Advanced Data Management (CSCI 490/680)

Provenance

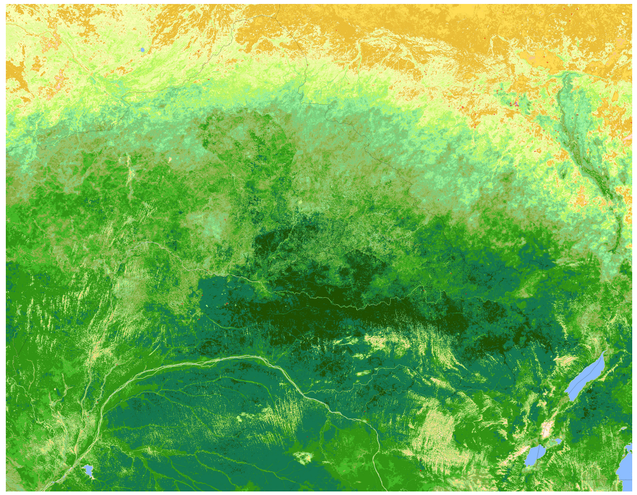
Dr. David Koop

Assignment 5

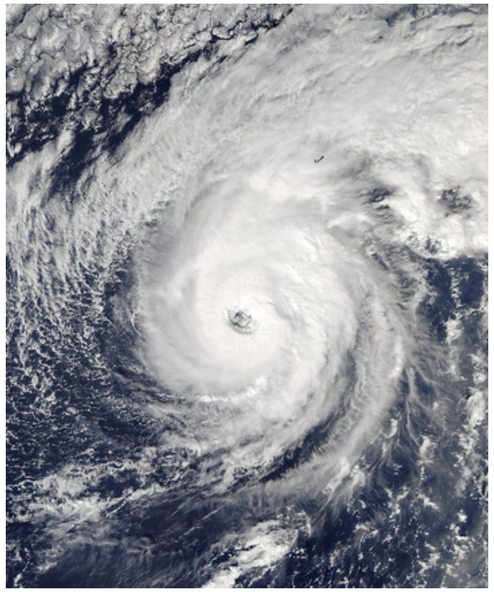
- Available soon
- Work with time series and spatial data
- Shorter assignment
- Due at the end of the semester

Spatial Data

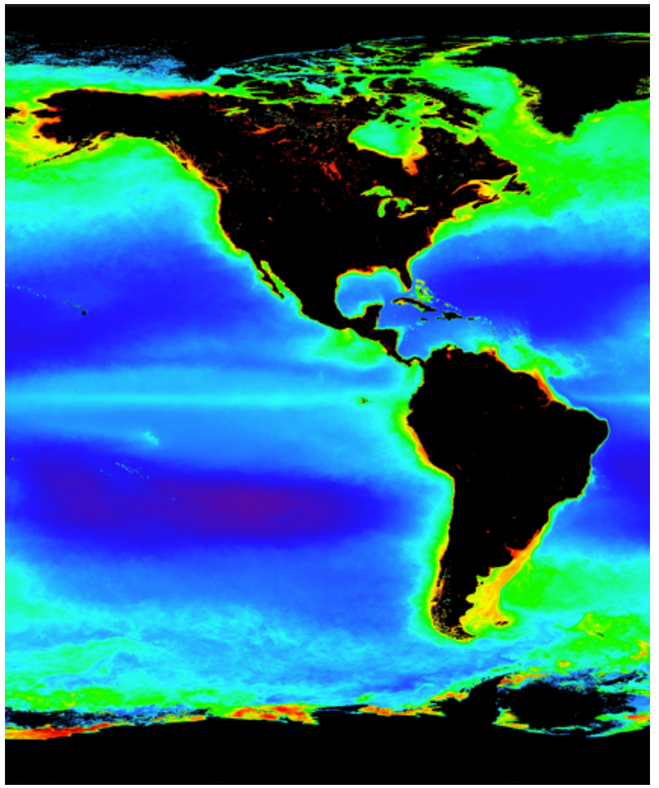
Measure vegetation density



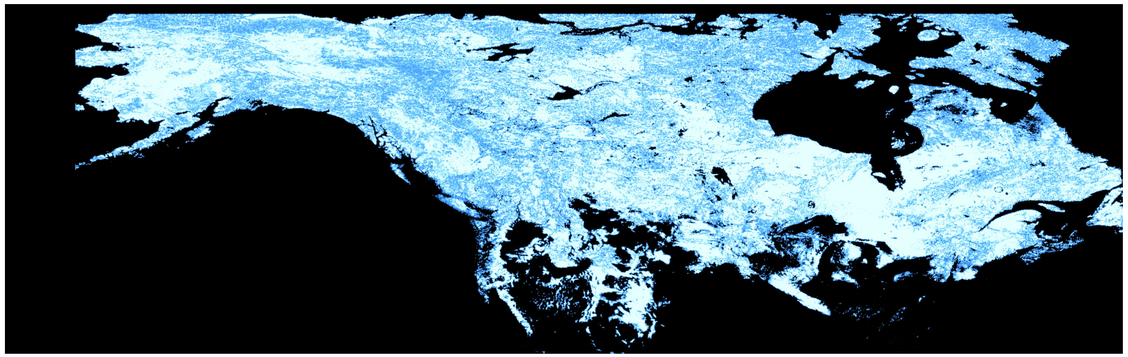
Track hurricanes



Track phytoplankton populations



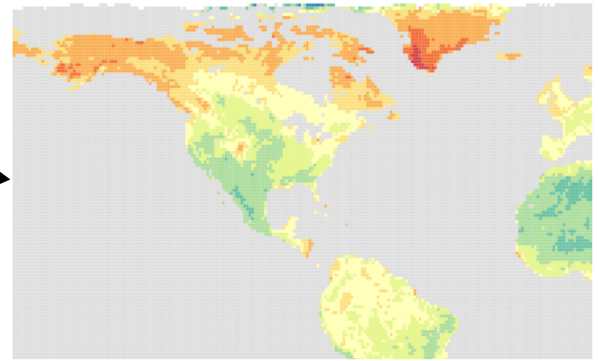
Measure snow melt



[L. Battle, 2017]

Interactive Exploration of Spatial Data

```
SELECT lat, lng, (b4-b6)/(b4+b6) as ndsi
FROM modis_data
WHERE ndsi >0.7
```



query

Client

Server

result

DBMS

PostgreSQL

MySQL

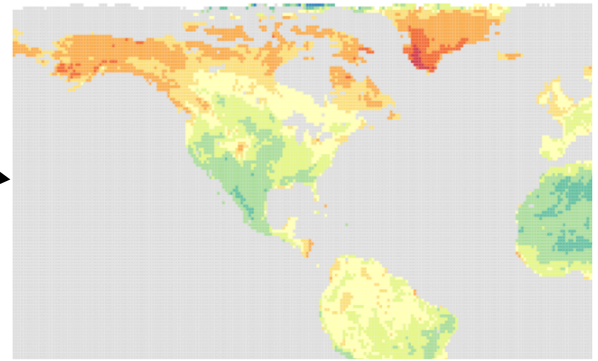
VERTICA An HP Company

SciDB

[L. Battle, 2017]

Interactive Exploration of Spatial Data

```
SELECT lat, lng, (b4-b6)/(b4+b6) as ndsi
FROM modis_data
WHERE ndsi >0.7
```



query

Client

Server

result

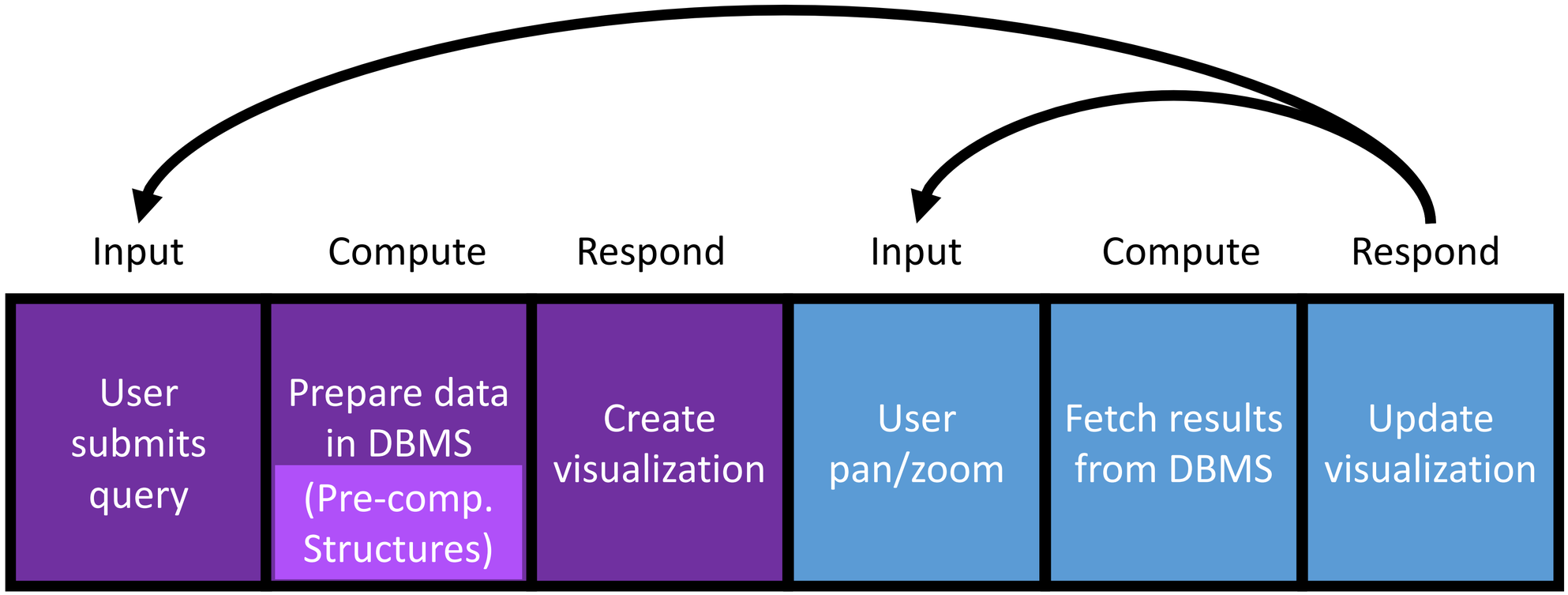
SLOW →

DBMS

PostgreSQL MySQL VERTICA An HP Company SciDB

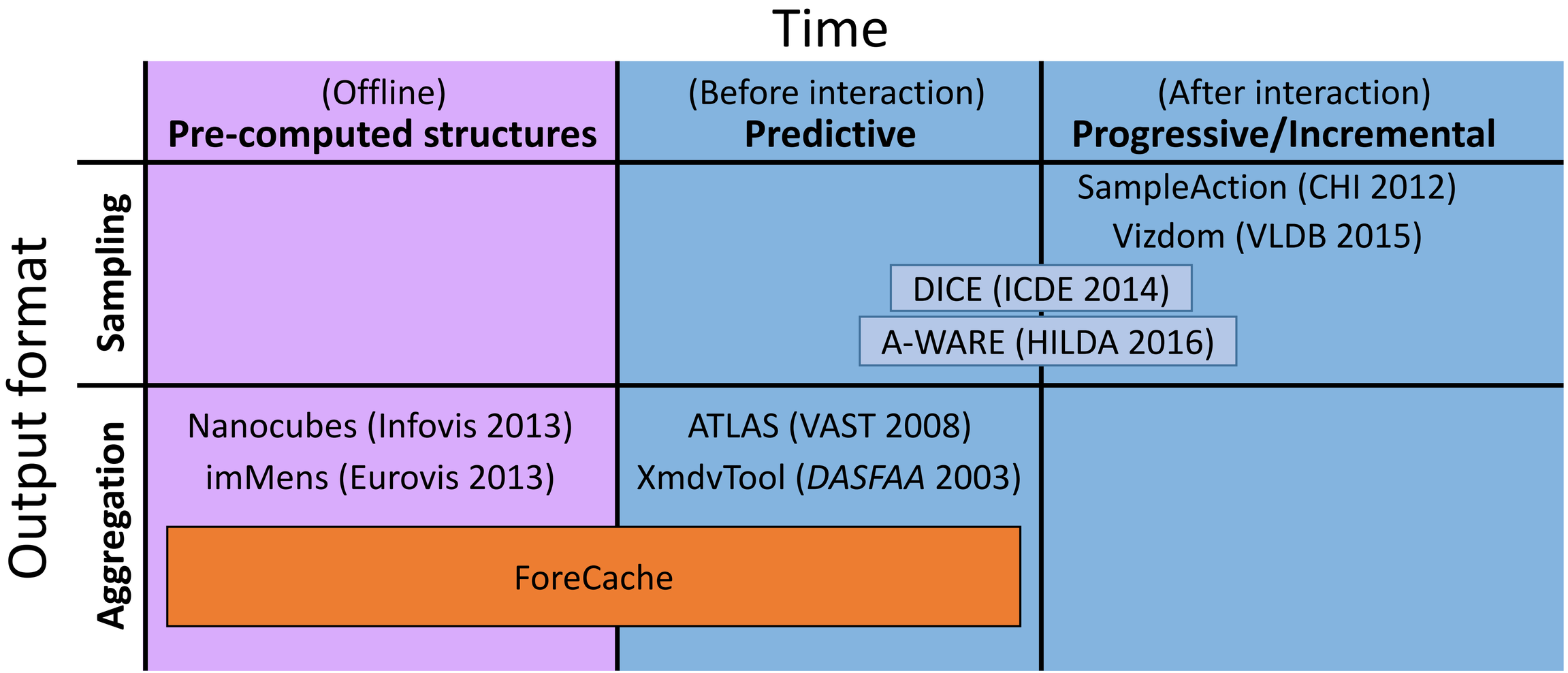
[L. Battle, 2017]

Two Inputs to Exploratory Browsing



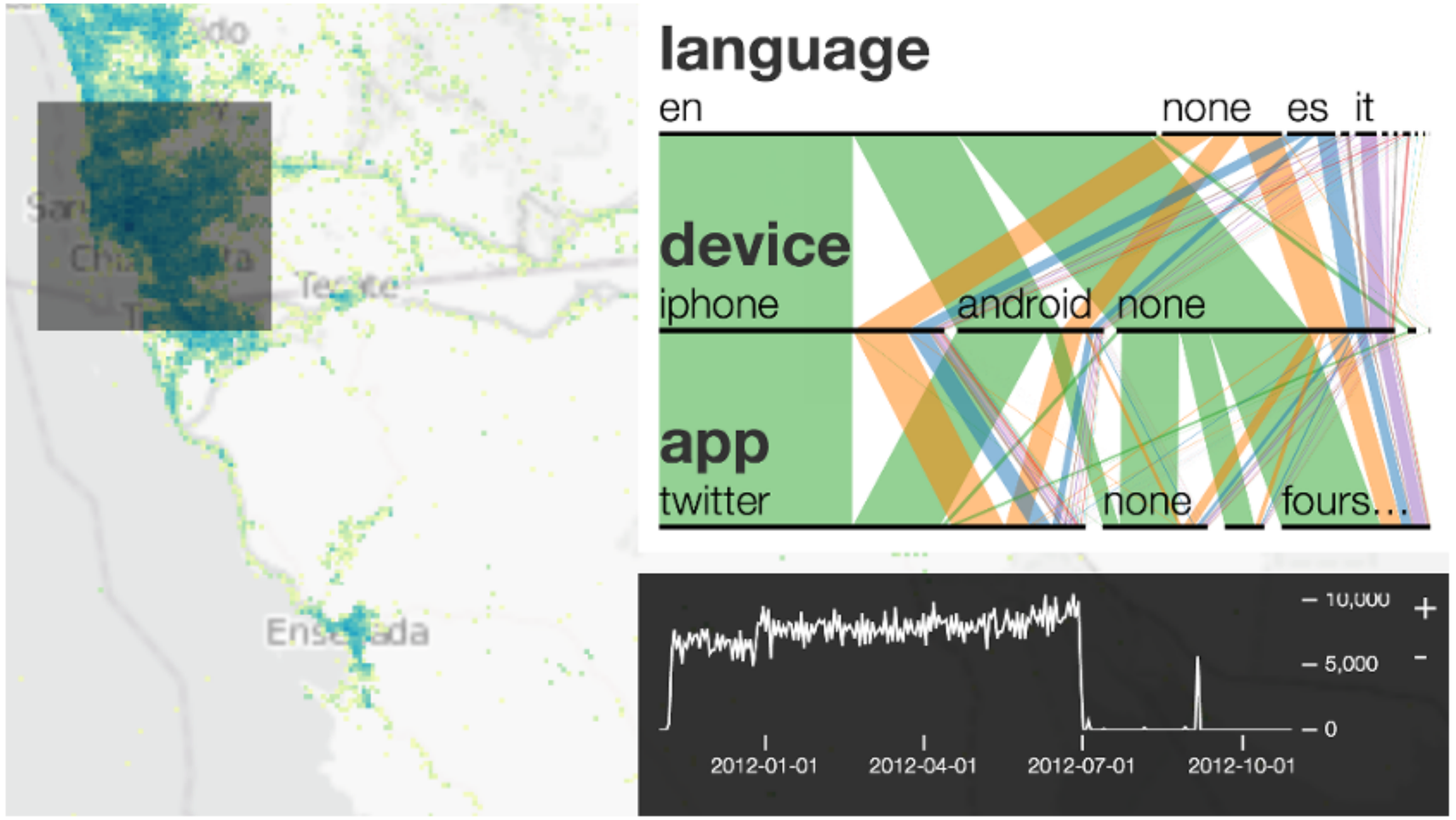
[L. Battle, 2017]

Systems for Interactive Exploration



[L. Battle, 2017]

Nanocubes



Linked view of tweets in San Diego, US

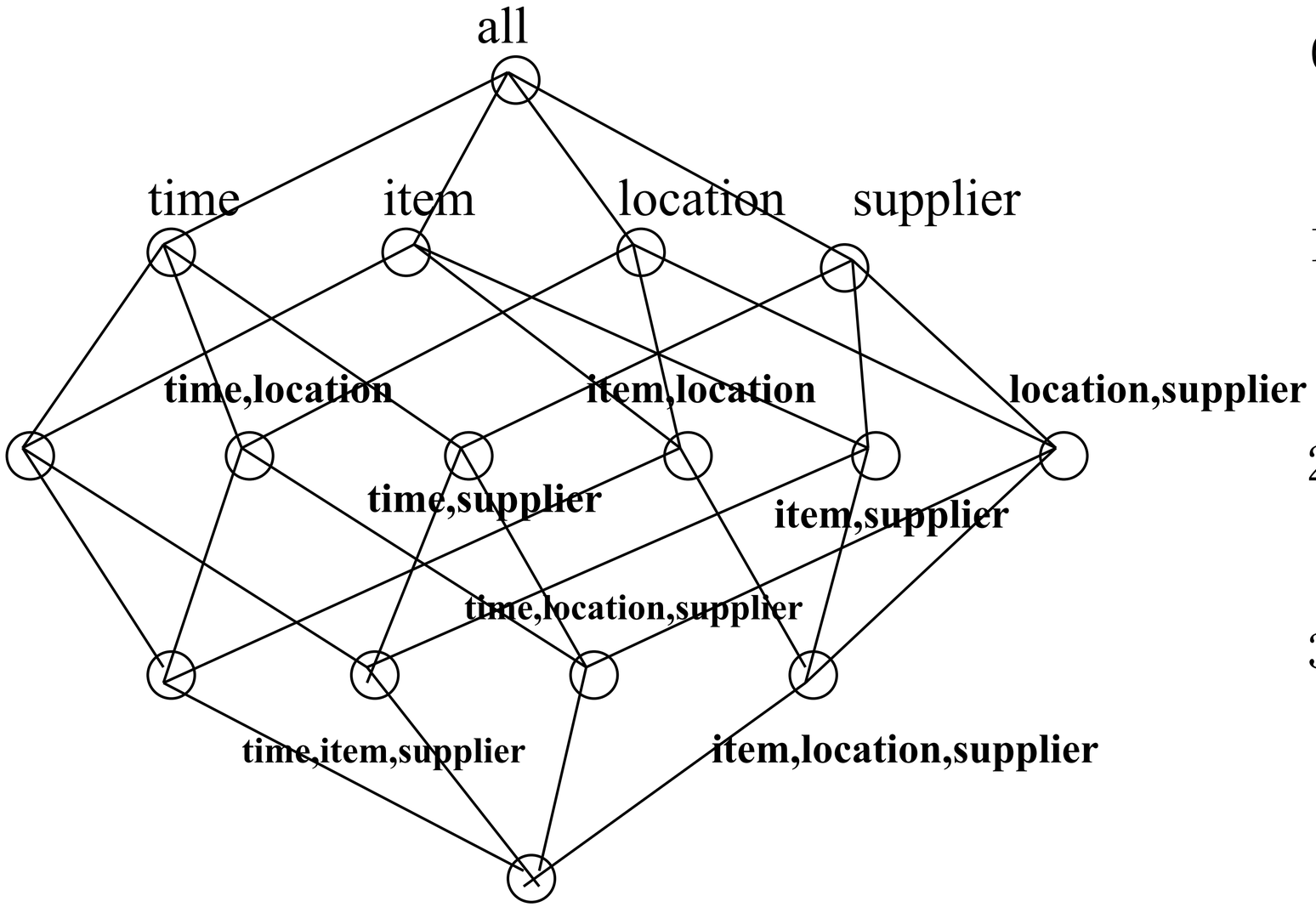
[Lins et. al, 2013]

From Tables and Spreadsheets to Data Cubes

- A **data warehouse** is based on a multidimensional data model which views data in the form of a data cube
- A **data cube**, such as sales, allows data to be modeled and viewed in multiple dimensions
  - **Dimension tables**, such as item (item_name, brand, type), or time(day, week, month, quarter, year)
  - **Fact table** contains **measures** (such as dollars_sold) and keys to each of the related dimension tables
- In data warehousing literature, an n-D base cube is called a **base cuboid**. The top most 0-D cuboid, which holds the highest-level of summarization, is called the **apex cuboid**. The lattice of cuboids forms a **data cube**.

[Han et al., 2011]

Data Cube: A Lattice of Cuboids



0−D (*apex*) cuboid

1−D cuboids

2−D cuboids

3−D cuboids

4−D (*base*) cuboid

[Han et al., 2011]

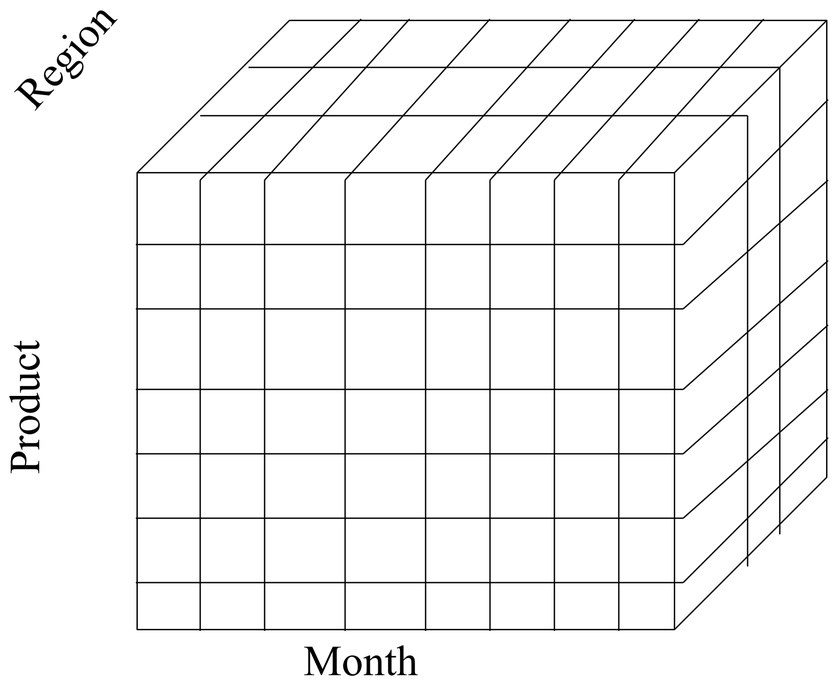
Data Cube Measures: Three Categories

- **Distributive**: if the result derived by applying the function to n aggregate values is the same as that derived by applying the function on all the data without partitioning
  - E.g., `count(), sum(), min(), max()`
- **Algebraic**: if it can be computed by an algebraic function with M arguments (where M is a bounded integer), each of which is obtained by applying a distributive aggregate function
  - E.g., `avg(), min_N(), standard_deviation()`
- **Holistic**: if there is no constant bound on the storage size needed to describe a subaggregate.
  - E.g., `median(), mode(), rank()`

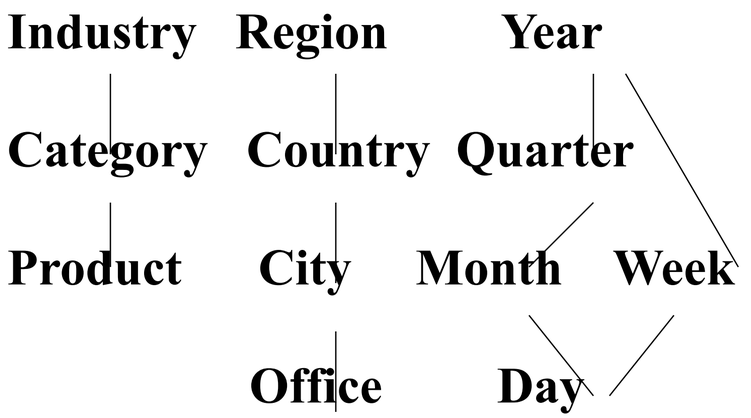
[Han et al., 2011]

Multidimensional Data

- Sales volume as a function of product, month, and region

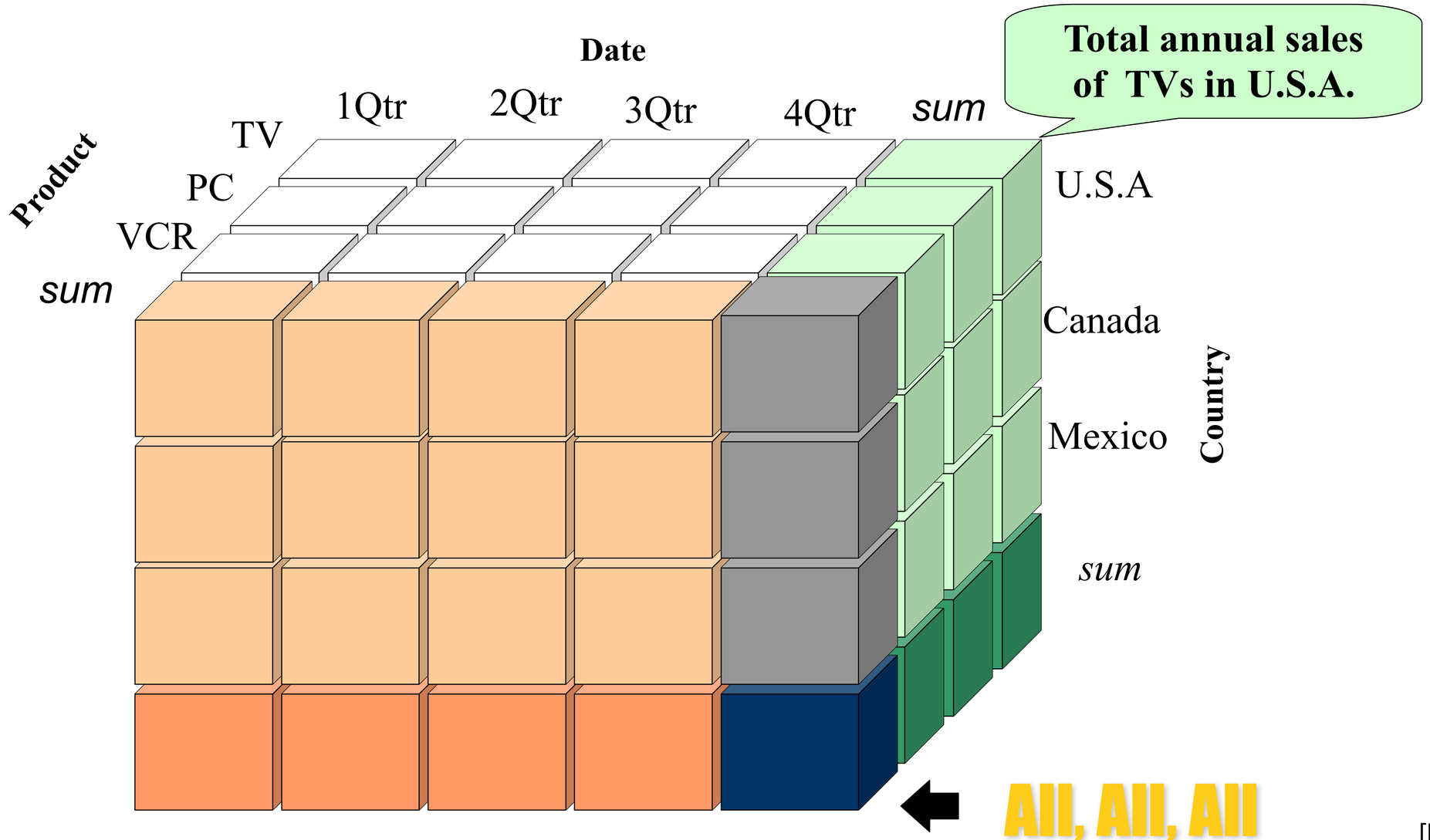


Dimensions: *Product, Location, Time*
Hierarchical summarization paths



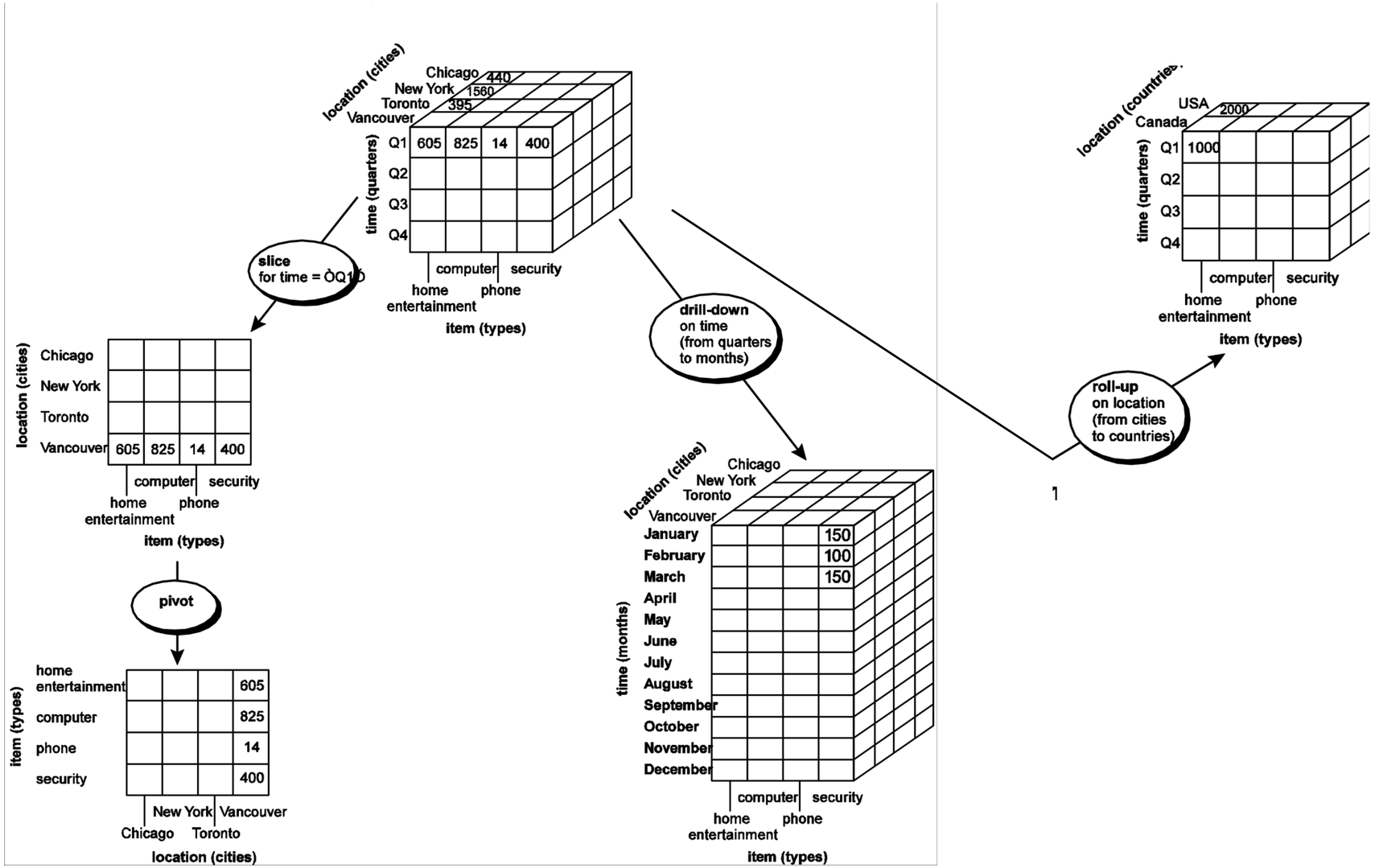
[Han et al., 2011]

A Sample Data Cube



[Han et al., 2011]

OLAP Operations



[Han et al., 2011]

Efficient Processing of OLAP Queries

- Determine which operations should be performed on the available cuboids
  - Transform drill, roll, etc. into corresponding SQL and/or OLAP operations, e.g., dice = selection + projection
- Determine which materialized cuboid(s) for OLAP operation:
  - Query: `{brand, province_or_state}` with `"year = 2004"`
  - 4 materialized cuboids available:
    1. `{year, item_name, city}`
    2. `{year, brand, country}`
    3. `{year, brand, province_or_state}`
    4. `{item_name, province_or_state}` where `year = 2004`
  - Which should be selected to process the query?

[Han et al., 2011]

Data Cube Aggregations

Relation A

Country	*Device*	*Language*
US	Android	en
US	iPhone	ru
South Africa	iPhone	en
India	Android	en
Australia	iPhone	en

Aggregation B

Country	*Device*	*Language*	*Count*
All	*All*	*All*	5

Group By on *Device, Language* C

Country	*Device*	*Language*	*Count*
All	Android	en	2
All	iPhone	en	2
All	iPhone	ru	1

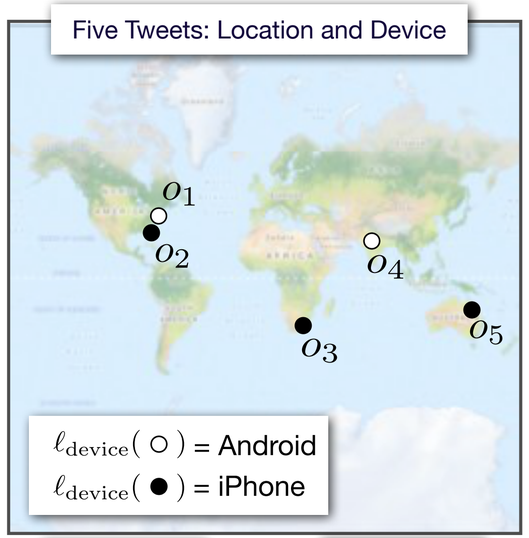
Cube on *Device, Language* D

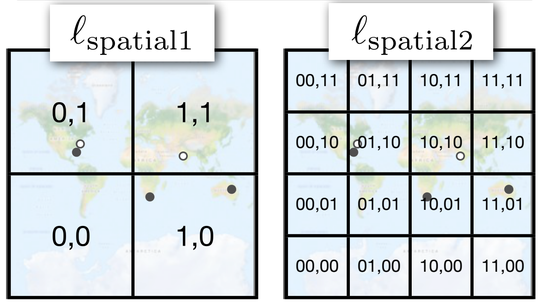
Country	*Device*	*Language*	*Count*
All	*All*	*All*	5
All	Android	*All*	2
All	iPhone	*All*	3
All	*All*	en	4
All	*All*	ru	1
All	iPhone	ru	1
All	Android	en	2
All	iPhone	en	2

Equivalent to Group By on all possible subsets of {*Device, Language*}

[Lins et. al, 2013]

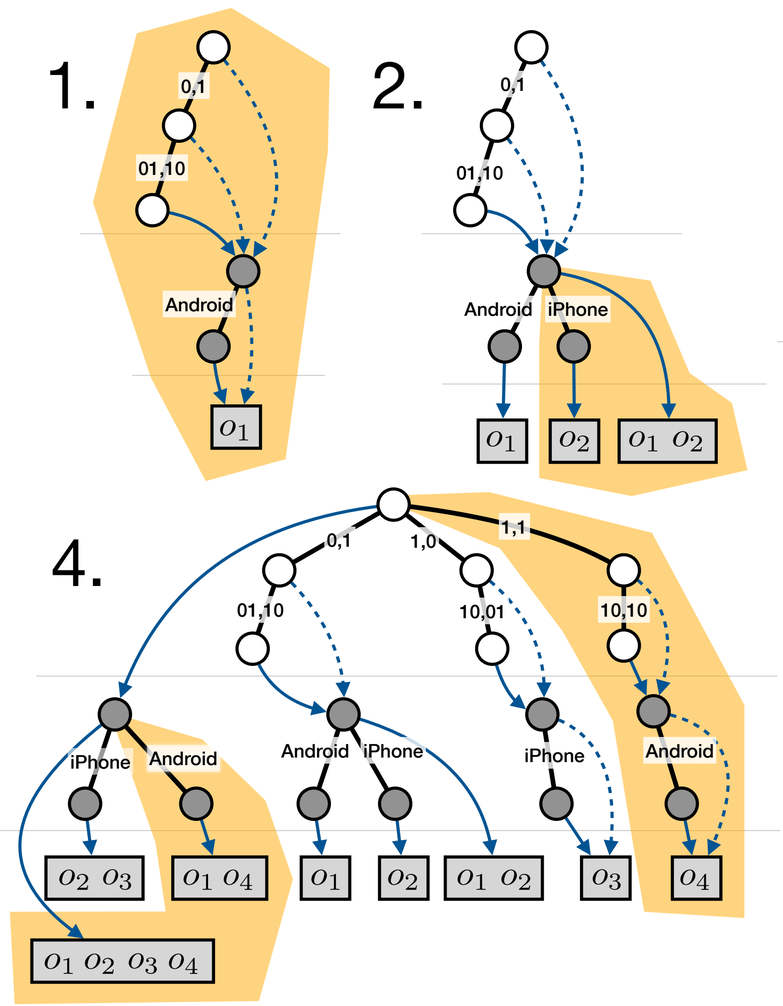
Building a Nanocube

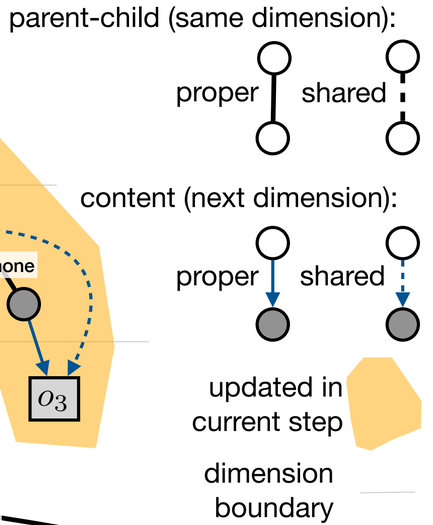




Indexing Schema

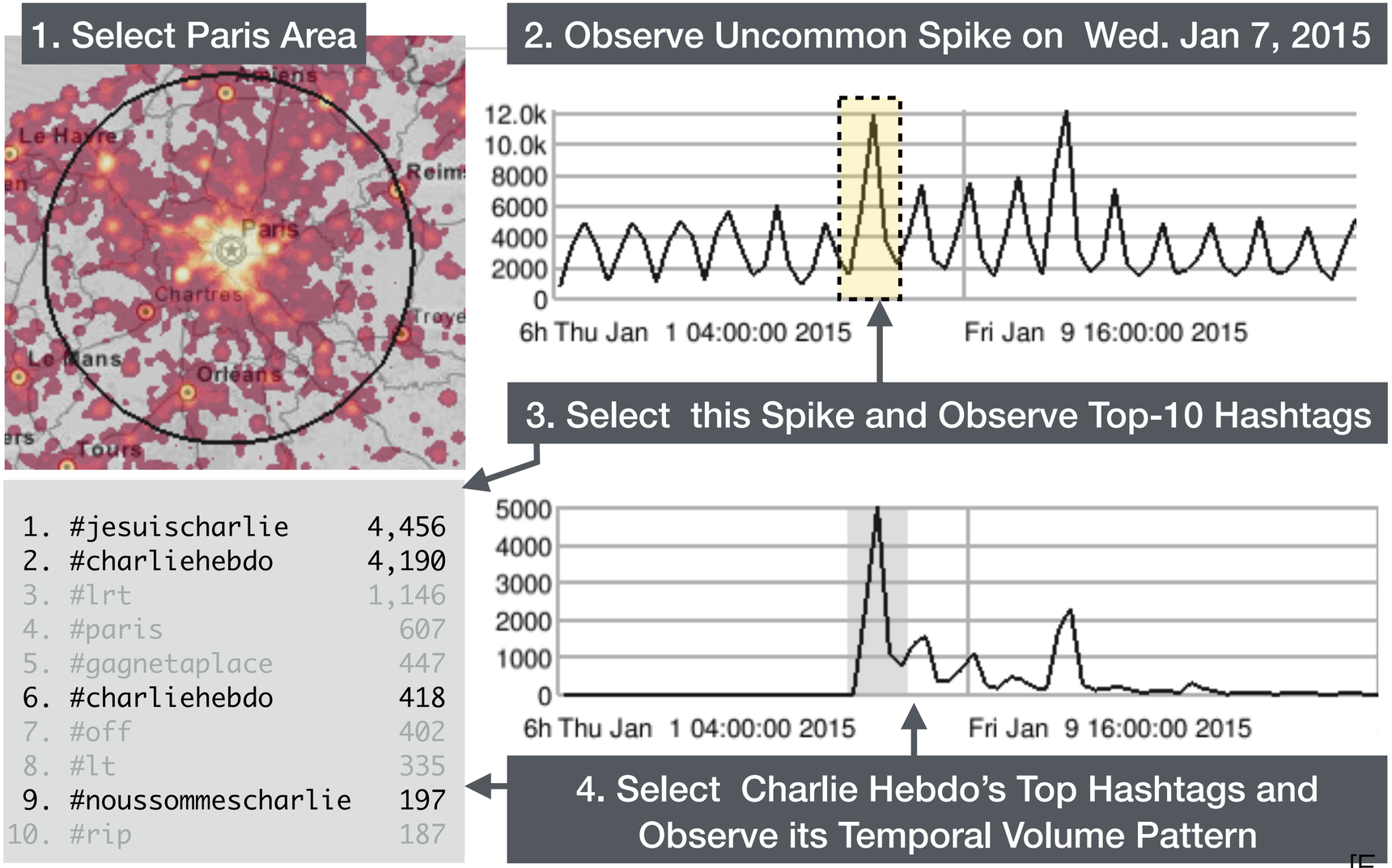
$$S = [\,[\ell_{\text{spatial1}}, \ell_{\text{spatial2}}], [\ell_{\text{device}}]\,]$$





[Lins et. al, 2013]

TopKube: Rankings and Top-k Queries



[F. Miranda et al., 2017]

TopKube vs. Nanocubes

- Product bin: the combination of selections from dimensions
- Nanocubes maps each product bin ((01,10), iPhone) to a **time series**

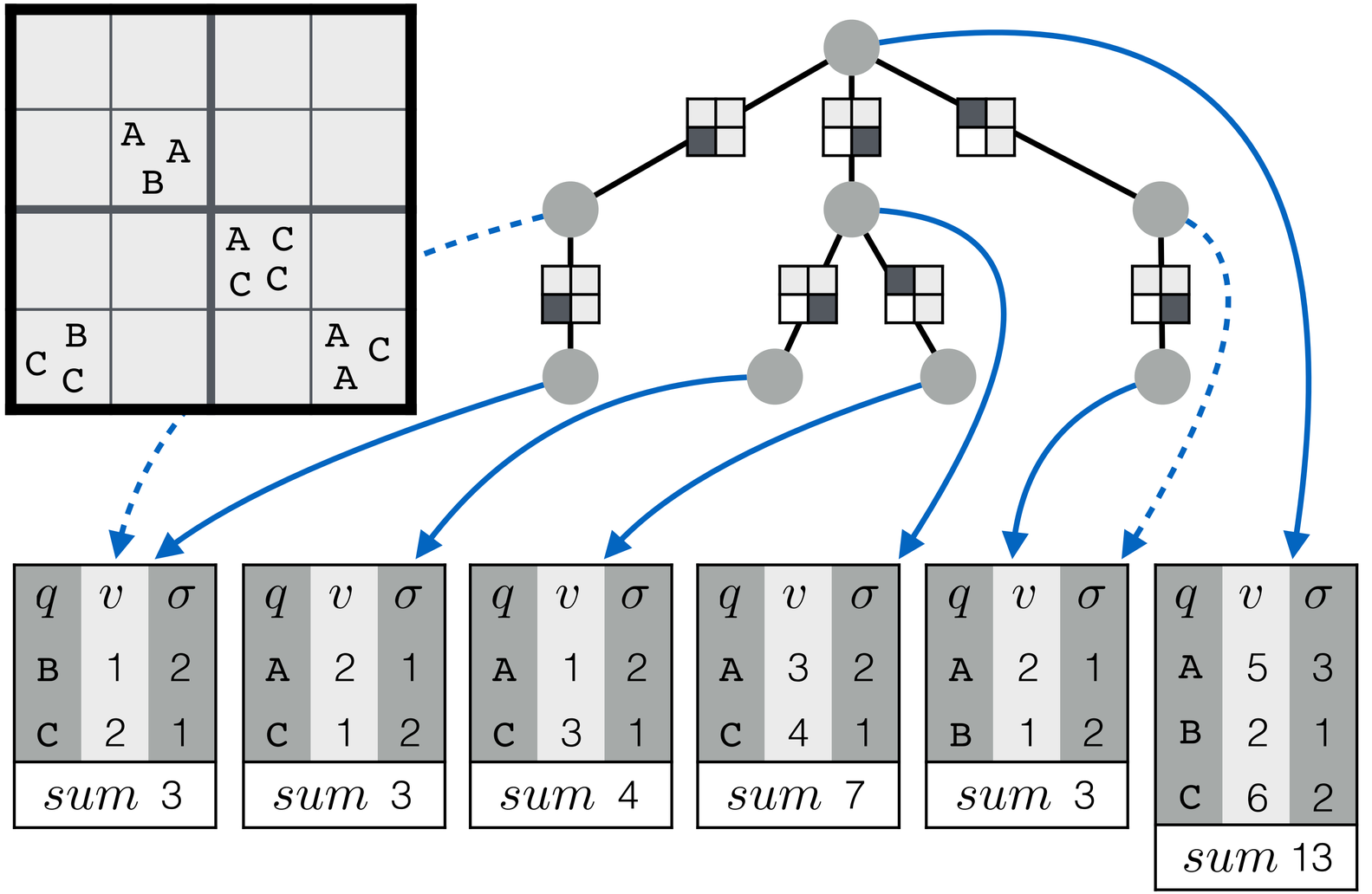
$$\beta \mapsto ((t_1, v_1), (t_2, v_1 + v_2), \ldots, (t_m, v_1 + \ldots + v_m))$$

- TopKube maps each product bin to **rank-aware multi-set**

$$\beta \mapsto \left\{ \text{lst} = ((q_1, v_1, \sigma_1), \ldots, (q_j, v_j, \sigma_j)), \text{sum} = \sum_{i=1}^{j} v_i \right\}$$

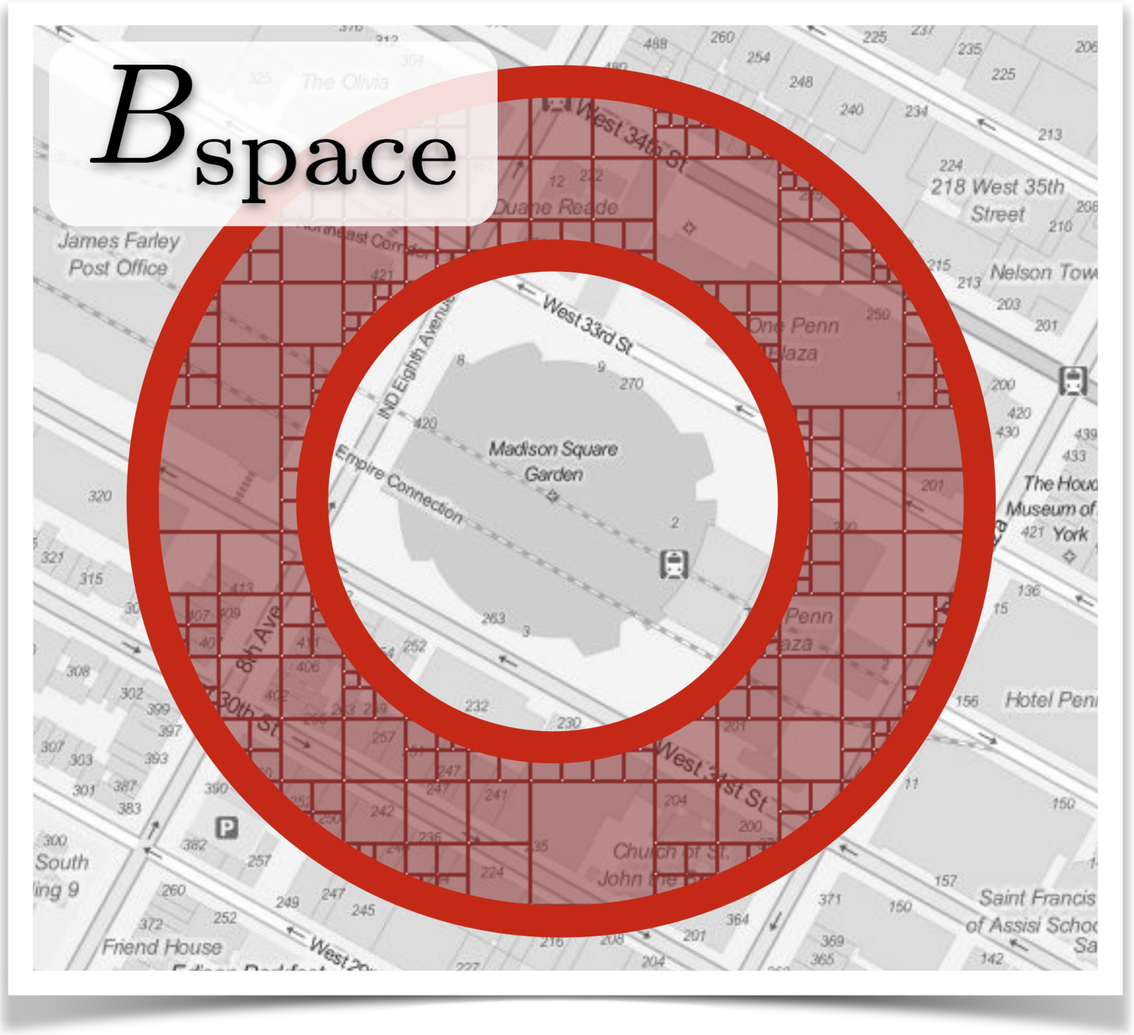
- q_i is the ith smallest key that appears in product bin
- v_i is the value of the measure for key q_i in the product bin
- σ_i is the index of the key with its largest value

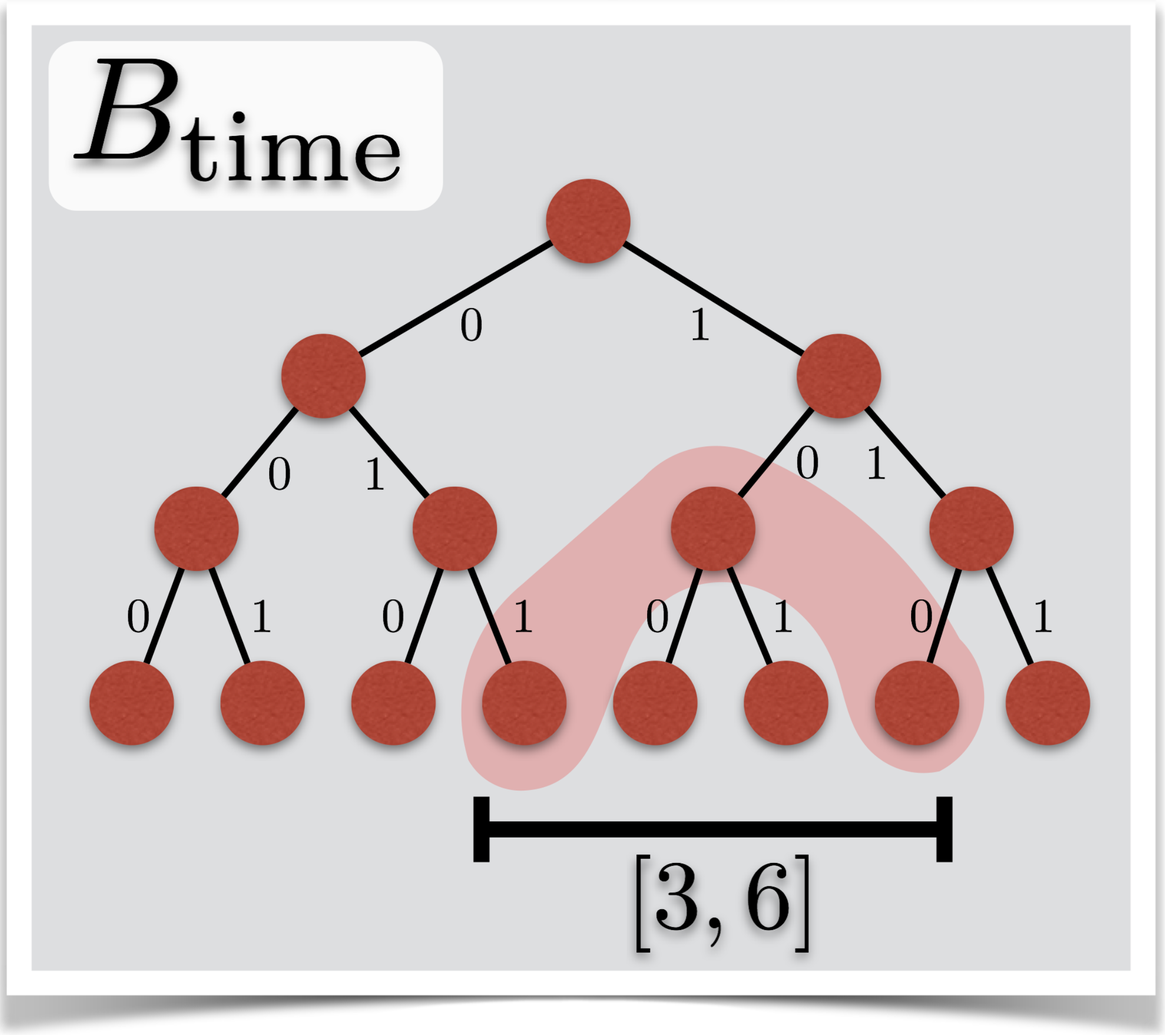
Example: One Spatial Dim. and A,B,C events



[F. Miranda et al., 2017]

Problem: Lots of Bins!





[F. Miranda et al., 2017]

Three Algorithms to Merge Bins

- Threshold: don't do a full scan, use extra information about ranking
- Sweep: Use a priority queue where the product bin with the current smallest key is on the top
- Hybrid:
  - Threshold has best theoretical guarantee but some sparse cases can be faster
  - Use Sweep on small input lists, Threshold on denser problem

What actually happened in a computational experiment?

Provenance in Art

Rembrandt van Rijn
Dutch, 1606 - 1669
Self-Portrait, 1659
oil on canvas
Andrew W. Mellon Collection
1937.1.72

Provenance

George, 3rd Duke of Montagu and 4th Earl of Cardigan [d. 1790], by 1767;[1] by inheritance to his daughter, Lady Elizabeth, wife of Henry, 3rd Duke of Buccleuch of Montagu House, London; John Charles, 7th Duke of Buccleuch; (P. & D. Colnaghi & Co., New York, 1928); (M. Knoedler & Co., New York); sold January 1929 to Andrew W. Mellon, Pittsburgh and Washington, D.C.; deeded 28 December 1934 to The A.W. Mellon Educational and Charitable Trust, Pittsburgh; gift 1937 to NGA.

[1] This early provenance is established by presence of a mezzotint after the portrait by R. Earlom (1743-1822), dated 1767. See John Charrington, A Catalogue of the Mezzotints After, or Said to Be After, Rembrandt, Cambridge, 1923, no. 49.

Associated Names

- Buccleuch, Henry, 3rd Duke of
- Buccleuch, John Charles, 7th Duke of
- Colnaghi & Co., Ltd., P. & D.
- Knoedler & Company, M.
- Mellon, Andrew W.
- Mellon Educational and Charitable Trust, The A.W.
- Montagu, and 4th Earl of Cardigan, George, 3rd Duke of

[National Gallery of Art]

Provenance in Art

Rembrandt van Rijn
Dutch, 1606 - 1669
Self-Portrait, 1659
oil on canvas
Andrew W. Mellon Collection
1937.1.72

Provenance

George, 3rd Duke of Montagu and 4th Earl of Cardigan [d. 1790], by 1767;[1] by inheritance to his daughter, Lady Elizabeth, wife of Henry, 3rd Duke of Buccleuch of Montagu House, London; John Charles, 7th Duke of Buccleuch; (P. & D. Colnaghi & Co., New York, 1928); (M. Knoedler & Co., New York); sold January 1929 to Andrew W. Mellon, Pittsburgh and Washington, D.C.; deeded 28 December 1934 to The A.W. Mellon Educational and Charitable Trust, Pittsburgh; gift 1937 to NGA.

[1] This early provenance is established by presence of a mezzotint after the portrait by R. Earlom (1743-1822), dated 1767. See John Charrington, A Catalogue of the Mezzotints After, or Said to Be After, Rembrandt, Cambridge, 1923, no. 49.

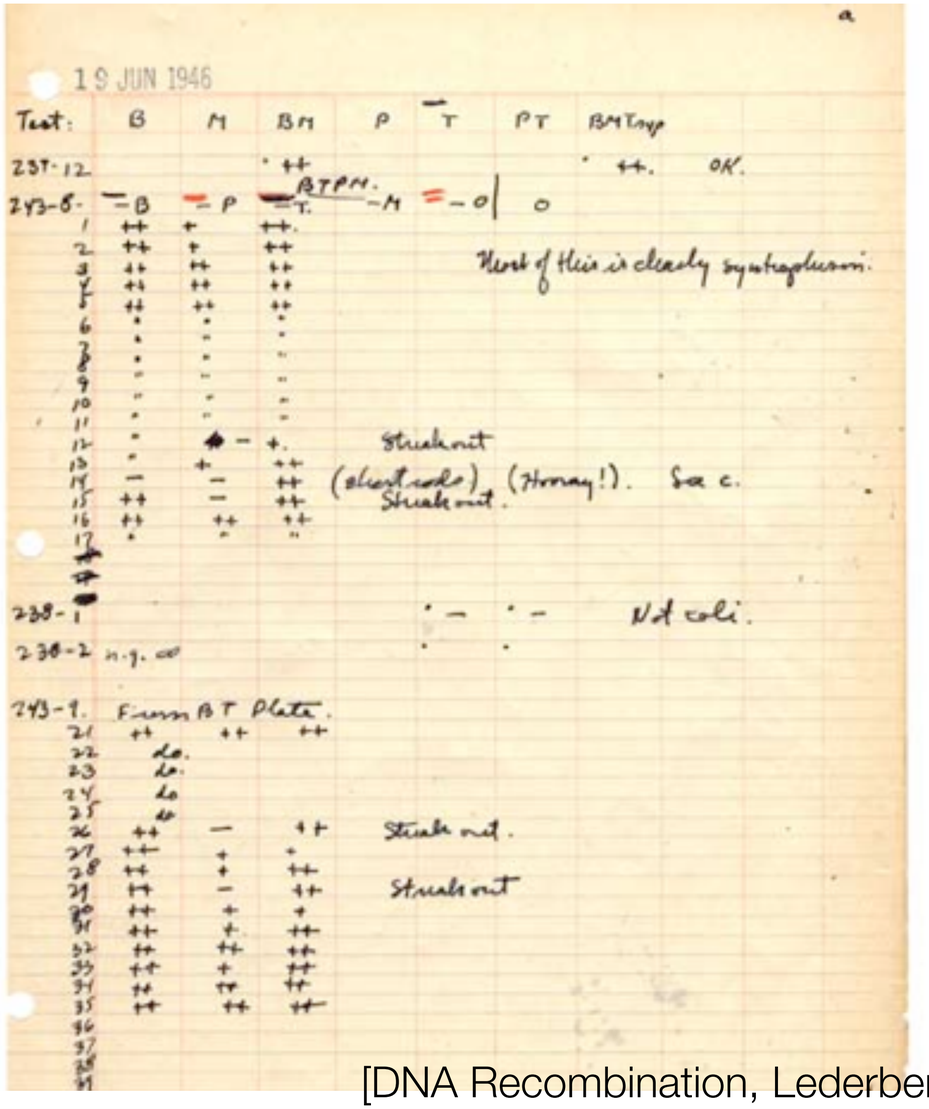
Associated Names

- Buccleuch, Henry, 3rd Duke of
- Buccleuch, John Charles, 7th Duke of
- Colnaghi & Co., Ltd., P. & D.
- Knoedler & Company, M.
- Mellon, Andrew W.
- Mellon Educational and Charitable Trust, The A.W.
- Montagu, and 4th Earl of Cardigan, George, 3rd Duke of

[National Gallery of Art]

Provenance in Science

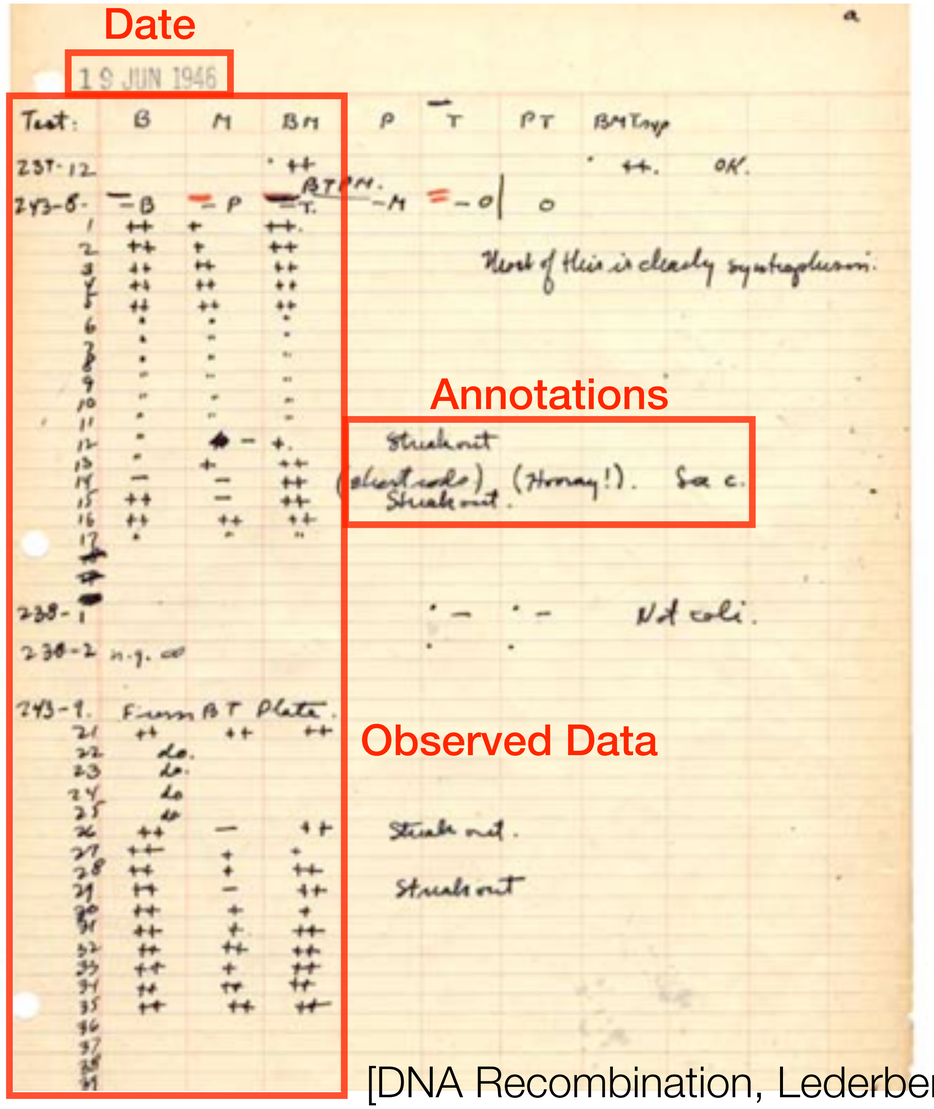
- Provenance: the lineage of data, a computation, or a visualization
- **Provenance is as (or more) important as the result!**
- Old solution:
  - Lab notebooks
- New problems:
  - Large volumes of data
  - Complex analyses
  - Writing notes doesn't scale



[DNA Recombination, Lederberg]

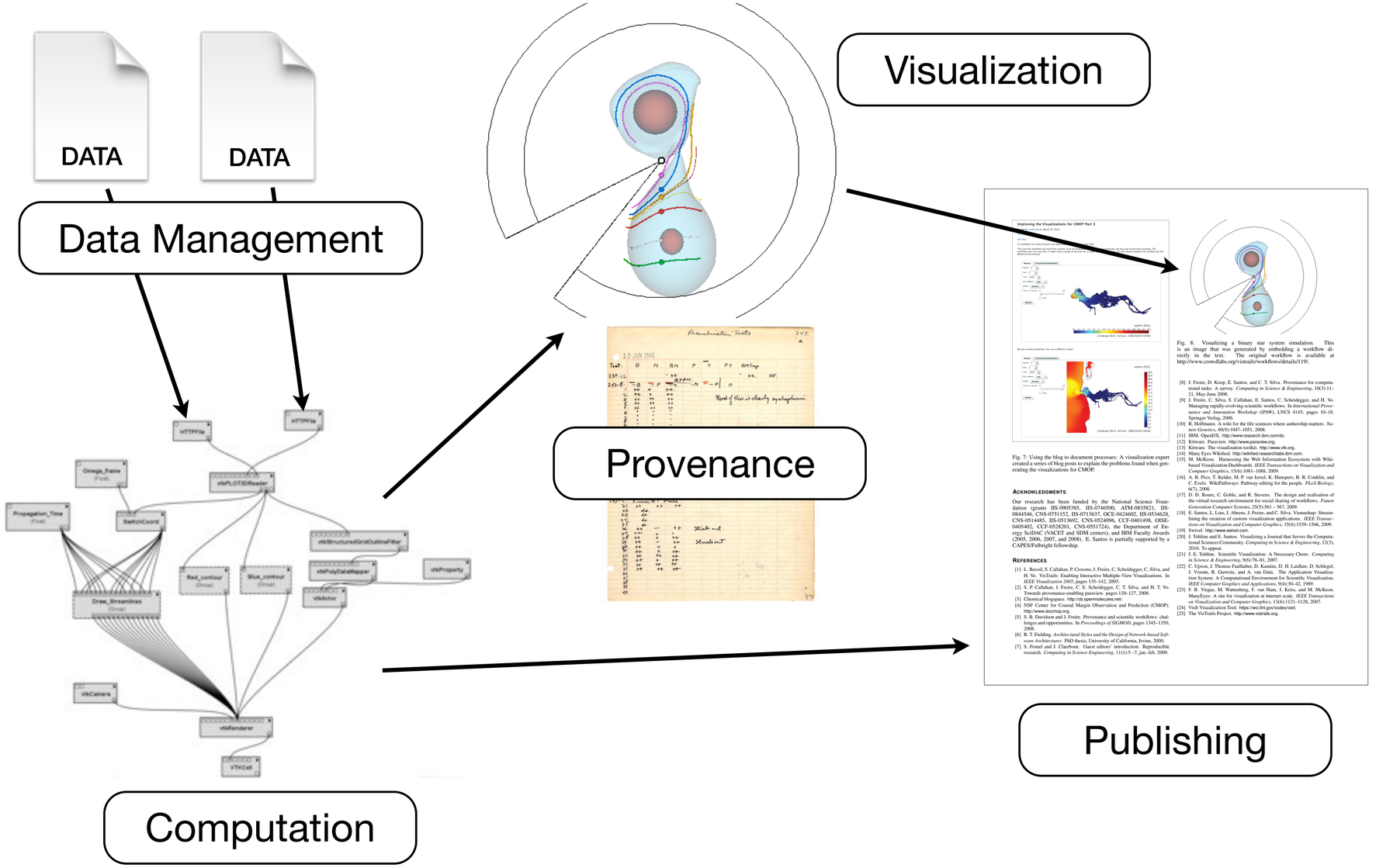
Provenance in Science

- Provenance: the lineage of data, a computation, or a visualization
- **Provenance is as (or more) important as the result!**
- Old solution:
  - Lab notebooks
- New problems:
  - Large volumes of data
  - Complex analyses
  - Writing notes doesn't scale



[DNA Recombination, Lederberg]

Provenance in Computational Science



Evolution of Publication

- Publish paper
- Publish code
- Publish computational experiments/tests
- Publish provenance (what actually happens during your runs)

Provenance-Rich Publication

Galois Conjugates of Topological Phases

M. H. Freedman,[1] J. Gukelberger,[2] M. B. Hastings,[1] S. Trebst,[1] M. Troyer,[2] and Z. Wang[1]
[1]*Microsoft Research, Station Q, University of California, Santa Barbara, CA 93106, USA*
[2]*Theoretische Physik, ETH Zurich, 8093 Zurich, Switzerland*
(Dated: July 6, 2011)

Galois conjugation relates unitary conformal field theories (CFTs) and topological quantum field theories (TQFTs) to their non-unitary counterparts. Here we investigate Galois conjugates of quantum double models, such as the Levin-Wen model. While these Galois conjugated Hamiltonians are typically non-Hermitian, we find that their ground state wave functions still obey a generalized version of the usual code property (local operators do not act on the ground state manifold) and hence enjoy a generalized topological protection. The key question addressed in this paper is whether such non-unitary topological phases can also appear as the ground states of Hermitian Hamiltonians. Specific attempts at constructing Hermitian Hamiltonians with these ground states lead to a loss of the code property and topological protection of the degenerate ground states. Beyond this we rigorously prove that no local change of basis (IV.5) can transform the ground states of the Galois conjugated doubled Fibonacci theory into the ground states of a topological model whose Hermitian Hamiltonian satisfies Lieb-Robinson bounds. These include all gapped local or quasi-local Hamiltonians. A similar statement holds for many other non-unitary TQFTs. One consequence is that the "Gaffnian" wave function cannot be the ground state of a gapped fractional quantum Hall state.

PACS numbers: 05.30.Pr, 73.43.-f

I. INTRODUCTION

Galois conjugation, by definition, replaces a root of a polynomial by another one with identical algebraic properties. For example, i and $-i$ are Galois conjugate (consider $z^2+1=0$) as are $\phi = \frac{1+\sqrt{5}}{2}$ and $-\frac{1}{\phi} = \frac{1-\sqrt{5}}{2}$ (consider $z^2-z-1=0$), as well as $\sqrt[3]{2}$, $\sqrt[3]{2}e^{2\pi i/3}$, and $\sqrt[3]{2}e^{-2\pi i/3}$ (consider $z^3-2=0$). In physics Galois conjugation can be used to convert non-unitary conformal field theories (CFTs) to unitary ones, and vice versa. One famous example is the non-unitary Yang-Lee CFT, which is Galois conjugate to the Fibonacci CFT $(G_2)_1$, the even (or integer-spin) subset of $\mathrm{su}(2)_3$.

In statistical mechanics non-unitary conformal field theories have a venerable history.[1,2] However, it has remained less clear if there exist physical situations in which non-unitary models can provide a useful description of the low energy physics of a quantum mechanical system – after all, Galois conjugation typically destroys the Hermitian property of the Hamiltonian. Some non-Hermitian Hamiltonians, which surprisingly have totally real spectrum, have been found to arise in the study of PT-invariant one-particle systems[3] and in some Galois conjugate many-body systems[4] and might be seen to open the door a crack to the physical use of such models. Another situation, which has recently attracted some interest, is the question whether non-unitary models can describe 1D edge states of certain 2D bulk states (the edge holographic for the bulk). In particular, there is currently a discussion on whether or not the "Gaffnian" wave function could be the ground state for a *gapped* fractional quantum Hall (FQH) state albeit with a non-unitary "Yang-Lee" CFT describing its edge.[5–7] We conclude that this is not possible, further restricting the possible scope of non-unitary models in quantum mechanics.

We reach this conclusion quite indirectly. Our main thrust is the investigation of Galois conjugation in the simplest non-Abelian Levin-Wen model.[8] This model, which is also called "DFib", is a topological quantum field theory (TQFT) whose states are string-nets on a surface labeled by either a trivial or "Fibonacci" anyon. From this starting point, we give a rigorous argument that the "Gaffnian" ground state cannot be locally conjugated to the ground state of any topological phase, within a Hermitian model satisfying Lieb-Robinson (LR) bounds[9] (which includes but is not limited to gapped local and quasi-local Hamiltonians).

Lieb-Robinson bounds are a technical tool for local lattice models. In relativistically invariant field theories, the speed of light is a strict upper bound to the velocity of propagation. In lattice theories, the LR bounds provide a similar upper bound by a velocity called the LR velocity, but in contrast to the relativistic case there can be some exponentially small "leakage" outside the light-cone in the lattice case. The Lieb-Robinson bounds are a way of bounding the leakage outside the light-cone. The LR velocity is set by microscopic details of the Hamiltonian, such as the interaction strength and range. Combining the LR bounds with the spectral gap enables us to prove locality of various correlation and response functions. We will call a Hamiltonian a *Lieb-Robinson Hamiltonian* if it satisfies LR bounds.

We work primarily with a single example, but it should be clear that the concept of Galois conjugation can be widely applied to TQFTs. The essential idea is to retain the particle types and fusion rules of a unitary theory but when one comes to writing down the algebraic form of the F-matrices (also called $6j$ symbols), the entries are now Galois conjugated. A slight complication, which is actually an asset, is that writing an F-matrix requires a gauge choice and the most convenient choice may differ before and after Galois conjugation.

Our method is not restricted to Galois conjugated $\mathrm{DFib}^{\mathcal{G}}$ and its factors $\mathrm{Fib}^{\mathcal{G}}$ and $\overline{\mathrm{Fib}^{\mathcal{G}}}$, but can be generalized to infinitely many non-unitary TQFTs, showing that they will not arise as low energy models for a gapped 2D quantum mechan-

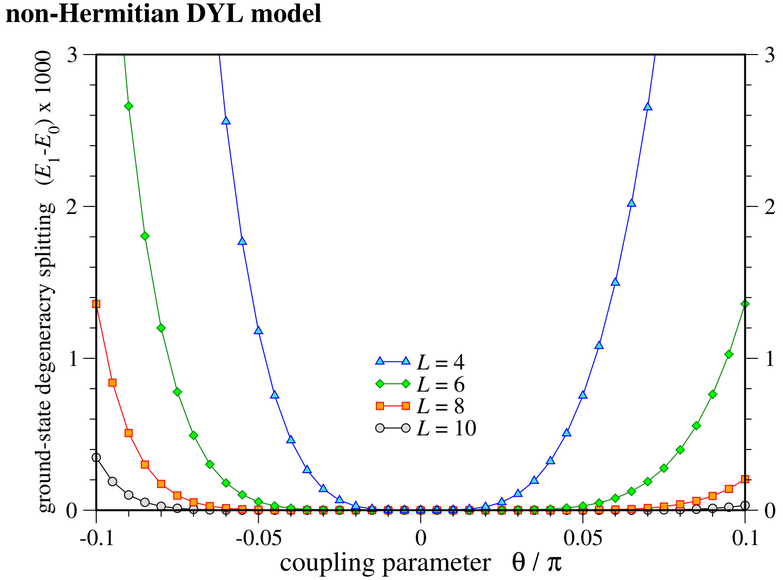


FIG. 6. (color online) Ground-state degeneracy splitting of the non-Hermitian doubled Yang-Lee model when perturbed by a string tension ($\theta \neq 0$).

[Freedman et al., 2012]

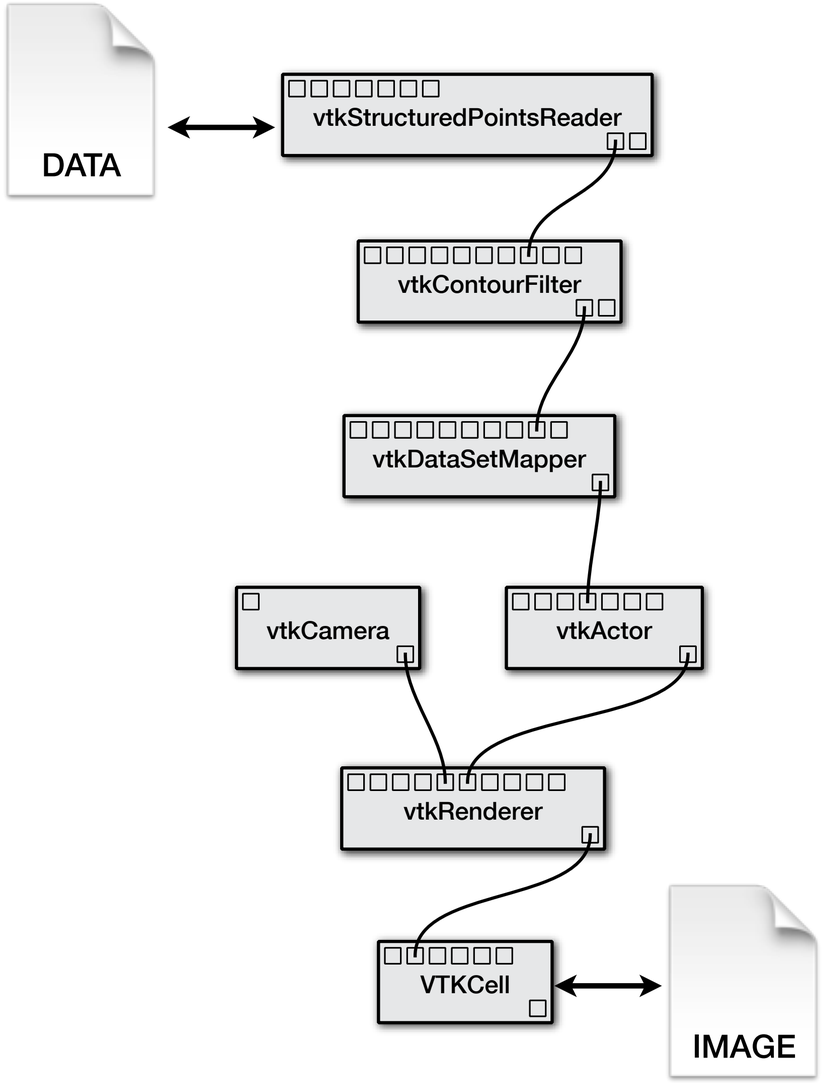
Benefits of Provenance-Rich Publications

- Produce more knowledge–not just text
- Allow scientists to stand on the shoulders of giants (and their own)
- Science can move faster!
- Higher-quality publications
- Authors will be more careful
- Many eyes to check results
- Describe more of the discovery process: people only describe successes, can we learn from mistakes?
- Expose users to different techniques and tools: expedite their training; and potentially reduce their time to insight

Provenance Definitions

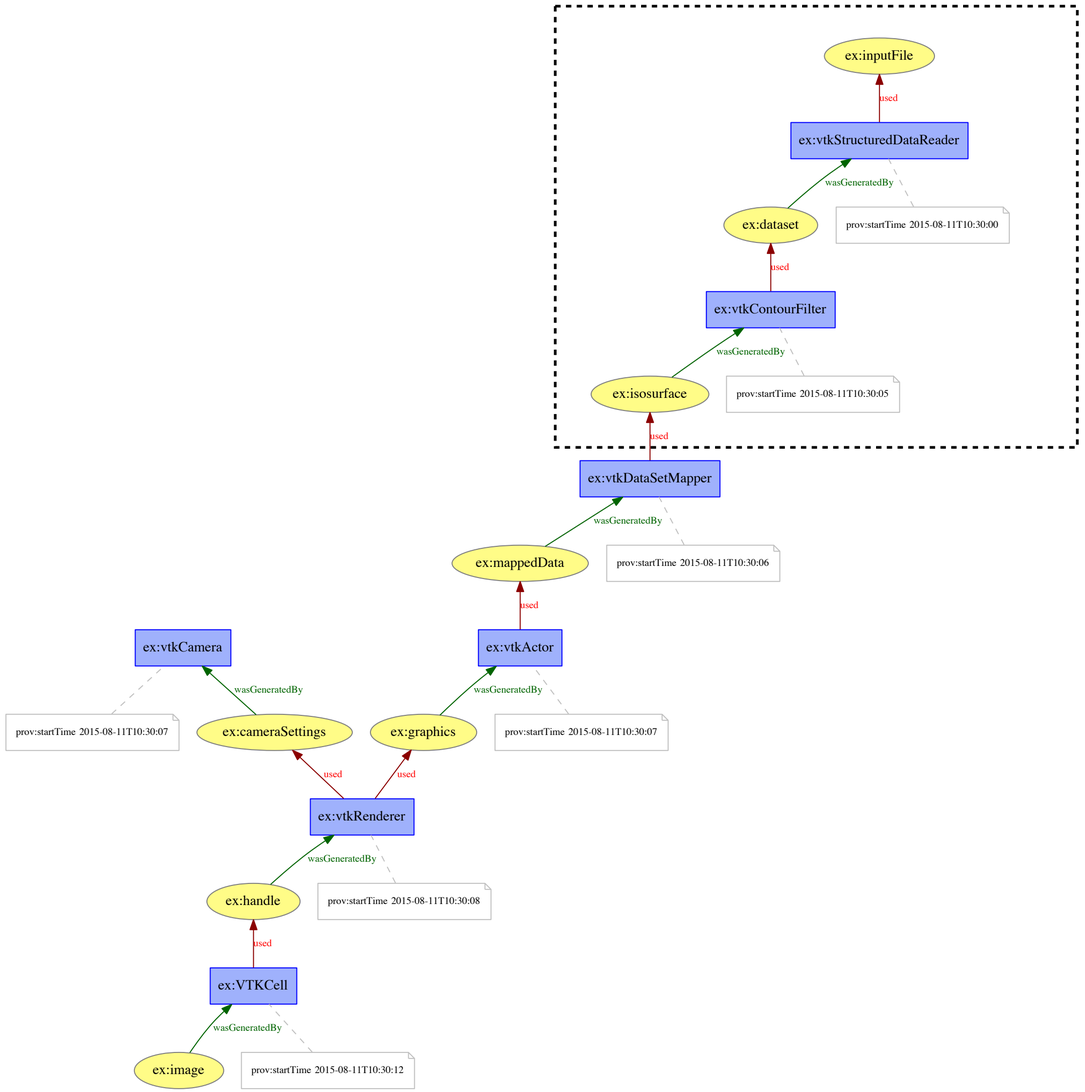
- Dictionary: "the source or origin of an object; its history and pedigree; a record of the ultimate derivation and passage of an item through its various owners."
- Focus on **causality**—the sequence of steps that detail how a result was generated and/or **derivation**—what data a result depended on
- Provenance itself is **data**, this list of steps along with metadata for each step: when it occurred, who initiated it, notes about it
- Can be used to preserve information about an experiment and to answer many questions

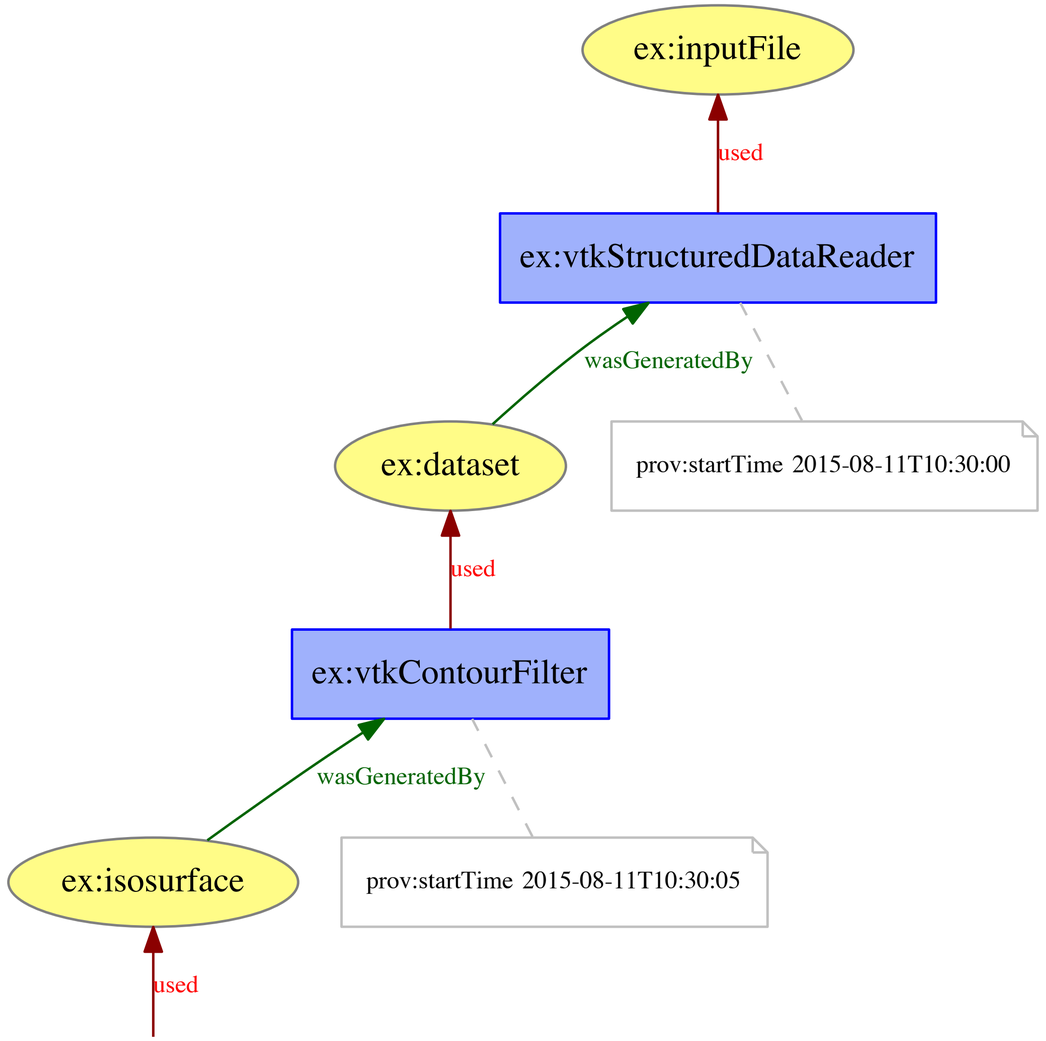
Workflows



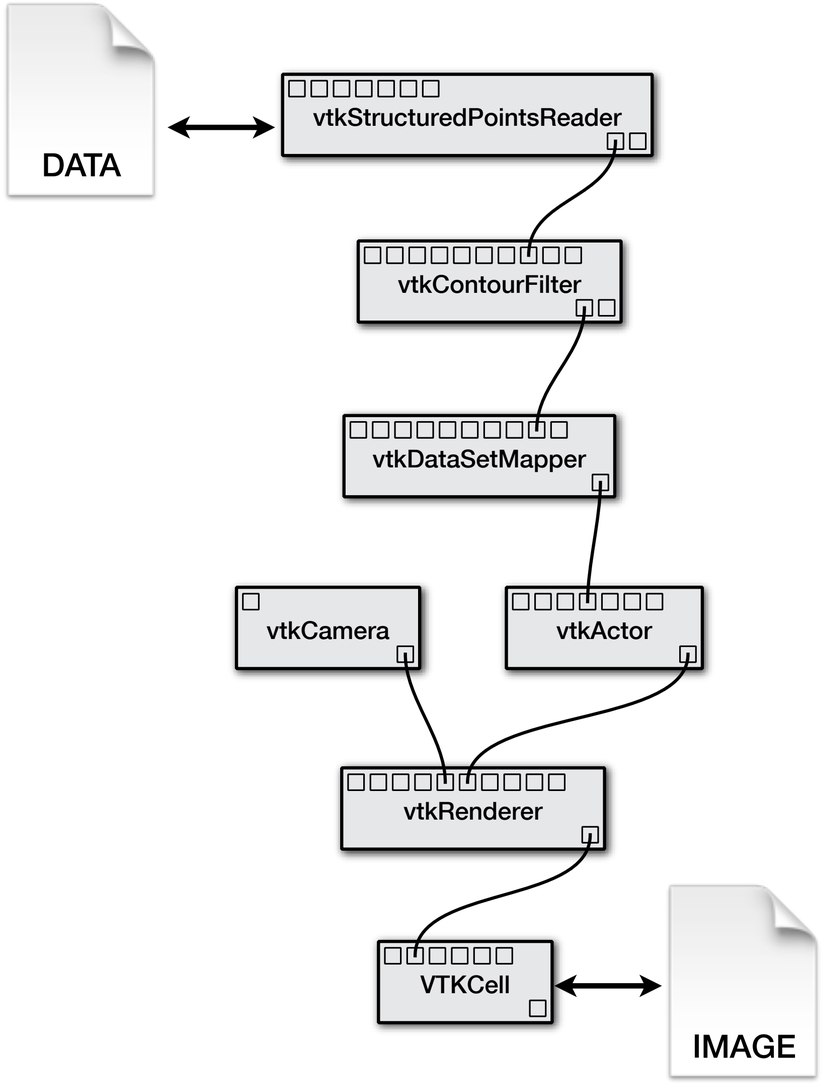
- Abstract computation
- Computational modules connected through input and output ports
- Data flows along the connections

Provenance Graph





Provenance Questions



- What process led to the output image?
- What input datasets contributed to the output image?
- What workflows create an isosurface with isovalue `57`?
- Who create this data product?
- When was this data file created?
- Why was `vtkCamera` used?
- Why do two output images differ?

Questions about Provenance

- How does one capture provenance?
- How does one manage provenance for later use?
- How do we answer questions about our provenance?
- How do we use provenance for good?

Provenance Management

- Provenance can be generated from tasks/programs/scripts/etc.
- Properties of provenance are related to the **computational model**
  - a specific application with a graphical interface
  - a script that automates the use of several command-line tools
  - a scientific workflow that combines several tools

Provenance & Causality

- Knowing what data/steps influenced other data/steps is important!
- Data dependencies: this output file depended on this input file
- Data-process dependencies: this output figure depended on these processes
- Causality can often be represented as a **graph** where connections represent dependencies

User-defined provenance

- Goal: capture lots of provenance automatically based on what steps are executed
- Problem: not everything can be captured automatically
- Annotations offer ability to keep notes about processes
- Users might also specify known causal links that cannot be automatically determined (e.g. a step depends on three system files that were not specified as inputs in the workflow)

Provenance Management

- What is needed to capture, store, and use provenance?
  1. Capture mechanism
  2. Model for representing provenance
  3. Tools to store, query, and analyze provenance

Provenance Capture Mechanisms

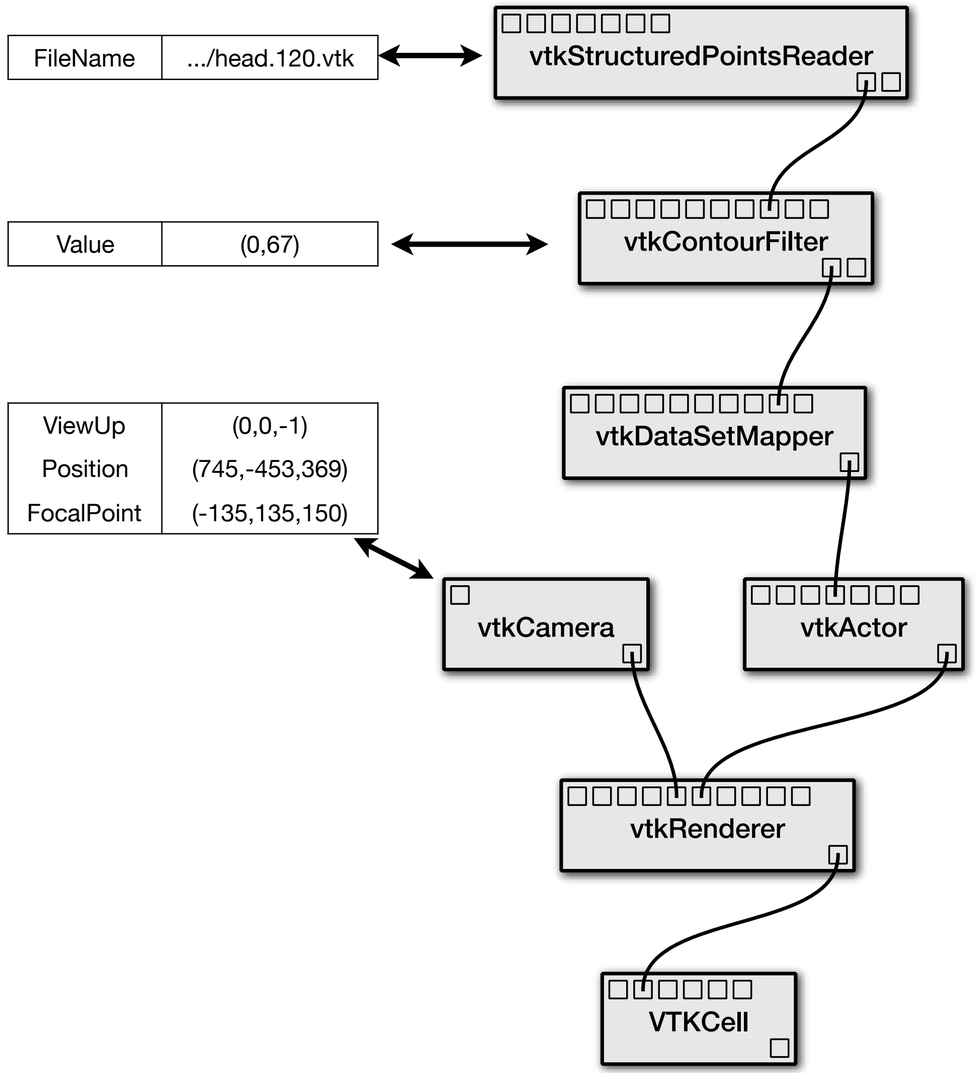
- **Workflow-based**: Since workflow execution is controlled, keep track of all the workflow modules, parameters, etc. as they are executed
- **Process-based**: Each process is required to write out its own provenance information (not centralized like workflow-based)
- **OS-based**: The OS or filesystem is modified so that any activity it does it monitored and the provenance subsystem organizes it
- Tradeoffs:
  - Workflow- and process-based have better abstraction
  - OS-based requires minimal user effort once installed and can capture "hidden dependencies"

Provenance Granularity

- How detailed should our provenance be?
  - **Coarse**: "This program ran with inputs x, y, z and produced outputs a, b, c"
  - **Fine**: "Input x was read into register 4, input y was read in register 5, add operation was performed using registers 4 and 5, …"
- More queries are possible with fine-grained provenance, but…
  - Storage concerns
  - Performance concerns
- Abstraction can help here

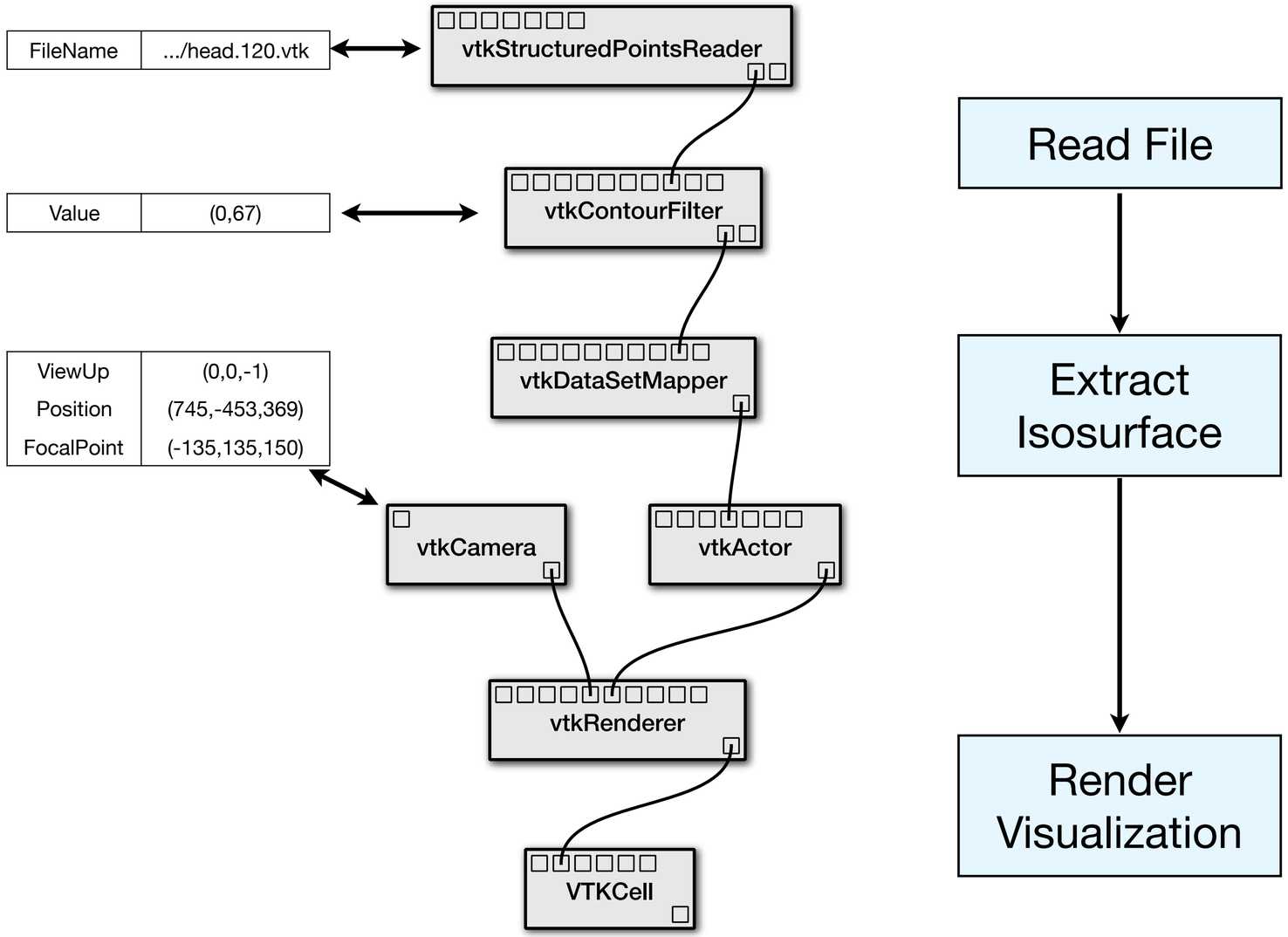
Abstraction: Script, Workflow, Abstract Workflow

```
data = vtk.vtkStructuredPointsReader()
data.SetFileName(../examples/data/head.120.vtk)

contour = vtk.vtkContourFilter()
contour.SetInput(data.GetOutput())
contour.SetValue(0, 67)

mapper = vtk.vtkPolyDataMapper()
mapper.SetInput(contour.GetOutput())
mapper.ScalarVisibilityOff()

actor = vtk.vtkActor()
actor.SetMapper(mapper)

cam = vtk.vtkCamera()
cam.SetViewUp(0,0,-1)
cam.SetPosition(745,-453,369)
cam.SetFocalPoint(135,135,150)
cam.ComputeViewPlaneNormal()

ren = vtk.vtkRenderer()
ren.AddActor(actor)
ren.SetActiveCamera(cam)
ren.ResetCamera()
renwin = vtk.vtkRenderWindow()
renwin.AddRenderer(ren)

style = vtk.vtkInteractorStyleTrackballCamera()
iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renwin)
iren.SetInteractorStyle(style)
iren.Initialize()
iren.Start()
```

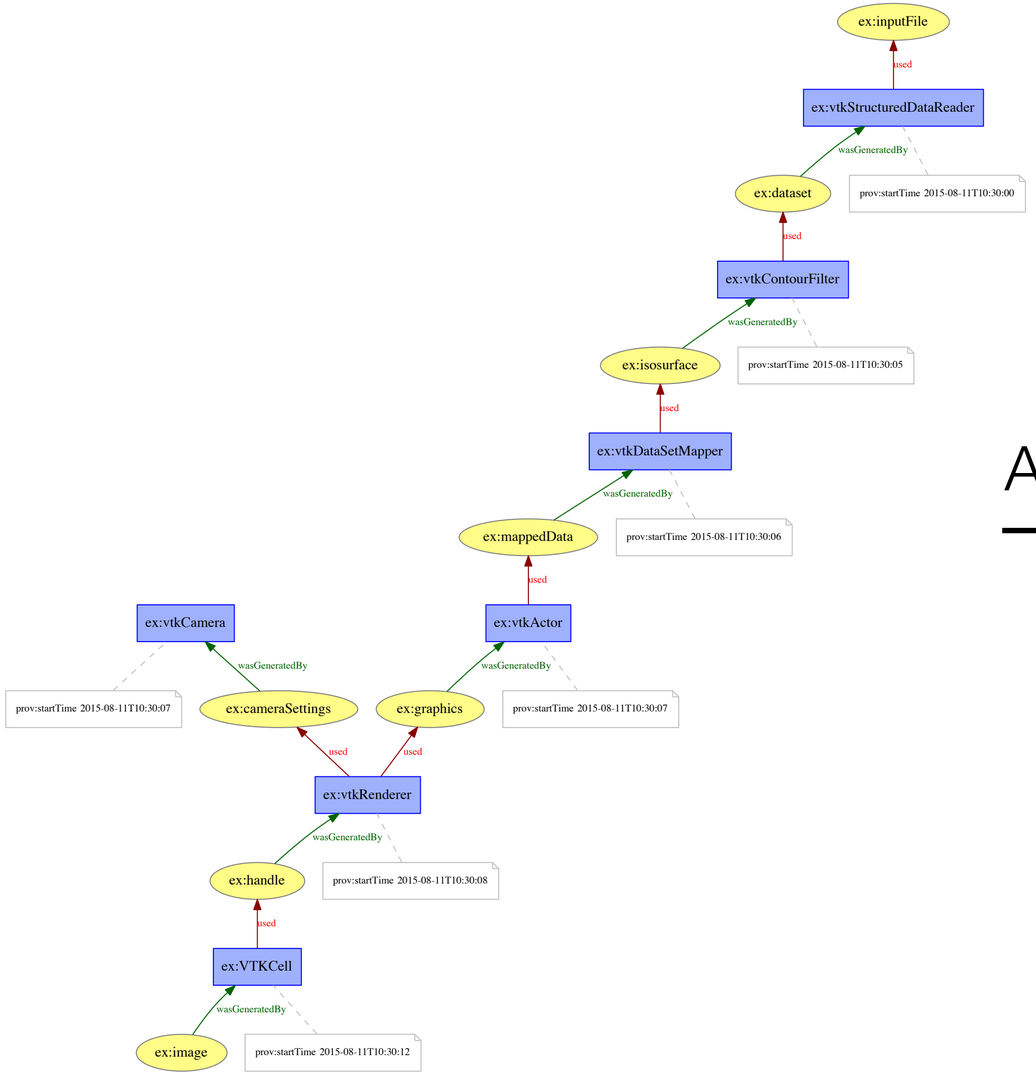


Abstraction: Script, Workflow, Abstract Workflow

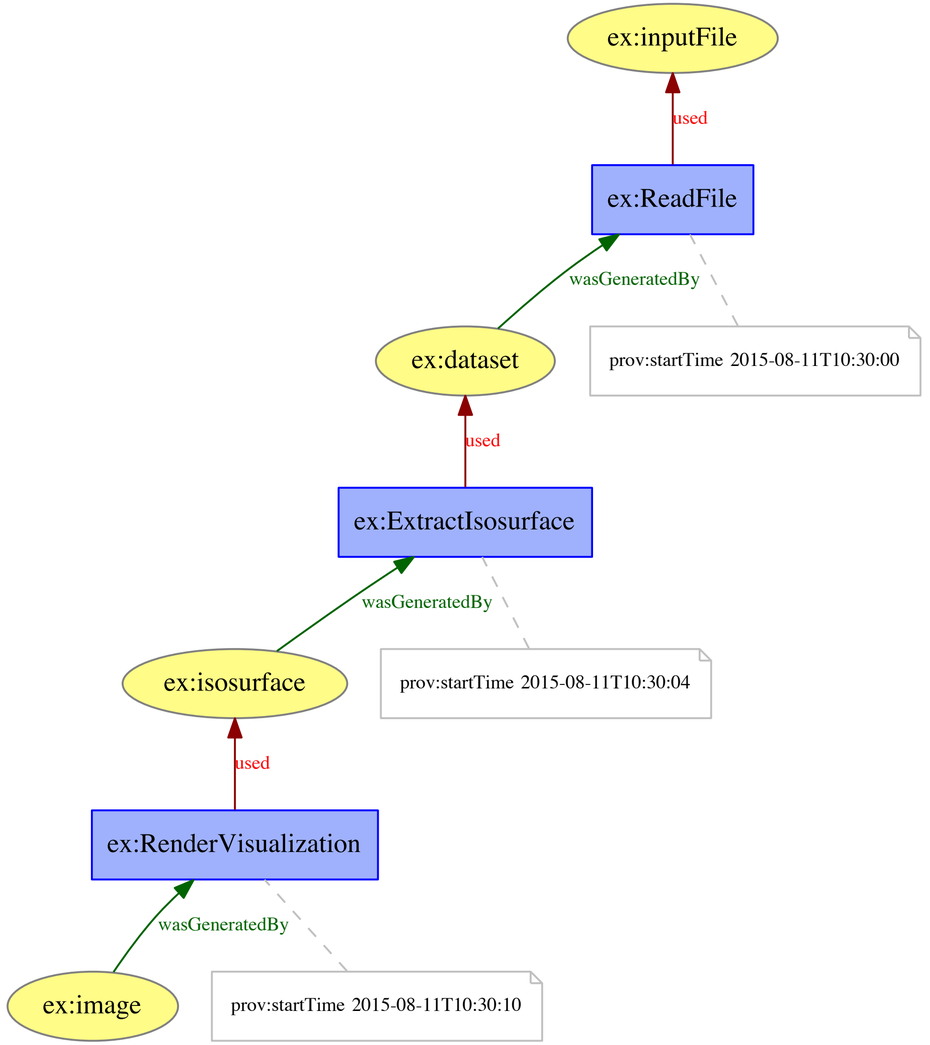
```
data = vtk.vtkStructuredPointsReader()
data.SetFileName(../examples/data/head.120.vtk)

contour = vtk.vtkContourFilter()
contour.SetInput(data.GetOutput())
contour.SetValue(0, 67)

mapper = vtk.vtkPolyDataMapper()
mapper.SetInput(contour.GetOutput())
mapper.ScalarVisibilityOff()

actor = vtk.vtkActor()
actor.SetMapper(mapper)

cam = vtk.vtkCamera()
cam.SetViewUp(0,0,-1)
cam.SetPosition(745,-453,369)
cam.SetFocalPoint(135,135,150)
cam.ComputeViewPlaneNormal()

ren = vtk.vtkRenderer()
ren.AddActor(actor)
ren.SetActiveCamera(cam)
ren.ResetCamera()
renwin = vtk.vtkRenderWindow()
renwin.AddRenderer(ren)

style = vtk.vtkInteractorStyleTrackballCamera()
iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renwin)
iren.SetInteractorStyle(style)
iren.Initialize()
iren.Start()
```



Abstraction: Provenance Views



Abstract



Provenance Storage

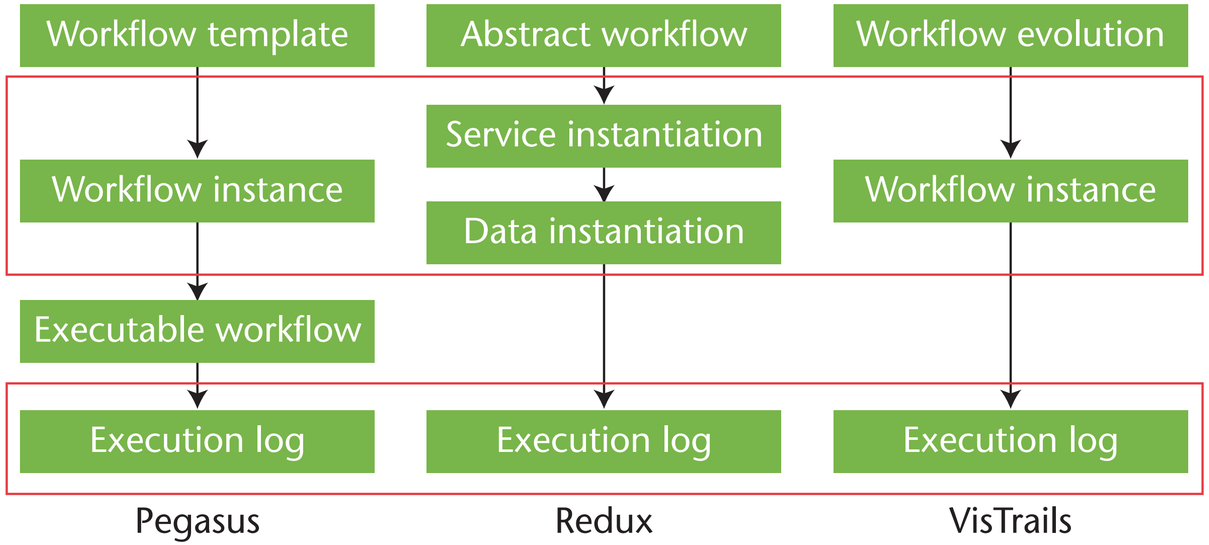
- Keeping provenance for each data item means lots of **repetition**
- Nested data storage also induces repetition
- Coarse provenance is naturally more compact, but how to decide what (not) to store?
- Repeated provenance is not uncommon:
  - Repeating the same computation with a different parameter
  - Creating a new computation that has a very similar structure to one that was run two weeks ago
- Provenance compression/factorization techniques (e.g. [Chapman et al., 2008], [Anand et al., 2009]) take advantage of that to reduce storage costs

Provenance Storage Formats

- Files, relational databases, XML databases, RDF (linked data)
- Log files are good for preserving data but can be bad to query or analyze
- Relational databases are great for column-specific queries but can be bad for dependency queries
- XML databases are more portable than relational databases but are usually less efficient for queries
- RDF triples are better for dependencies and integrating domain-specific knowledge but can be slower

Layered Provenance

- As with relational databases, want to normalize provenance to **minimize redundant information**
- Example: Don't store workflow specification each time that workflow is executed–store it once and reference it
- Also allow different layers for different aspects of provenance



[Freire et. al, 2008]

Provenance Models

- How provenance is represented (more abstract than the details of how it is actually stored)
- PROV (W3C Standard) has different storage backends for provenance but all of it conforms to the same model
- Model the objects involved and their relationships (e.g. activities, dependencies)
- Interoperability is a concern
  - Why? May use multiple tools/techniques to achieve a result, want to analyze the entire provenance chain

Prospective and Retrospective Provenance

- Prospective provenance is what was specified/intended
  - a workflow, script, list of steps
- Retrospective provenance is what actually happened
  - actual data, actual parameters, errors that occurred, timestamps, machine information
- **Do not need** prospective provenance to have retrospective provenance!
- Retrospective provenance is often the same type of information as prospective plus more
- Could have multiple retrospective provenance traces for one prospective provenance listing

Prospective and Retrospective Provenance

- **Example:** Baking a Cake
- Prospective Provenance (Recipe):
  1. Gather ingredients (3/4 cup butter, 3/4 cocoa, 3/4 cup flour, ...)
  2. Preheat oven to 350 degrees
  3. Grease cake pan
  4. Mix wet ingredients in large bowl
  5. Mix dry ingredients in a separate bowl
  6. Add dry mixture to wet mixture
  7. Pour batter into cake pan
  8. Put pan in the oven and bake for 30 minutes
  9. Take cake out of oven and let it cool

Prospective and Retrospective Provenance

- Retrospective Provenance (What actually happened)

1. **Went to store to buy butter**
2. Gathered ingredients (3/4 cup butter, 3/4 cocoa, **1 cup flour**, ...)
3. Greased cake pan
4. Preheated oven to 350 degrees
5. Mixed wet ingredients in large bowl
6. Mixed dry ingredients in a separate bowl
7. Added **wet** mixture to **dry** mixture
8. Poured batter into cake pan
9. Put pan in the oven and baked for **35 minutes**
10. Took cake out of oven and let it cool for **10 minutes**

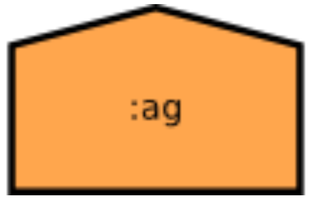
Provenance Model History

- Community organized provenance challenges (2006-2009)
- First Provenance Challenge assessed capabilities of systems
- Second Provenance Challenge examined interoperability
- Led to development of Open Provenance Model (OPM), (2007)
  - Sought to establish interchange format for provenance
- Further work led to PROV W3C Recommendations (2013)
  - Some confusion from name changes from OPM to PROV even though concepts are similar
  - Focus is on **model** not formats

PROV: Three Key Classes

An **entity** is a physical, digital, conceptual, or other kind of thing with some fixed aspects; entities may be real or imaginary.

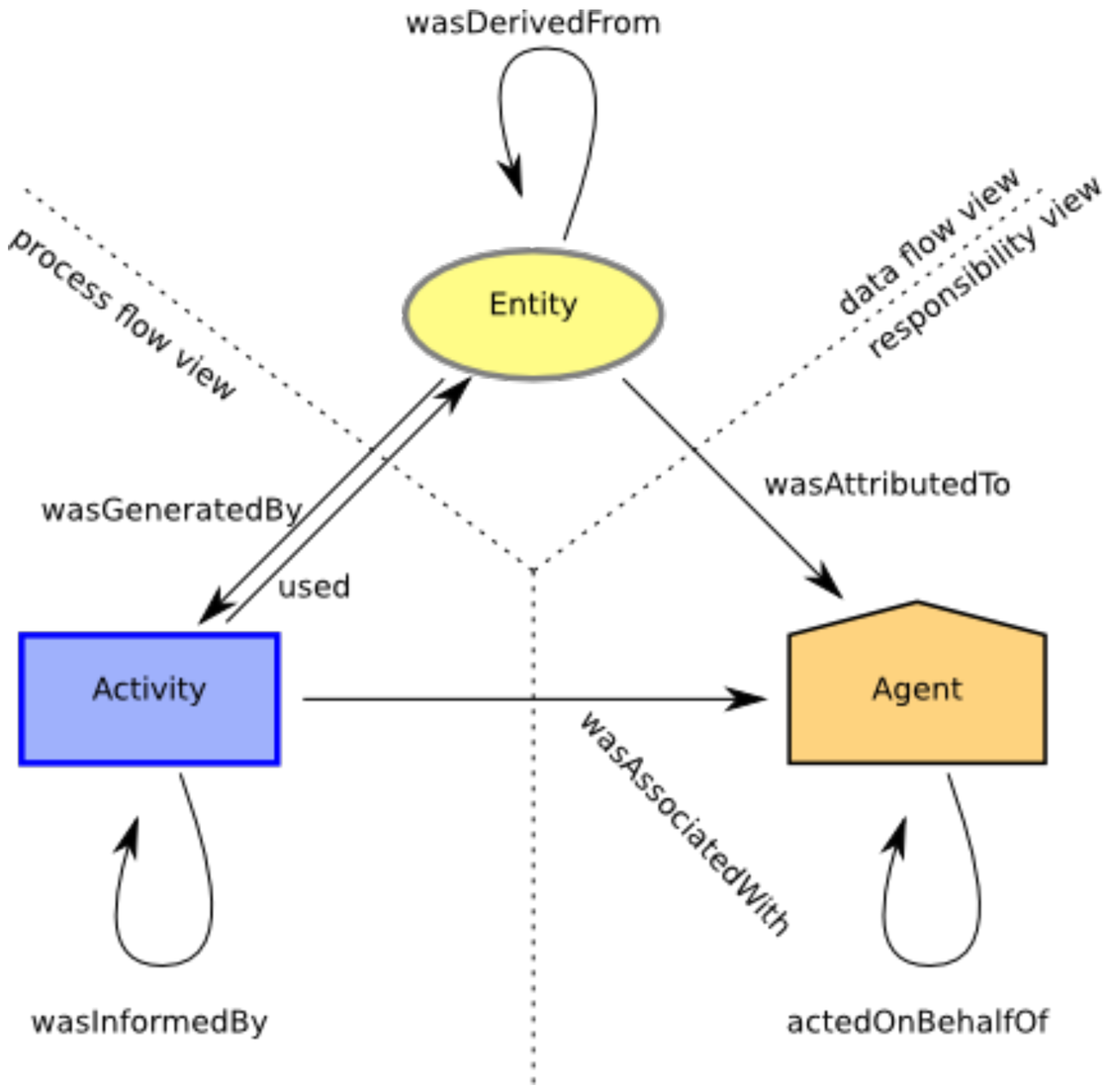
An **activity** is something that occurs over a period of time and acts upon or with entities; it may include consuming, processing, transforming, modifying, relocating, using, or generating entities.



An **agent** is something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity.

[Moreau et al., 2014]

PROV: Three Views of Provenance



[Moreau et al., 2014]

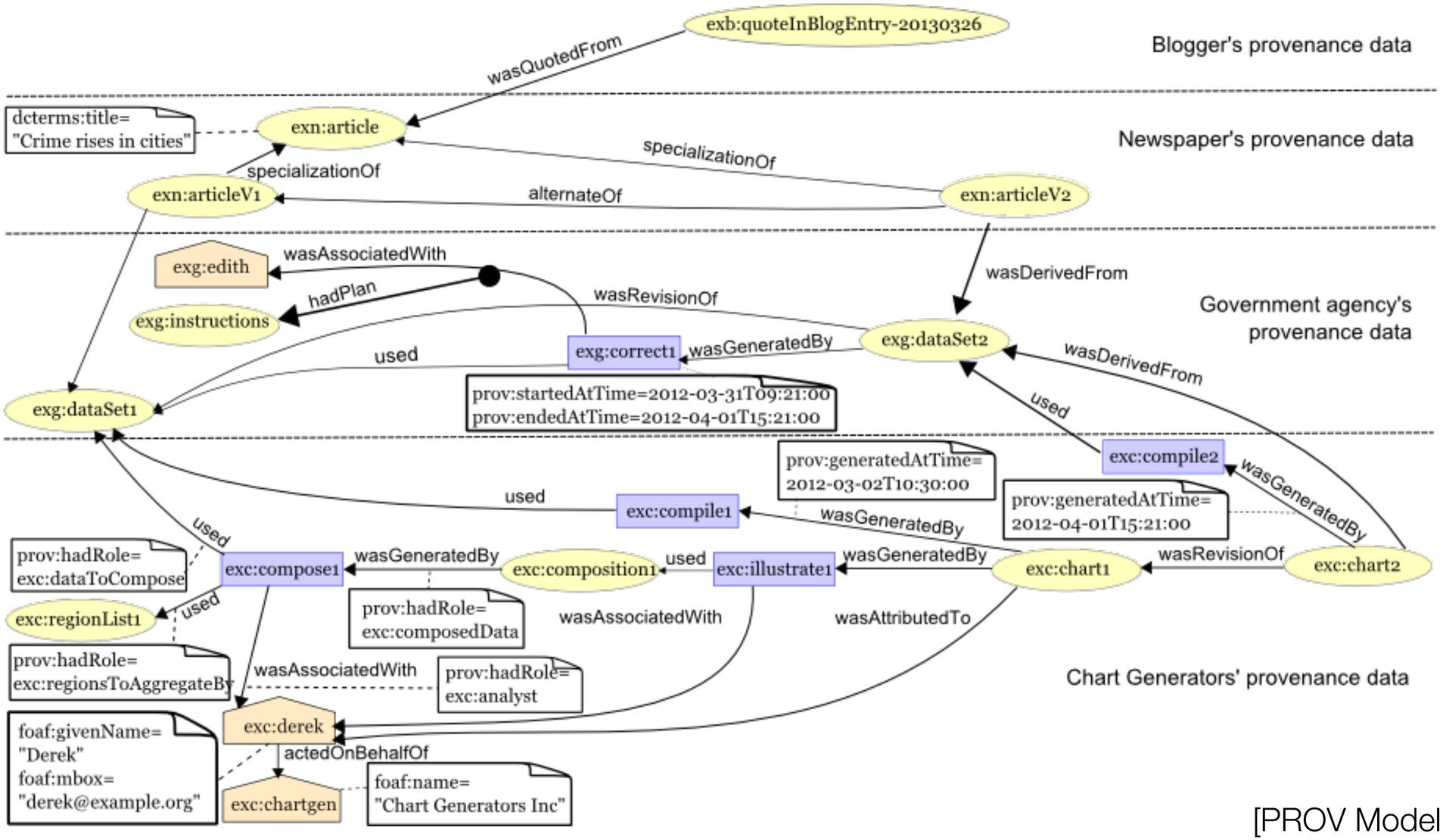
PROV Edges: Derivation

- Derivation Edges:
  - wasGeneratedBy: entity ⟶ activity
  - used: activity ⟶ entity

  - wasDerivedFrom: entity ⟶ entity

[PROV Model Primer, 2013]

PROV Example



[PROV Model Primer, 2013]

Querying Provenance

- Query methods are often tied to storage backend
- SQL, XQuery, Prolog, SPARQL, ...

REDUX

```
SELECT Execution.ExecutableWorkflowId, Execution.ExecutionId, Event.EventId, ExecutableActivity.ExecutableActivityId
from Execution, Execution_Event, Event, ExecutableWorkflow_ExecutableActivity, ExecutableActivity,
    ExecutableActivity_Property_Value, Value, EventType as ET
where Execution.ExecutionId=Execution_Event.ExecutionId
and Execution_Event.EventId=Event.EventId
and ExecutableActivity.ExecutableActivityId=ExecutableActivity_Property_Value.ExecutableActivityId
and ExecutableActivity_Property_Value.ValueId=Value.ValueId and Value.Value=Cast('-m 12' as binary)
and ((CONVERT(DECIMAL, Event.Timestamp)+0)%7)=0 and Execution_Event.ExecutableWorkflow_ExecutableActivityId=
   ExecutableWorkflow_ExecutableActivity.ExecutableWorkflow_ExecutableActivityId
and ExecutableWorkflow_ExecutableActivity.ExecutableWorkflowId=Execution.ExecutableWorkflowId
and ExecutableWorkflow_ExecutableActivity.ExecutableActivityId=ExecutableActivity.ExecutableActivityId
and Event.EventTypeId=ET.EventTypeId and ET.EventTypeName='Activity Start';
```

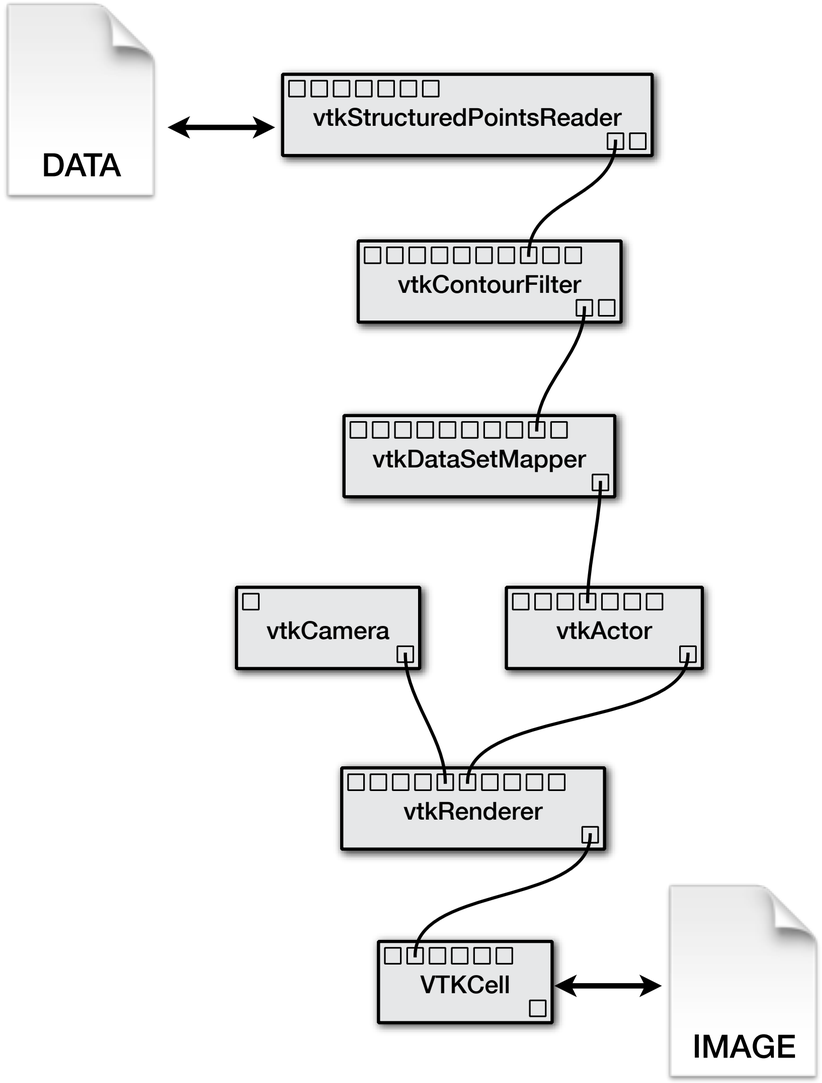
VisTrails

```
wf{*}: x where x.module='AlignWarp' and x.parameter('model')='12'
        and (log{x}: y where y.dayOfWeek='Monday')
```

MyGrid

```
SELECT ?p
where (?p <http://www.mygrid.org.uk/provenance#startTime> ?time) and (?time > date)
using ns for <http://www.mygrid.org.uk/provenance#> xsd for <http://www.w3.org/2001/XMLSchema#>

SELECT ?p
where <urn:lsid:www.mygrid.org.uk:experimentinstance:HXQOVQA2ZI0>
(?p <http://www.mygrid.org.uk/provenance#runsProcess> ?processname .
?p <http://www.mygrid.org.uk/provenance#processInput> ?inputParameter .
?inputParameter <ont:model> <ontology:twelfthOrder>)
using ns for <http://www.mygrid.org.uk/provenance#> ont for <http://www.mygrid.org.uk/ontology#>
```

Querying Provenance

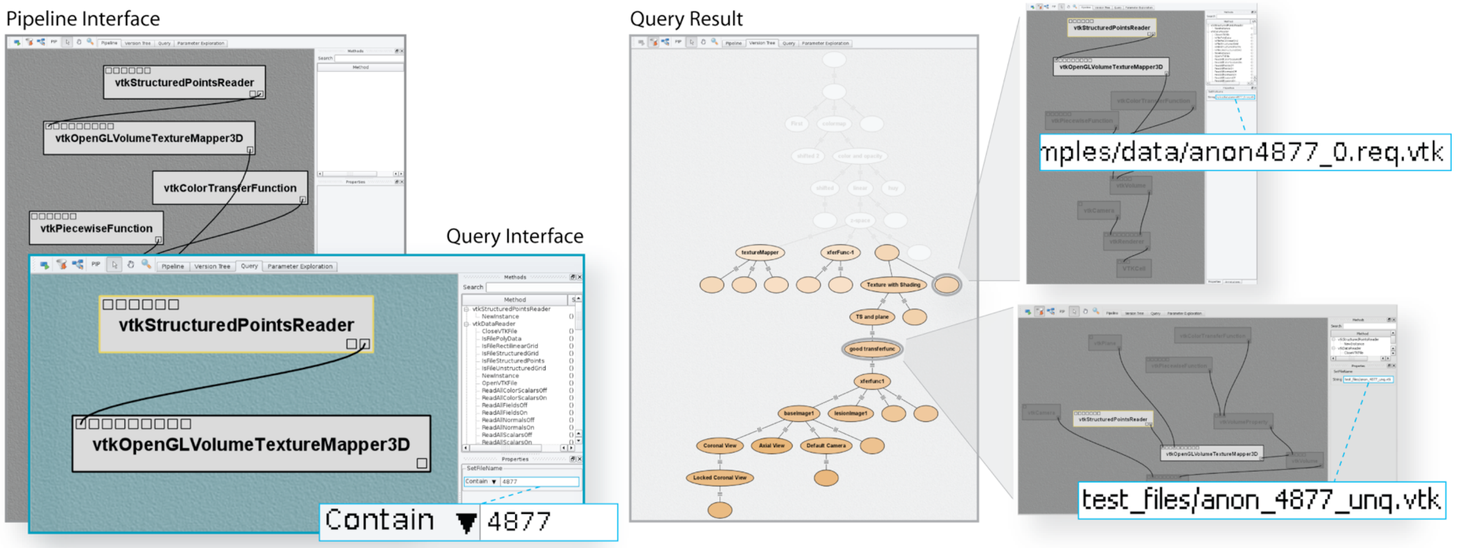


- *What process led to the output image?*
- *What input datasets contributed to the output image?*
- *What workflows include resampling and isosurfacing with isovalue* `57`*?*

- Graph traversal or graph patterns
  - How do we write such queries?

Querying Provenance by Example

- Provenance is represented as graphs: hard to specify queries using text!
- Querying workflows by example [Scheidegger et al., TVCG 2007; Beeri et al., VLDB 2006; Beeri et al. VLDB 2007]
  - WYSIWYQ -- What You See Is What You Query
  - Interface to create workflow is same as to query



Stronger Links Between Provenance and Data

```
<workflow_exec id="1">
  <m_exec id="5"
          name="vtkStructuredDataReader"
          package="edu.utah.sci.vistrails
          version="5.6.0">
    <param id="2" name="SetFile"
           value="/MyData/05-12-sc2.dat"/>
  </m_exec>
  <m_exec id="6"
          name="vtkContourFilter"
          package="edu.utah.sci.vistrails.vtk"
          version="5.6.0">
    <param id="3" name="SetValue"
           value="[1, 57]"/>
    <param id="4" name="ComputeScalarsOn"
           value="True"/>
  </m_exec>

  ...

  <m_exec id="11"
          name="FileSink"
          package="edu.utah.sci.vistrails.basic"
          version="1.5">
    <param id="15" name="path"
           value="/home/a/results/23.out"/>
  </m_exec>
```

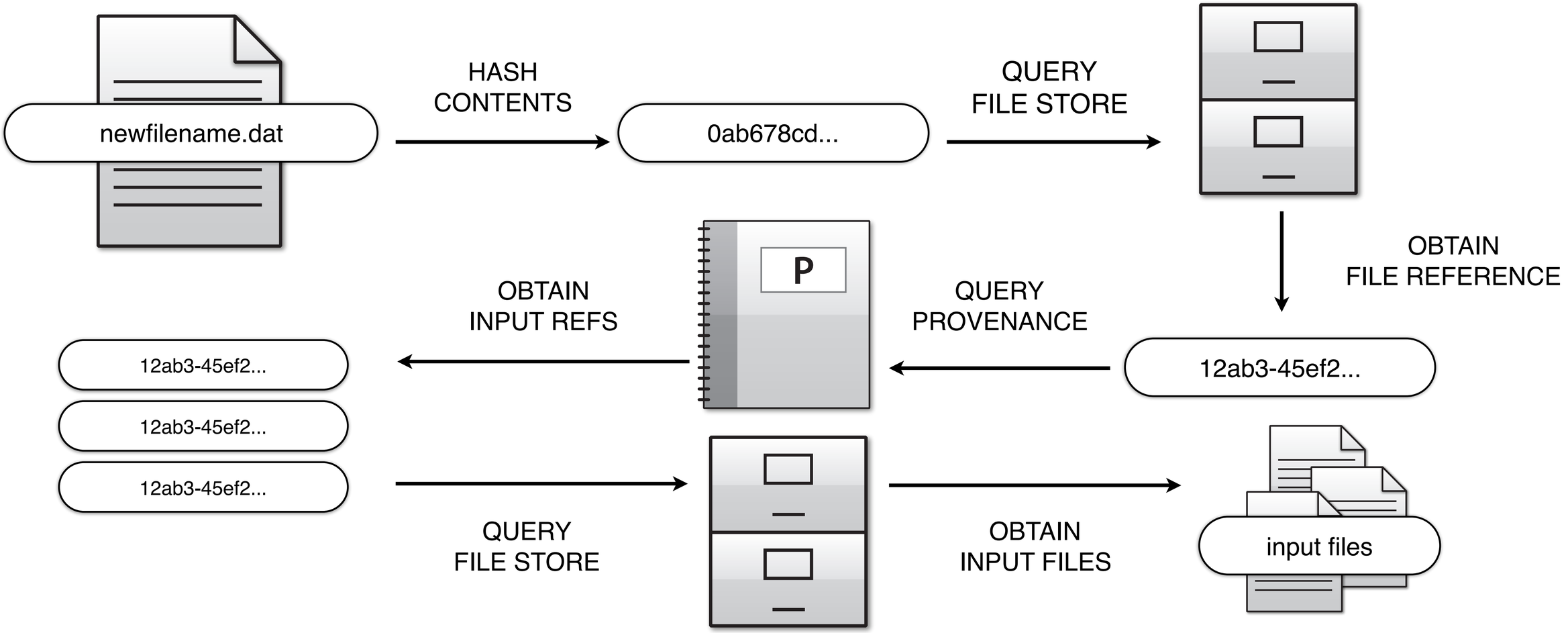
FILE NOT FOUND

FILE NOT FOUND

- Filenames are often the mode of identification in data exploration
- We might also use URIs or access curated data stores
  - Always expected for exploratory tasks?
  - What happens if offline?
- Solution:
  - Managed store for data associated with computations
  - Improved data identification
  - Automatic versioning

[Koop et. al, 2010]

Provenance from Data



[Koop et. al, 2010]

Provenance-Enabled Systems

Table 1. Provenance-enabled systems.

System	Capture mechanism	Prospective provenance	Retrospective provenance	Workflow evolution
REDUX	Workflow-based	Relational	Relational	No
Swift	Workflow-based	SwiftScript	Relational	No
VisTrails	Workflow-based	XML and relational	Relational	Yes
Karma	Workflow- and process-based	Business Process Execution Language	XML	No
Kepler	Workflow-based	MoML	MoML variation	Under development
Taverna	Workflow-based	Scufl	RDF	Under development
Pegasus	Workflow-based	OWL	Relational	No
PASS	OS-based	N/A	Relational	No
ES3	OS-based	N/A	XML	No
PASOA/PreServ	Process-based	N/A	XML	No

[Freire et. al, 2008]

Provenance-Enabled Systems

Table 1. Provenan

System	Storage	Query support	Available as open source?
REDUX	Relational database management system (RDBMS)	SQL	No
Swift	RDBMS	SQL	Yes
VisTrails	RDBMS and files	Visual query by example, specialized language	Yes
Karma	RDBMS	Proprietary API	Yes
Kepler	Files; RDBMS planned	Under development	Yes
Taverna	RDBMS	SPARQL	Yes
Pegasus	RDBMS	SPARQL for metadata and workflow; SQL for execution log	Yes
PASS	Berkeley DB	nq (proprietary query tool)	No
ES3	XML database	XQuery	No
PASOA/PreServ	Filesystem, Berkeley DB	XQuery, Java query API	Yes

[Freire et. al, 2008]

Provenance-Enabled Systems

Table 1. Provenan

System	Storage	Query support	Available as open source?
REDUX	Relational database management system (RDBMS)	SQL	No
Swift	RDBMS	SQL	Yes
VisTrails	RDBMS and files	Visual query by example, specialized language	Yes
Karma	RDBMS	Proprietary API	Yes
Kepler	Files; RDBMS planned	Under development	Yes
Taverna	RDBMS	SPARQL	Yes
Pegasus	RDBMS	SPARQL for metadata and workflow; SQL for execution log	Yes
PASS	Berkeley DB	nq (proprietary query tool)	No
ES3	XML database	XQuery	No
PASOA/PreServ	Filesystem, Berkeley DB	XQuery, Java query API	Yes

[Freire et. al, 2008]