Protecting privacy in practice

2017-06-13

Lars Albertsson
www.mapflat.com

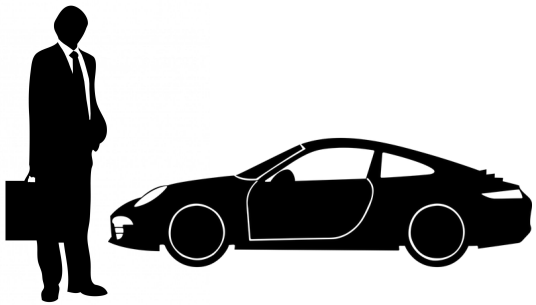
Who’s talking?

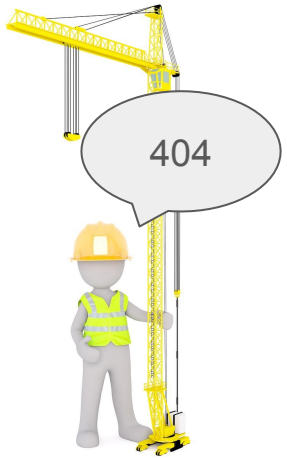
- KTH-PDC Center for High Performance Computing (MSc thesis)
- Swedish Institute of Computer Science (distributed system test+debug tools)
- Sun Microsystems (building very large machines)
- Google (Hangouts, productivity)
- Recorded Future (natural language processing startup)
- Cinnober Financial Tech. (trading systems)
- Spotify (data processing & modelling)
- Schibsted Media Group (data processing & modelling)
- Mapflat (independent data engineering consultant)

Privacy protection resources

All of this might go wrong. Large fine.

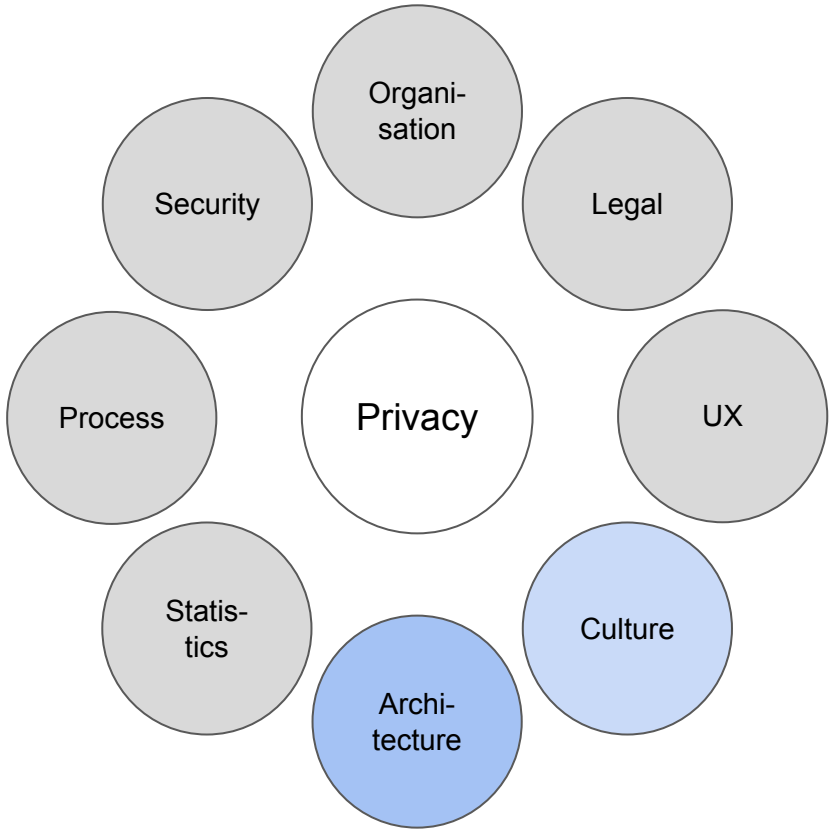
Pour your data into our product.





Privacy-driven design

- Technical scope
  - Toolbox
  - Not complete solutions
- Assuming that you solve:
  - Legal requirements
  - Security primitives
  - ...
- Not description of any company

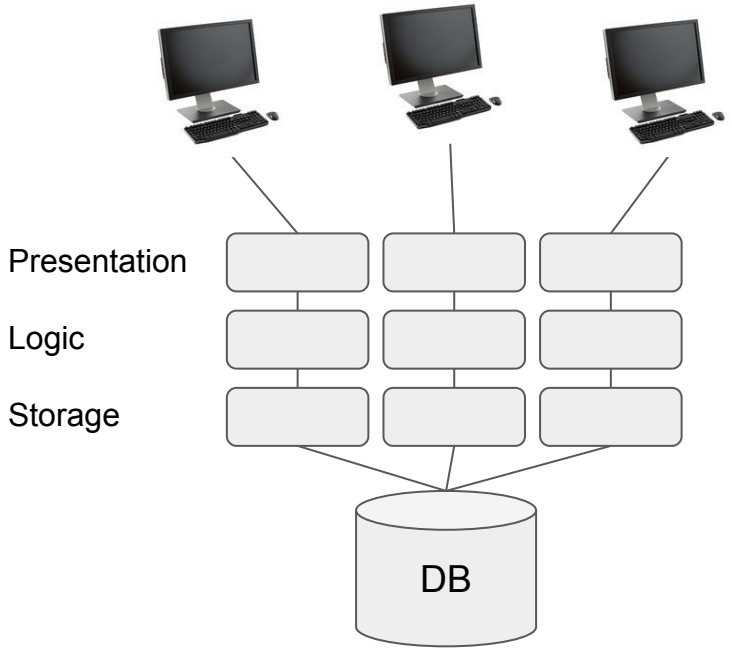


Requirements, engineer’s perspective

- Right to be forgotten
- Limited collection
- Limited retention
- Limited access
  - From employees
  - In case of security breach
- Right for explanations
- User data enumeration
- User data export

Ancient data-centric systems

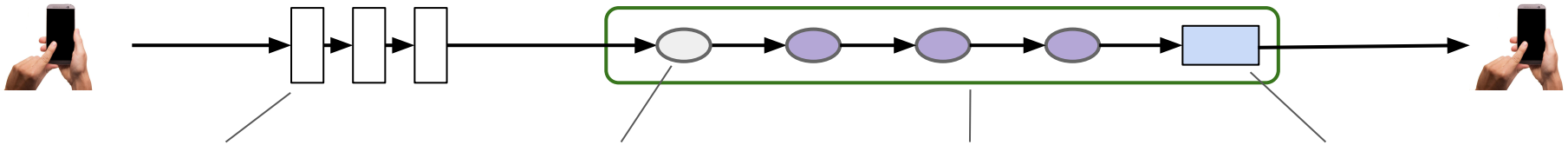
- The monolith
- All data in one place
- Analytics + online serving from single database
- Current state, mutable



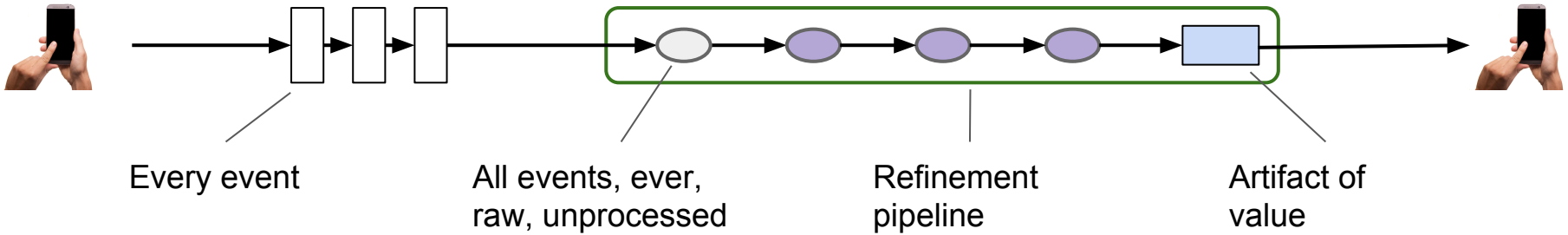
- Please delete me?
- What data have you got on me?

- Sure, no problem!

Event oriented systems



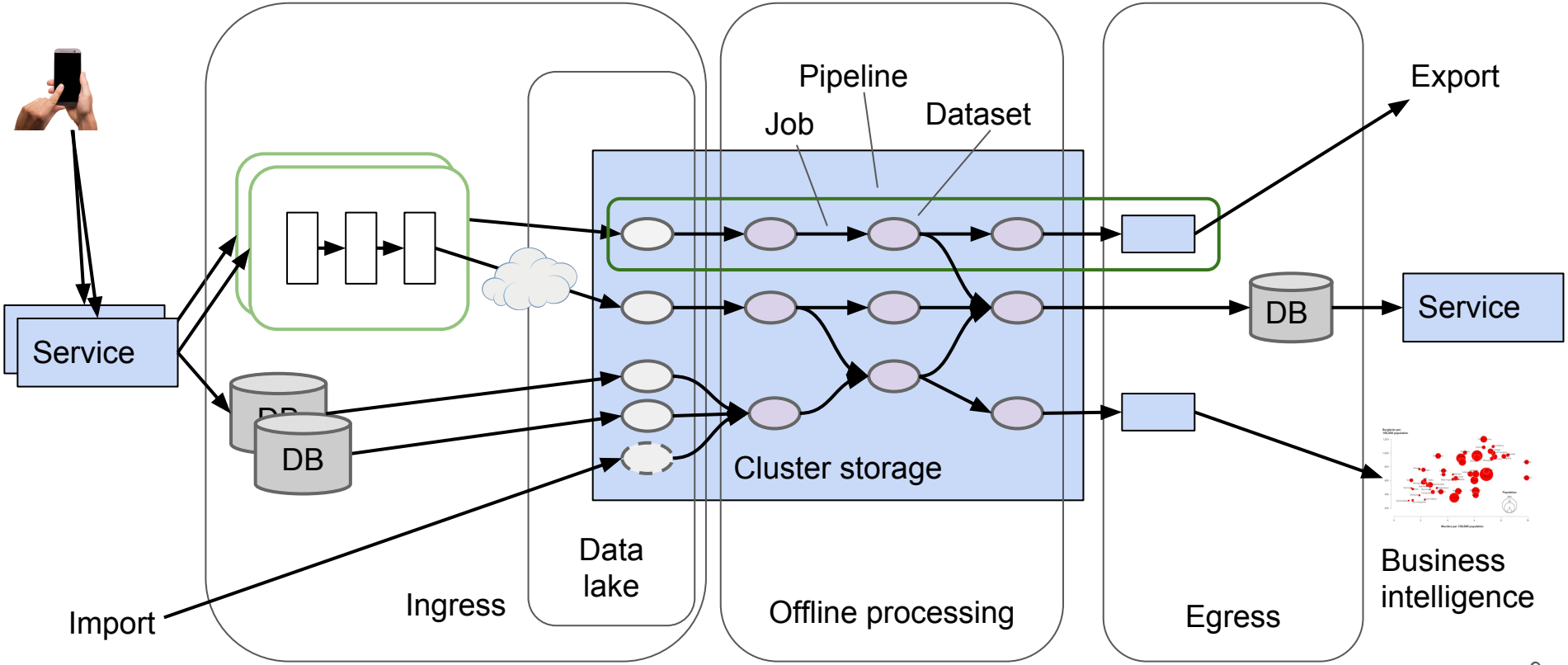
Event oriented systems



- Motivated by
  - New types of data-driven (AI) features
  - Quicker product iterations
    - Data-driven product feedback (A/B tests)
    - Fewer teams involved in changes
  - Robustness - scales to more complex business logic

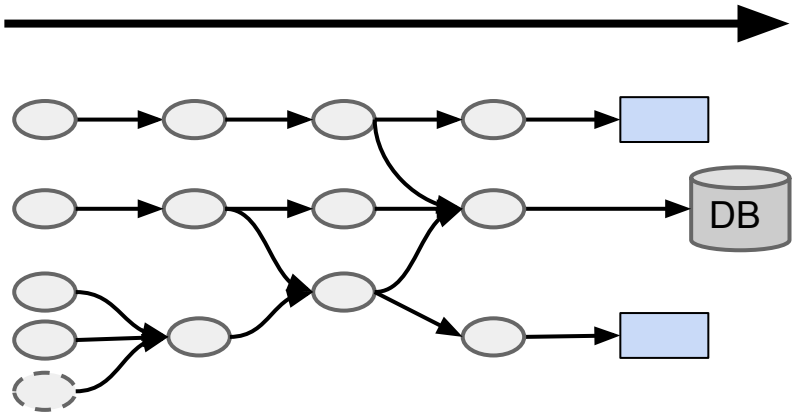
Enable disruption

Data processing at scale



Workflow manager

- Dataset “build tool”
- Run job instance when
  - input is available
  - output missing
  - resources are available
- Backfill for previous failures
- DSL describes DAG
  - Includes ingress & egress



Recommended: Luigi / Airflow

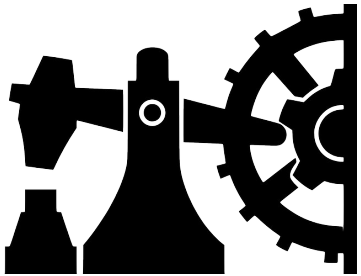
Factors of success

- Event-oriented - append only
- Immutability
- At-least-once semantics
- Reproducibility
  - Through 1000s of copies
- Redundancy

- Please delete me?
- What data have you got on me?

- Hold on a second...

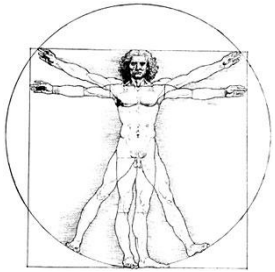
Solution space



Technical feasibility

Easy to do the right thing

Awareness culture



Personal information (PII) classification

- Red - sensitive data
  - Messages
  - GPS location
  - Views, preferences
- Yellow - personal data
  - IDs (user, device)
  - Name, email, address
  - IP address
- Green - insensitive data
  - Not related to persons
  - Aggregated numbers

- Grey zone
  - Birth date, zip code
  - Recommendation / ads models?

Nota bene: This is only an example classification

PII arithmetics

Red + green = red red + yellow = red yellow + green = yellow

Aggregate(red/yellow) = green ?

Green + green + green = yellow ?

Yellow + yellow + yellow = red ?

Machine_learning_model(yellow) = yellow ?

Overfitting => persons could be identified

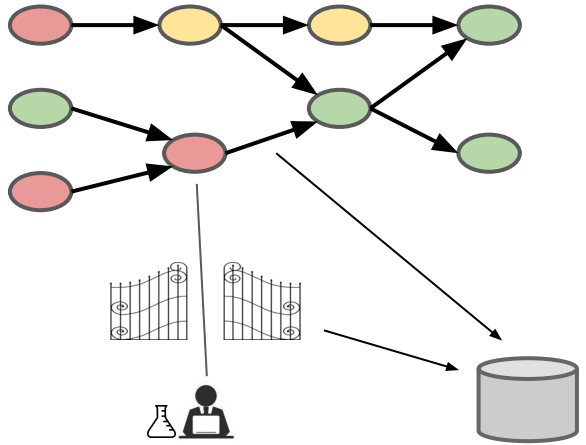
Make privacy visible at ground level

- In dataset names
  - hdfs://red/crm/received_messages/year=2017/month=6/day=13
  - s3://yellow/webshop/pageviews/year=2017/month=6/day=13
- In field names
  - response.y_text = “Dear ” + user.y_name + “, thanks for contacting us …”
- In credential / service / table / ... names

- Spreads awareness
- Catch mistakes in code review
- Enables custom tooling for violation warnings

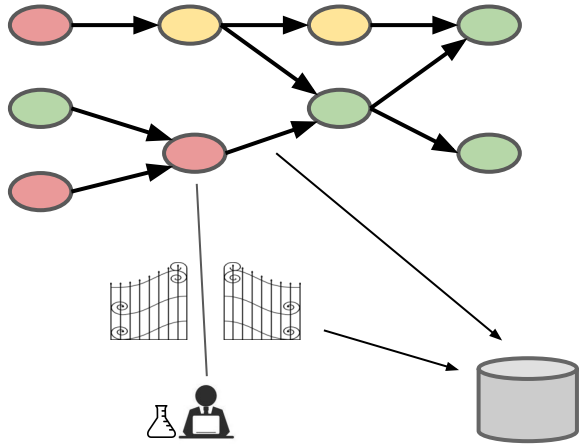
Eye of the needle tool

- Provide data access through gateway tool
  - Thin wrapper around Spark/Hadoop/S3/...
  - Hard-wired configuration
- Governance
  - Access audit, verification
  - Policing/retention for experiment data



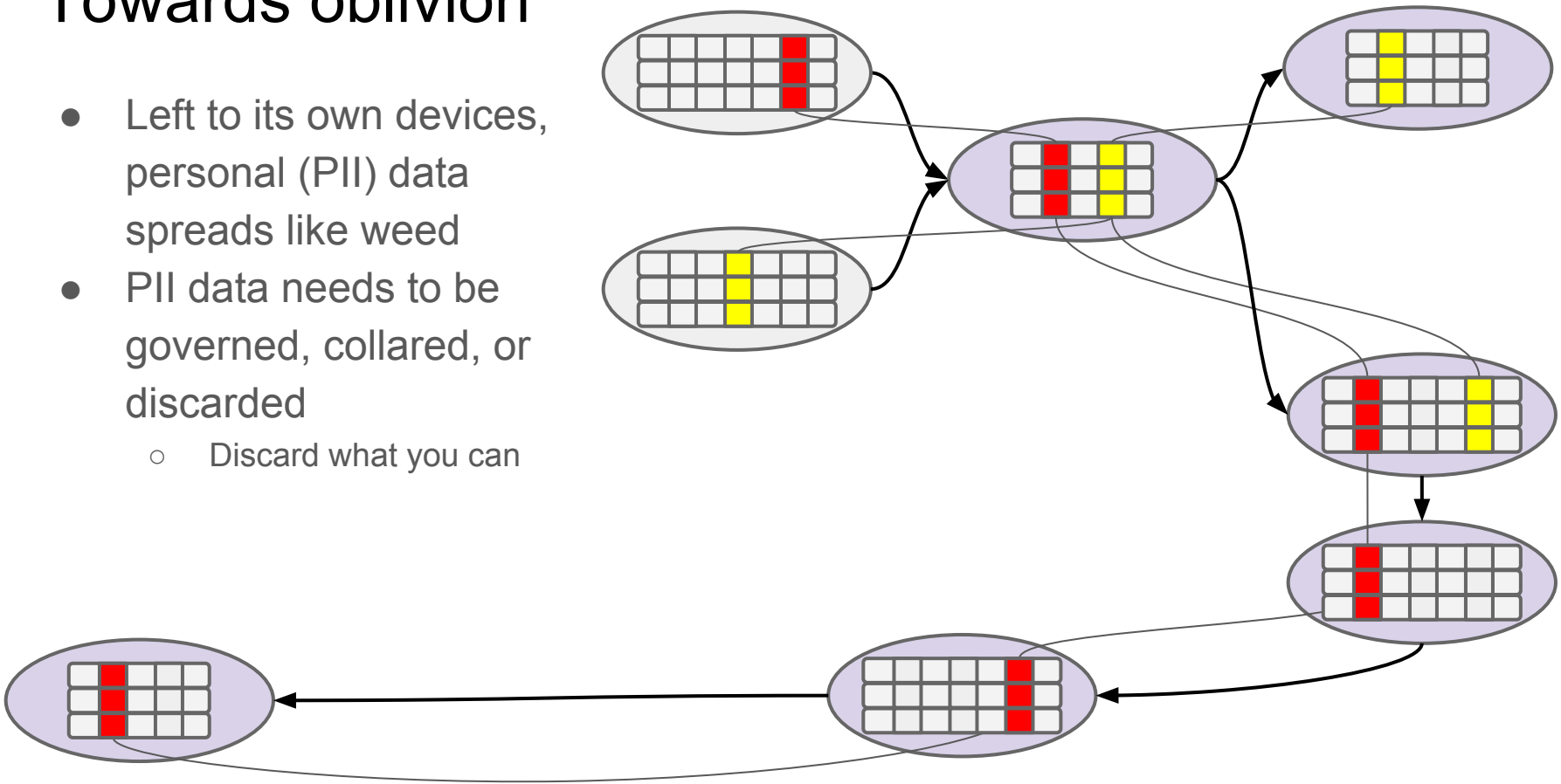
Eye of the needle tool

- Easy to do the right thing
  - Right resource choice, e.g. “allocate temporary cluster/storage”
  - Enforce practices, e.g. run jobs from central repository code
  - No command for data download
- Enabling for data scientists
  - Empowered without operations
  - Directory of resources



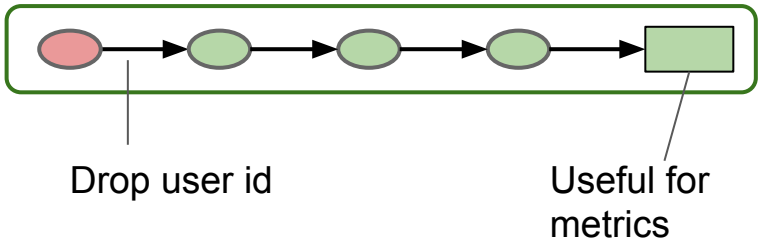
Towards oblivion

- Left to its own devices, personal (PII) data spreads like weed
- PII data needs to be governed, collared, or discarded
  - Discard what you can

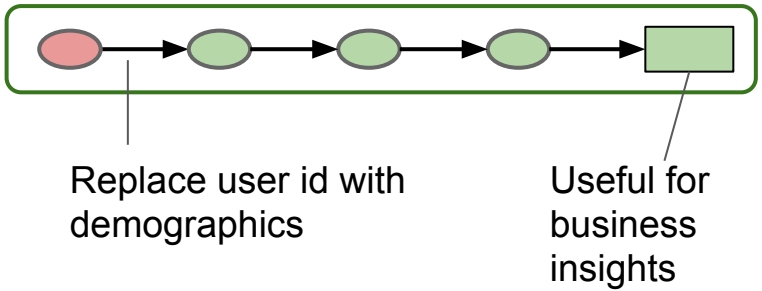


Discard: Anonymisation

- Discard all PII
  - User id in example
- No link between records or datasets

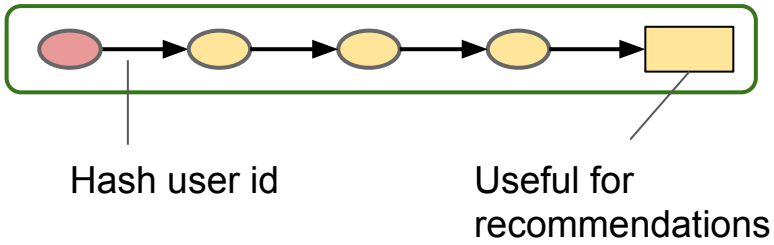


- Replace with non-PII
  - E.g. age, gender, country
- Still no link
  - Beware: rare combination => not anonymised

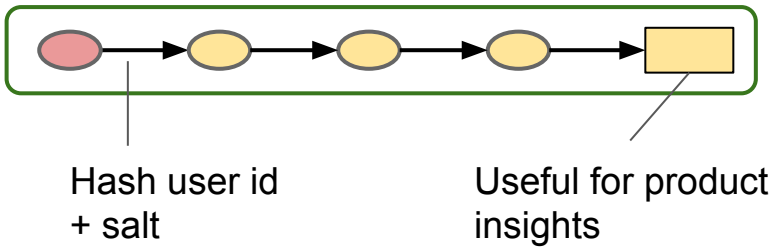


Partial discard: Pseudonymisation

- Hash PII
- Records are linked
  - Across datasets
  - Still PII, GDPR applies
  - Persons can be identified (with additional data)
  - Hash recoverable from PII

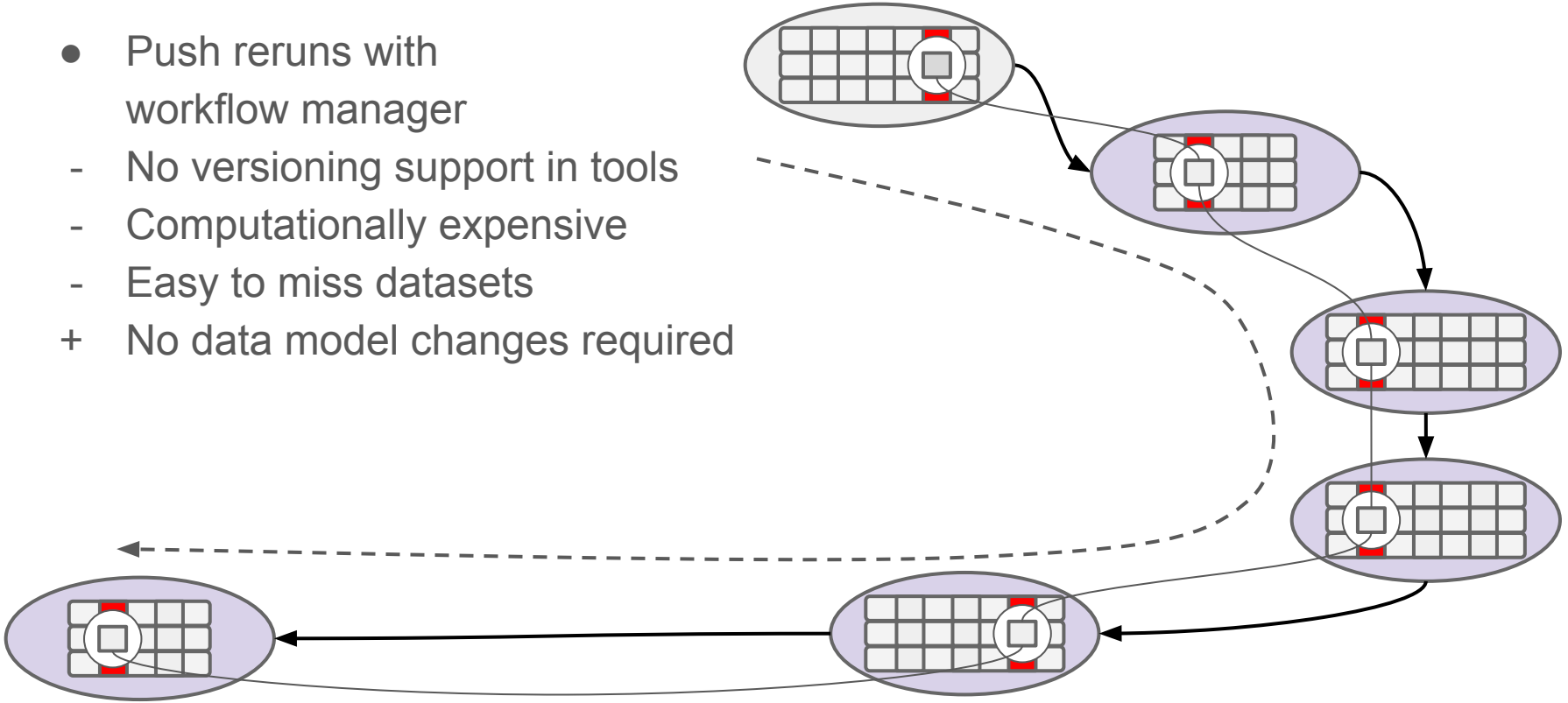


- Hash PII + salt
  - Hash not recoverable
- Records are still linked
  - Across datasets if salt is constant



Governance: Recomputation

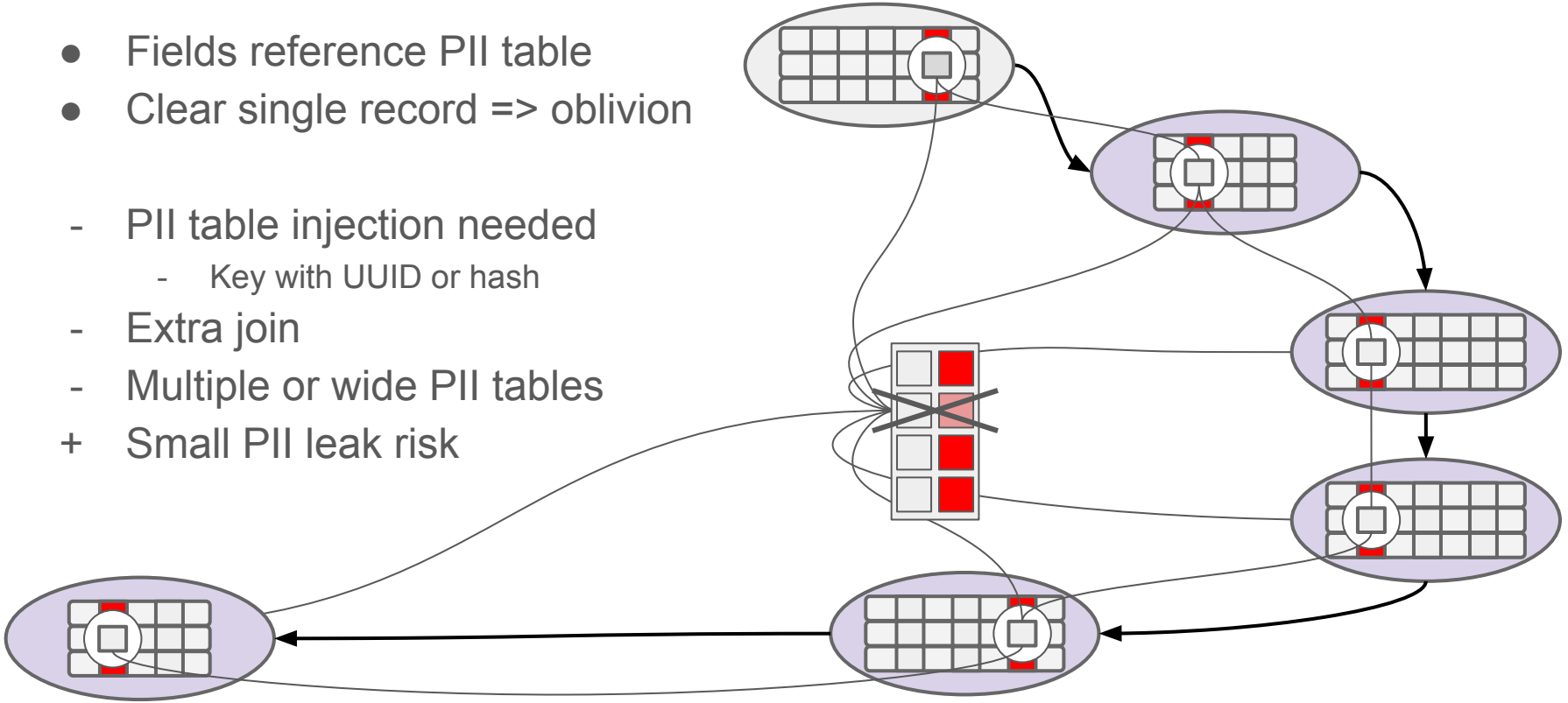
- Push reruns with workflow manager
- \- No versioning support in tools
- \- Computationally expensive
- \- Easy to miss datasets
- \+ No data model changes required



Ejected record pattern

- Fields reference PII table
- Clear single record => oblivion

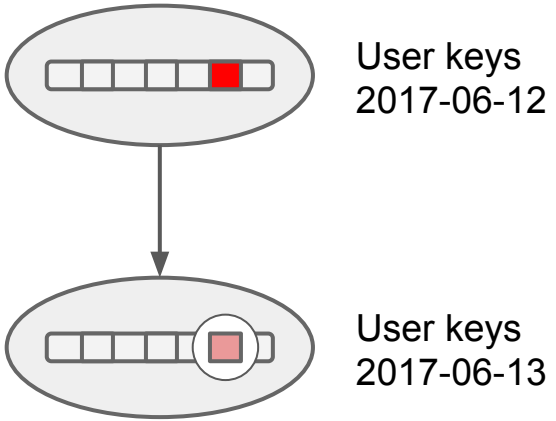
- PII table injection needed
  - Key with UUID or hash
- Extra join
- Multiple or wide PII tables
+ Small PII leak risk



Record removal in pipelines

- Datasets are immutable
- Version n+1 of raw dataset lacks record
- Short retention of old versions
- *Always depend on latest version*

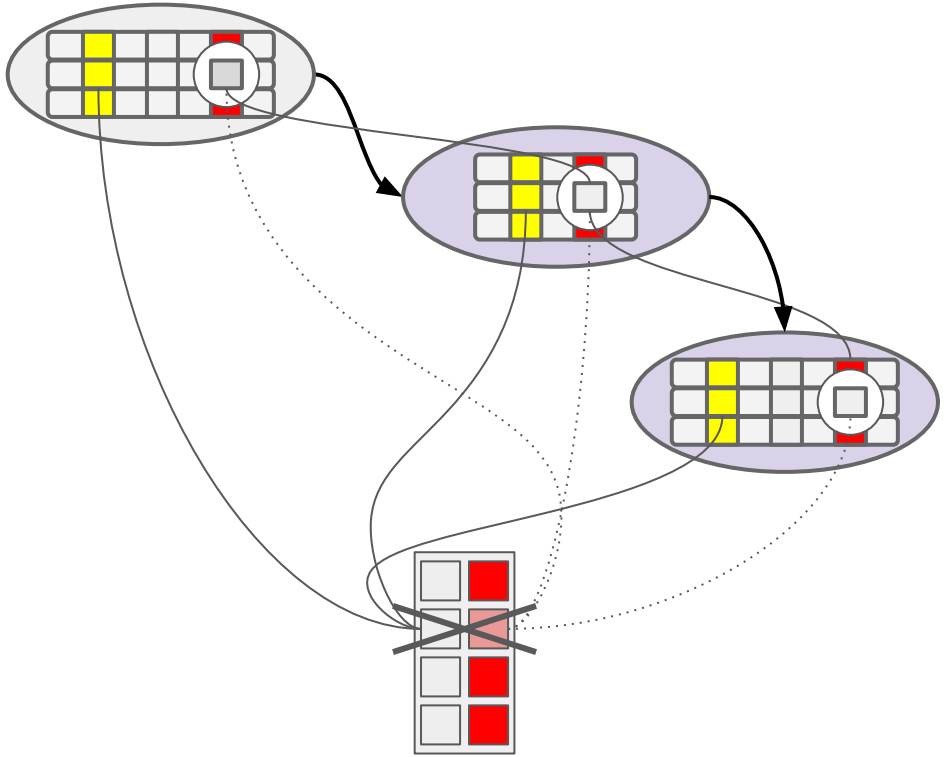
```
class Purchases(Task):
  date = DateParameter()

  def requires(self):
    return [Users(self.date),
            Orders(self.date),
            UserKeys.latest()]
```



Lost key pattern

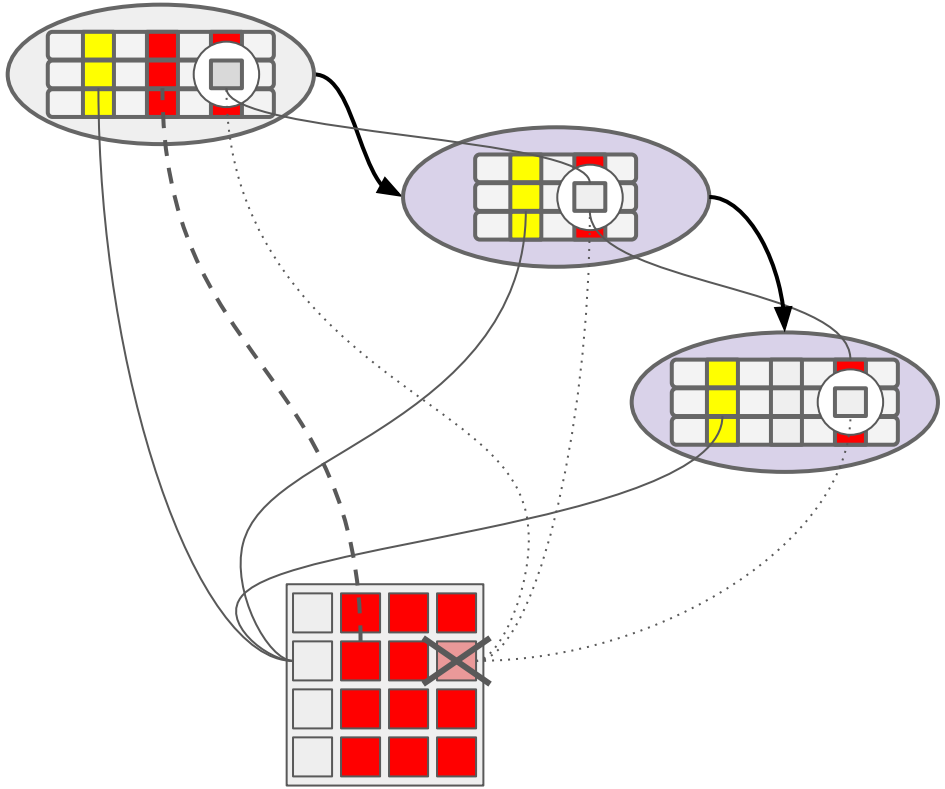
- PII fields encrypted
- Per-user decryption key table
- Clear single user key => oblivion

- Extra join + decrypt
- Decryption (user) id needed
+ Multi-field oblivion
+ Small PII leak risk



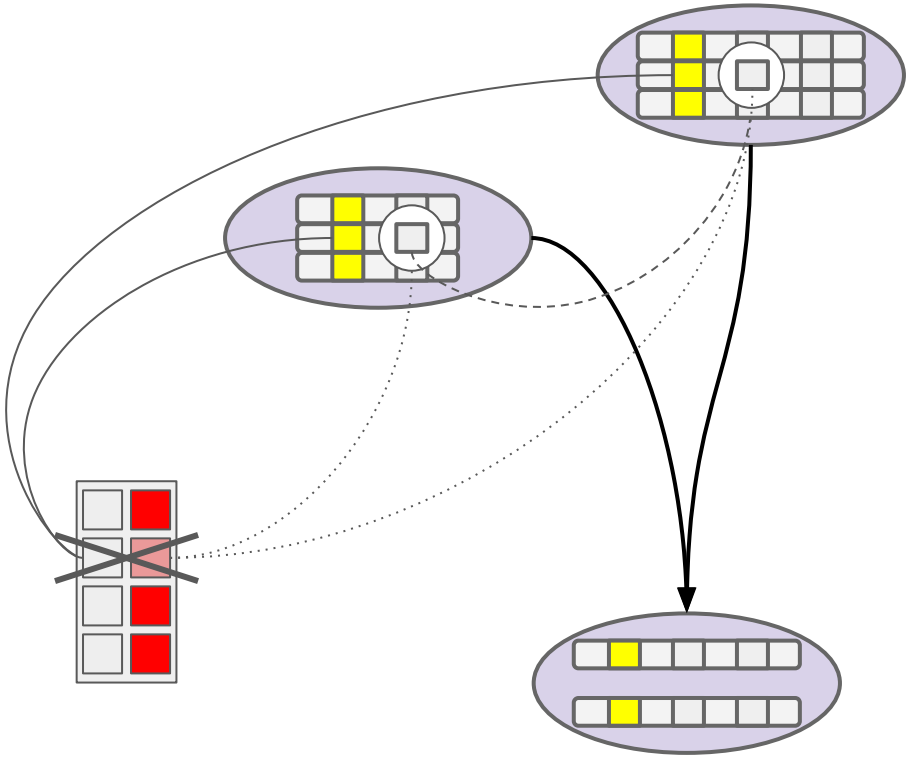
Lost key partial oblivion

- Different fields encrypted with different keys
- Partial user oblivion
  - E.g. forget my GPS coordinates



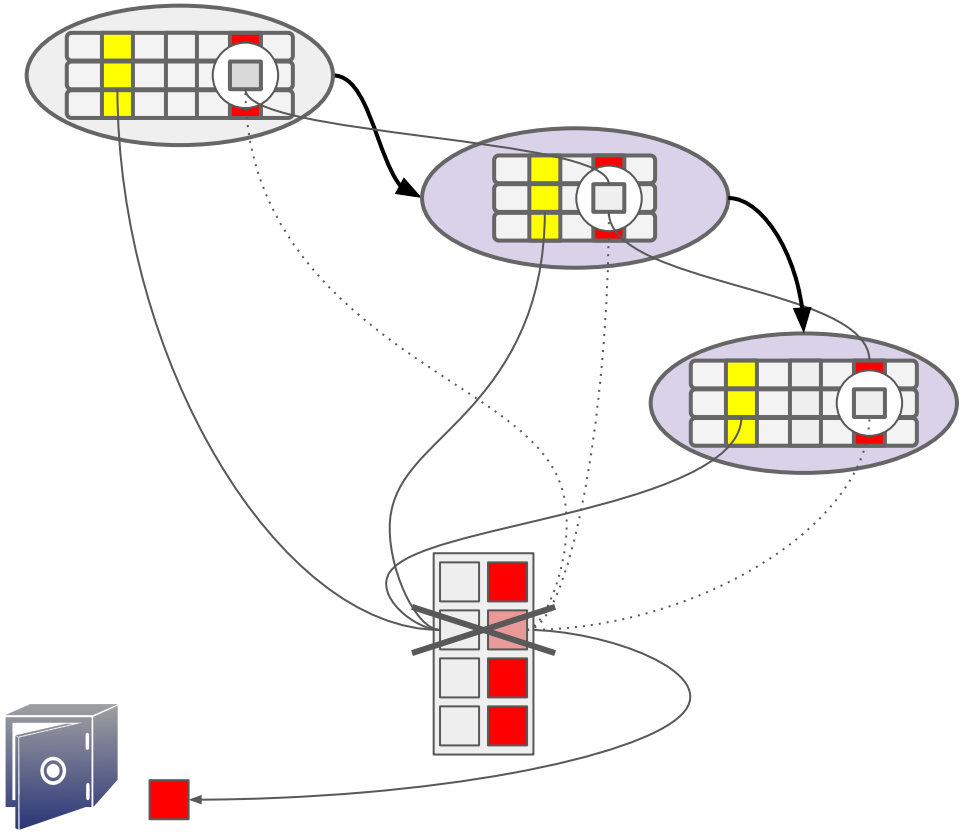
Lost link key

- Encrypt key fields that link datasets
- Ability to join is lost
- No data loss
  - Salt => anonymous data
  - No salt => pseudonymous data



Reversible oblivion

- Lost key pattern
- Give ejected record key to third party
  - User
  - Trusted organisation
- Destroy company copies

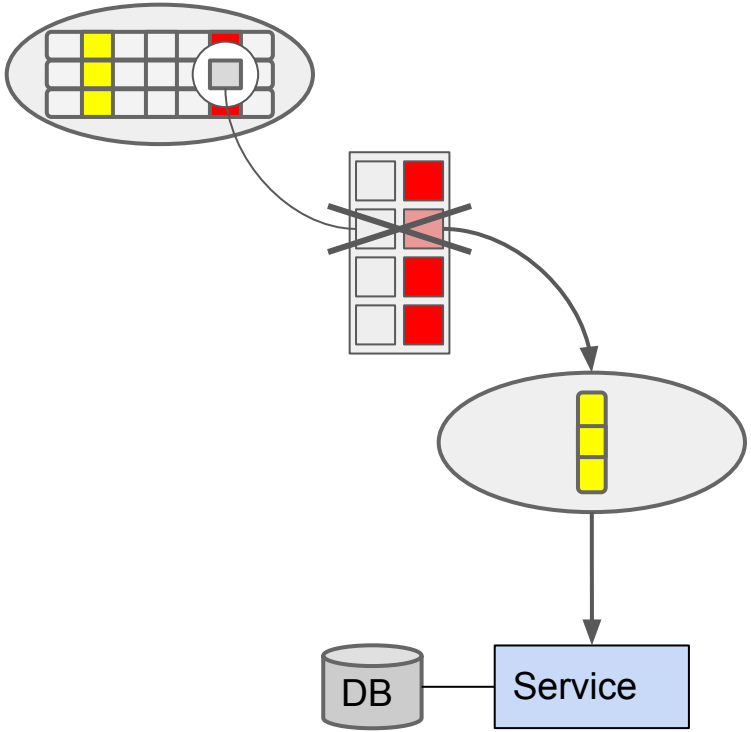


Data model deadly sins

- Using PII data as key
  - Username, email
- Publishing entity ids containing PII data
  - E.g. user shared resources (favourites, compilations) including username
- Publishing pseudonymised datasets
  - They can be de-pseudonymised with external data
  - E.g. AOL, Netflix, ...

Tombstone line

- Produce dataset/stream of forgotten users
- Egress components, e.g. online service databases, may need push for removal.
  - Higher PII leak risk



The art of deletion

- Example: Cassandra
- Deletions == tombstones
- Data remains
  - Until compaction
  - In disconnected nodes

Component-specific expertise necessary

Deletion layers

- Every component adds deletion burden
  - Minimise number of components
  - Ephemeral >> dedicated. Recycle machines.
- Every storage layer adds deletion burden
  - Minimise number of storage layers
  - Cloud storage requires documented erasure semantics + agreements.
- Invent simple strategies
  - Example: Cycle Cassandra machines regularly, erase block devices.

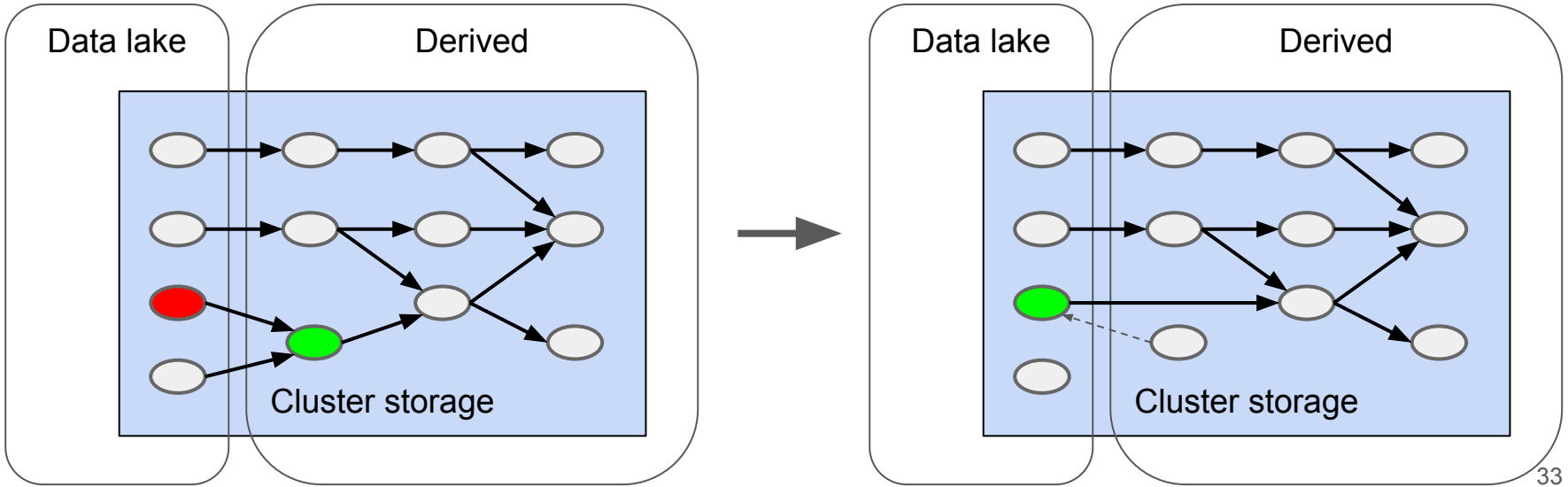
Increasing cost of heterogeneity

Retention limitation

- Best solved in workflow manager
  - Connect creation and destruction
- Short default retention, whitelist exceptions
- In conflict with technical ideal of immutable raw data

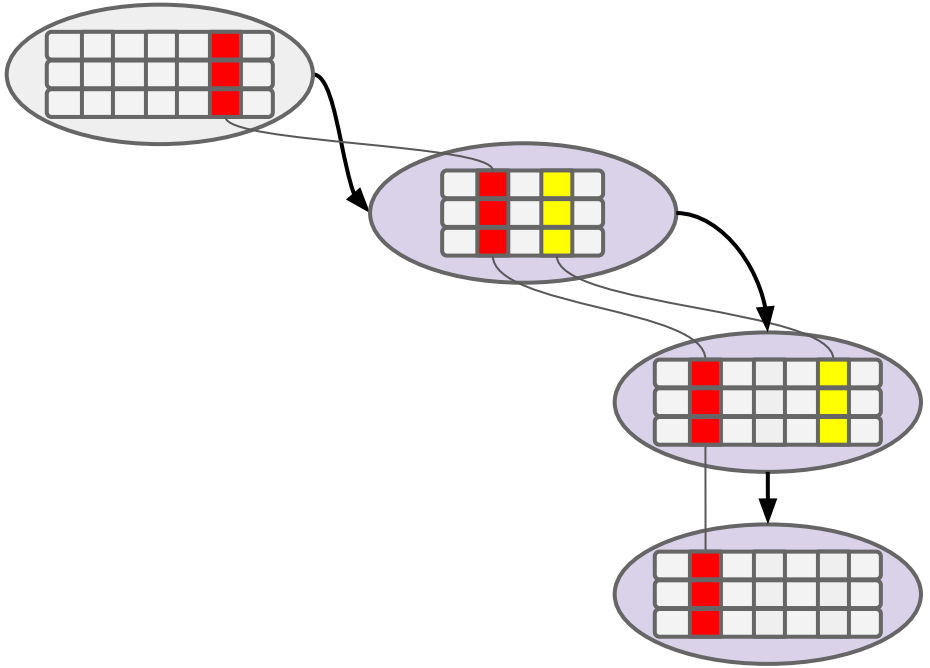
Lake promotion

- Remove expire raw dataset, promote derived datasets to lake
- First derived dataset = washed(raw)?
- Workflow DAG still works



Lineage

- Tooling for tracking data flow
- Dataset granularity
  - Workflow manager?
- Field granularity
  - Framework instrumentation?
- Multiple use cases
  - (Discovering data)
  - (Pipeline change management)
  - Detecting dead end data flows
  - Right to export data
  - Explanation of model decisions



Solicitation: PII & lineage type systems

- Idea: decorate (scala) types
  - PII classification (red/yellow/green)
  - Lineage (e.g. processing class id + commit id + dataset revision)
- Assistance with PII arithmetics
  - PII[Red, String] + PII[Green, String] => PII[Red, String]
  - PII[Red, Int] + PII[Red, Int] => PII[Green, Int]
- Detect unused PII fields
- Assist with recomputation
  - For PII cleaning
  - Bug fixes

Resources

- http://www.slideshare.net/lallea/data-pipelines-from-zero-to-solid
- http://www.mapflat.com/lands/resources/reading-list
- https://ico.org.uk/
- EU Article 29 Working Party

Credits

- Alexander Kjeldaas, independent
- Lena Sundin, Spotify
- Oscar Söderlund, Spotify
- Oskar Löthberg, Spotify
- Sofia Edvardsen,
  Sharp Cookie Advisors
- Øyvind Løkling,
  Schibsted Media Group