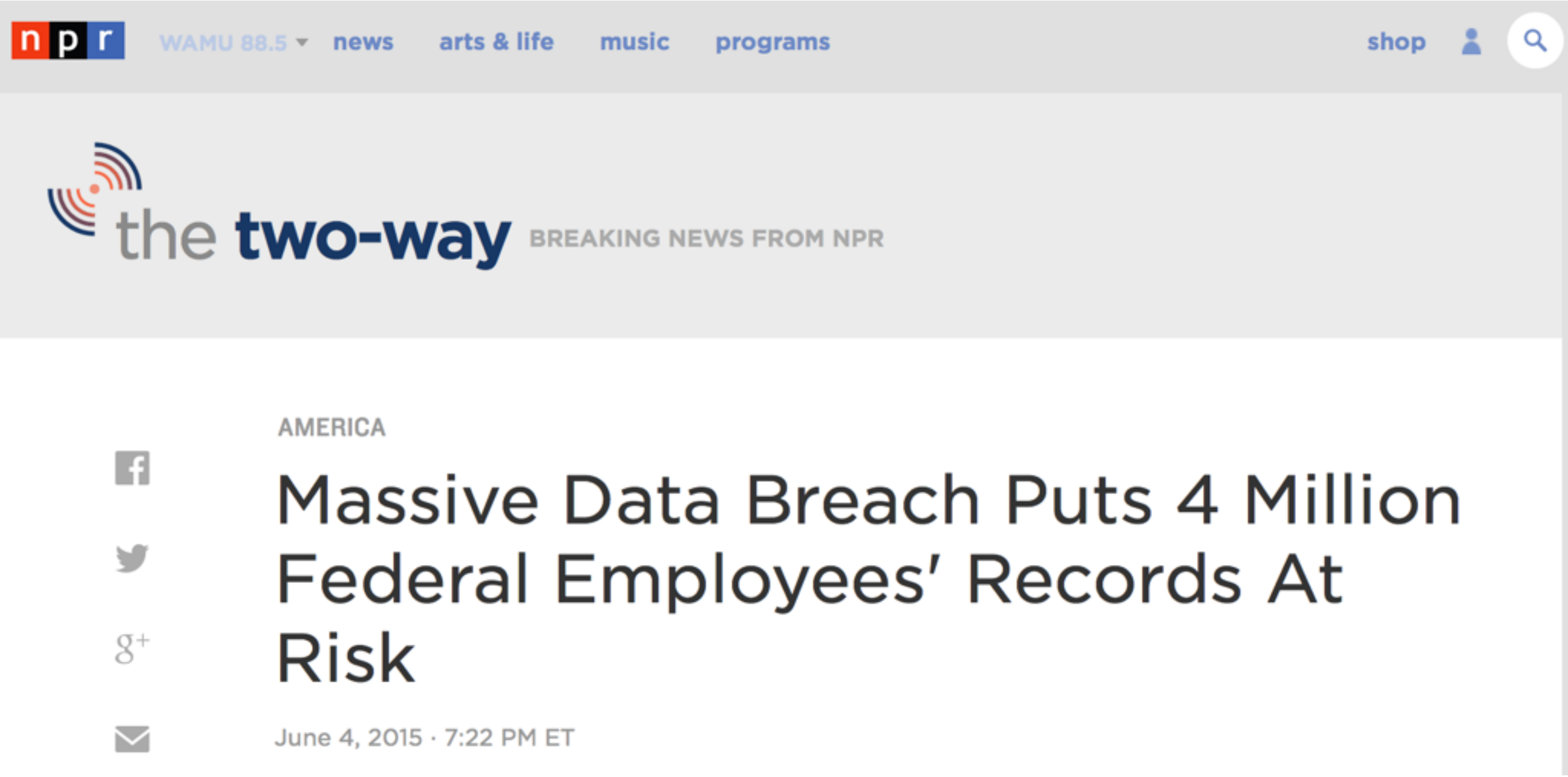
Can we make systems more secure by recording and analyzing detailed provenance records?

IRDS
October 27, 2016
James Cheney



AMERICA

Massive Data Breach Puts 4 Million Federal Employees' Records At Risk

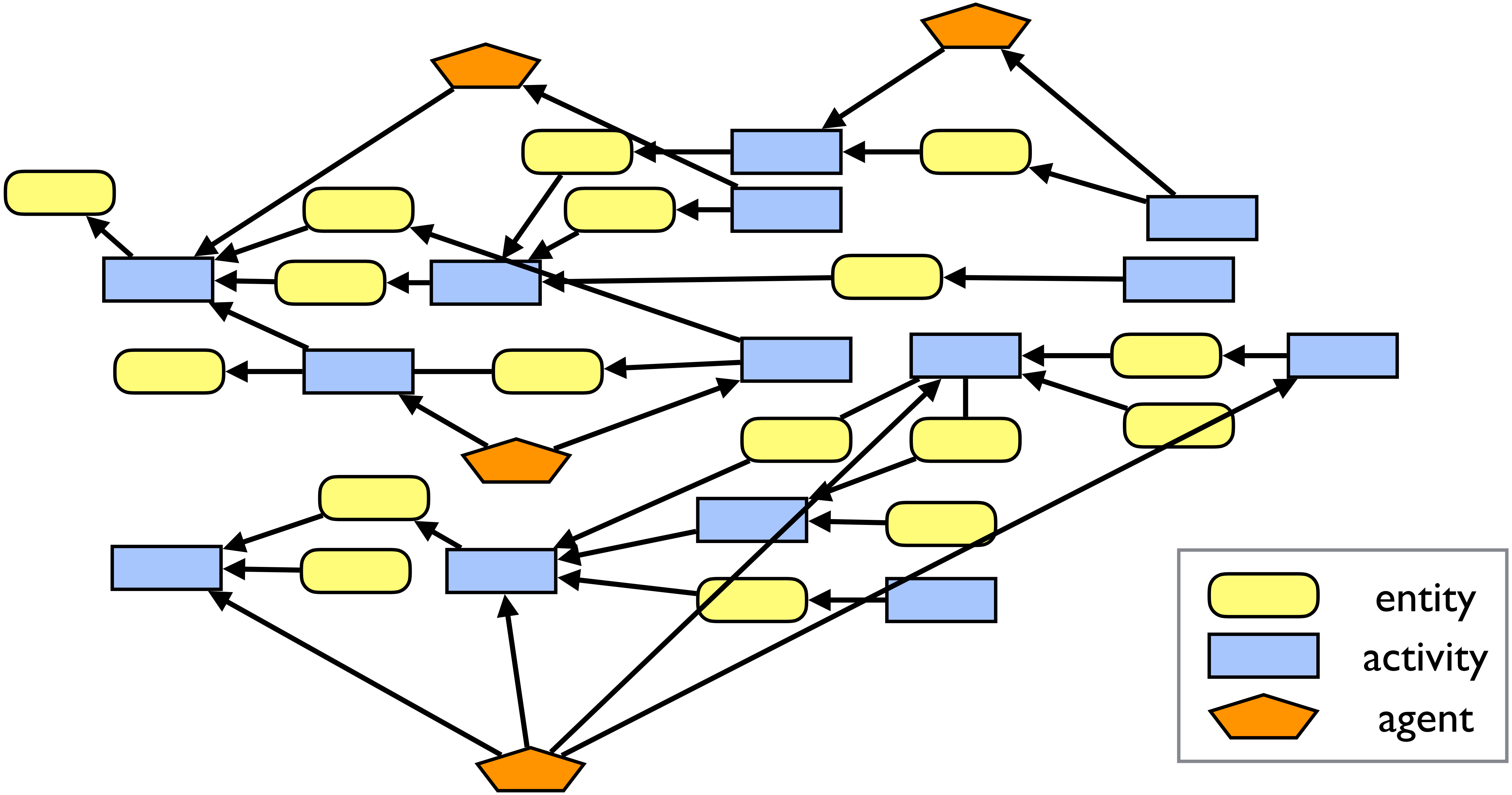
June 4, 2015 · 7:22 PM ET

Office of Personnel Management breach (2015)

Context

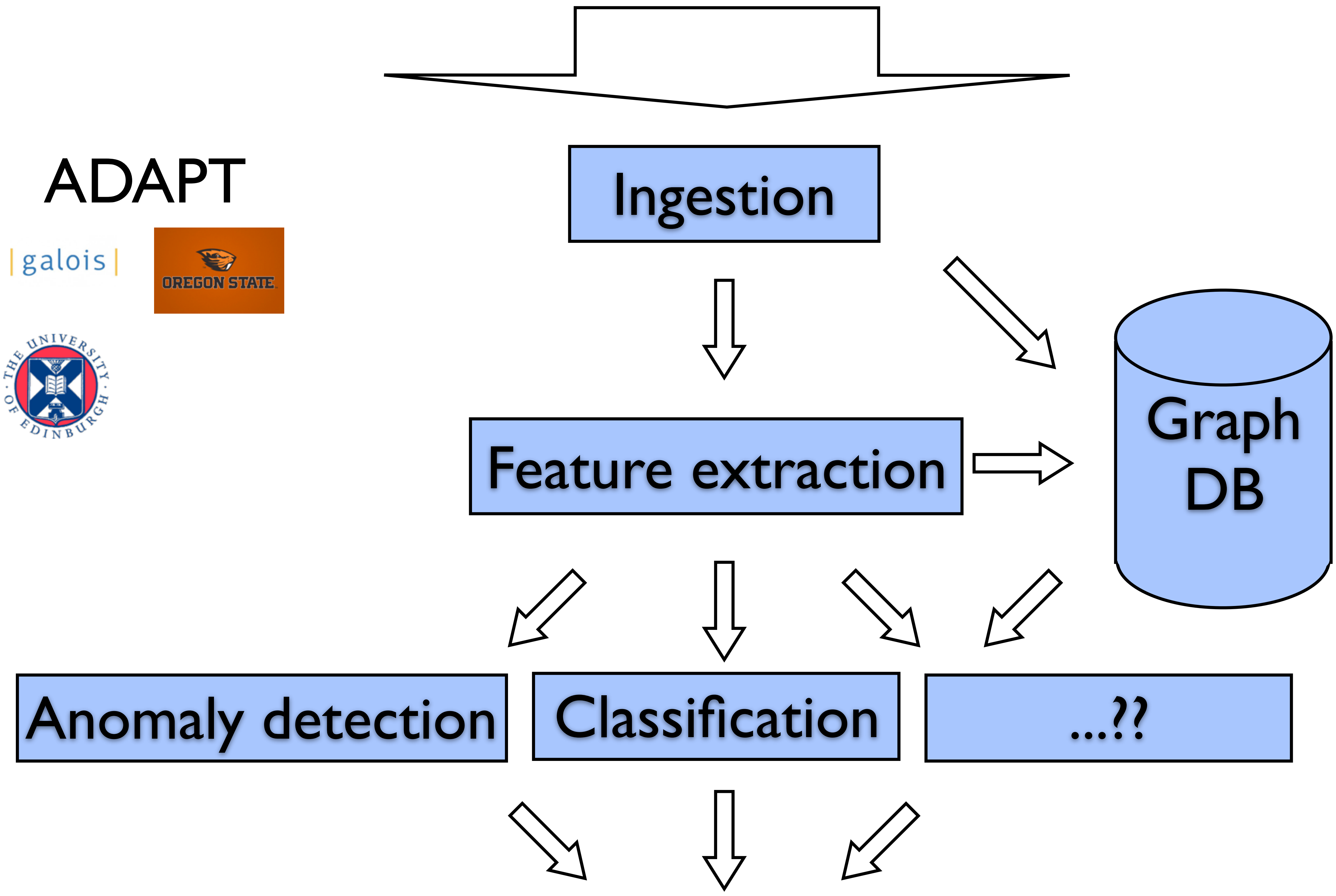
- "Advanced persistent threats" (APTs) are stealthy, long-term, resourceful attackers
    - Simulate normal user behavior most of the time
    - Lateral attacks, avoid violating fixed security policy
    - Think government, not garage
- *Transparent Computing*: try to fight APTs through pervasive recording and analysis of provenance
    - aka "event traces/logs on steroids"
    - $60m DARPA research program (2015-19)

"Provenance" - representation of the origin, history or ownership of data



Incoming provenance data (up to 2MB/s)

ADAPT



Attack subgraph / warning

Case study

- First TC evaluation was in September
- Recorded various systems for 3-4 days
  - **while being attacked by friendly colleagues**
- Up to 175GB of raw data per system
  - just loading the data into graph DBMSs took hours/days...
- We can make (subsets of) the data available for projects!

Needle in a haystack

- We then had to **find** the attacks
- It turns out to be very challenging to detect "attack subgraphs" automatically
  - We did not have labeled training data
  - We did not necessarily know what features would be most relevant
- So far: anomaly detection to find "suspicious" starting nodes
  - Followed by manual exploration / querying of the graph
- Much more could be done!

Mo' provenance = mo' problems

- Expect data rates of up to **2MB/sec**, but still struggling to manage / analyze GBs of prov in **minutes**
- General-purpose graph DBs (neo4j, titan/cassandra) not quite there yet for this scenario
  - Fast/continuous data loading seems challenging
  - Query languages/optimization also have many idiosyncrasies
- Compression or other ways to minimize / discard uninteresting "normal" activity?
- Streaming analysis/summarization to make it easier to find unusual activity patterns?
- Any other interesting off-the-shelf unsupervised learning, or graph algorithms that can run on graphs with millions/billions of edges?

Query language matters

- *Gremlin* query language (Apache)
  - "navigational", not "declarative"

```
X = g.V().has(‘pid’,2304);
g.V(x).until(both().hasLabel('EDGE_EVENT_AFFECTS_SUBJECT').count().is(0)).repeat(has('subjectType
',0).in().hasLabel('EDGE_EVENT_AFFECTS_SUBJECT').in().hasLabel('Subject').has('subjectType',
4).out().hasLabel('EDGE_EVENT_ISGENERATEDBY_SUBJECT').out().hasLabel('Subject').has('subjectT
ype',0).as('b')).select('b').unfold()"
```

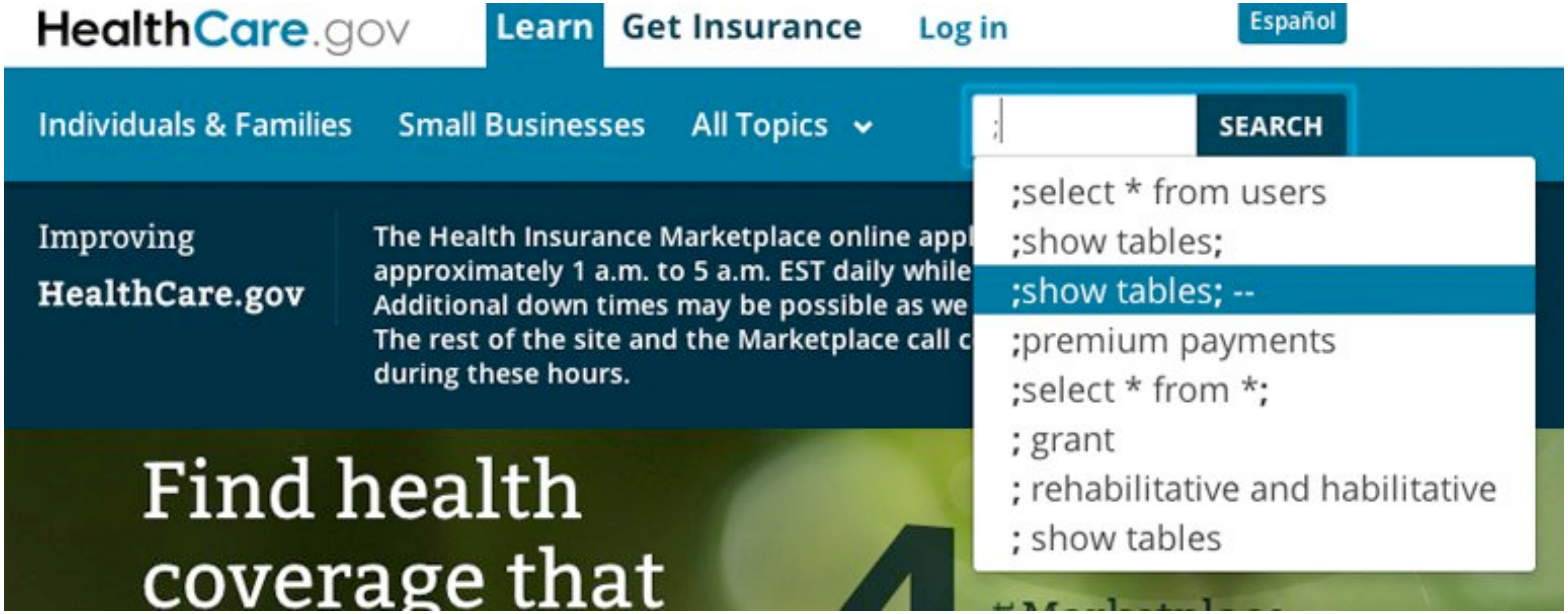
- Maybe something like this would be better?

```
MATCH (x:Process) WHERE x.pid = “2304”
MATCH (y:Process)
MATCH path =
  (x)<-{[:AFFECTS_SUBJECT]-(:Event)<-[:ISGENERATED_BY]-(y)*}
RETURN path.y
```

Related issue

- In the TC program, there are 5 "provider" projects and 3 "analyzer" projects
  - This led almost immediately to the need for a "common data model".
  - But this model only specifies the common "syntax"
- Only now are we starting to discuss "semantic" alignment across the providers
- Open question: Entity resolution/duplicate elimination? Defining consistent standards for "completeness" of the provenance?

Research question: Web/Database programming

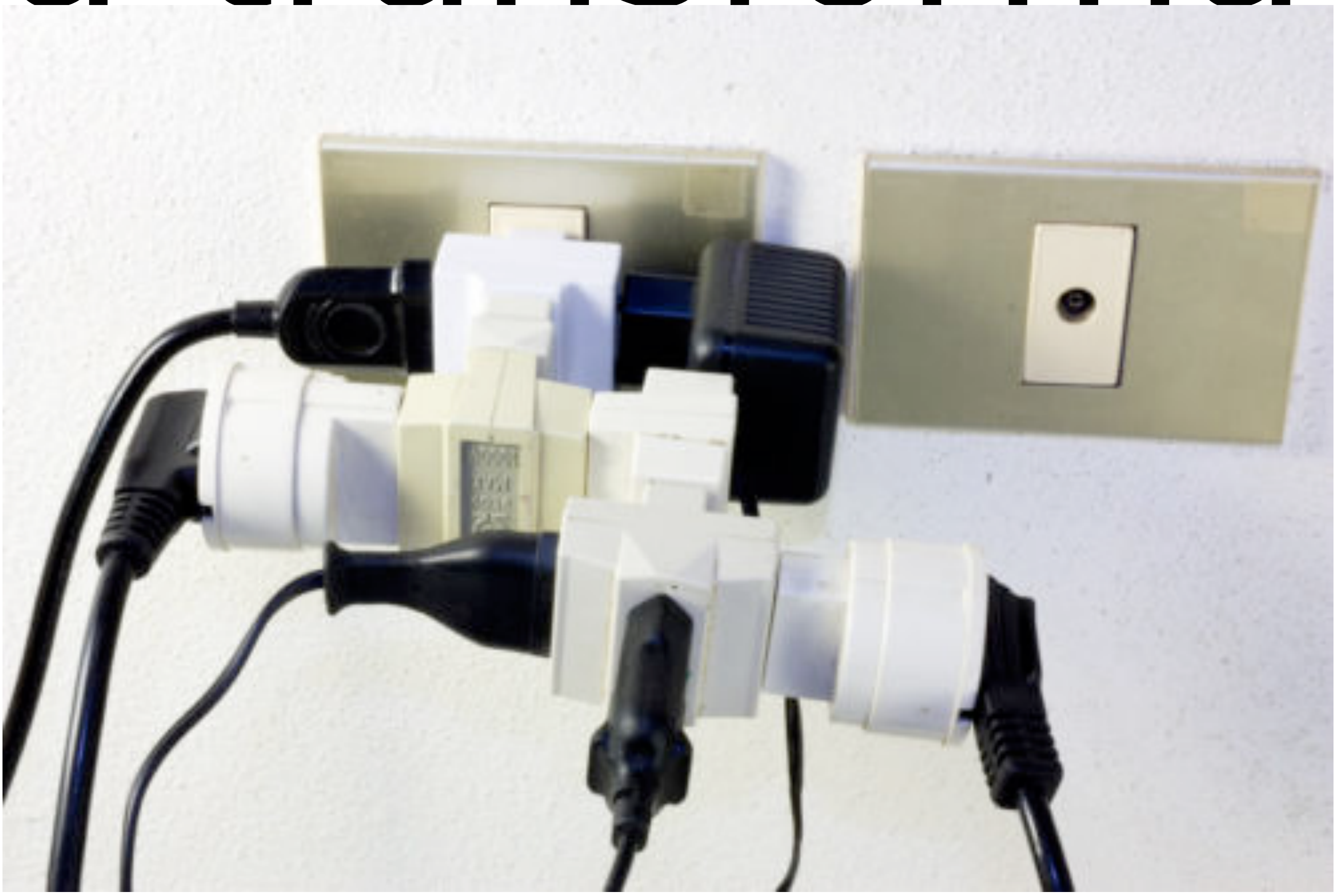


The Health Insurance Marketplace online appl
approximately 1 a.m. to 5 a.m. EST daily while
Additional down times may be possible as we
The rest of the site and the Marketplace call c
during these hours.

;select * from users
;show tables;
;show tables; --
;premium payments
;select * from *;
; grant
; rehabilitative and habilitative
; show tables

How can we (safely/securely) program multiple layers (database, browser, regular PL)?

Research question: Data transformation



- How do I make use of data in format X with tools that expect Y?
- What if some of X is missing or Y requires information that X doesn't provide?

Summary

- My work explores the interaction between language design, semantics, and data management.
- A current focus is the interaction of provenance and security
  - Transparent Computing/ADAPT: analyzing provenance to find attacks
  - Also interested in security/privacy for provenance
- Other interests: integrating data management into programming languages, data transformation & synchronization
- to improve understanding of, address complex data management problems
  - Also relevant to "science data", though maybe not what people currently think of as "data science"