retrieval

Query:

Collection:





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

NCSA

Two Main Components

Core

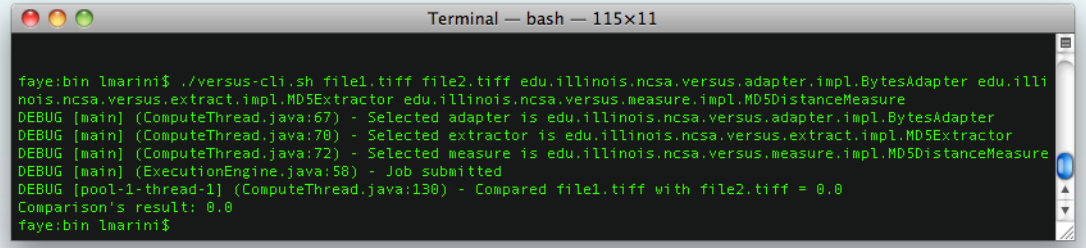
- A set of Java interfaces
- Multithreaded Execution Engine
- Registry to register and query for methods

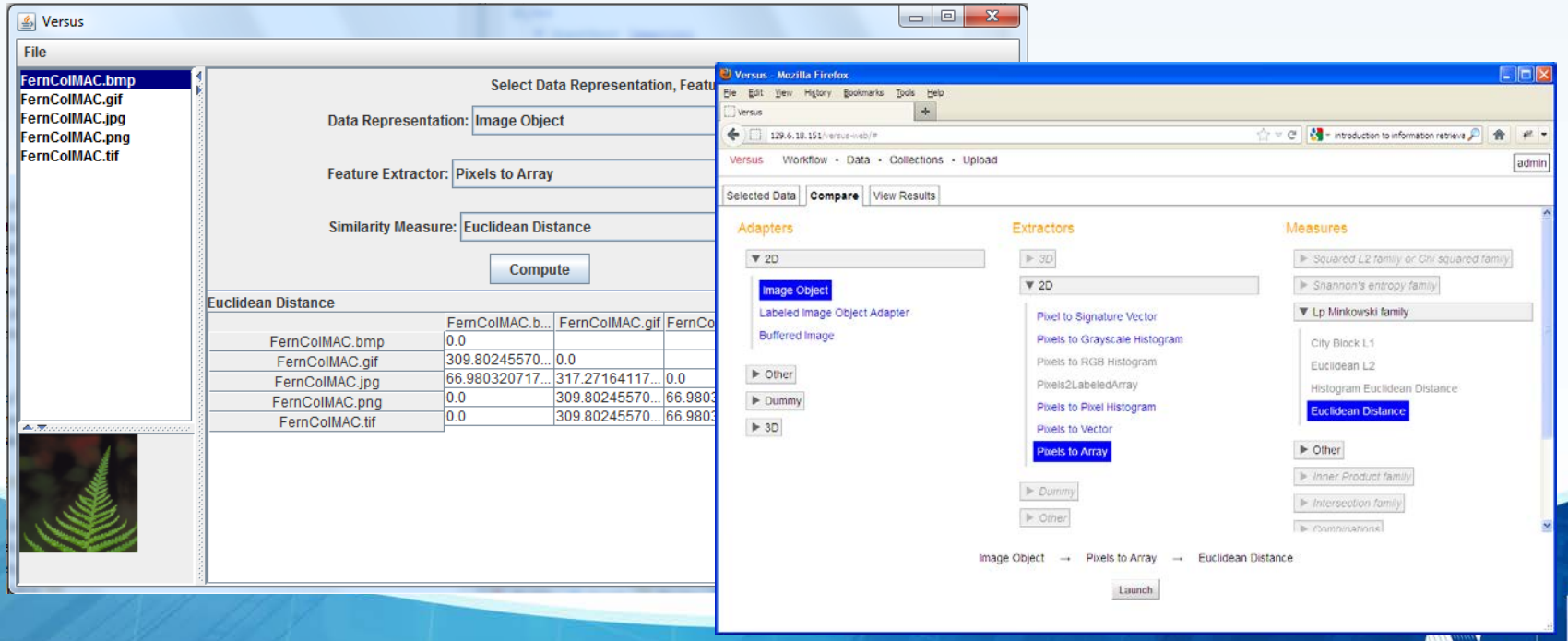
Web Service

- HTTP API wrapping Core
- Master/slave architecture

Several Clients

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



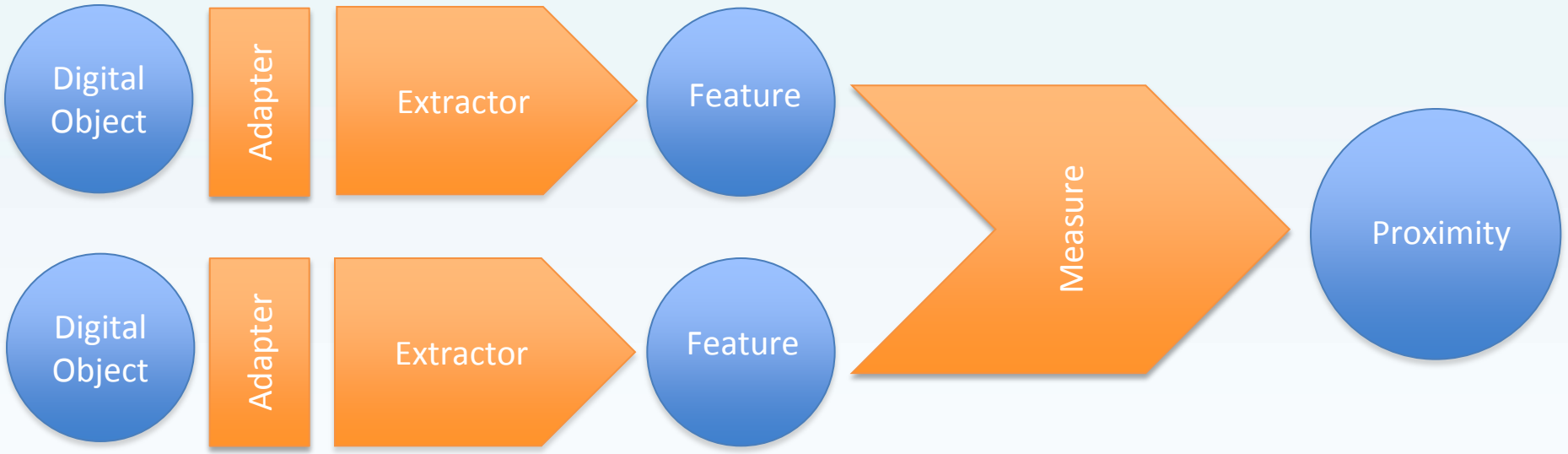


NCSA

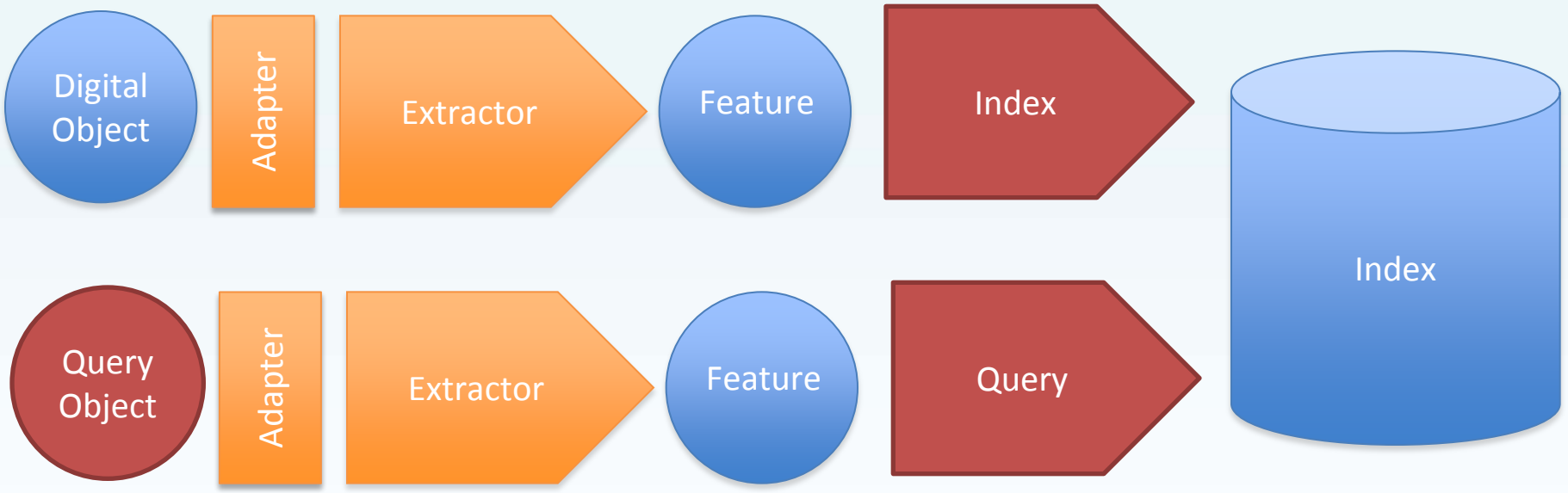
Why would one use Versus instead of writing specific implementations as need be?

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

Pairwise Comparison API



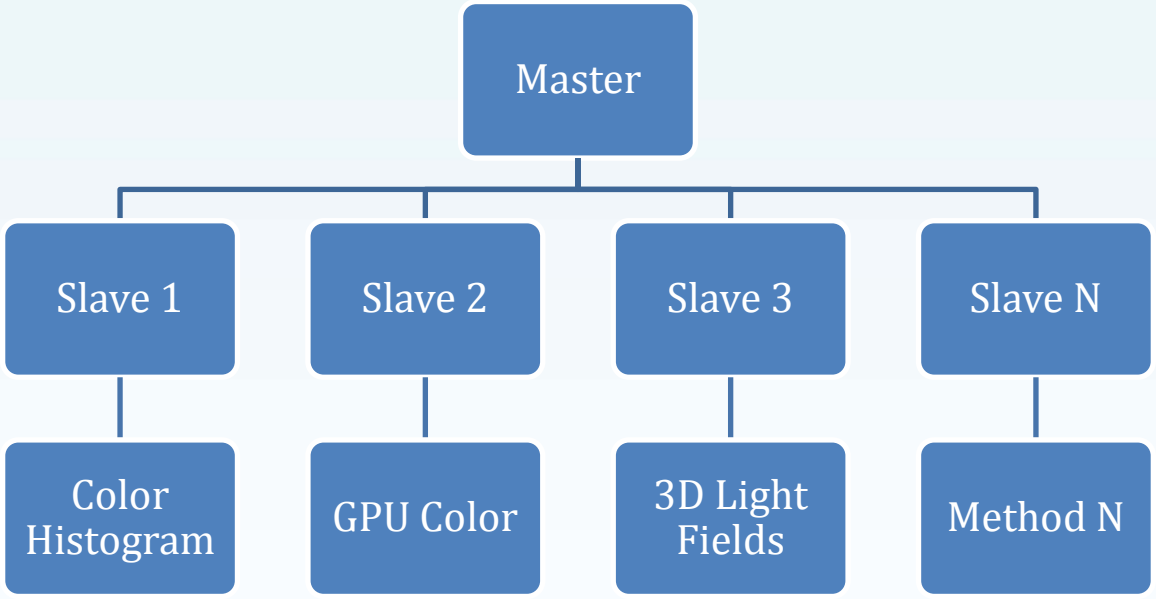
Indexing API



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