An Integrated Purpose Control Framework for Enhanced Privacy Protection

API-based System Implementation

Hari Siswantoro - siswantoro@fbk.eu

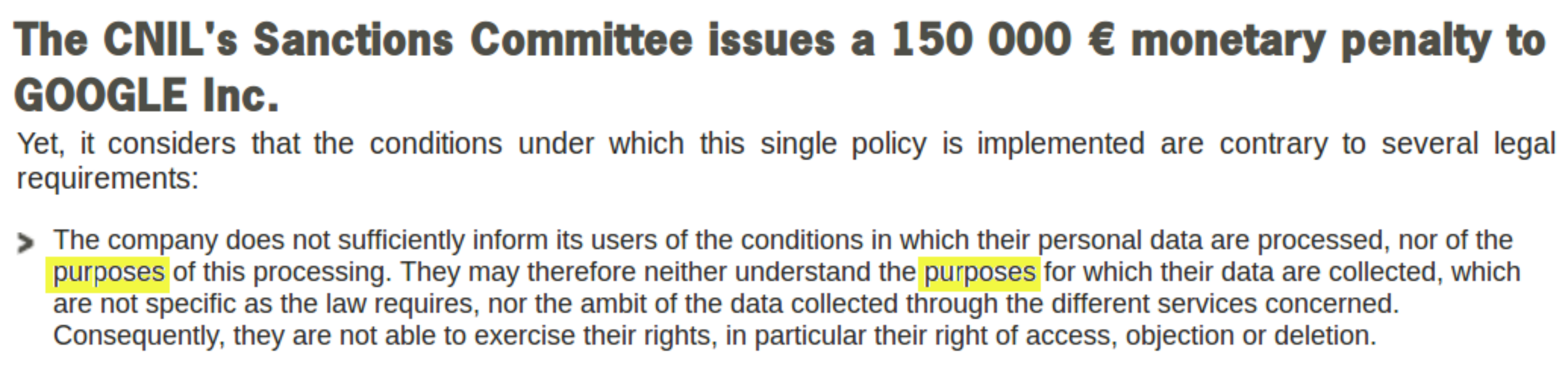
ST Retreat - October 21, 2014

Outline

- Purpose control motivation
- Purpose-based user policy
- API-based implementation
- Summary

Purpose Control Motivation

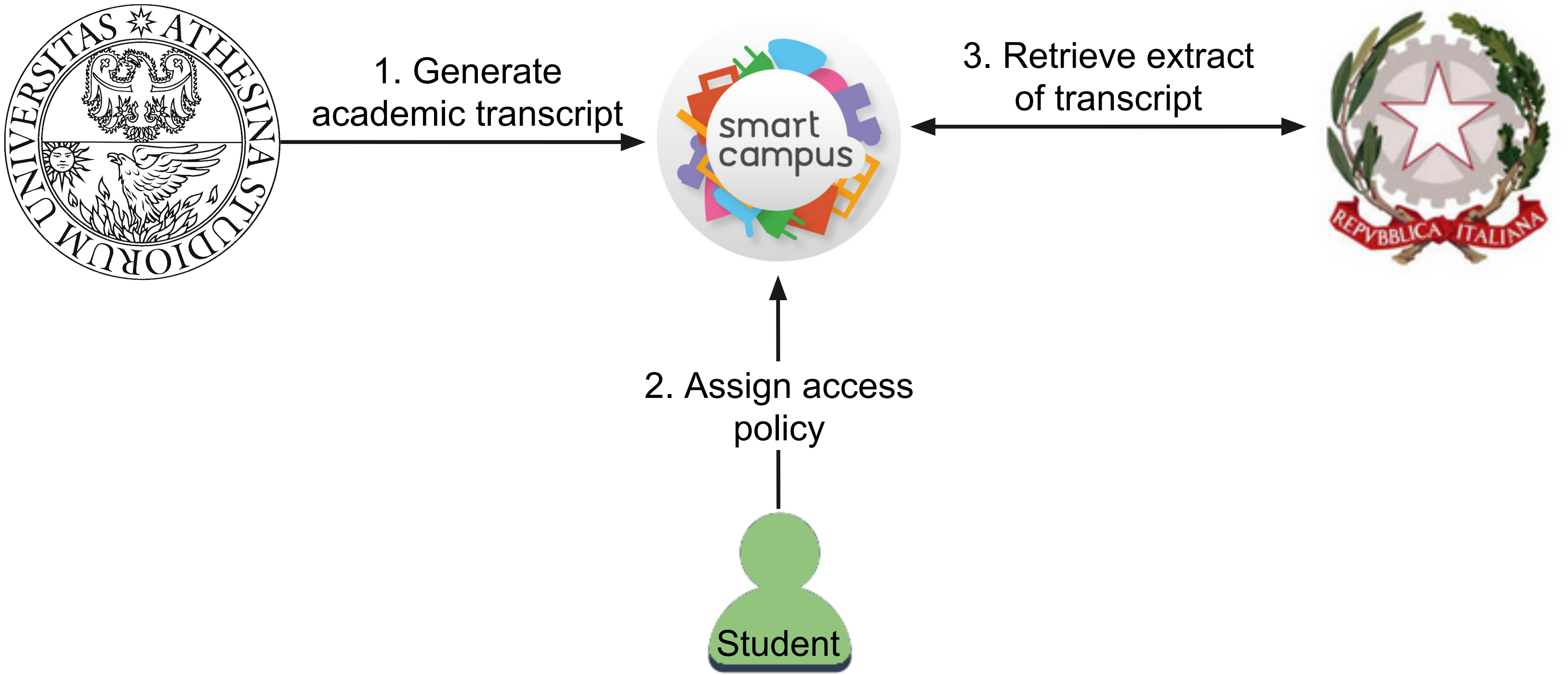
- Explicitly regulated by major legislations
- Lack of user participation
- Consider usability



The CNIL's Sanctions Committee issues a 150 000 € monetary penalty to GOOGLE Inc.

Yet, it considers that the conditions under which this single policy is implemented are contrary to several legal requirements:

- The company does not sufficiently inform its users of the conditions in which their personal data are processed, nor of the purposes of this processing. They may therefore neither understand the purposes for which their data are collected, which are not specific as the law requires, nor the ambit of the data collected through the different services concerned. Consequently, they are not able to exercise their rights, in particular their right of access, objection or deletion.



1. Generate academic transcript

2. Assign access policy

3. Retrieve extract of transcript

Student

Use Case Example

Purpose-based User Policy

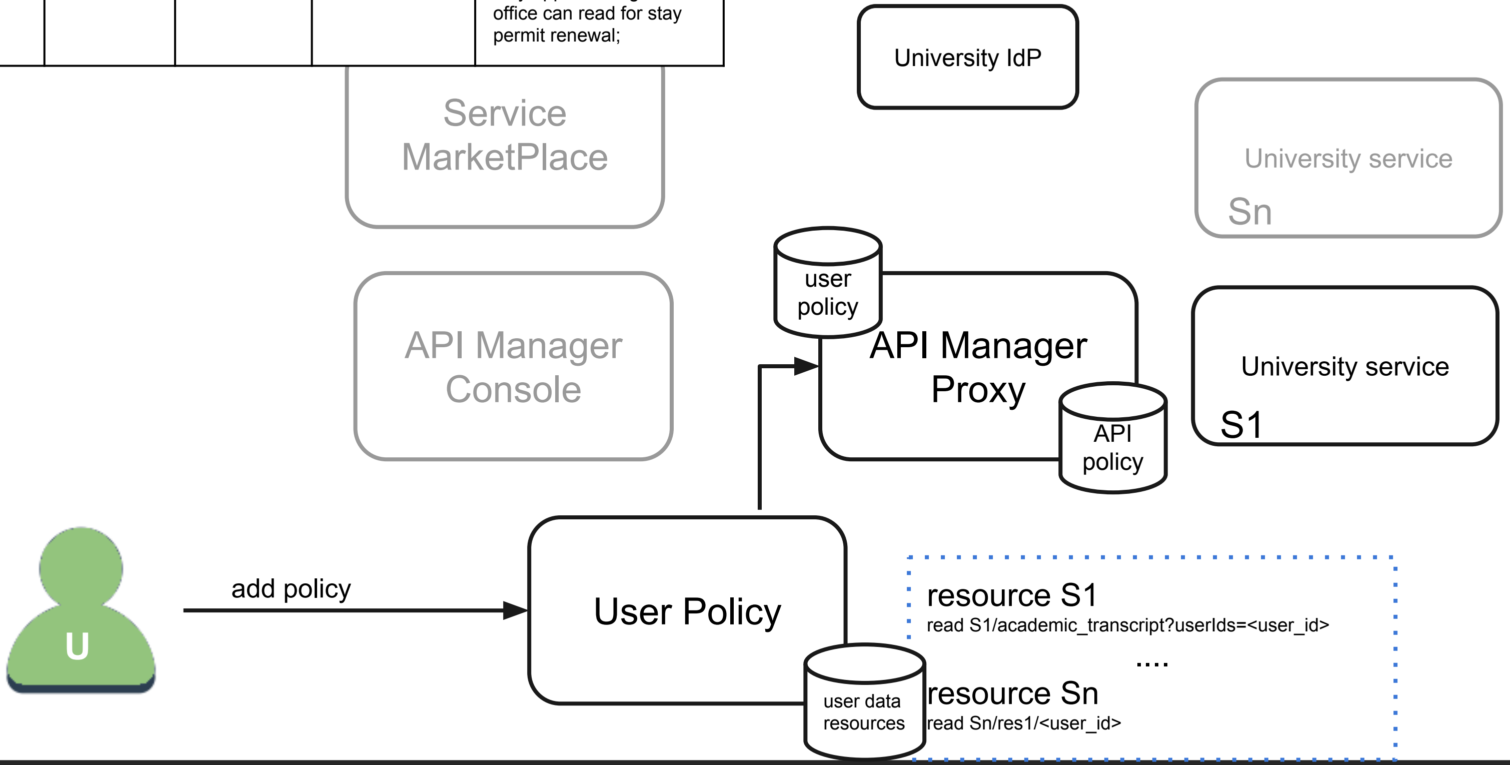
```
<recipients> CAN <actions> FOR <purposes> [IF
<gen_conditions>] [PROVIDED <provisions>] [FOLLOW
<obligations>]
```

where:

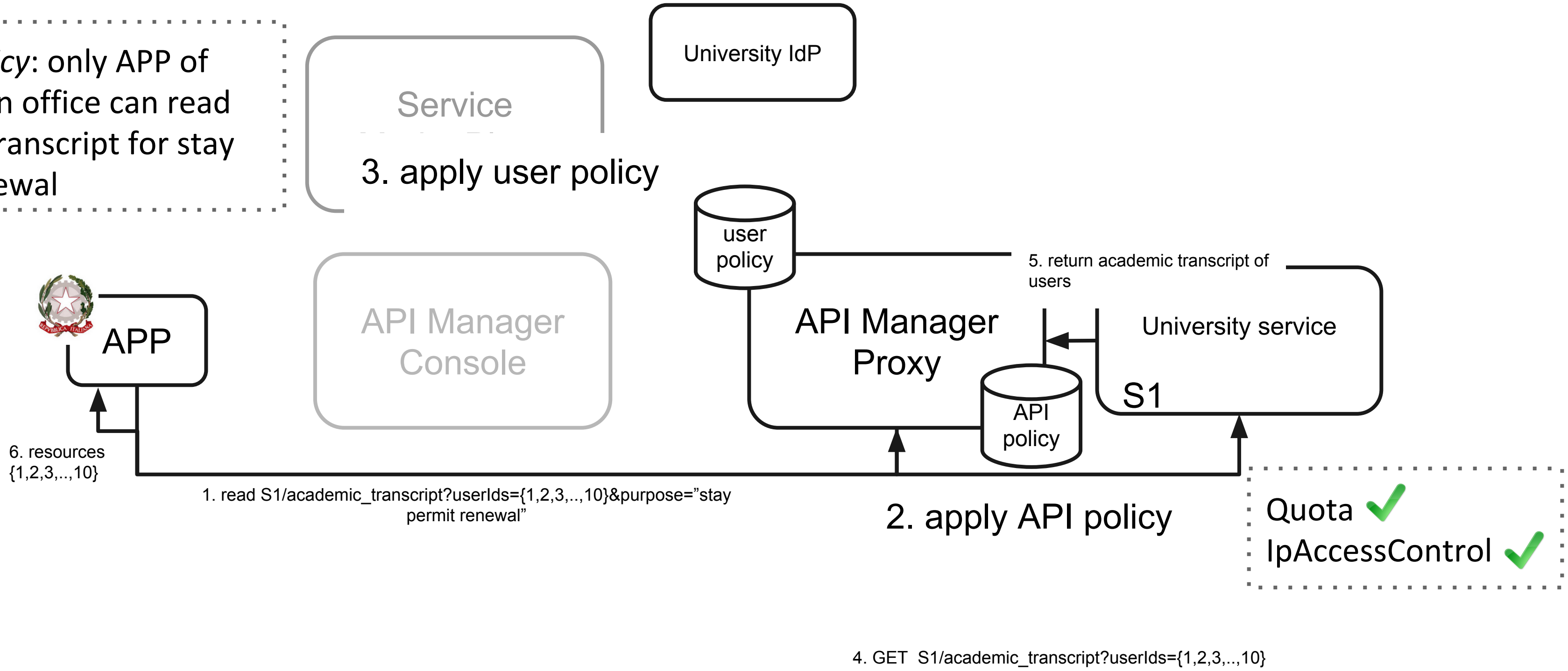
- recipient can be the requesting apps
- action is the set of actions
- purpose is the allowed intended purposes
- generic conditions, provisions and obligations are optional

API-based Implementation: SmartCampus

IdP	user id	API id	resource id	policy
unitn	1	1	2	only app of immigration office can read for stay permit renewal;



USER 1 policy: only APP of immigration office can read academic transcript for stay permit renewal



Service

3. apply user policy

University IdP

user policy

5. return academic transcript of users

API Manager Console

APP

API Manager Proxy

University service

S1

API policy

6. resources {1,2,3,..,10}

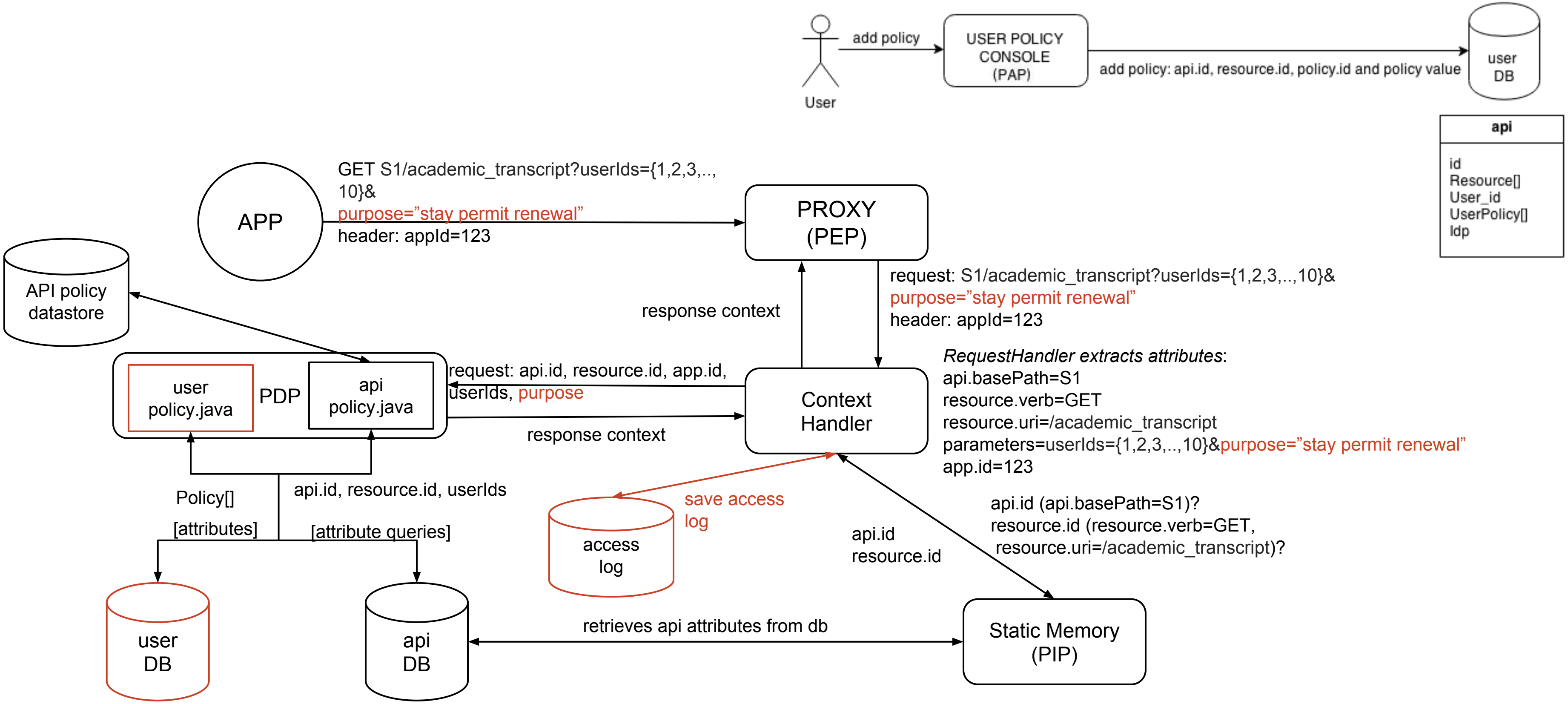
1. read S1/academic_transcript?userIds={1,2,3,..,10}&purpose=”stay permit renewal”

2. apply API policy

Quota ✓
IpAccessControl ✓

4. GET S1/academic_transcript?userIds={1,2,3,..,10}

Use Case Example



User

add policy

USER POLICY CONSOLE (PAP)

add policy: api.id, resource.id, policy.id and policy value

user DB

api

id
Resource[]
User_id
UserPolicy[]
Idp

APP

GET S1/academic_transcript?userIds={1,2,3,..,10}&
purpose="stay permit renewal"
header: appId=123

PROXY (PEP)

response context

request: S1/academic_transcript?userIds={1,2,3,..,10}&
purpose="stay permit renewal"
header: appId=123

API policy datastore

RequestHandler extracts attributes:
api.basePath=S1
resource.verb=GET
resource.uri=/academic_transcript
parameters=userIds={1,2,3,..,10}&purpose="stay permit renewal"
app.id=123

user policy.java

PDP

api policy.java

request: api.id, resource.id, app.id, userIds, purpose

response context

Context Handler

Policy[]

[attributes]

api.id, resource.id, userIds

[attribute queries]

save access log

access log

api.id
resource.id

api.id (api.basePath=S1)?
resource.id (resource.verb=GET,
resource.uri=/academic_transcript)?

user DB

api DB

retrieves api attributes from db

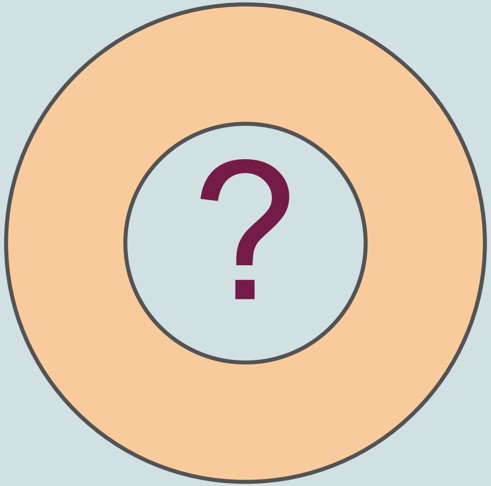
Static Memory (PIP)

API Manager Architecture

Summary

- Purpose control requirements: pre and post-release policy, involving user
- Challenge: post-release audit to multi parties system
- Future plan: protocol based purpose control framework

Grazie mille!



?