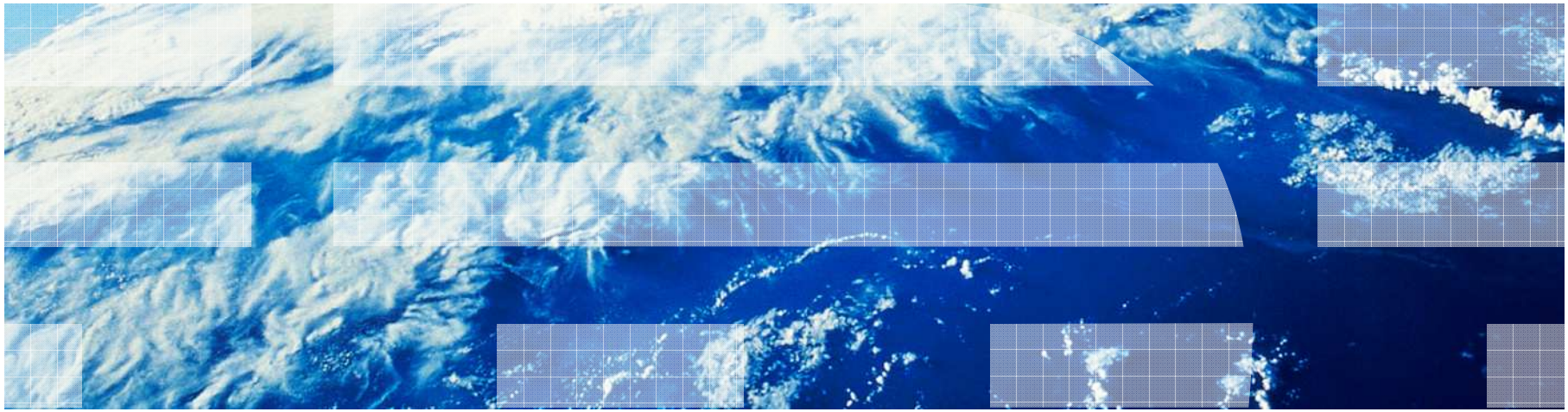
Gregory Neven, IBM Research – Zurich

joint work with Laurent Bussard (EMIC) and Jan Schallaböck (ULD)

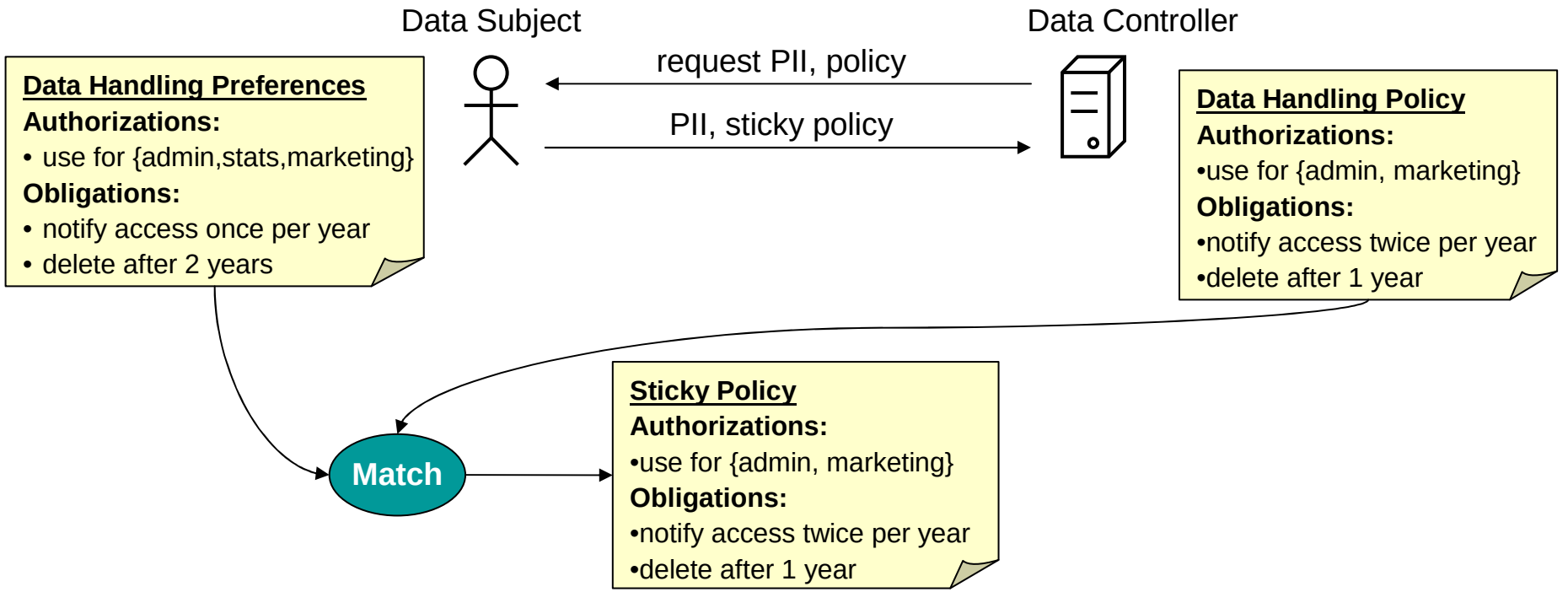
W3C Workshop on Privacy and Data Usage Control,

October 4-5, 2010, Cambridge, MA, USA

Dependencies between Authorizations and Obligations



Two-sided data handling in PPL



Authorization = right to perform an action
violated when non-authorized action is performed

Obligation = duty to perform an action
violated when obliged action is *not* performed

Types of dependencies

- Obligation ≈ authorization for complementary action, e.g.,
  - Obligation to delete within 1 year
    - ≈ authorization to store up to 1 year
  - Authorization to use for {admin, marketing}
    - ≈ obligation not to use for other purposes than {admin, marketing}
- Execution of authorization can trigger obligation, e.g.

  e.g., each access (= authorization) must be logged (= obligation)
- Execution of obligation requires authorization to do so

  e.g., obligation to notify data subject

this presentation

The problem

Data Controllers “overdoing” their obligations so that

- becomes a nuisance

  e.g., notify at least once per year vs. notify twice per day

- degrades quality of service

  e.g., delete data after 10 years vs. delete data immediately

How to protect Data Subjects from overzealous Data Controllers?

Legal perspective

- Example: credit scoring agencies
- Processing credit data prohibited by European data protection law (95/46/EC) unless legal basis
  e.g., contract, explicit consent
- Makes sense for credit agency to notify data subjects about stored information:
  - privacy of data subject
  - quality of information
- Too frequent notifications
  - are experienced as spam
  - partially defeat goal of quality control
- Obligation to notify may be embedded in contract (privacy as a competitive advantage?) or, in near future, in the law protection against overzealous Data Controllers?

Policy matching in PPL

Automated matching of data handling preferences/policies via **"more or equally permissive than"** operator ($\trianglerighteq$) defined on

- authorizations, e.g.

  use for purposes $P \trianglerighteq$ use for purposes $P' \Leftrightarrow P \supseteq P'$

- obligations, e.g.

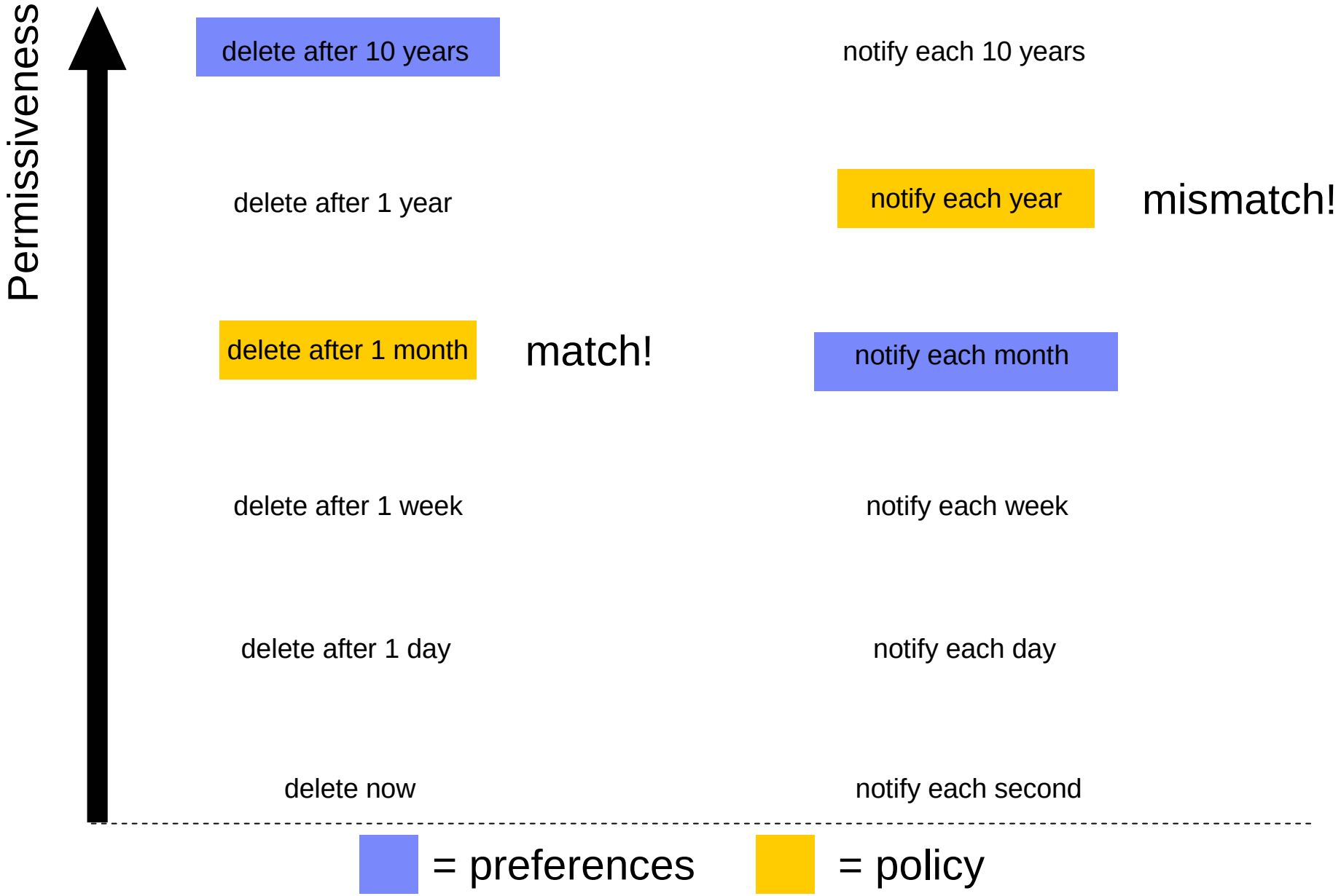
  send usage summary with freqency $f \trianglerighteq$ with frequency $f' \Leftrightarrow f \leq f'$

  delete within time $t \trianglerighteq$ delete within time $t' \Leftrightarrow t \geq t'$

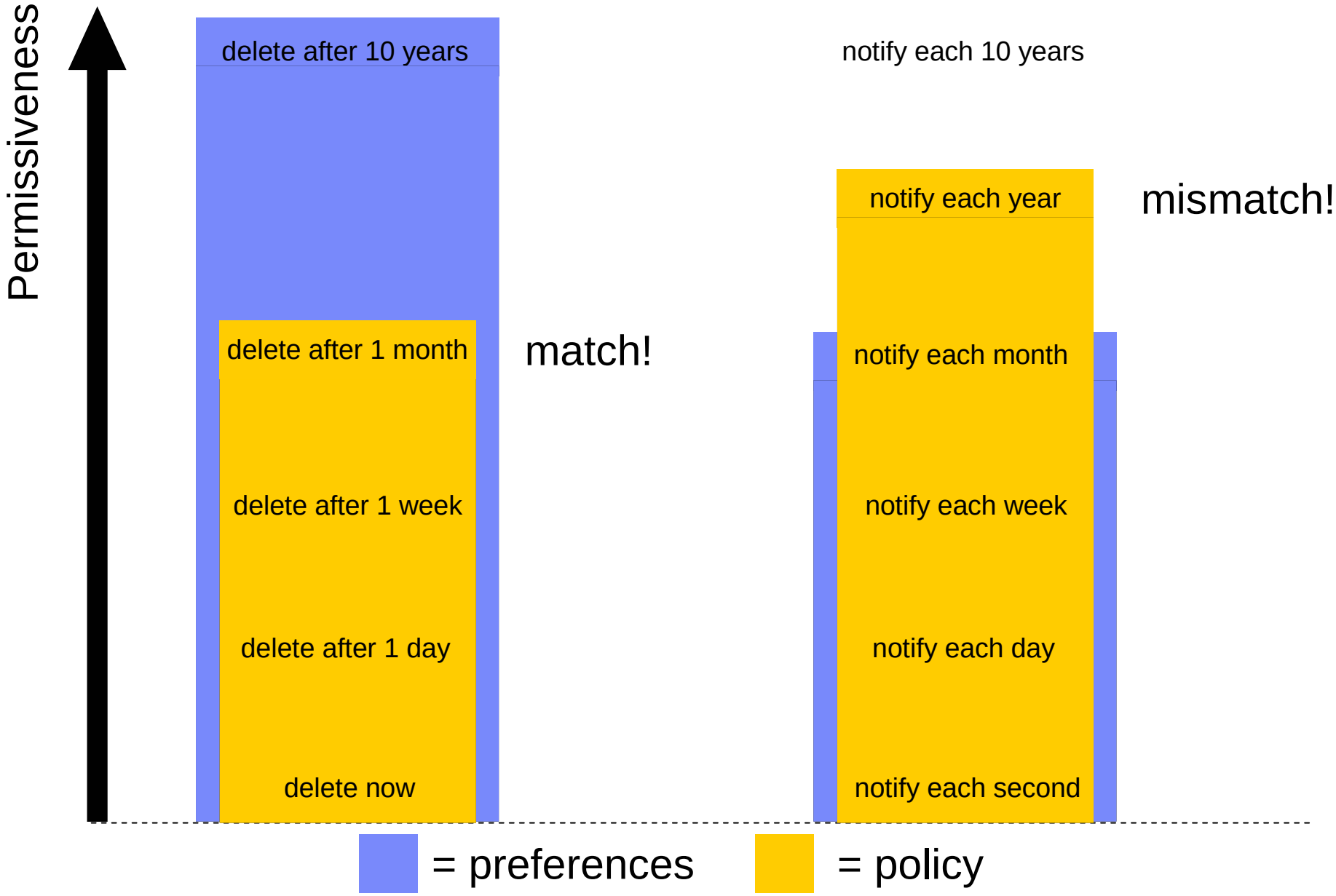
Authorizations vs. obligations issues avoided by

1. separating authorizations/obligations vocabulary
2. assuming imposing obligation implies authorization to execute it
3. assuming more privacy is always better

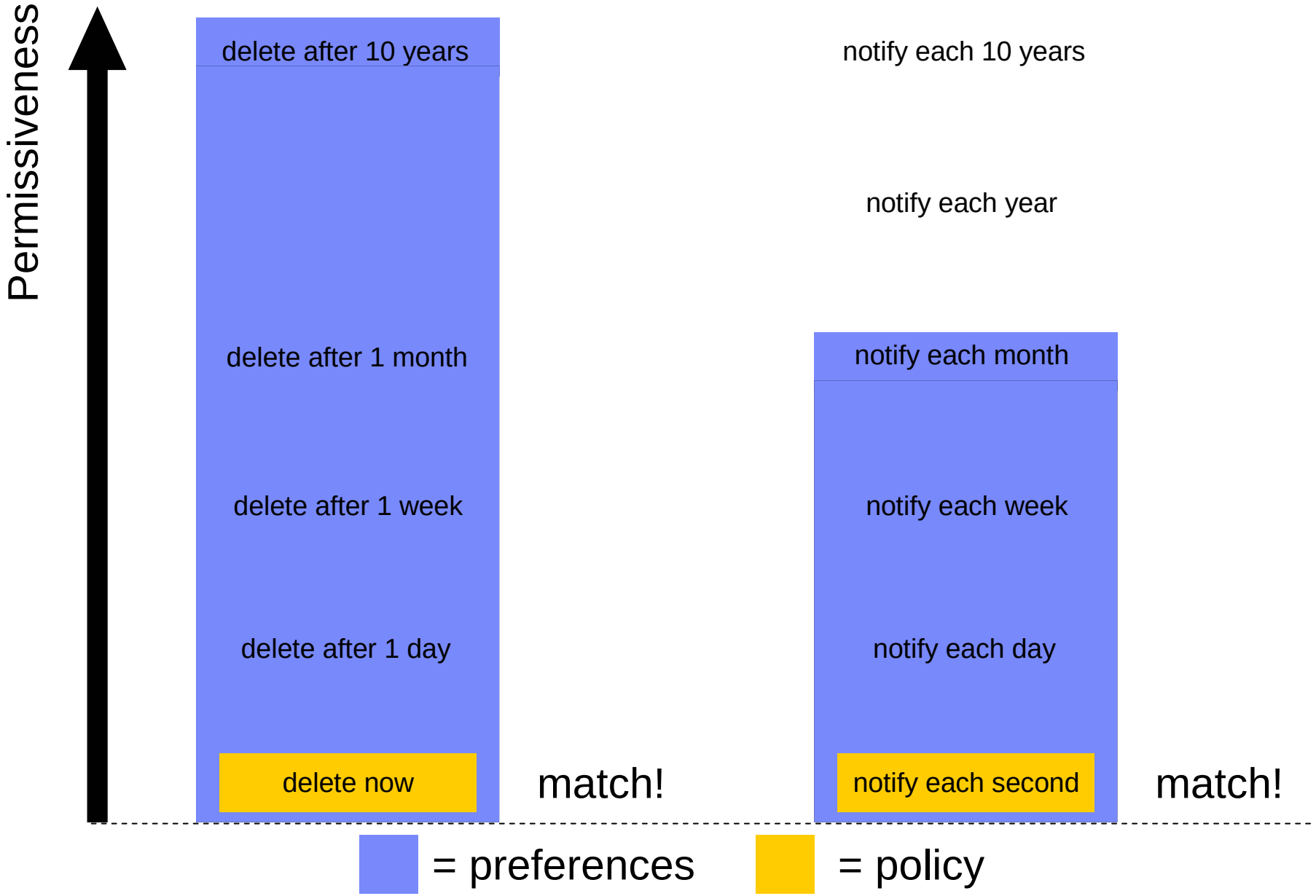
Graphical representation of matching



Graphical representation of matching



Graphical representation of matching



Solution

Avoid authorizations vs. obligations issues by

1. separate authorizations/obligations vocabulary
2. assuming imposing obligation implies authorization to execute it
3. ~~assuming more privacy is always better~~
3. explicitly specify **ranges** of permitted parameter values

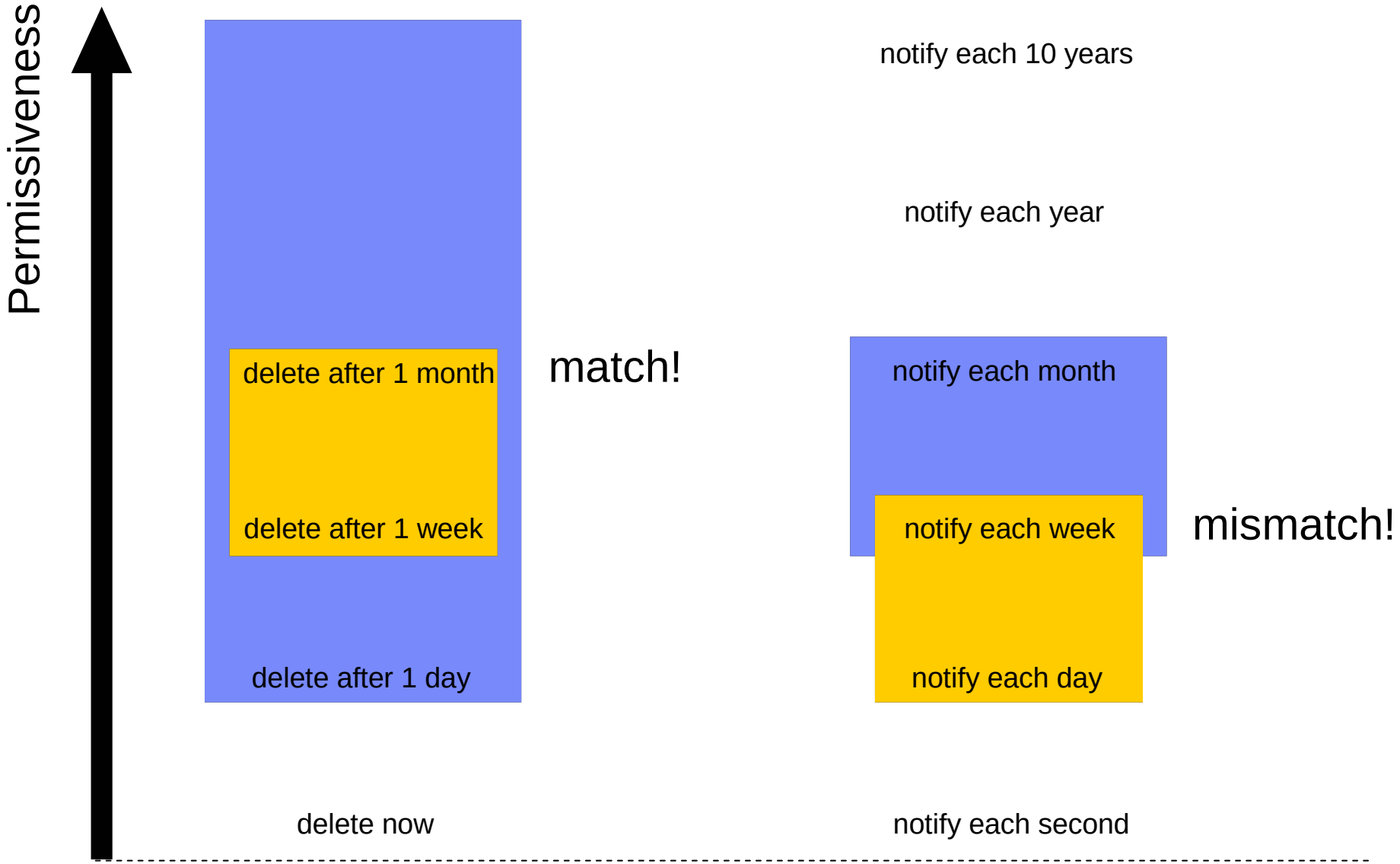
e.g.,

send usage summary with frequency $[f_{min},f_{max}]$

delete within time $[t_{min},t_{max}]$

use for purpose {admin}, {stats}, {marketing}, {admin,stats}, {admin,marketing}

Graphical representation of matching



Optional obligations

PPL matching definition:

data handling policy *Pol* **matches** preferences *Prefs* iff

$$Prefs \trianglerighteq Pol \iff$$
$$\forall A \in Pol.Auths \;\; \exists A' \in Prefs.Auths \;:\; A' \trianglerighteq A$$
$$\wedge \;\; \forall O \in Prefs.Obls \;\; \exists O' \in Pol.Obls \;:\; O \trianglerighteq O'$$

How to protect against adhering to *more* obligations than required?

Same approach works: specify **range of sets** of obligations

Simpler: specify **mandatory** and **optional** obligations

$$Prefs \trianglerighteq Pol \iff$$
$$\forall A \in Pol.Auths \;\; \exists A' \in Prefs.Auths \;:\; A' \trianglerighteq A$$
$$\wedge \;\; \forall O \in Prefs.MObls \;\; \exists O' \in Pol.MObls \;:\; O \trianglerighteq O'$$
$$\wedge \;\; \forall O \in (Pol.MObls \cup Pol.OObls)$$
$$\exists O' \in (Prefs.MObls \cup Prefs.OObls) \;:\; O' \trianglerighteq O$$

Conclusion

- Dependencies exist between authorizations and obligations
- Partially solved by strictly separating vocabularies
- Replacing obligation parameters values with ranges helps

  (at cost of more complicated policies and matching)