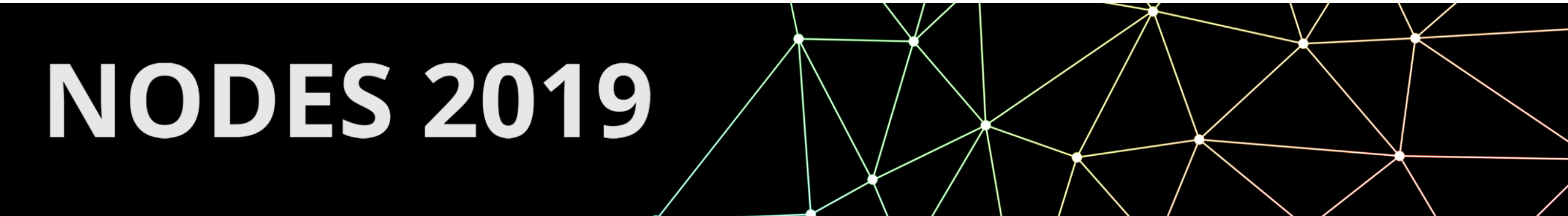
Collaboration insights from data access analytics
"Follow the data"

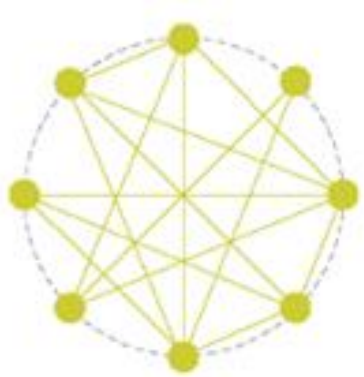
Ravi Krishnaswamy

Autodesk Inc.



How Valuable is a Network ?

Sarnoff
$V=n$



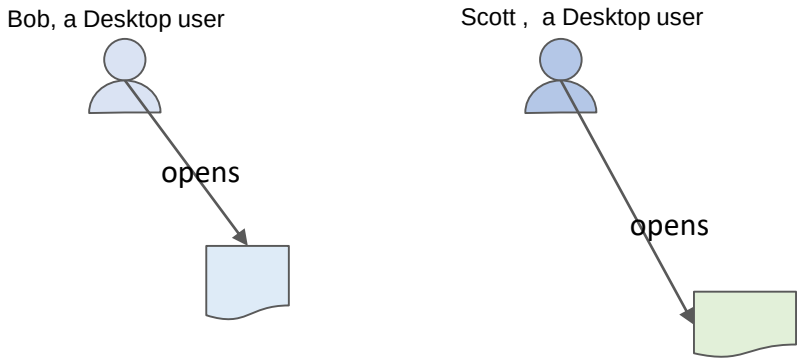
Metcalfe
$V=n^2$

Reed
$V=2^n$

Reed: the utility of large networks, particularly social networks, can scale exponentially with the size of the network

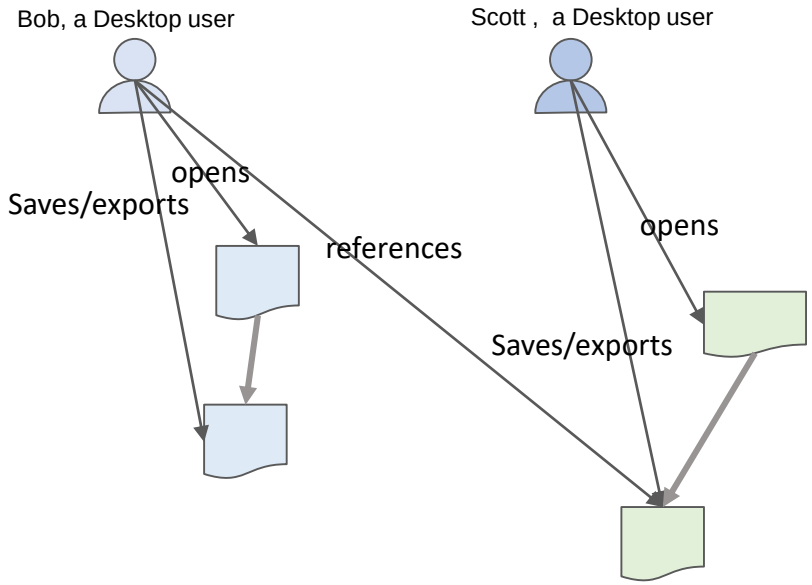
Community detection through data

Core concepts



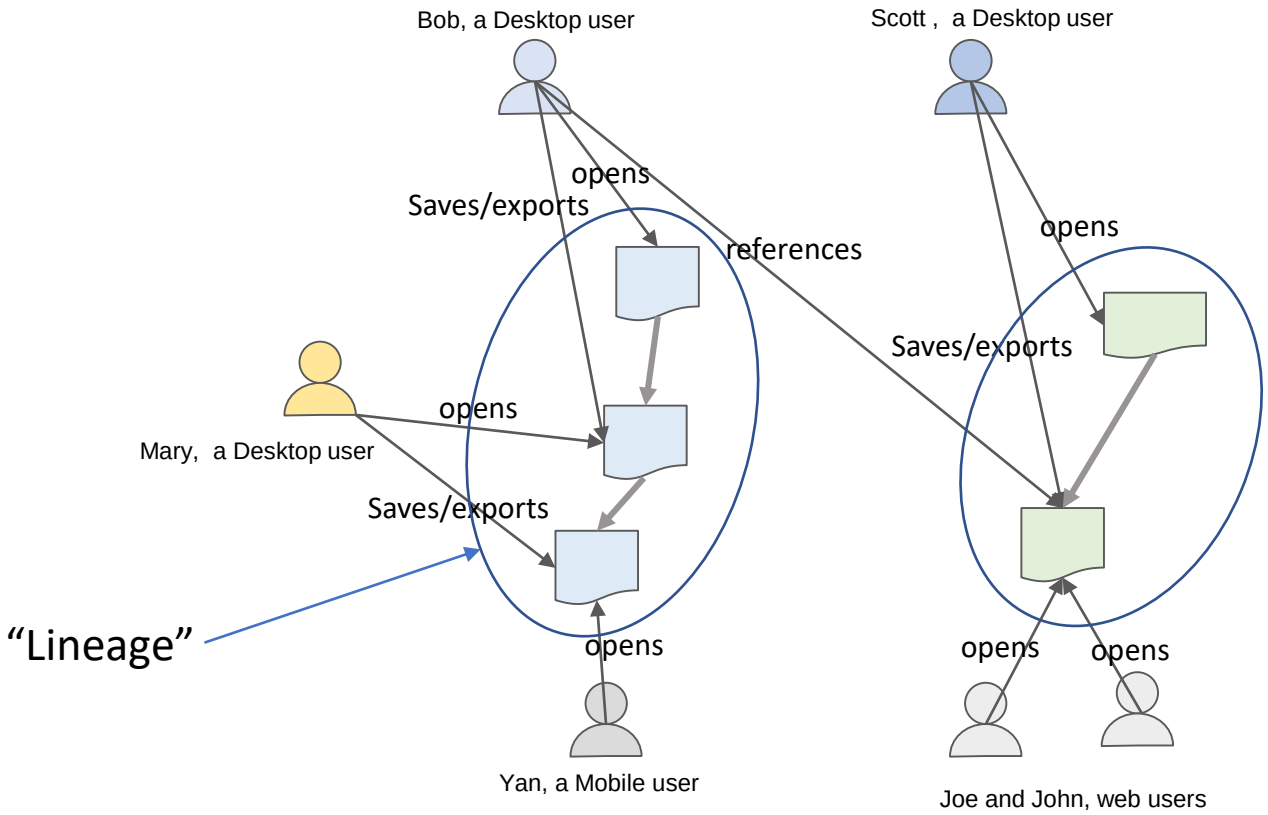
Community detection through data

Core concepts



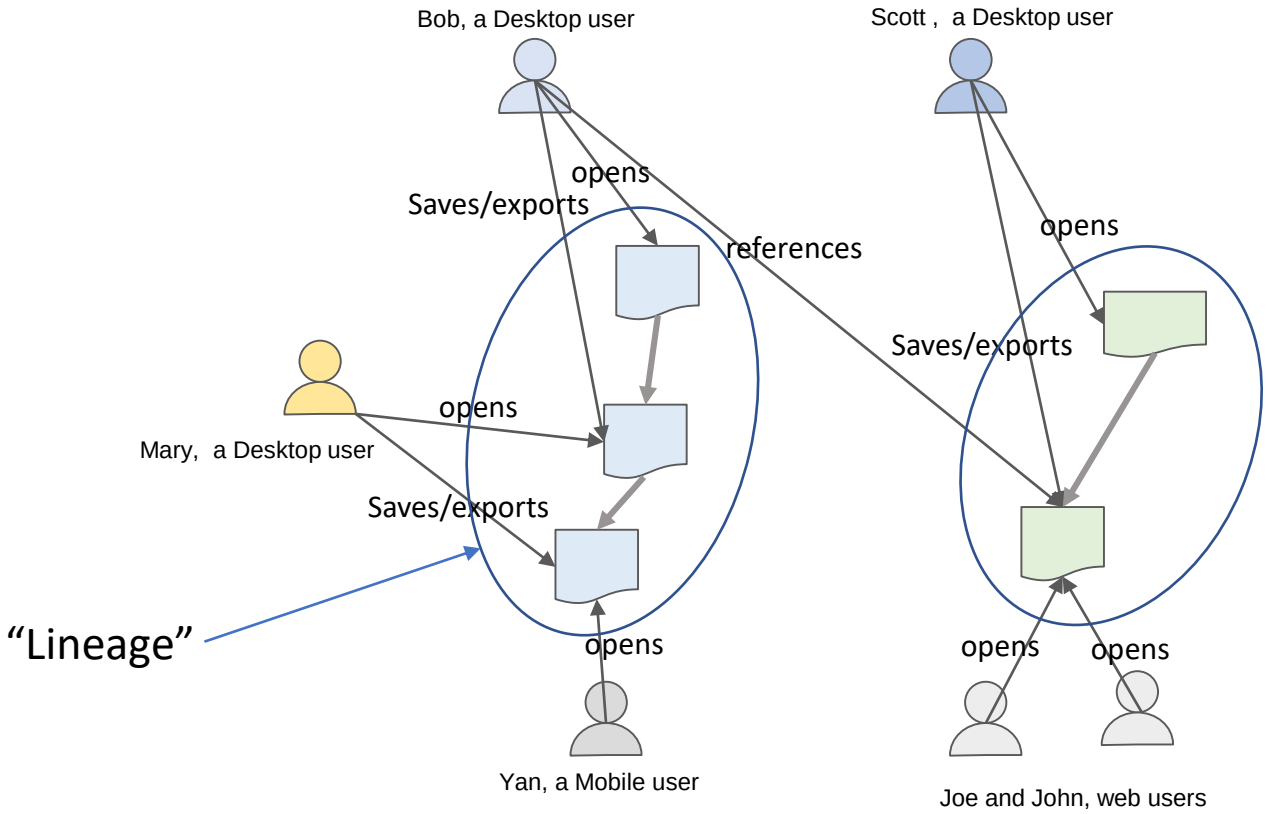
Community detection through data

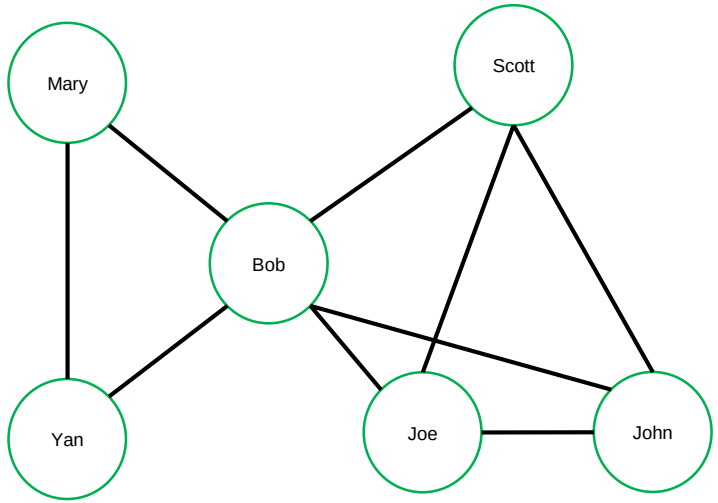
Core concepts



Community detection through data

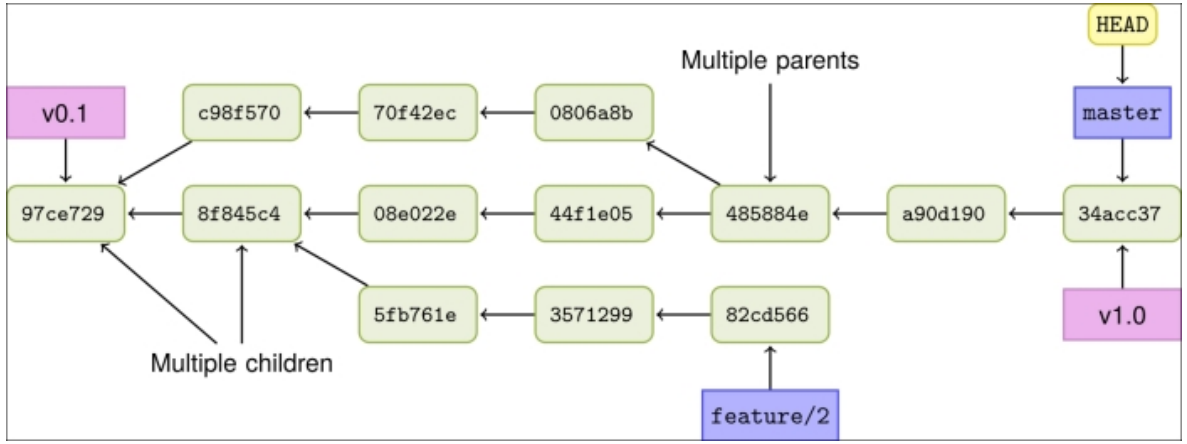
Core concepts

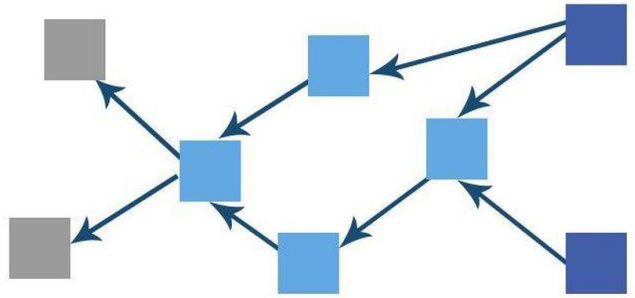


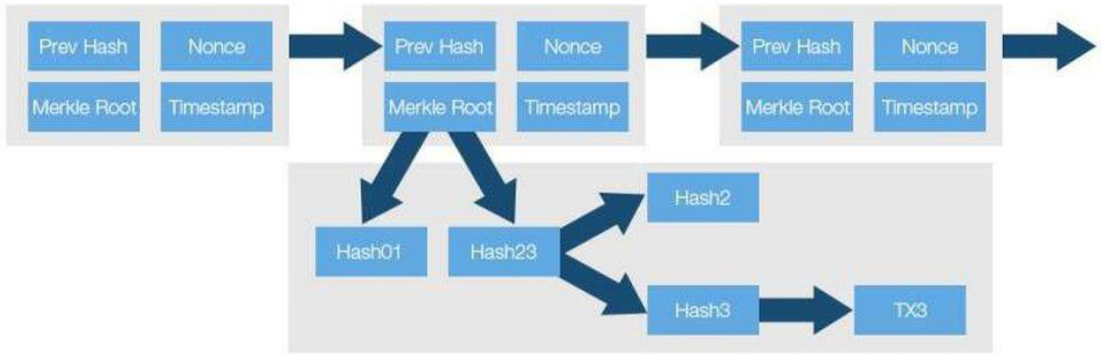


Hash fingerprints to connect versions

Existing use cases

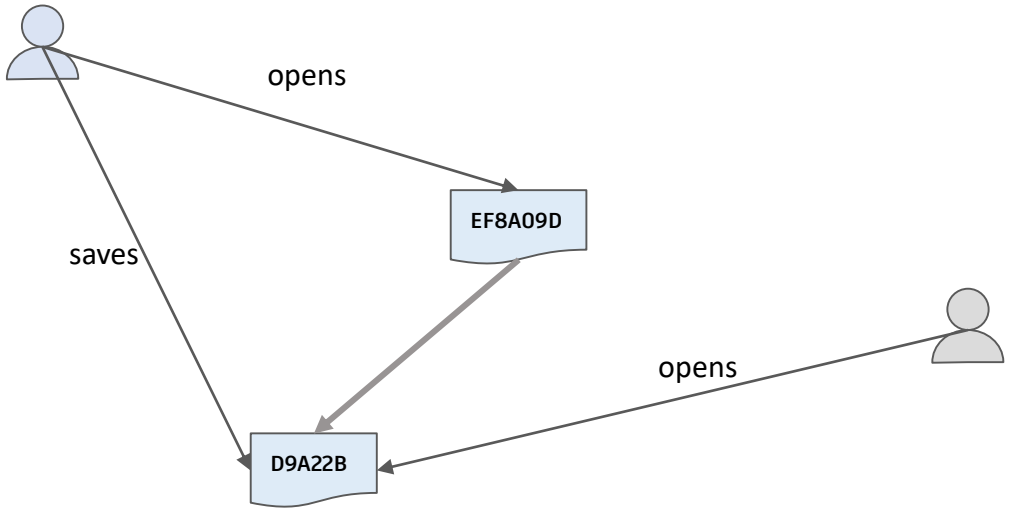






Connecting versions through hashes

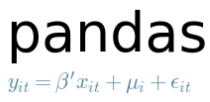
Ours is another use case

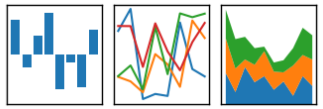


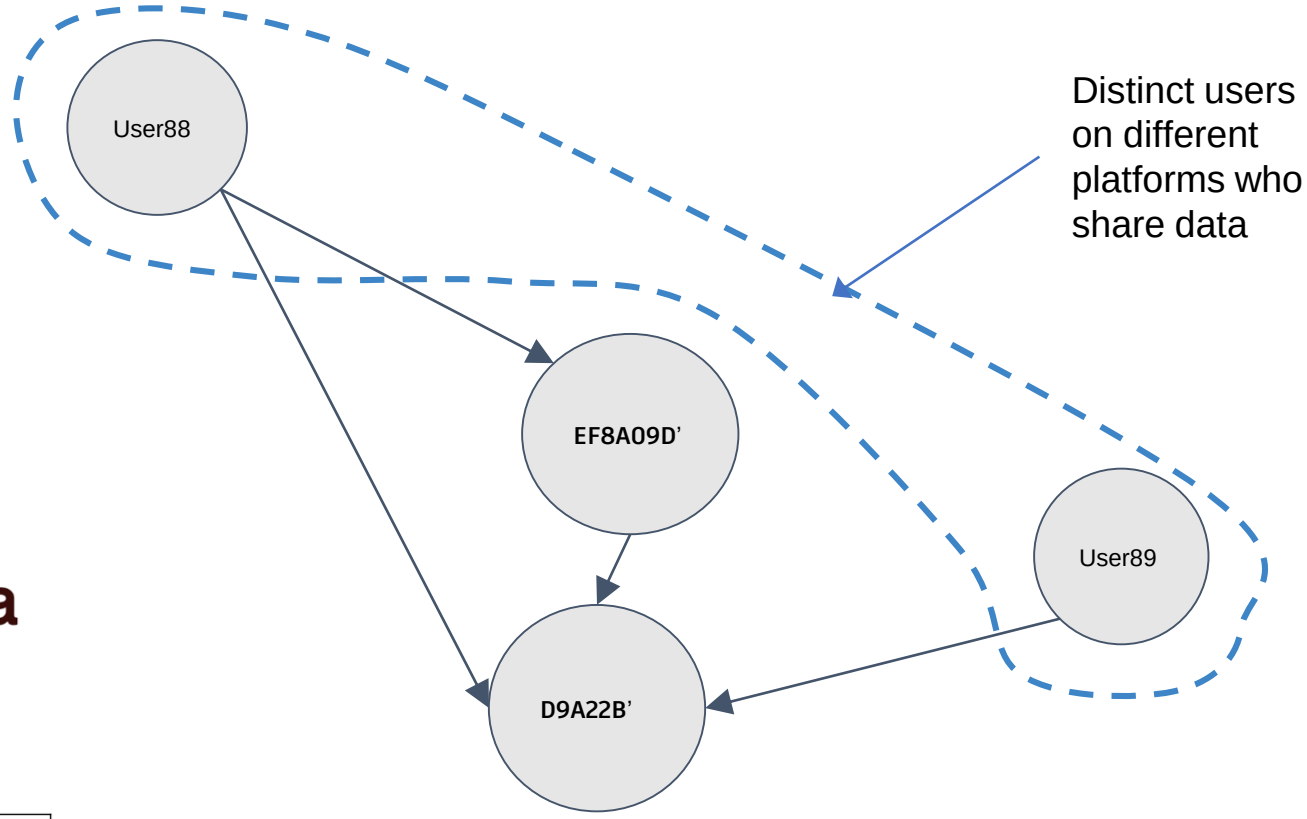
Log Item: (anonymized-user-id, platform, file-operation, **hash-before, hash-after**, time)

(u88, ‘desktop-win’, ‘save’, **‘EF8A09D’, ‘D9A22B’**, 9320031)
(u89, ‘mobile-ios’, ‘open’, **‘D9A22B’, ‘D9A22B’,** 10311299)

Connecting by hashes at scale

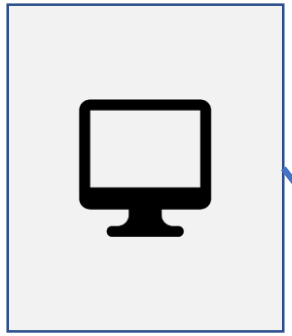


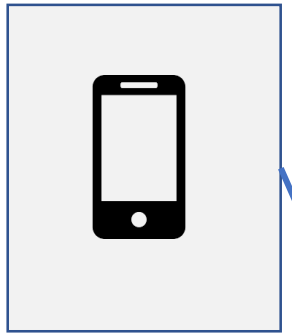


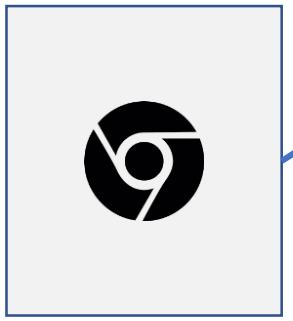


User88	Desktop-win	Save	**EF8A09D**	**D9A22B**
User89	Ios	Open	**D9A22B**	n/a

The Pipeline







Mixpanel

mixpanel

Data Lake

Data Lake

Spark

Query/
Extract to CSV

SPARK/
Qubole

Qubole

Scala

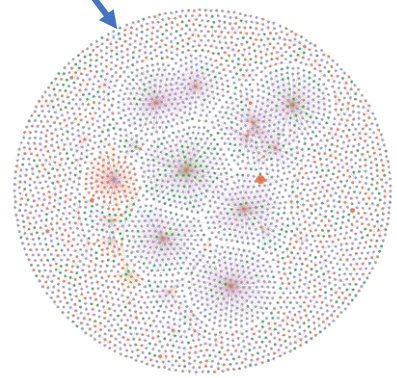
Spark

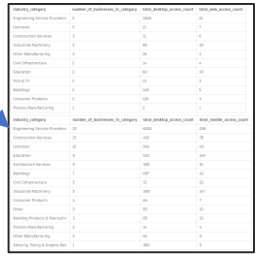
Bulk Import
to Neo4j

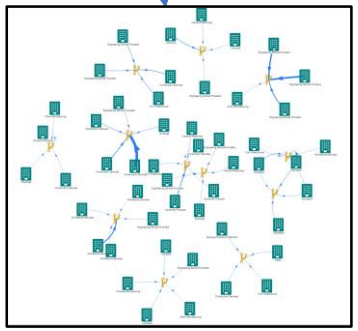
Query
/Output Query
Results to
GraphML/ CSV

pandas

Layout and
Visuzalization
Tool (Gephi)



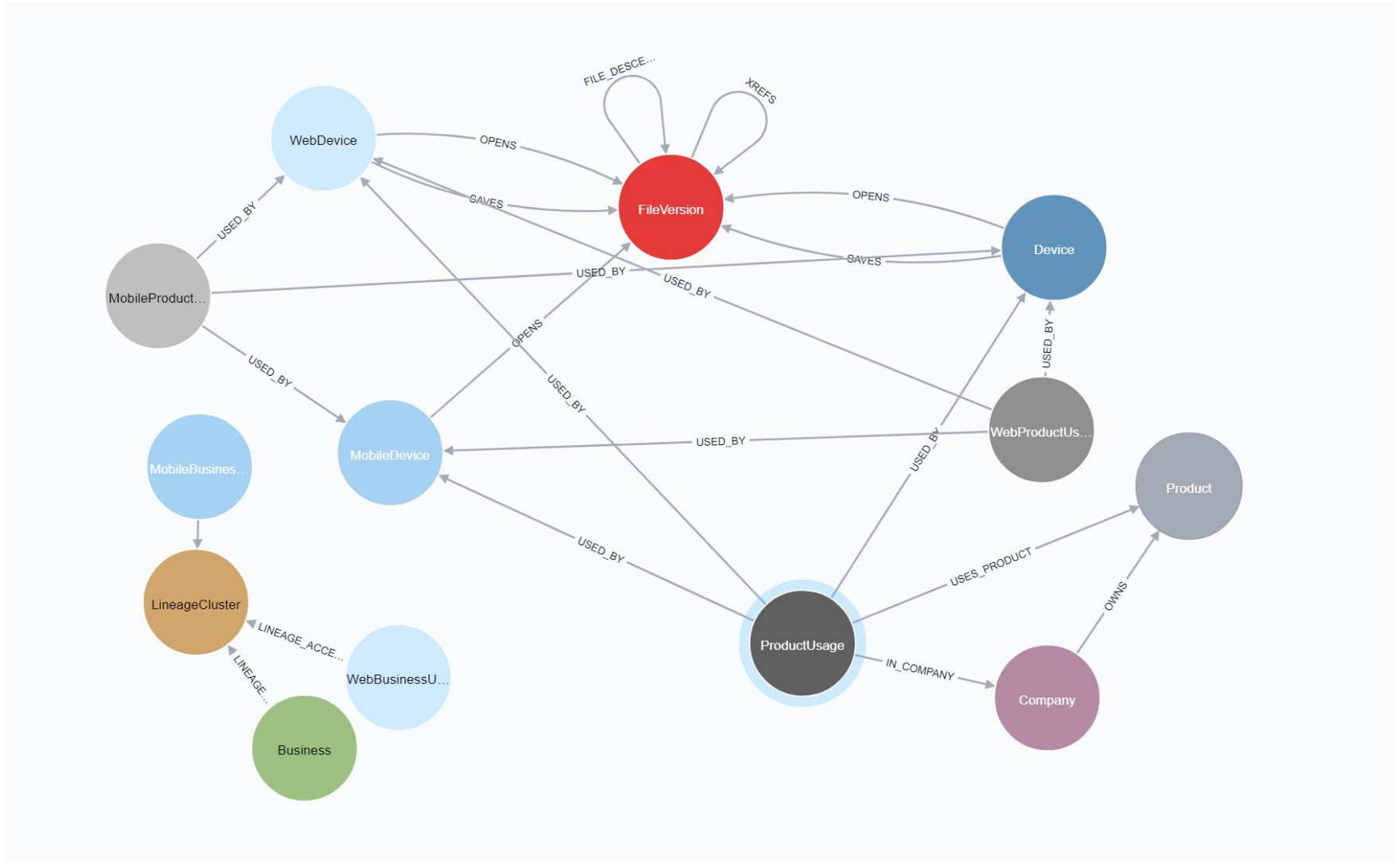




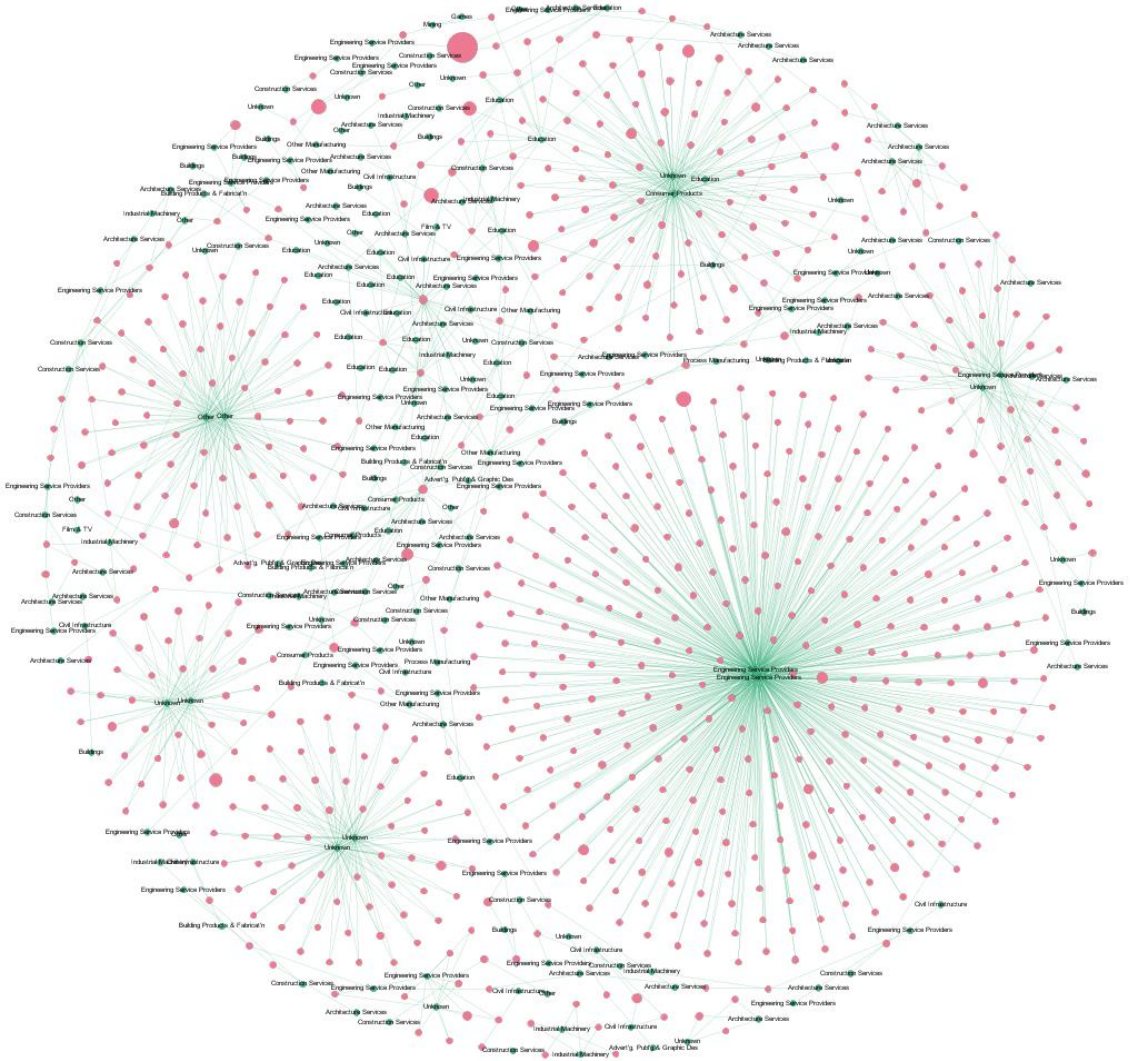
Elements of the pipeline

- Hive data processed in Spark 2.4 cluster
- Scala scripts to clean and export edgelists
- Scala scripts to import to Neo4j with loadCSV
- Postprocess graph to build lineages, interval information, access counts
- Data Exploration: Cypher queries to answer basic questions
- Data Exploration: Visualize graphs (Neovis, Gephi)
- Export queries (Cypher) for more post processing (Pandas)

Db Schema


FILE_DESCE...
XREFS
WebDevice
OPENS
FileVersion
OPENS
Device
SAVES
SAVES
USED_BY
USED_BY
MobileProduct...
USED_BY
USED_BY
USED_BY
OPENS
USED_BY
MobileBusines...
MobileDevice
USED_BY
WebProductUs...
USED_BY
Product
USED_BY
USES_PRODUCT
LineageCluster
LINEAGE_ACCE...
LINEAGE...
WebBusinessU...
ProductUsage
IN_COMPANY
OWNS
Company
Business

Industry types that interact



Identify lineages with algo.unionFind()

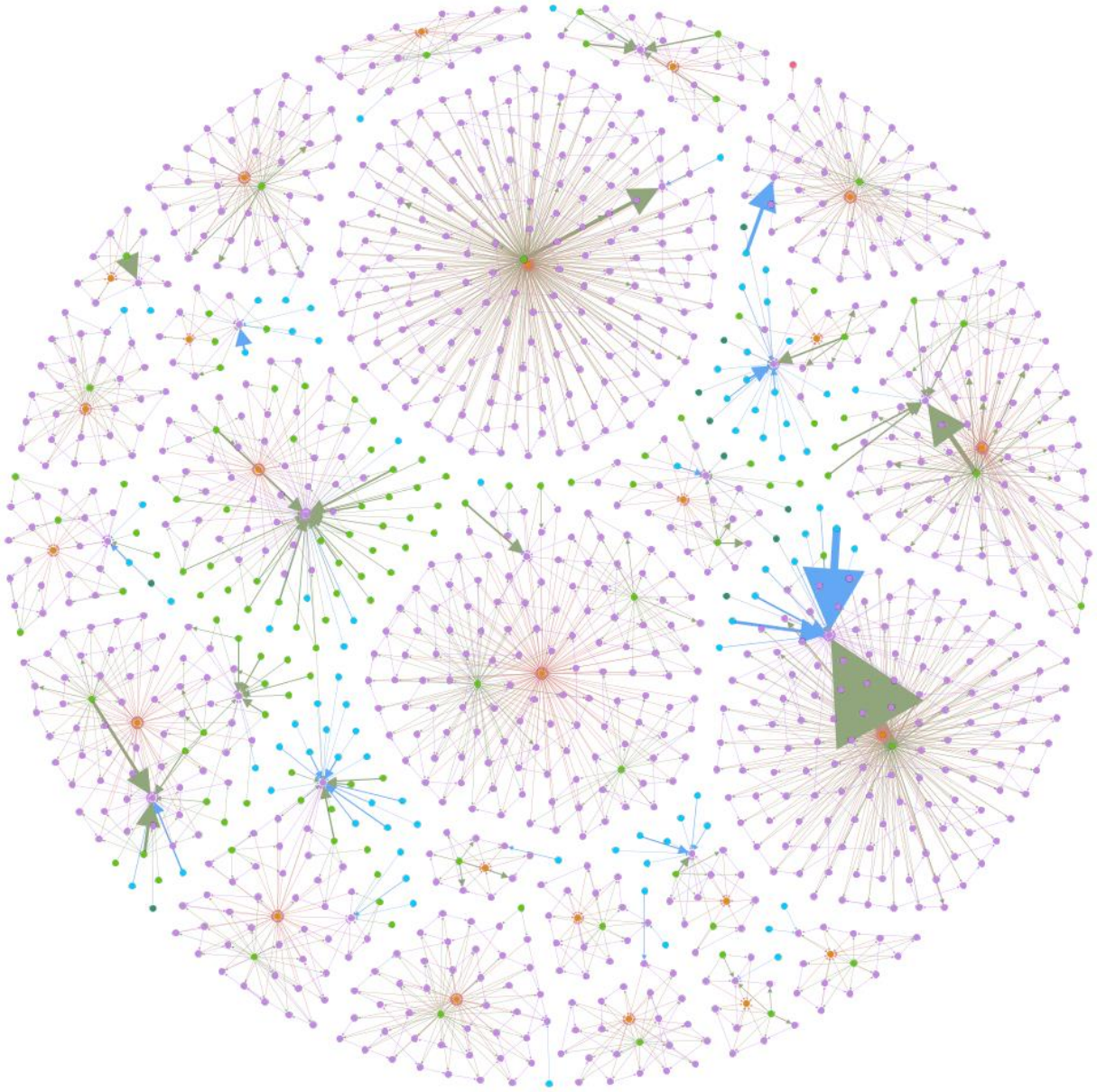
Web/Mobile/Desktop interaction

Purple: Fingerprint of specific file version

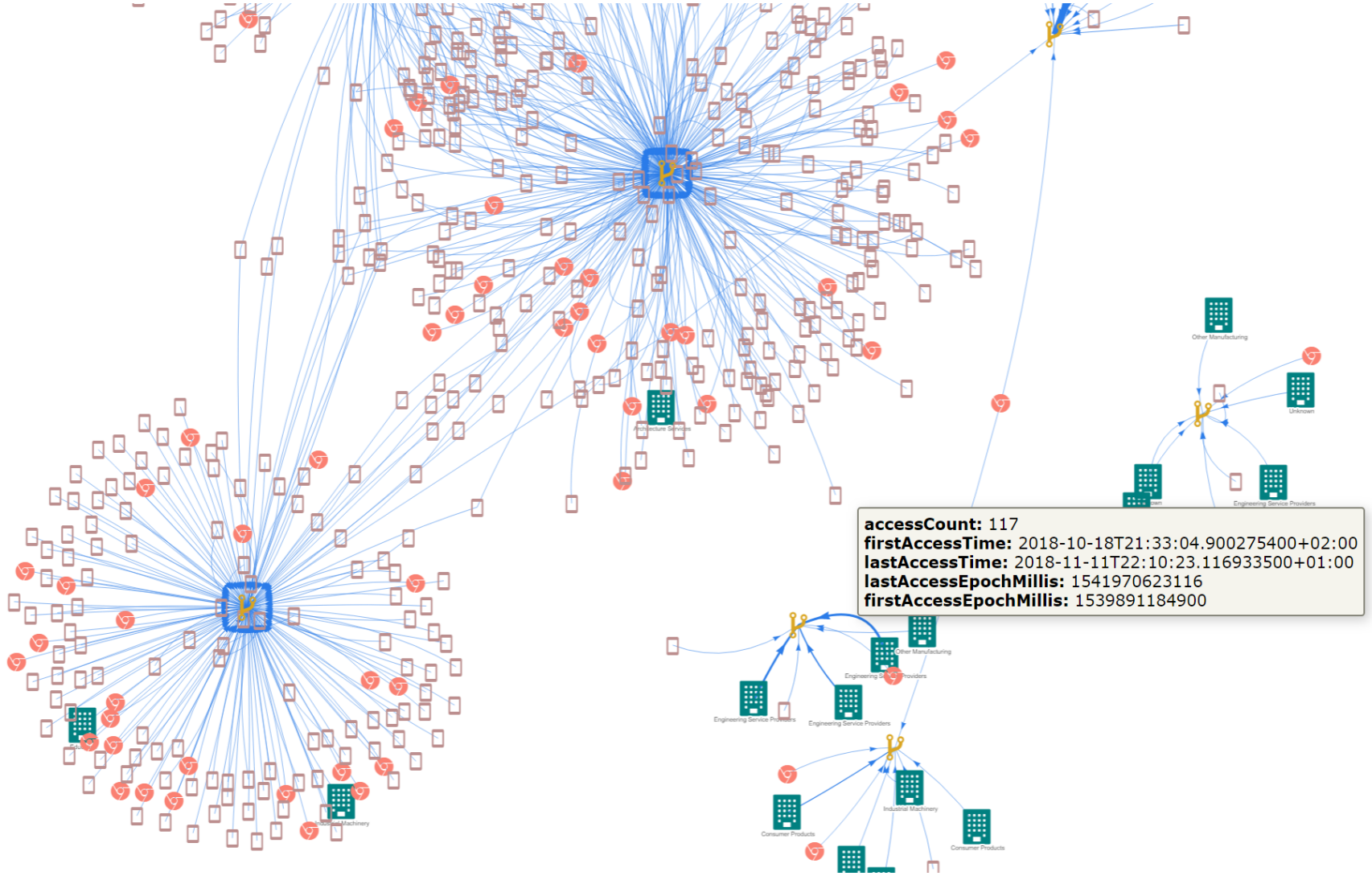
Chain of purple nodes: Lineages

Size of arrow: Number of accesses to specific fingerprint version

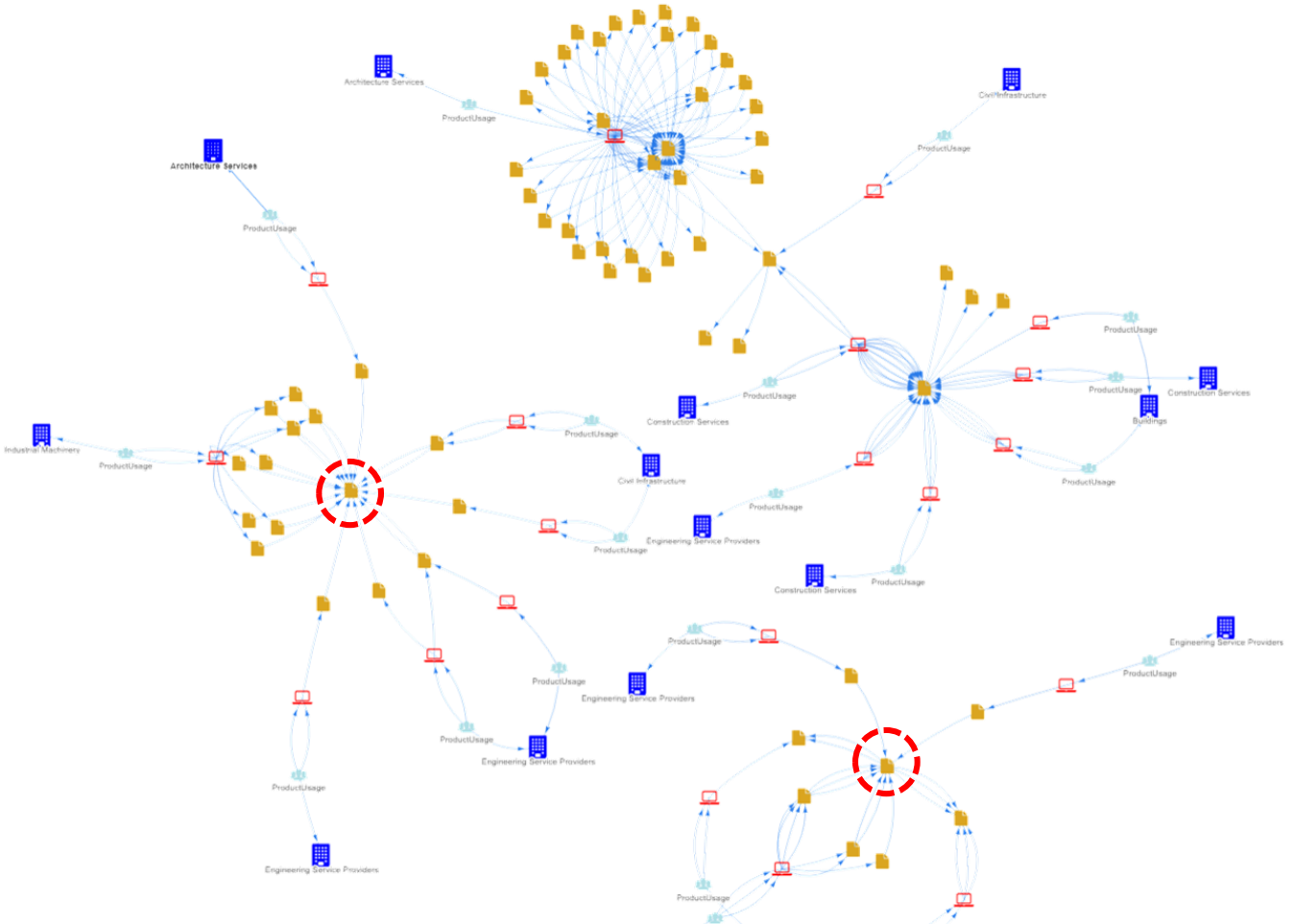
Green: Desktop; Red: Web; Blue: Mobile



Lineages and access patterns



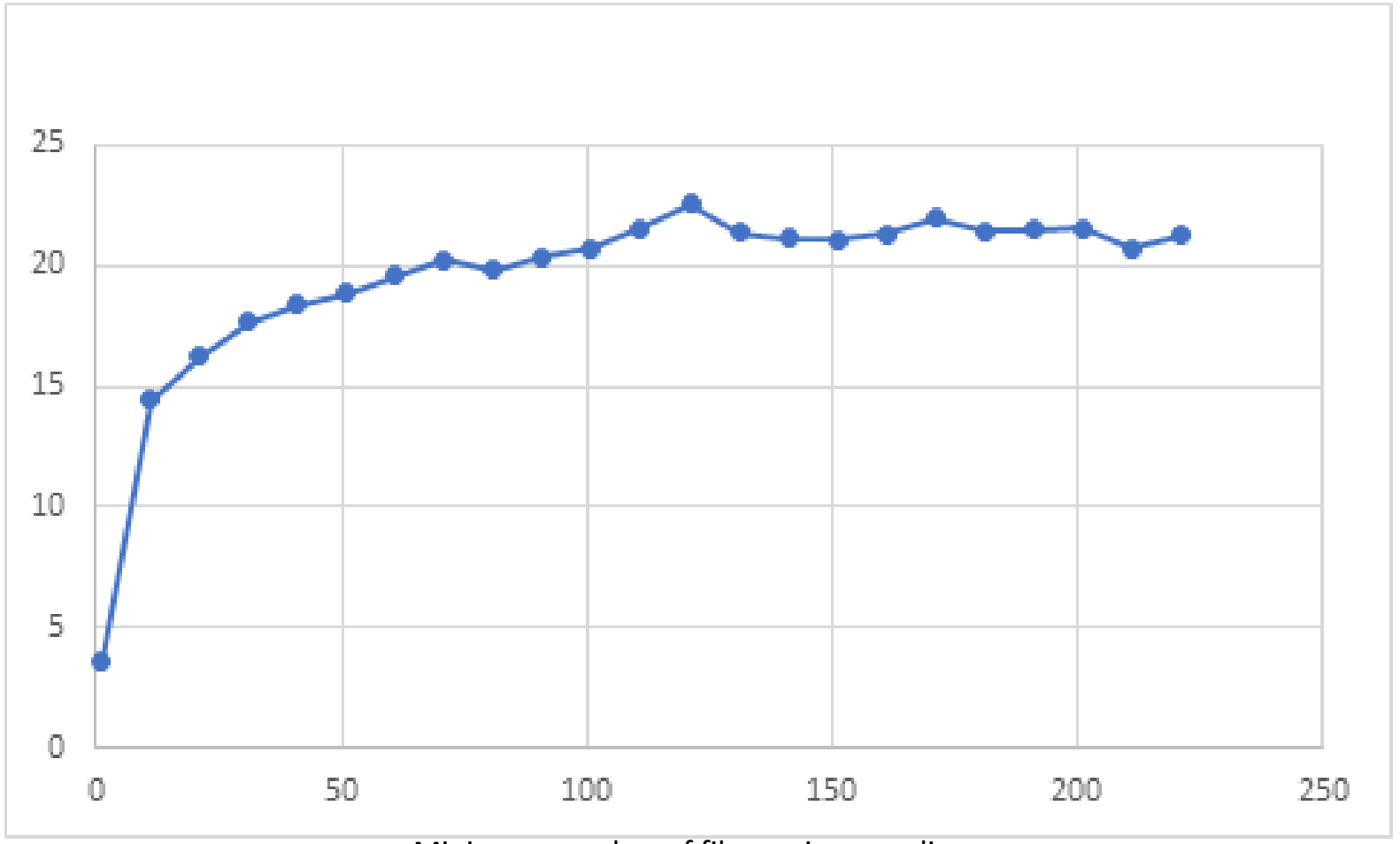
Connections by indirect reference to data




```
$ match (c:Company)-[]-(pu:ProductUsage)-[]-(d:Device)-[:OPENS]->(fv:FileVersion)-[:XREFS]-(xf:FileVersion) where
  not exists(xf.lineagePartition) with  distinct id(xf) as idx, c.hashedName as cname  with  idx, count(*) as ct
   where ct > 2 and ct < 10 match  (c:Company)-[u]-(pu:ProductUsage)-[r]-(d:Device)-[o:OPENS]->(fv:FileVersion)-
  [x:XREFS]-(xf:FileVersion) where id(xf) = idx return c, pu, d, fv, xf, u, r, o, x
```

What fraction of data is accessed by distinct devices?

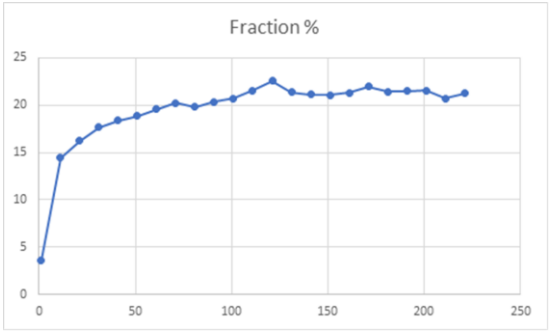
(%) lineages accessed by more than 1 device



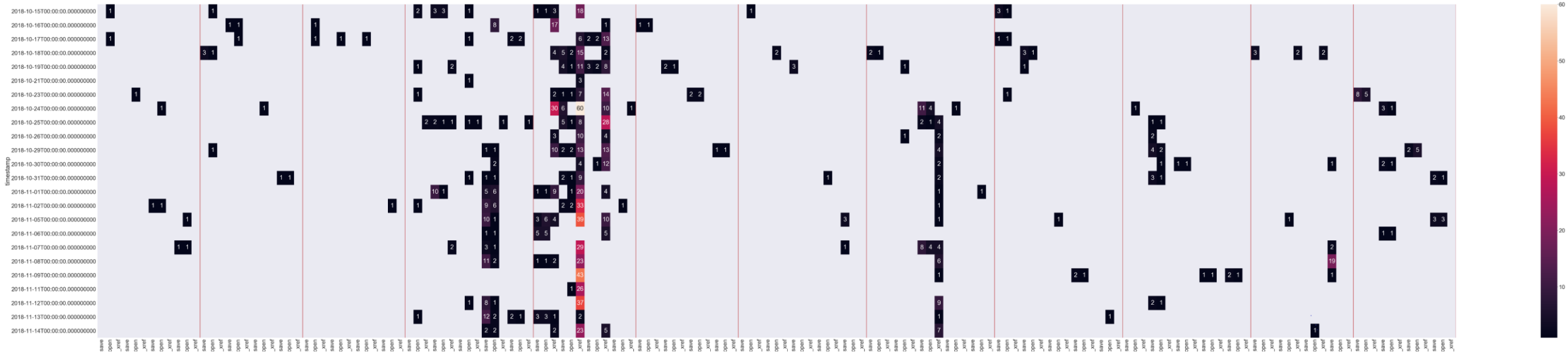
Minimum number of file versions per lineage

What fraction of data is accessed by distinct devices?

```
$ match (c)<-[:IN_COMPANY]-(:ProductUsage)-[]-(d:Device)-[r:OPENS]->
(fv:FileVersion) with fv, r, c.usageType as em, c.hashedName as name,
r.storageTechnology as st where (em = "Commercial") with distinct
fv.lineagePartition as lp  match (l:LineageCluster) where l.partitionId = lp
 with l match (d:Device)-[]->(ffv:FileVersion)  where ffv.lineagePartition =
l.partitionId  with count(distinct d) as dd, l, range(1, 221, 10) as rg unwind
rg as min_in_lineage with dd,l, min_in_lineage where l.numberOfVersions >=
min_in_lineage and dd > 1 return min_in_lineage, count(dd) order by
min_in_lineage
```



Time Series: access patterns



Takeaways

- Relatively easy to integrate into spark pipelines
- ‘Sweet spot’ size for data sets
- Flexibility of Graphs: Augmenting/Changing schema
- Rich set of queries possibly by Cypher and plugins algo and apoc
- Rich set of queries to provide input to advanced Analytics/ML

Questions

1. Efficient load of external file data into Neo4j can be achieved with which of the clauses?

(a) MERGE

(b) SET

(c) LOAD CSV

2. The value of a social network of n nodes using Reeds law can be thought to be

(a) O (n)

(b) O (n^2)

(c) O (2^n)

3. Name the procedure used in this talk to determine the connected components of the graph