

# DAE: A Platform for Document Analysis Research

Dezhao Song  
Dec. 10, 2010

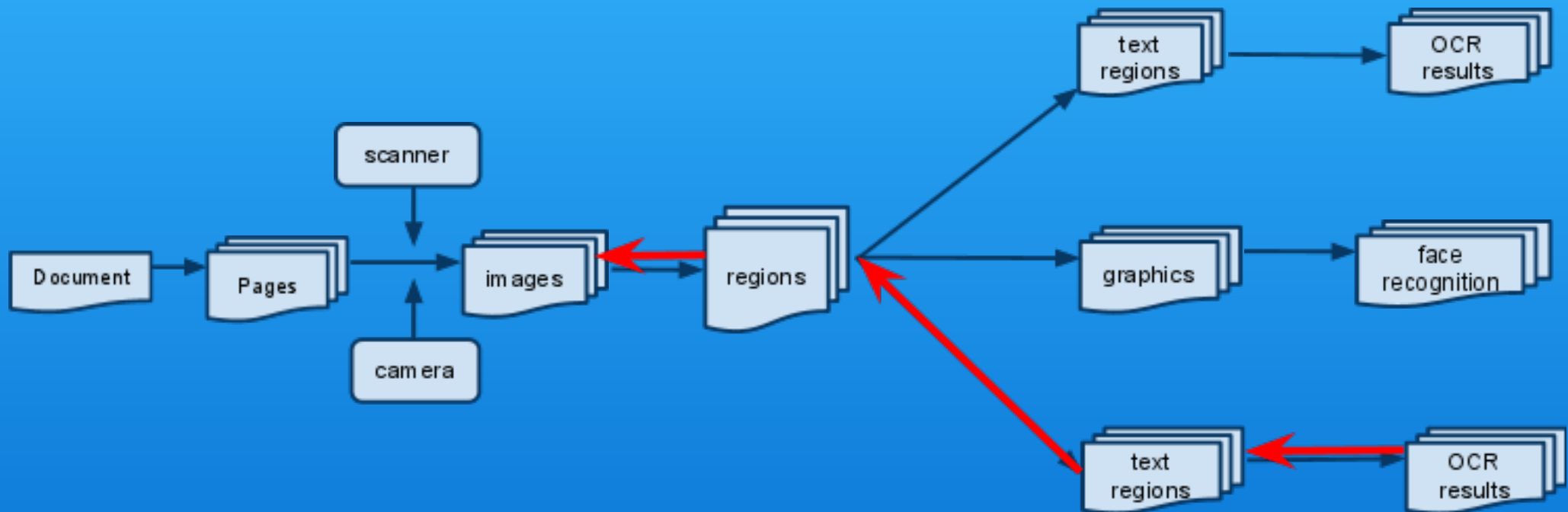
# Agenda

- Motivation
- System Architecture
- Database Design
  - ER Model
  - Data Upload
- Algorithm Execution
  - Local Binaries
  - Web Services
- System Demo
- Conclusion

# Motivation

- Build a shared platform for researchers to perform research of topic X.
  - But why?
  - Keep track of activities and data provenance
    - Algorithm executions
    - Data associations
  - Reference data from databases
    - Store data and their associations
- In the DAE project, this X is Document Analysis

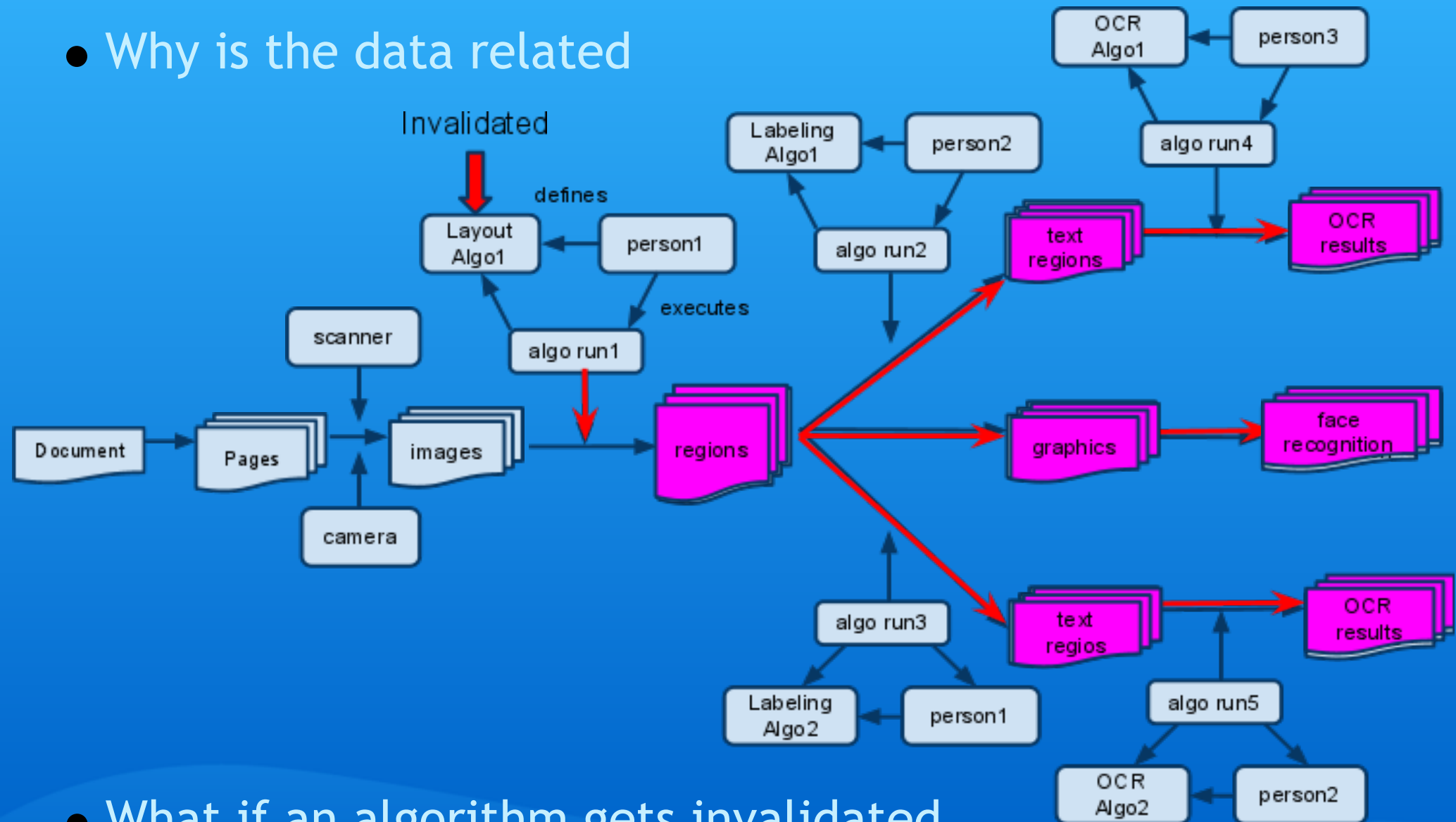
# A Typical Document Analysis Workflow



- Sample query:
  - Return images whose text regions contain some particular textual information.
- The data itself has certain associations

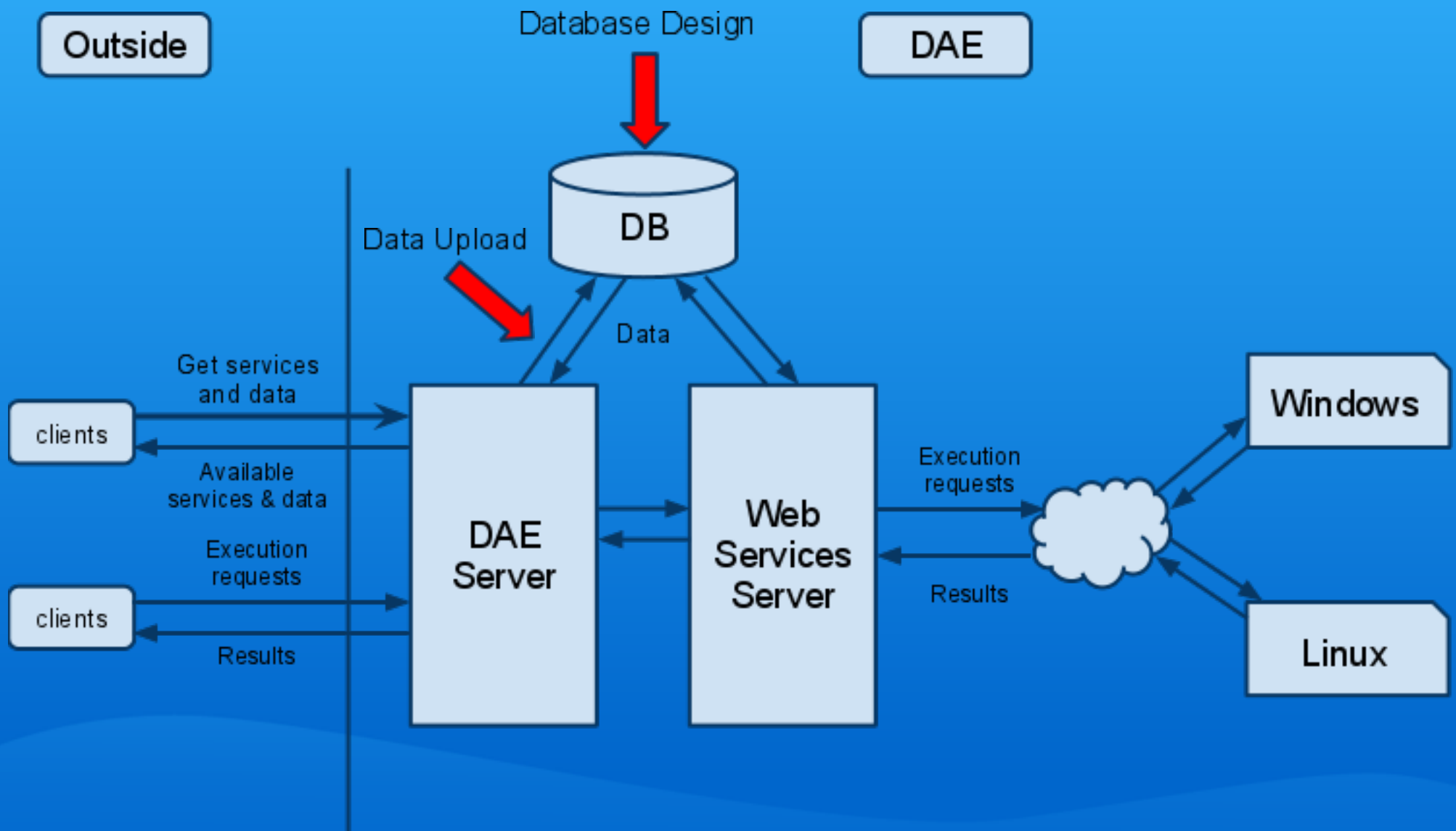
# Need More?

- Why is the data related

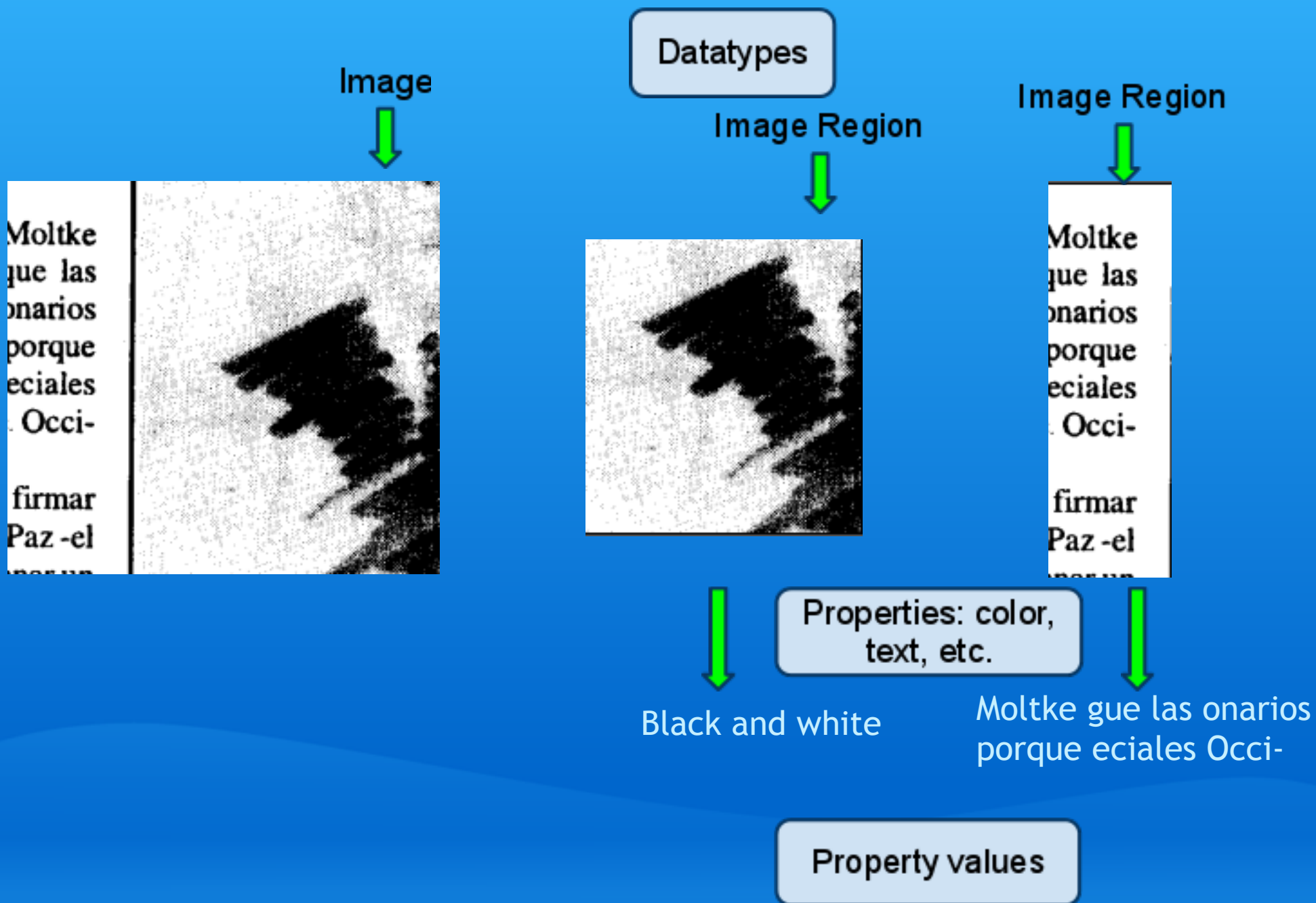


- What if an algorithm gets invalidated

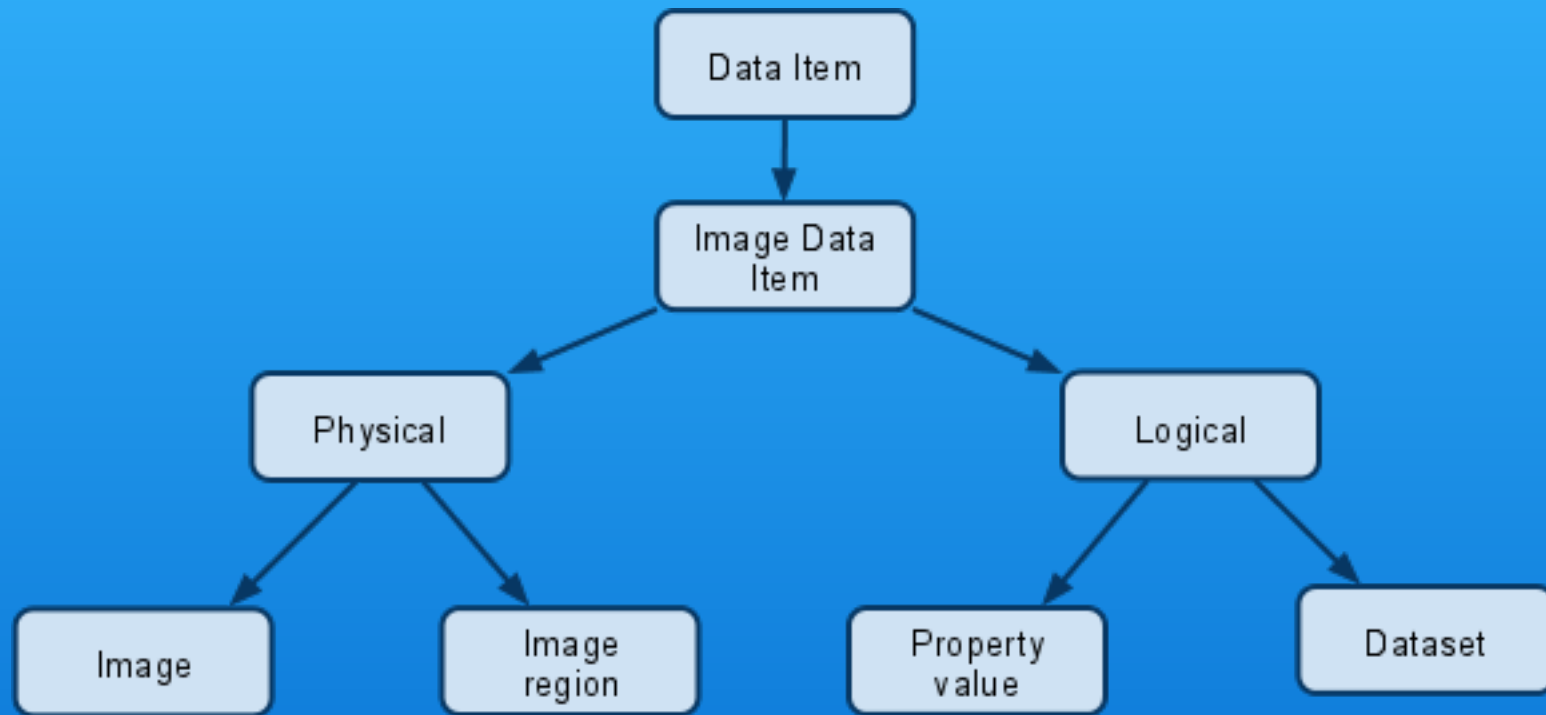
# System Architecture



# Database Design - Data Item

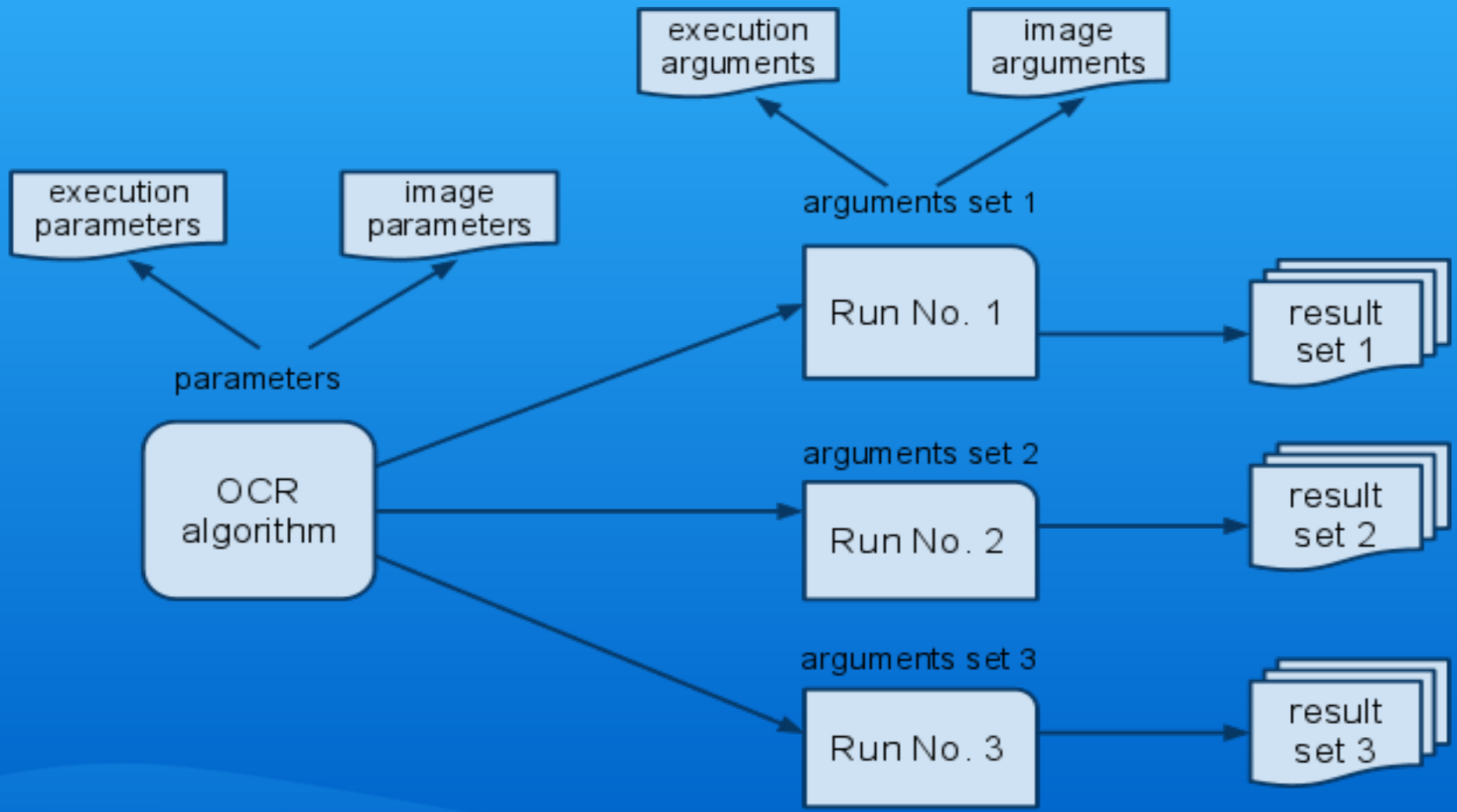


# Data Item Hierarchy

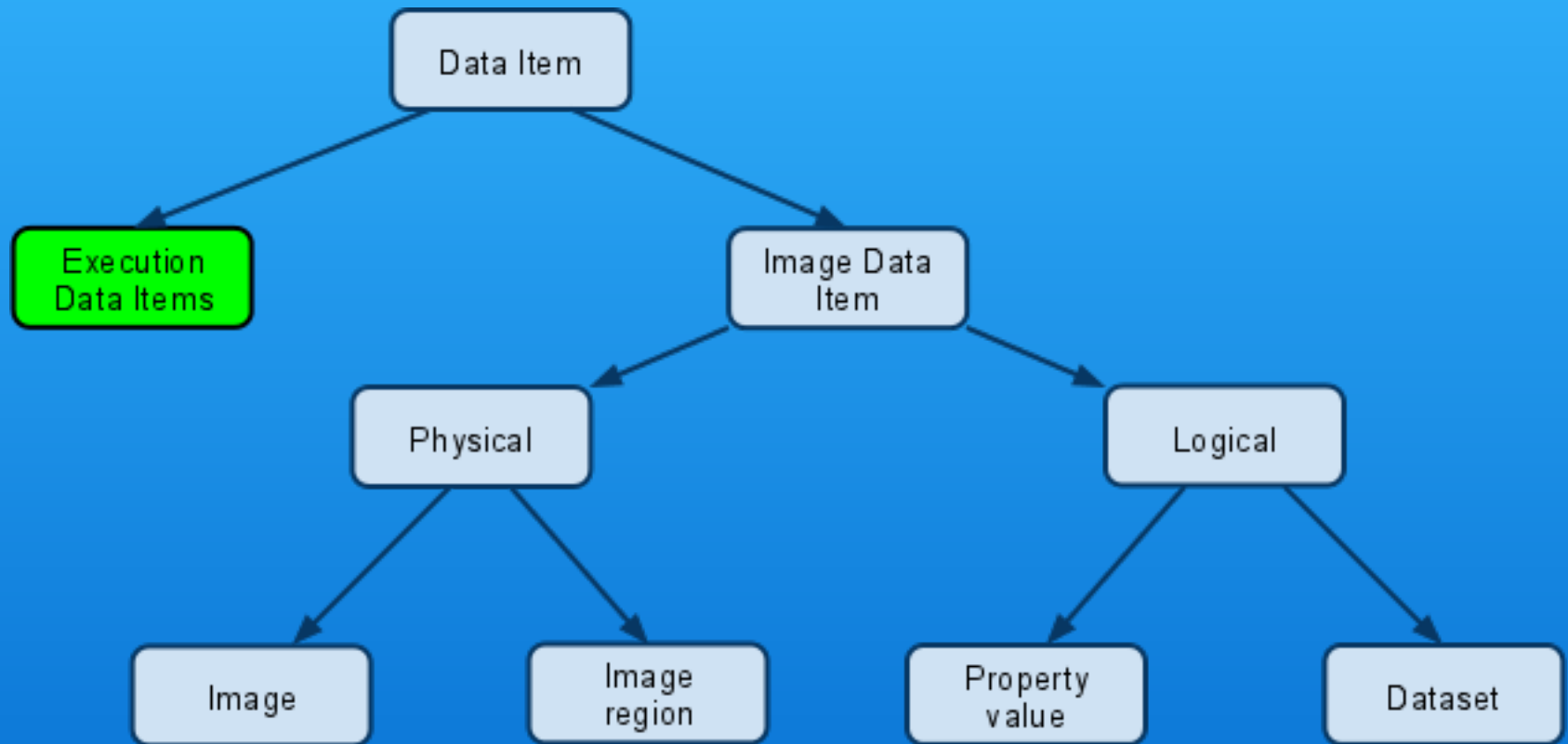


- Data Items are associated
  - Dataset contains images
  - Image includes regions
  - Regions have property values

# Database Design - Algorithm

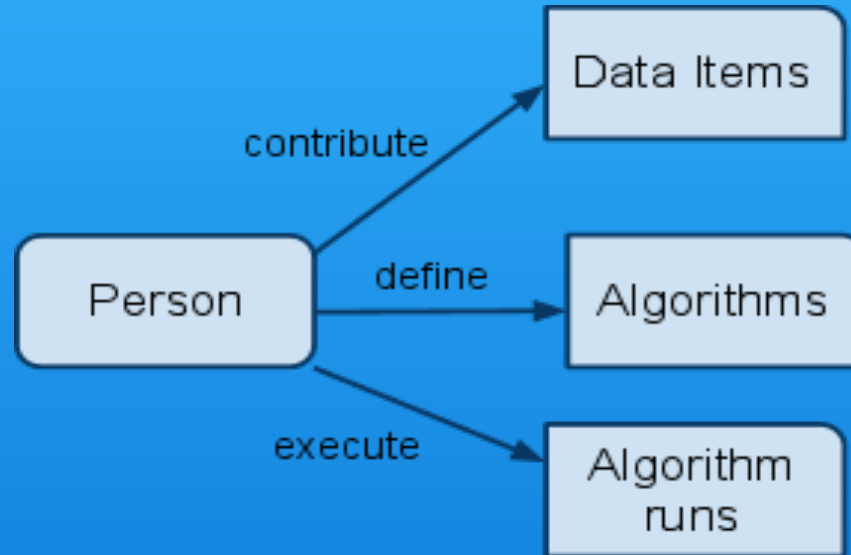


# Completing Data Item Hierarchy



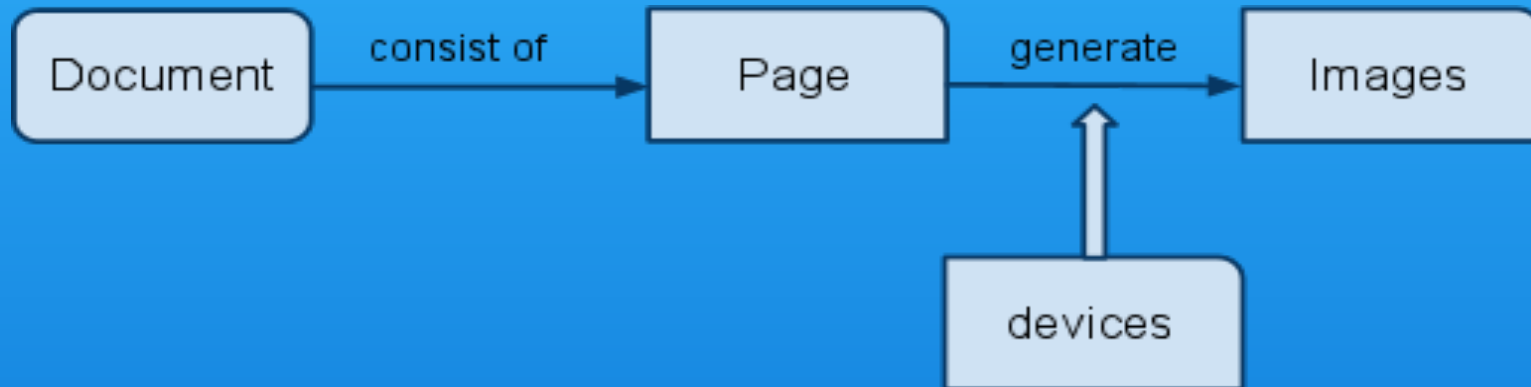
- More provenance information is now recored

# Person Entity



- People can
  - submit their algorithms.
  - contribute data items: images, regions (through executing algorithms).
  - all these will be recorded and be recognized by other users.

# Document Entity



- What is a document?
  - A book
  - A technical paper
  - A pile of pages about the same subject
- physical or logical?



# Contribute Data

- Data is fundamental to the system.
  - browsing
  - executing algorithms
  - examining provenance
- How to store data into the DAE database
  - SQL queries - sure!
  - But we should not expect users to totally understand the data model, form SQL queries and contribute data.
- The way users upload data should be user friendly and flexible.

# How to Upload Data

- We represent the ER model in XML (DAEX)
  - well structured and accepted
  - easier to understand
- Users that want to upload data need to represent the metadata with DAEX
  - Metadata represented in such XML files can be parsed and automatically stored into database
- Oracle XSU
  - Object relational view
  - Instead-Of trigger

# Sample Metadata File

```
<page_image>
  <id/>
  <path/>
  <hdpi/>
  <vdpi/>
  <width/>
  <height/>
  <skew/>
  <type_list>
    <type_list_item>
      <id/>
      <name/>
      <description/>
    </type_list_item>
  </type_list>
  <page_element_list>
    <page_element_list_item>
      <id/>
      <description/>
      <toleftx/>
      <tolefty/>
      <width/>
      <height/>
    </page_element_list_item>
  </page_element_list>
</page_image>
```

```
<dataset>
  <id/>
  <name/>
  <associating_dataset_list>
    <associating_dataset_list_item>
      <id/>
    </associating_dataset_list_item>
    <associating_dataset_list_item>
      <id/>
    </associating_dataset_list_item>
  </associating_dataset_list>
  <contributor_list>
    <contributor_list_item>
      <person_id/>
    </contributor_list_item>
  </contributor_list>
  <copyright_list>
    <copyright_list_item>
      <id/>
    </copyright_list_item>
  </copyright_list>
  <type_list>
    <type_list_item>
      <id/>
      <name/>
      <description/>
    </type_list_item>
  </type_list>
</dataset>
```

# Metadata Transformation

- There is existing data and relevant XML metadata in the document analysis domain
  - Owners of such data should not re-generate metadata from scratch.
- Transform with XSLT
  - APTI
  - GEDI
  - CVC
- How about non-XML metadata
  - Transform with additional programs
  - UNLV

# Web Services

- A recent accomplishment by other project members
  - Yingjie Li, Xingjian Zhang and Professor Bart Lamiroy
- Represent our currently hosted algorithms as web services
- Execute algorithms by calling web services
- Allow us to call algorithms that we don't host but registered in the database
- Web services Vs. local binaries

# A Sample Web Service

```
<service name="tesseractwsdl">
  <port name="tesseractwsdlPort" binding="tns:tesseractwsdlBinding">
    <soap:address location="http://dae-sbx.cse.lehigh.edu/wsdl/tesseract.php"/>
  </port>
</service>

<portType name="tesseractwsdlPortType">
  <operation name="callback">
    <input message="tns:callbackRequest"/>
    <output message="tns:callbackResponse"/>
  </operation>
</portType>

<message name="callbackRequest">
  <part name="args" type="tns:tesseractInput" />
</message>
<message name="callbackResponse">
  <part name="return" type="tns:tesseractOutput" />
</message>

<types>
  <xsd:complexType name="tesseractInput">
    <xsd:element name="tif_image"/>
  </xsd:complexType>
  <xsd:complexType name="tesseractOutput">
    <xsd:element name="page_image"/>
  </xsd:complexType>
</types>
```

# System Demo

- The implementation is done by Mike Kot, Yingjie Li, Weidong Chen, Xingjian Zhang and Yang Yu.
- Browsing
  - datasets
  - dataset associations
  - images
  - page elements
  - page element property values
- Executing local binaries
- Executing web services

# Conclusion and Impact

- The platform
- Database
  - Designed a comprehensive and flexible database schema
  - host and search data, analytic results and data provenance
- Semi-automatic upload utility
  - Users provide XML files conforming to our design
  - The rest is done automatically
- Set up web services for running algorithms
- A standard way for identifying input and output
  - Avoid biases in data selection
  - Reproduce experiment results
- Go forward based upon previous achievements
- Facilitate collaborations between document analysis researchers
- What if we want to transform to other research domains?