

# Identity, Privacy, and Data Protection in the Cloud – XACML

David Brossard  
Product Manager, Axiomatics

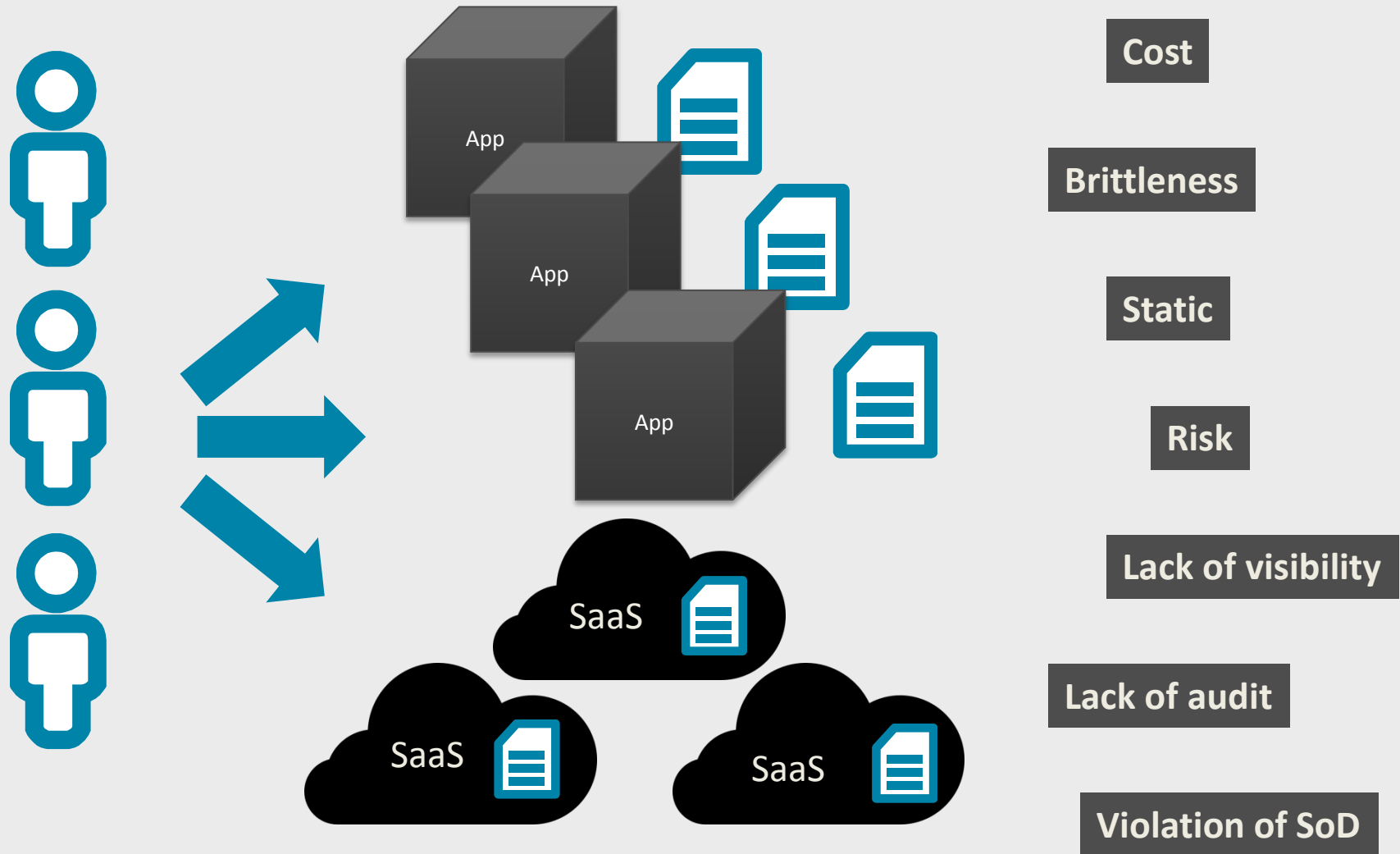
# What you will learn

- ❏ The issue with authorization in the cloud
- ❏ Quick background on XACML
- ❏ 3 strategies to extend authorization to the Cloud
- ❏ What it means for
  - ❏ customers
  - ❏ SaaS providers

# The issue with Authorization today

The black box challenge

# System growth leads to AuthZ challenges



# The Authorization Challenge

- ❏ What happens to my data?
- ❏ Who can access which information?
- ❏ How do I comply with (what the auditor will ask for)
  - ❏ Regulations?
    - ❏ E.g. Export Control
  - ❏ Contractual obligations?
- ❏ Going to the cloud doesn't make it easier
  - ❏ Do I need a different approach for cloud?

# Example: Manufacturing in the cloud

## Export Control

- Know the user (citizenship, location, affiliation)
- Know the end use (end location, purpose of use)



# XACML to the rescue

Implementing fine-grained  
authorization in the cloud

# Authorization is nearly always about

Role-based  
Access  
Control

Who?



Identity + role (+ group)



# Authorization should really be about...

Attribute-  
based  
Access  
Control

Who?

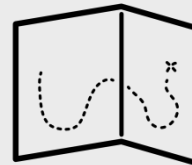
What?

When?

Where?

Why?

How?



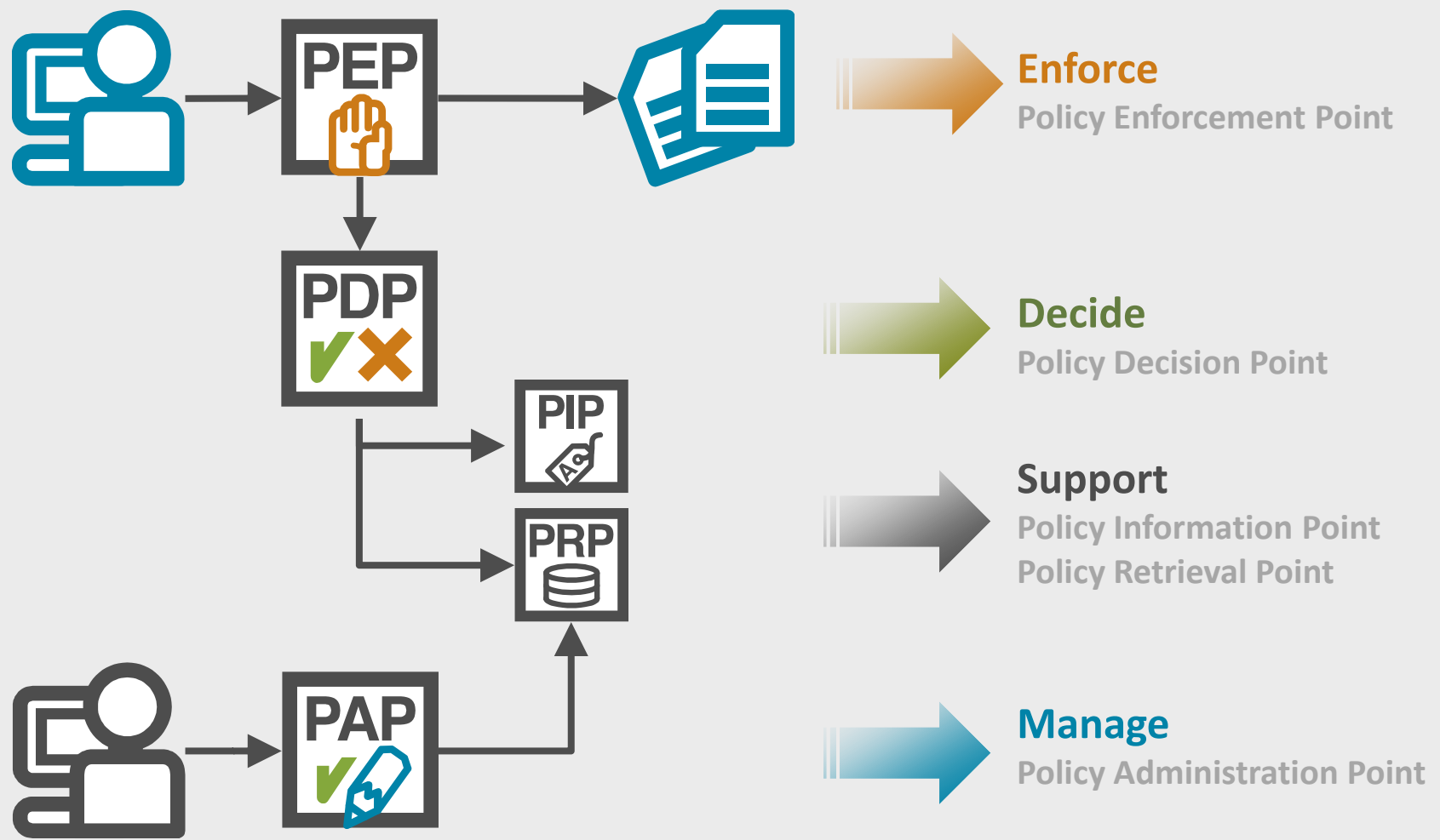
# Behold XACML, the standard for ABAC

- ❏ eXtensible Access Control Markup Language
- ❏ OASIS standard
- ❏ XACML is expressed as
  - ❏ A specification document (a [PDF](#)) + XML schema
- ❏ Policy-based & attribute-based language
  - ❏ Implement authorization based on object relations
  - ❏ Only employees of a given plant can see technical data linked to items assigned to the plant

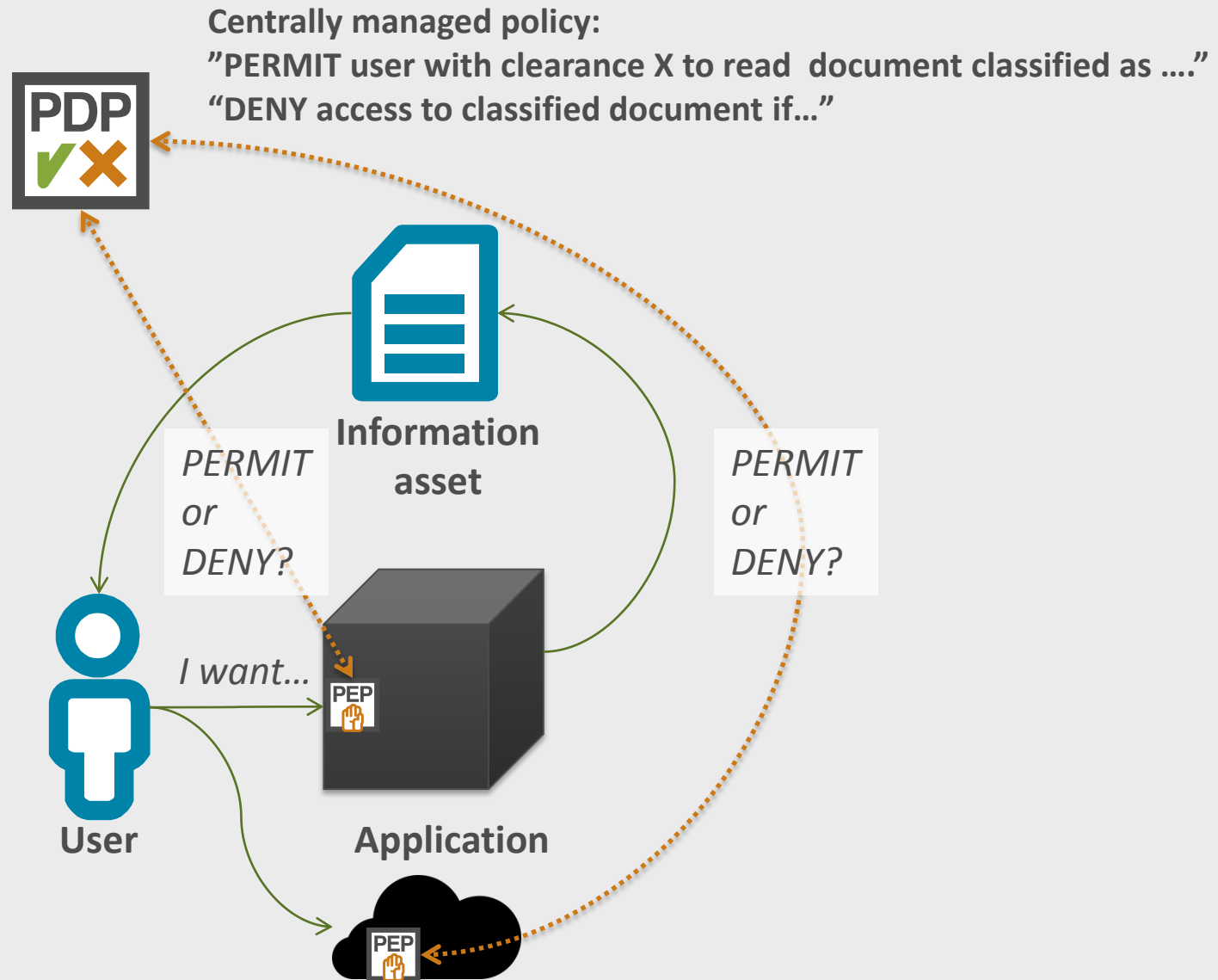
# Who's behind XACML?

- ❏ Oracle
- ❏ IBM
- ❏ Veterans Administration
- ❏ Axiomatics
- ❏ EMC<sup>2</sup>
- ❏ Bank of America
- ❏ The Boeing Company
- ❏ And many more...

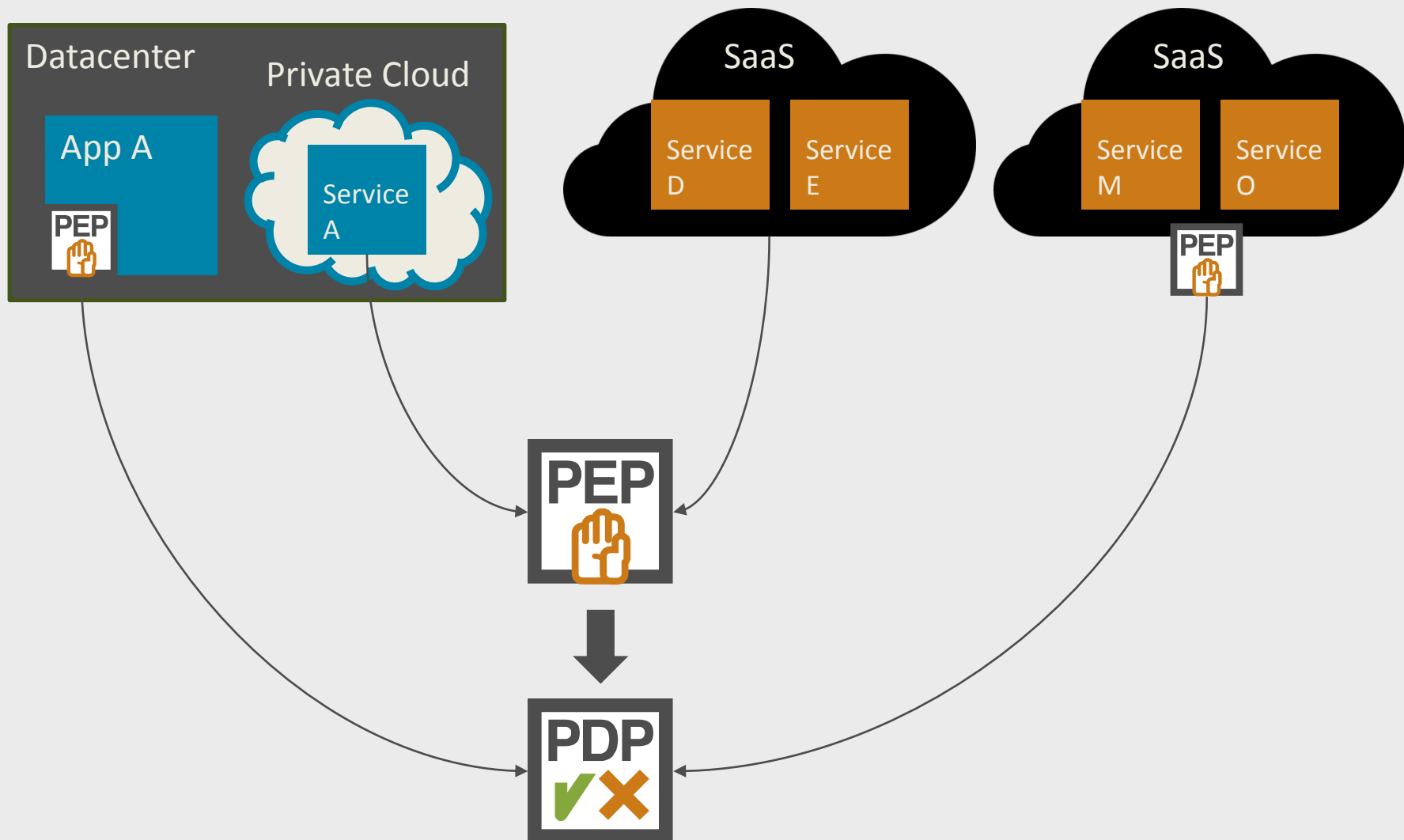
# Refresher: the XACML architecture



# XACML → Transparent & Externalized AuthZ



# XACML → Anywhere AuthZ & Architecture



# Fine-grained Authorization for the Cloud

Three strategies for externalized  
authorization in the cloud

# Option #1 – tell your provider to adopt XACML

- ❏ A SaaS provider should offer
  - ❏ Functional APIs (their core business)
  - ❏ Non-functional (Security) APIs
- ❏ Let customers push their own XACML policies
- ❏ Apply the administrative delegation profile
  - ❏ <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-administration-v1-spec-en.html>

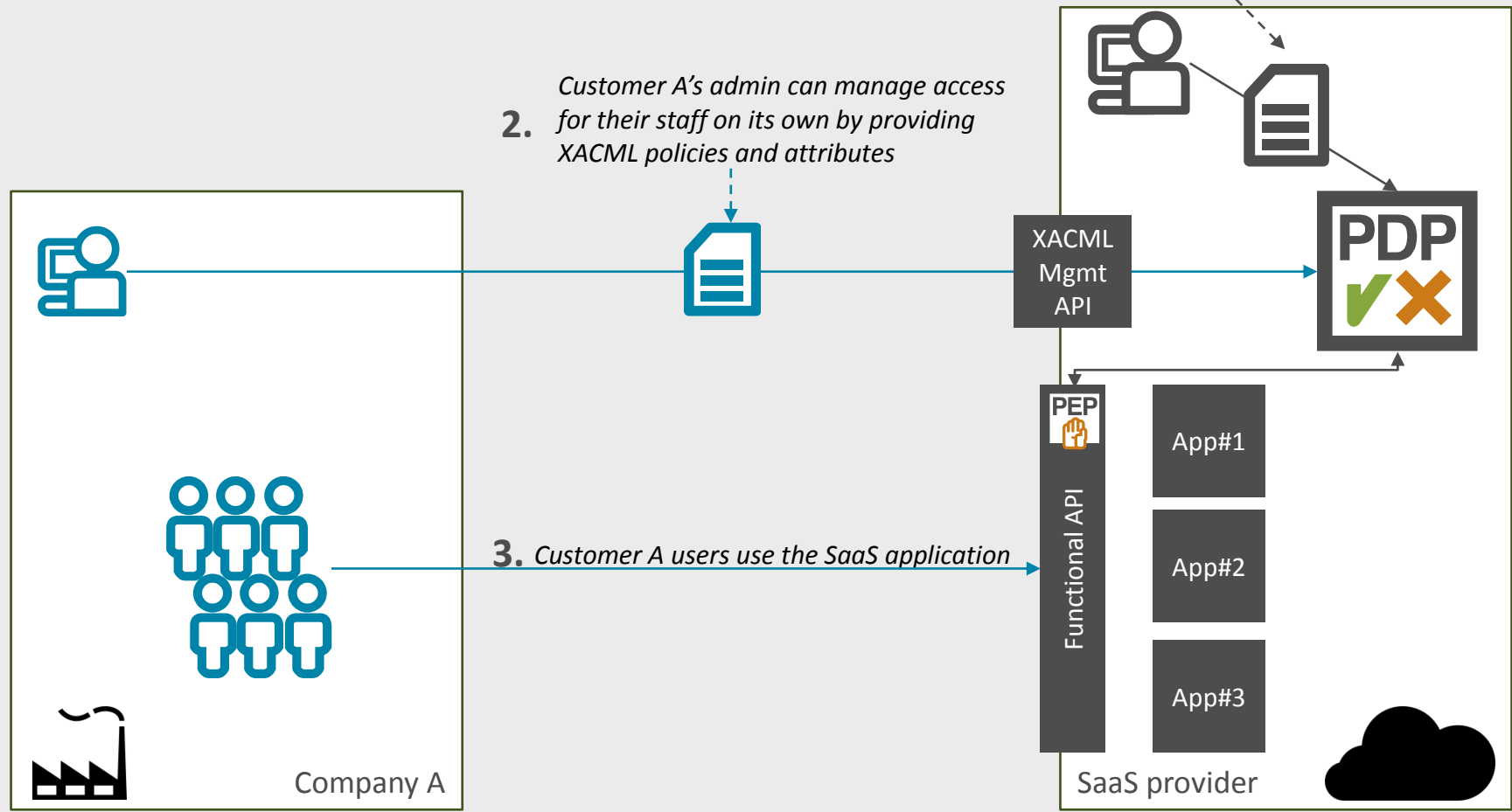


# Option #1 – Architecture

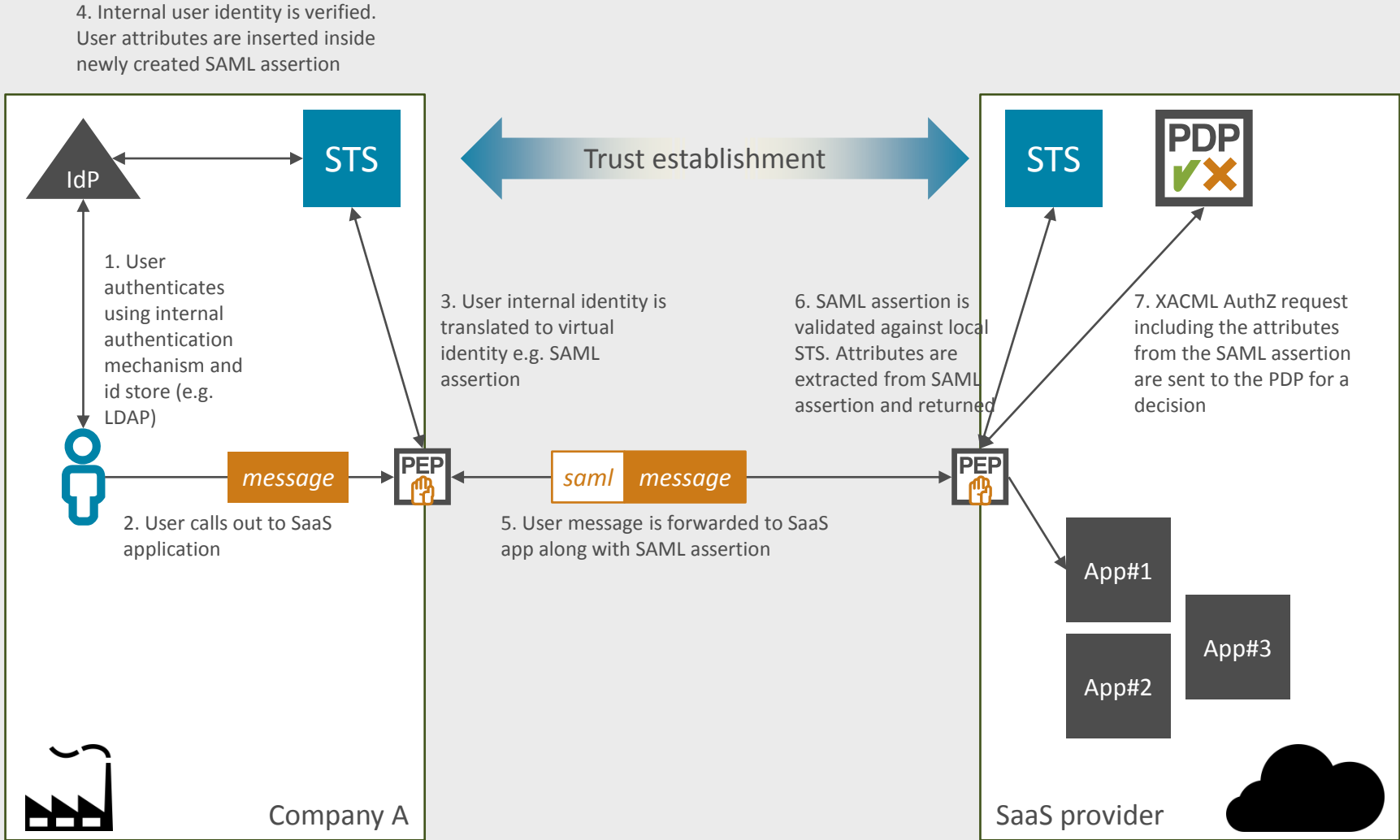
1. *SaaS Admin delegates rights to manage access control provided to customer A. The rights are restricted to only the applications and resources provided to this particular customer's users.*

2. *Customer A's admin can manage access for their staff on its own by providing XACML policies and attributes*

3. *Customer A users use the SaaS application*



# Option #1 – Architecture (including id. Federation)



# Option #1 – Pros & Cons

## Pros

- ▣ Consistent access control
- ▣ Fine-grained
- ▣ Risk-aware
- ▣ Future-proof
- ▣ SaaS vendor benefit
  - ▣ multi-tenancy

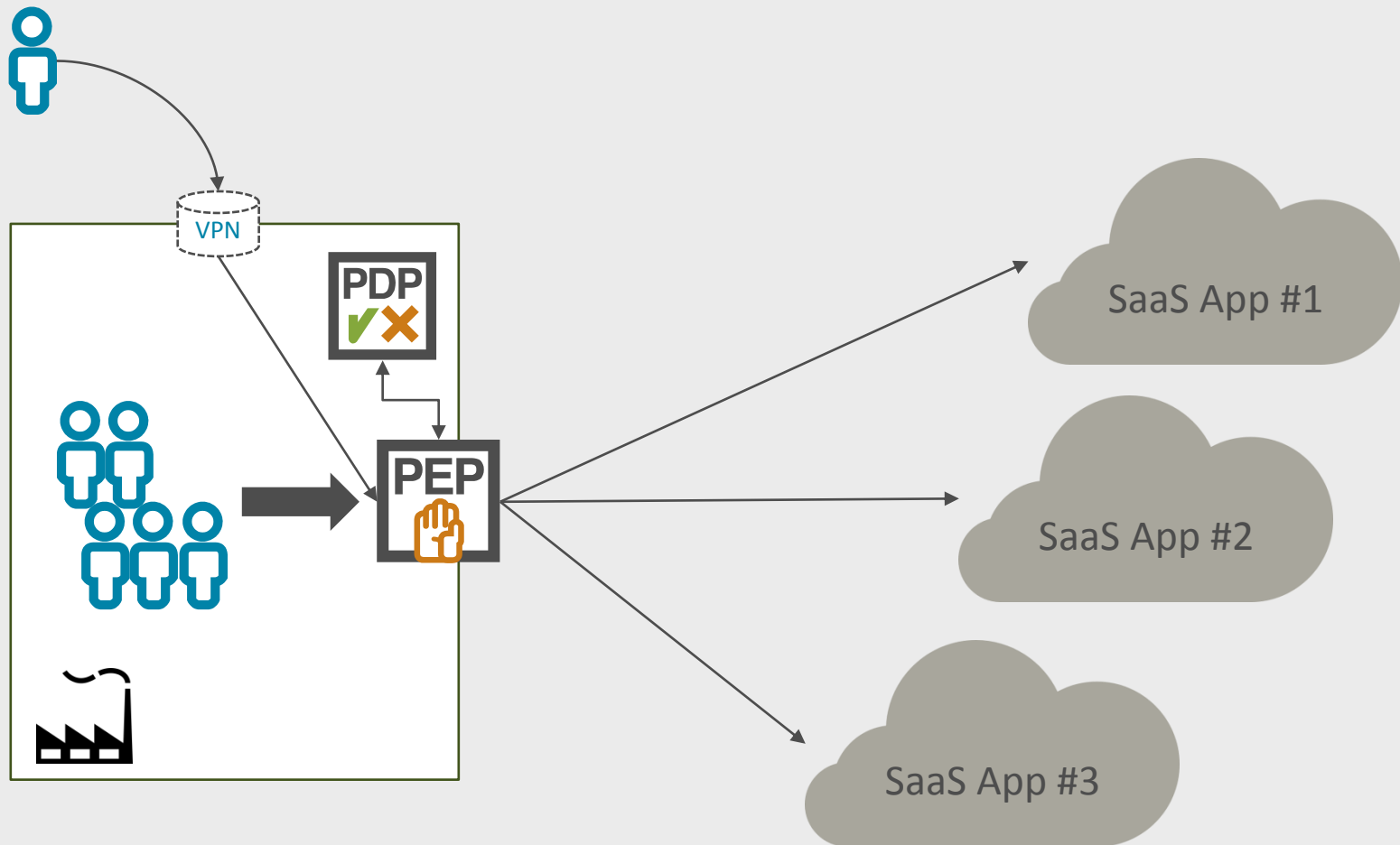
## Cons

- ▣ Not many SaaS vendors support XACML today

# Option #2 – Proxy your cloud connections

- ❏ If you can restrict access to SaaS applications from within the corporate network...
- ❏ All access to SaaS apps could be made to tunnel through a proxy

# Option #2 – Architecture



# Option #2 – Pros & Cons

## Pros

- ❏ Workaround current SaaS limitations
- ❏ Easy to deploy
- ❏ Available today

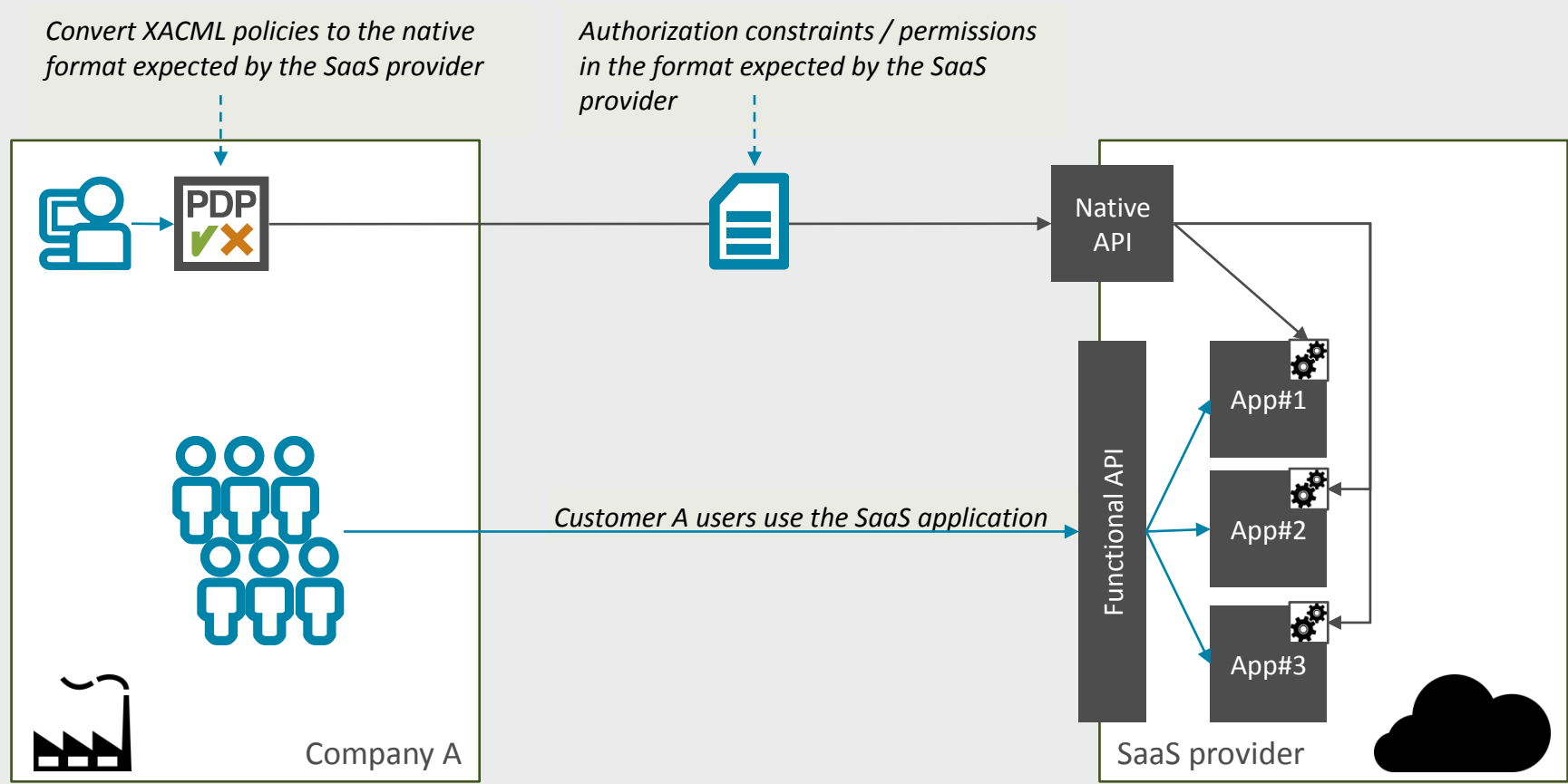
## Cons

- ❏ No direct access to SaaS app
  - ❏ Forces users to go via VPN
- ❏ Access may not be as fine grained as Option #1
  - ❏ Lack of visibility into the SaaS data

# Option #3 – Policy Provisioning based on XACML

- ❏ What if the provider is reluctant to adopt XACML?
- ❏ *“If the application won’t go to XACML then XACML will go to the application”*
  - ❏ Eve Maler, Forrester
- ❏ You still get
  - ❏ Centrally managed authorization
  - ❏ Standards-based (XACML)
- ❏ Approach
  - ❏ Convert from XACML to expected SaaS format
  - ❏ Push via SaaS management APIs

# Option #3 – Architecture





# Option #3 – Pros & Cons

## Pros

- ▣ Feasible today
- ▣ Viable solution
- ▣ Extends the customer's XACML-based authorization system's reach

## Cons

- ▣ Possible loss of XACML richness in access control
- ▣ Loss of dynamic nature

# Standards & the Cloud

- ▣ Standards are important for the cloud
  - ▣ It promotes vendor interoperability
  - ▣ It promotes layer interoperability
- ▣ Example
  - ▣ XACML authorization services can easily use SAML<br>OAUTH, OAUTH2, OpenID...
  - ▣ XACML can also use semantic web standards
- ▣ This leads to easier deployments and faster ROI

# To summarize

## ❏ Cloud requires eXtensible Authorization

- ❏ Fine-grained
- ❏ Externalized

## ❏ Traditional approaches

- ❏ #1: tell your SaaS provider to adopt XACML.
- ❏ #2: proxy your cloud connections.

## ❏ Extended approach

- ❏ #3: Policy Provisioning based on XACML
- ❏ Also works for business apps (SharePoint, Windows)

*Every cloud  
has a  
XACML  
lining*

# Online resources

- ❏ OASIS technical committee:

- ❏ <http://www.oasis-open.org/committees/xacml/>

- ❏ LinkedIn group

- ❏ <http://www.linkedin.com/groups/OASIS-XACML-3934718>