On the (Privacy) Harms of the European Digital Identity Framework

Christian Knabenhans and Shannon Veitch

CAW 2026

Joint work with Mathilde Raynal, Sylvain Chatel, Theresa Stadler, Wouter Lueks, and Carmela Troncoso

REGULATION

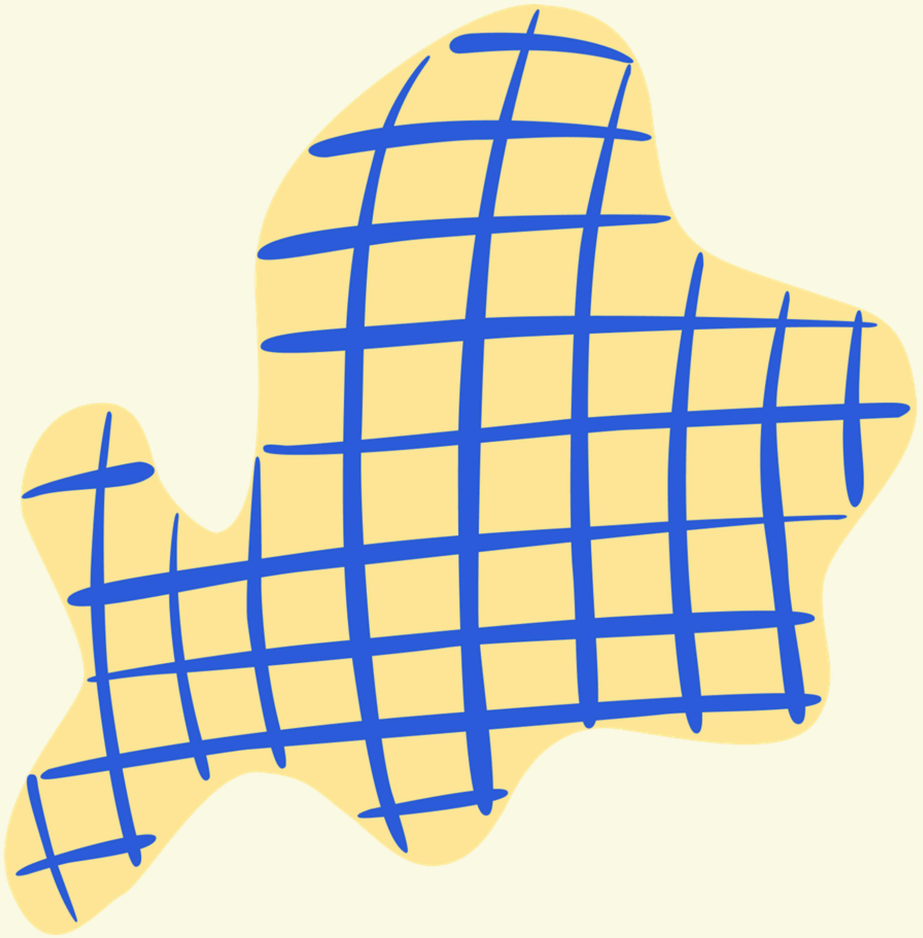
The EU Digital Identity Framework: a new era for Digital Identification

The European Digital Identity Regulation (EU) 2024/1183 was adopted on 20 May 2024.

It seeks to enhance the way citizens and businesses securely identify and authenticate themselves online, strengthening trust in cross-border services by ensuring the availability of a unified, secure, and user-friendly digital identity solution across the EU.

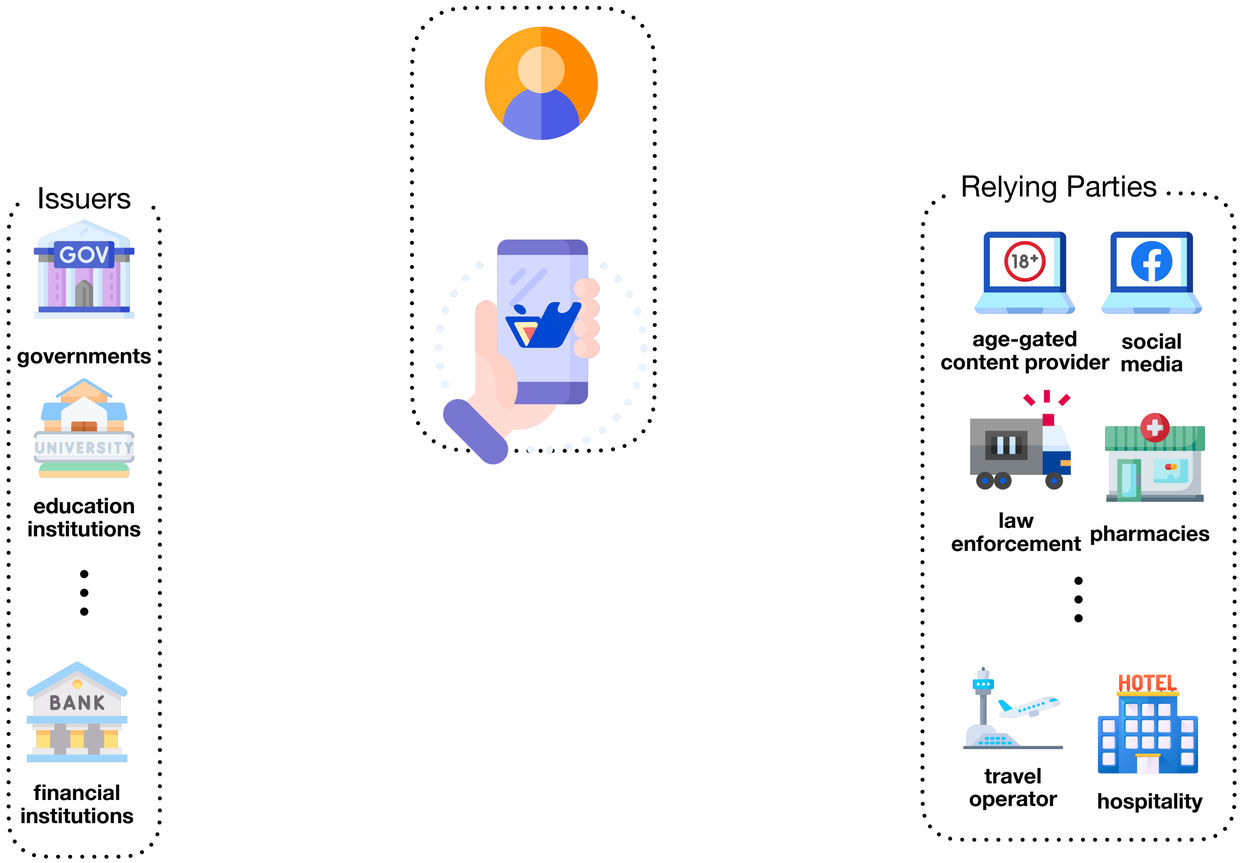
Learn more about the initiative ›

Explore the regulation

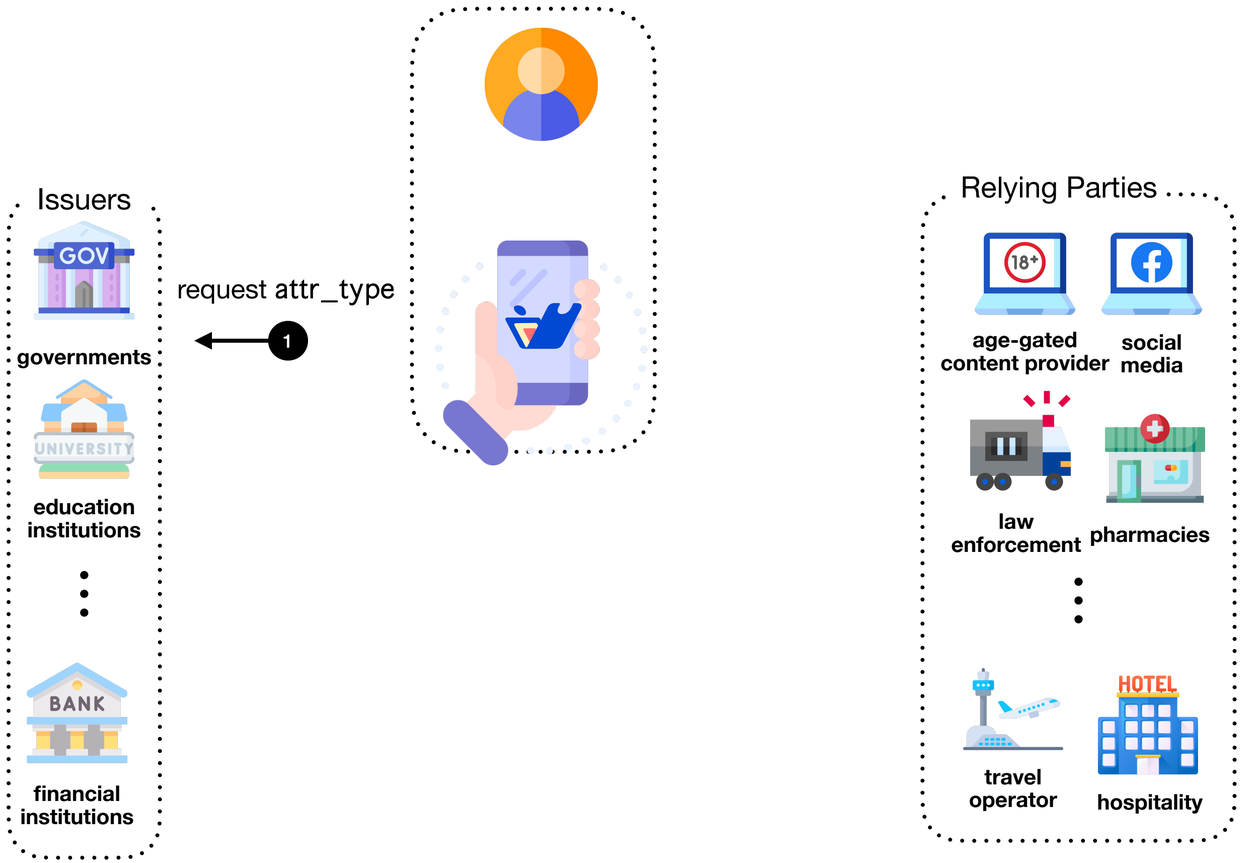


EUDI graphics from: https://ec.europa.eu/digital-building-blocks/sites/spaces/EUDIGITALIDENTITYWALLET/pages/694487738/EU+Digital+Identity+Wallet+Home

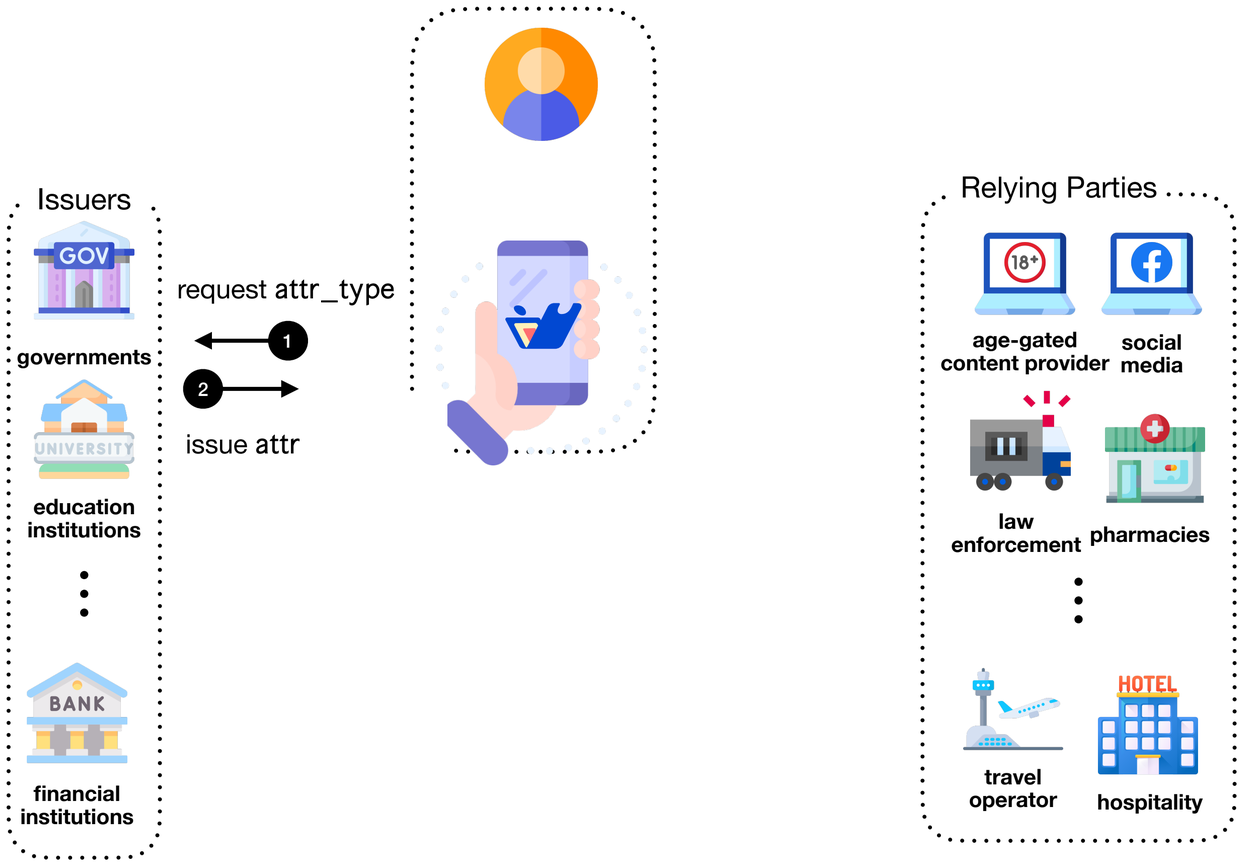
European Digital Identity Framework (EUDIF)



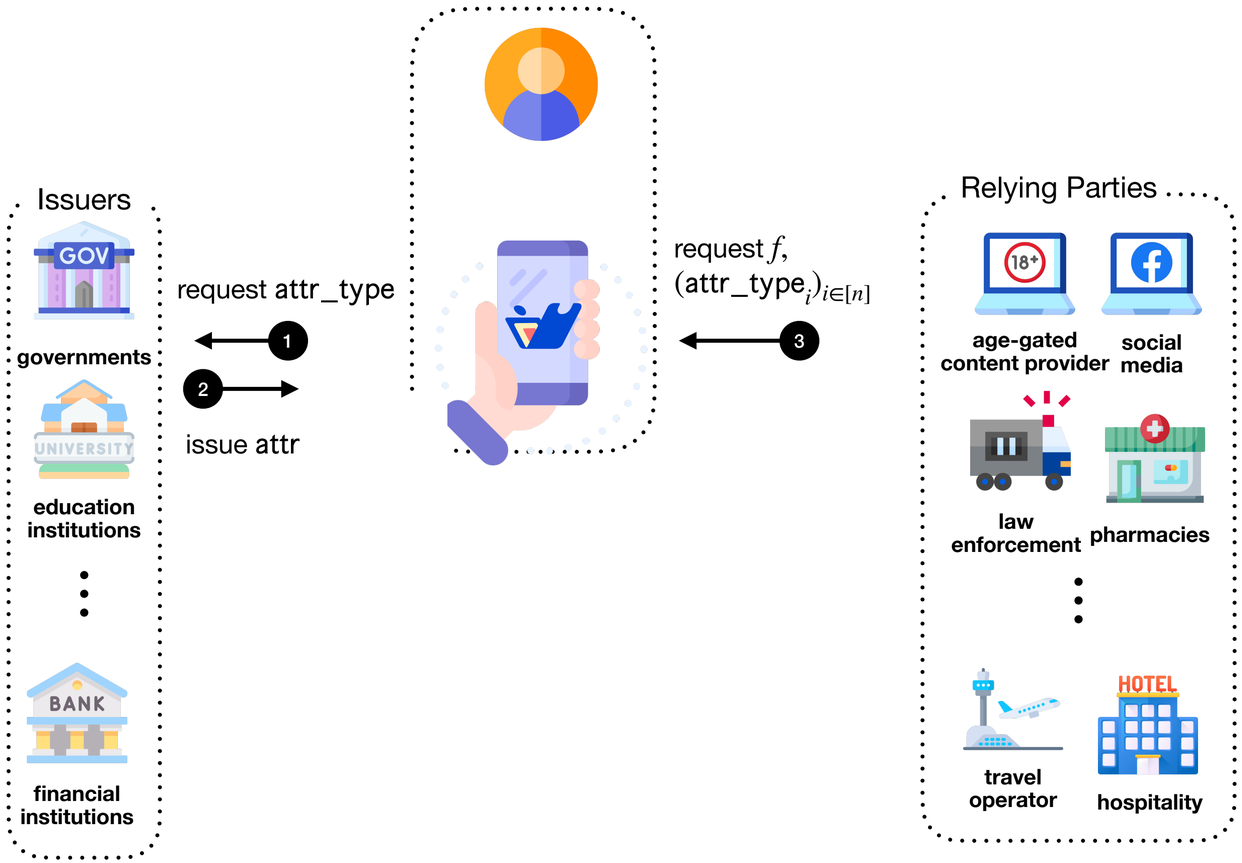
European Digital Identity Framework (EUDIF)



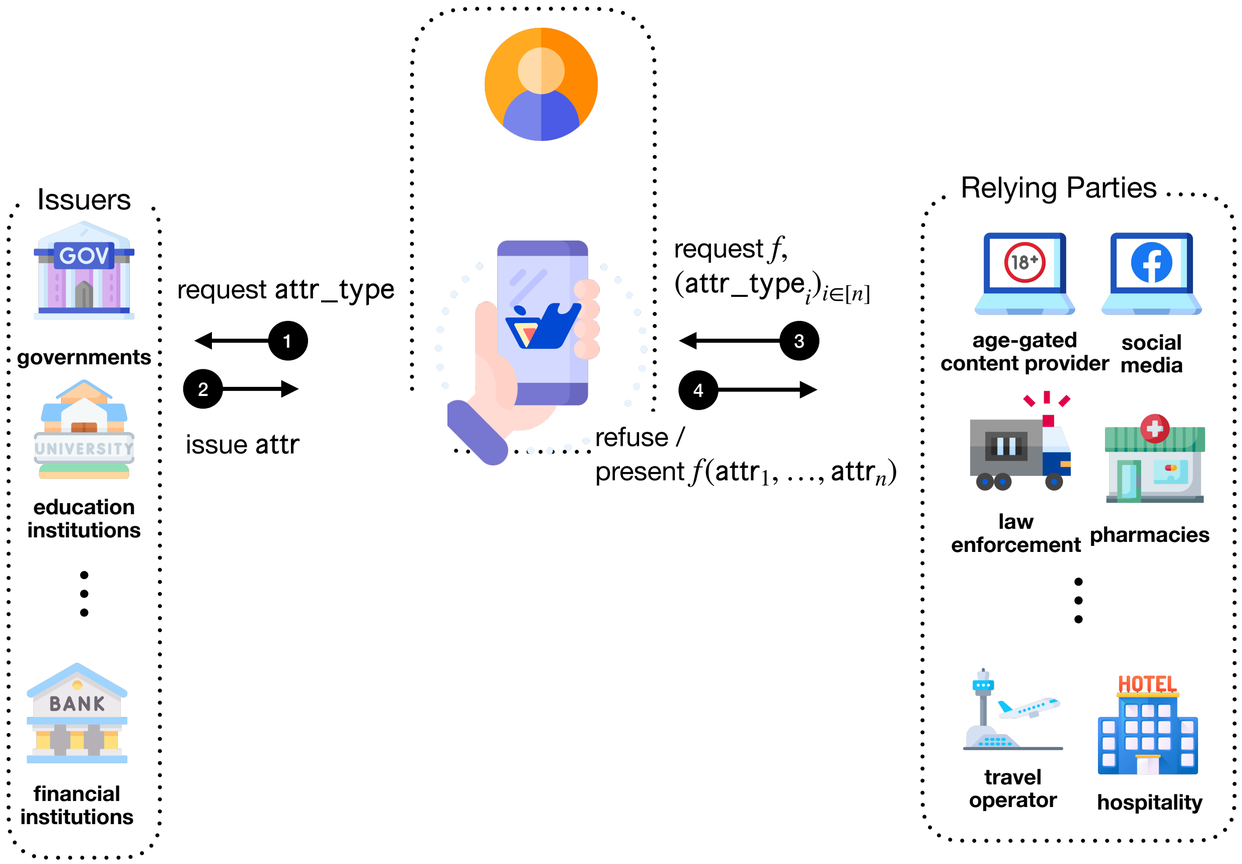
European Digital Identity Framework (EUDIF)



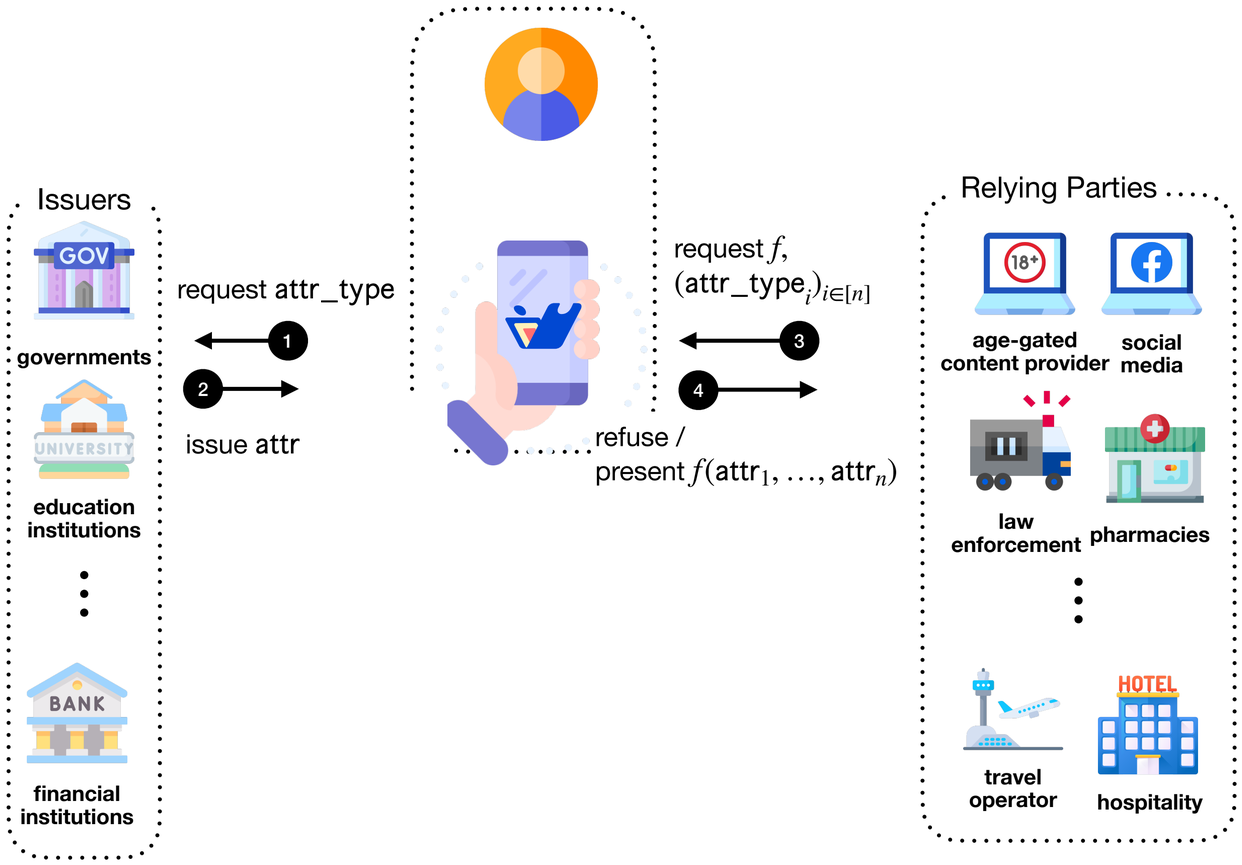
European Digital Identity Framework (EUDIF)

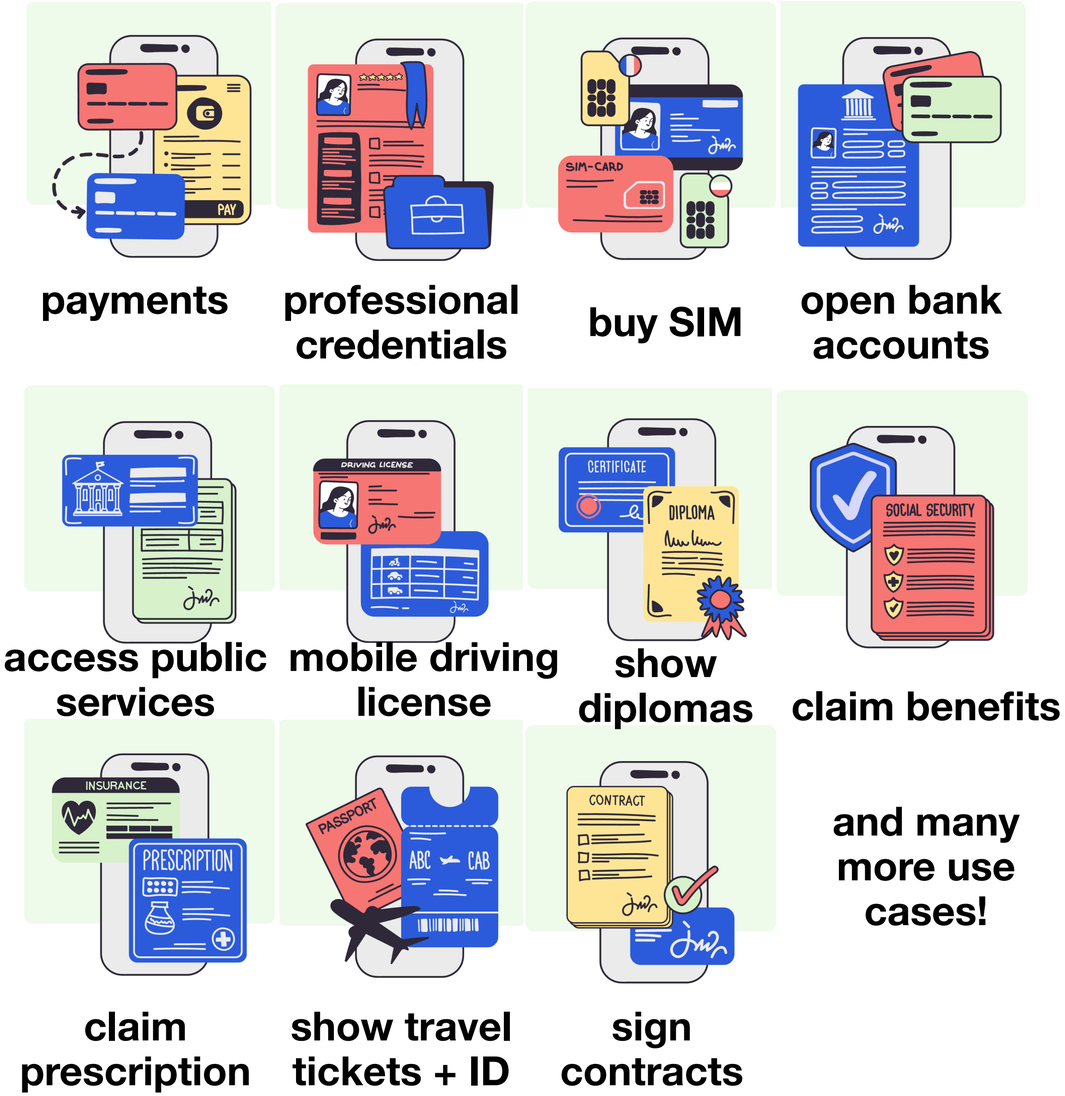


European Digital Identity Framework (EUDIF)



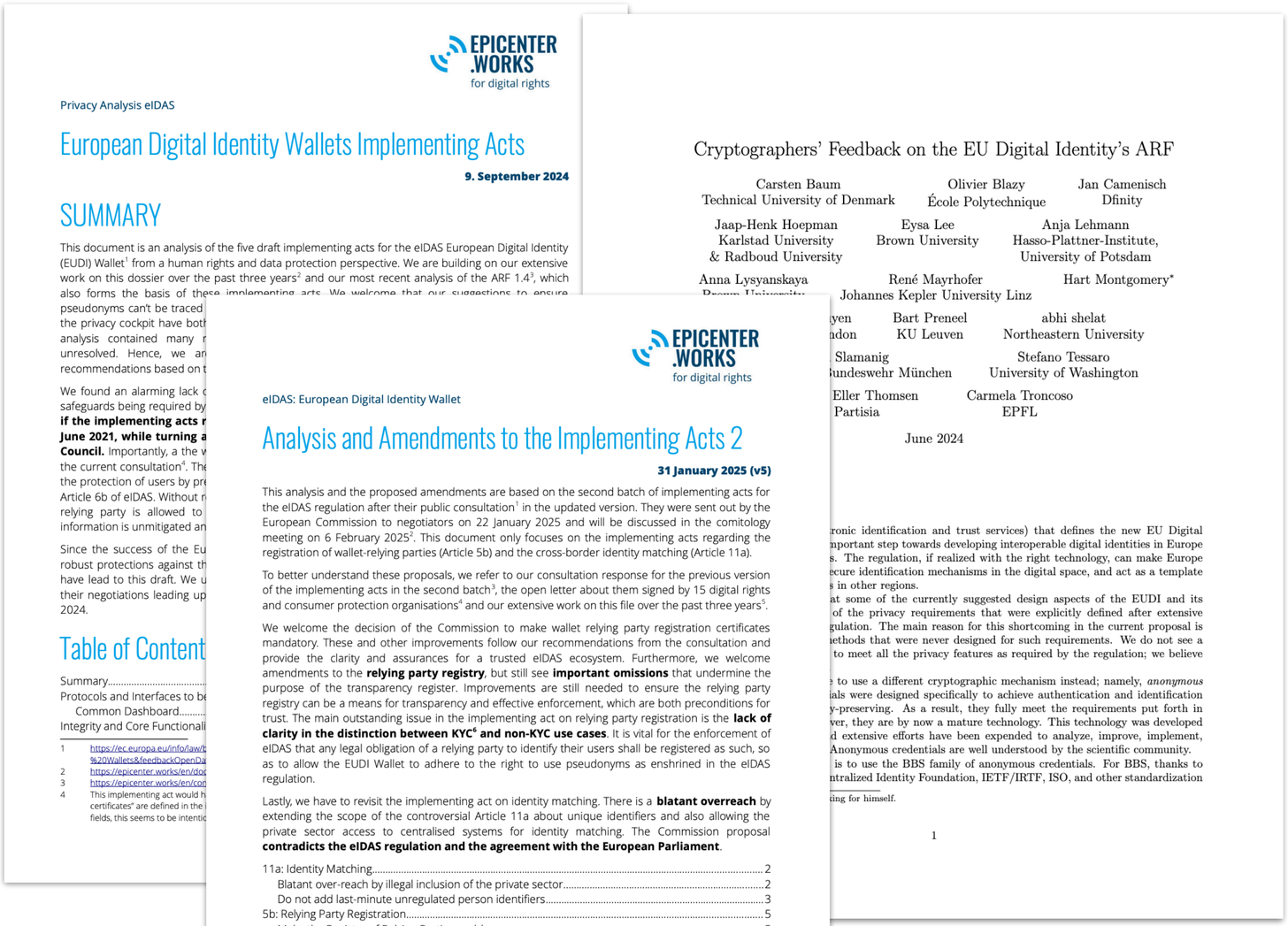
European Digital Identity Framework (EUDIF)





Privacy and the EUDIF

Privacy and the EUDIF



EPICENTER .WORKS for digital rights

Privacy Analysis eIDAS

European Digital Identity Wallets Implementing Acts

9. September 2024

SUMMARY

This document is an analysis of the five draft implementing acts for the eIDAS European Digital Identity (EUDI) Wallet[1] from a human rights and data protection perspective. We are building on our extensive work on this dossier over the past three years[2] and our most recent analysis of the ARF 1.4[3], which also forms the basis of these implementing acts. We welcome that our suggestions to ensure pseudonyms can't be traced
the privacy cockpit have both
analysis contained many r
unresolved. Hence, we ar
recommendations based on t

We found an alarming lack
safeguards being required by
if the implementing acts r
June 2021, while turning a
Council. Importantly, a the v
the current consultation[4]. The
the protection of users by pr
Article 6b of eIDAS. Without r
relying party is allowed to
information is unmitigated an

Since the success of the Eu
robust protections against th
have lead to this draft. We u
their negotiations leading up
2024.

Table of Content

1 https://ec.europa.eu/info/law/b
9%20Wallets&feedbackOpenDa
2 https://epicenter.works/en/do
3 https://epicenter.works/en/con
4 This implementing act would h
certificates" are defined in the
fields, this seems to be intentic

EPICENTER .WORKS for digital rights

eIDAS: European Digital Identity Wallet

Analysis and Amendments to the Implementing Acts 2

31 January 2025 (v5)

This analysis and the proposed amendments are based on the second batch of implementing acts for the eIDAS regulation after their public consultation[1] in the updated version. They were sent out by the European Commission to negotiators on 22 January 2025 and will be discussed in the comitology meeting on 6 February 2025[2]. This document only focuses on the implementing acts regarding the registration of wallet-relying parties (Article 5b) and the cross-border identity matching (Article 11a).

To better understand these proposals, we refer to our consultation response for the previous version of the implementing acts in the second batch[3], the open letter about them signed by 15 digital rights and consumer protection organisations[4] and our extensive work on this file over the past three years[5].

We welcome the decision of the Commission to make wallet relying party registration certificates mandatory. These and other improvements follow our recommendations from the consultation and provide the clarity and assurances for a trusted eIDAS ecosystem. Furthermore, we welcome amendments to the **relying party registry**, but still see **important omissions** that undermine the purpose of the transparency register. Improvements are still needed to ensure the relying party registry can be a means for transparency and effective enforcement, which are both preconditions for trust. The main outstanding issue in the implementing act on relying party registration is the **lack of clarity in the distinction between KYC[6] and non-KYC use cases**. It is vital for the enforcement of eIDAS that any legal obligation of a relying party to identify their users shall be registered as such, so as to allow the EUDI Wallet to adhere to the right to use pseudonyms as enshrined in the eIDAS regulation.

Lastly, we have to revisit the implementing act on identity matching. There is a **blatant overreach** by extending the scope of the controversial Article 11a about unique identifiers and also allowing the private sector access to centralised systems for identity matching. The Commission proposal **contradicts the eIDAS regulation and the agreement with the European Parliament**.

Cryptographers' Feedback on the EU Digital Identity's ARF

Carsten Baum, Technical University of Denmark
Olivier Blazy, École Polytechnique
Jan Camenisch, Dfinity
Jaap-Henk Hoepman, Karlstad University & Radboud University
Eysa Lee, Brown University
Anja Lehmann, Hasso-Plattner-Institute, University of Potsdam
Anna Lysyanskaya, Brown University
René Mayrhofer, Johannes Kepler University Linz
Hart Montgomery*
yen, ndon
Bart Preneel, KU Leuven
abhi shelat, Northeastern University
Slamanig, undeswehr München
Stefano Tessaro, University of Washington
Eller Thomsen, Partisia
Carmela Troncoso, EPFL

June 2024

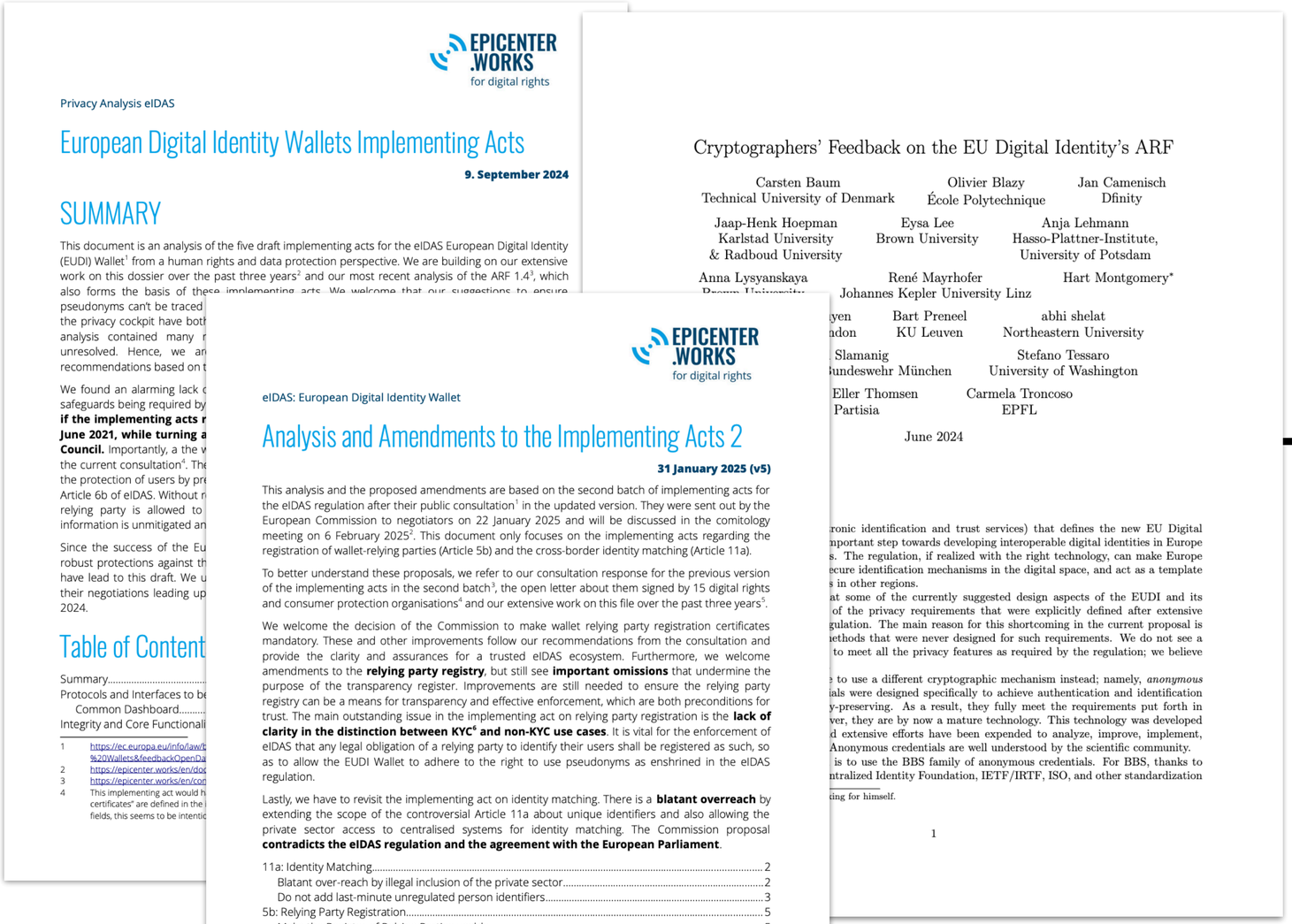
ronic identification and trust services) that defines the new EU Digital
important step towards developing interoperable digital identities in Europe
s. The regulation, if realized with the right technology, can make Europe
ecure identification mechanisms in the digital space, and act as a template
s in other regions.
at some of the currently suggested design aspects of the EUDI and its
of the privacy requirements that were explicitly defined after extensive
gulation. The main reason for this shortcoming in the current proposal is
ethods that were never designed for such requirements. We do not see a
to meet all the privacy features as required by the regulation; we believe

to use a different cryptographic mechanism instead; namely, *anonymous*
ials were designed specifically to achieve authentication and identification
y-preserving. As a result, they fully meet the requirements put forth in
ver, they are by now a mature technology. This technology was developed
d extensive efforts have been expended to analyze, improve, implement,
Anonymous credentials are well understood by the scientific community.
is to use the BBS family of anonymous credentials. For BBS, thanks to
ntralized Identity Foundation, IETF/IRTF, ISO, and other standardization

ting for himself.

1

Privacy and the EUDIF



EPICENTER .WORKS for digital rights

Privacy Analysis eIDAS

European Digital Identity Wallets Implementing Acts

9. September 2024

SUMMARY

This document is an analysis of the five draft implementing acts for the eIDAS European Digital Identity (EUDI) Wallet[1] from a human rights and data protection perspective. We are building on our extensive work on this dossier over the past three years[2] and our most recent analysis of the ARF 1.4[3], which also forms the basis of these implementing acts. We welcome that our suggestions to ensure pseudonyms can't be traced
the privacy cockpit have both
analysis contained many
unresolved. Hence, we ar
recommendations based on t

We found an alarming lack
safeguards being required by
if the implementing acts
June 2021, while turning a
Council. Importantly, a the
the current consultation[4]. The
the protection of users by pr
Article 6b of eIDAS. Without
relying party is allowed to
information is unmitigated an

Since the success of the Eu
robust protections against th
have lead to this draft. We u
their negotiations leading up
2024.

Table of Content

Summary.....
Protocols and Interfaces to b
Common Dashboard.....
Integrity and Core Functionali

1 https://ec.europa.eu/info/law/b
9%20Wallets&feedbackOpenDa
2 https://epicenter.works/en/do
3 https://epicenter.works/en/co
4 This implementing act would h
certificates" are defined in the
fields, this seems to be intentic

Cryptographers' Feedback on the EU Digital Identity's ARF

Carsten Baum, Technical University of Denmark
Olivier Blazy, École Polytechnique
Jan Camenisch, Dfinity
Jaap-Henk Hoepman, Karlstad University & Radboud University
Eysa Lee, Brown University
Anja Lehmann, Hasso-Plattner-Institute, University of Potsdam
Anna Lysyanskaya, Brown University
René Mayrhofer, Johannes Kepler University Linz
Hart Montgomery*
yen, ndon
Bart Preneel, KU Leuven
abhi shelat, Northeastern University
Slamanig, undeswehr München
Stefano Tessaro, University of Washington
Eller Thomsen, Partisia
Carmela Troncoso, EPFL

June 2024

ronic identification and trust services) that defines the new EU Digital
important step towards developing interoperable digital identities in Europe
s. The regulation, if realized with the right technology, can make Europe
ecure identification mechanisms in the digital space, and act as a template
s in other regions.
at some of the currently suggested design aspects of the EUDI and its
of the privacy requirements that were explicitly defined after extensive
gulation. The main reason for this shortcoming in the current proposal is
ethods that were never designed for such requirements. We do not see a
to meet all the privacy features as required by the regulation; we believe

to use a different cryptographic mechanism instead; namely, *anonymous*
ials were designed specifically to achieve authentication and identification
y-preserving. As a result, they fully meet the requirements put forth in
ver, they are by now a mature technology. This technology was developed
d extensive efforts have been expended to analyze, improve, implement,
Anonymous credentials are well understood by the scientific community.
is to use the BBS family of anonymous credentials. For BBS, thanks to
ntralized Identity Foundation, IETF/IRTF, ISO, and other standardization

ding for himself.

1

EPICENTER .WORKS for digital rights

eIDAS: European Digital Identity Wallet

Analysis and Amendments to the Implementing Acts 2

31 January 2025 (v5)

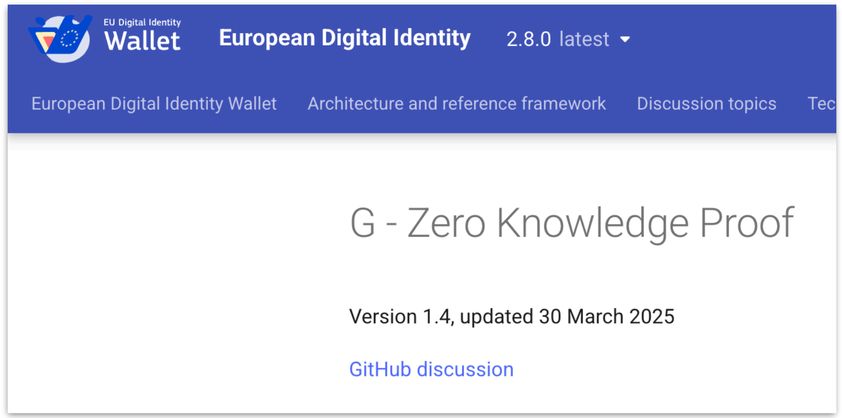
This analysis and the proposed amendments are based on the second batch of implementing acts for the eIDAS regulation after their public consultation[1] in the updated version. They were sent out by the European Commission to negotiators on 22 January 2025 and will be discussed in the comitology meeting on 6 February 2025[2]. This document only focuses on the implementing acts regarding the registration of wallet-relying parties (Article 5b) and the cross-border identity matching (Article 11a).

To better understand these proposals, we refer to our consultation response for the previous version of the implementing acts in the second batch[3], the open letter about them signed by 15 digital rights and consumer protection organisations[4] and our extensive work on this file over the past three years[5].

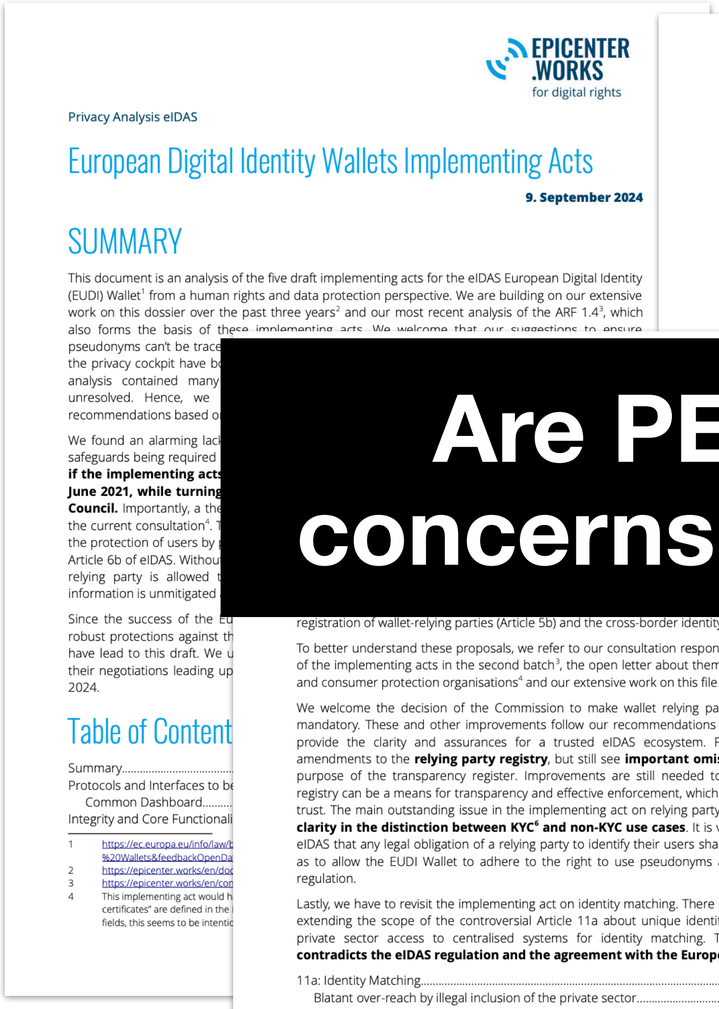
We welcome the decision of the Commission to make wallet relying party registration certificates mandatory. These and other improvements follow our recommendations from the consultation and provide the clarity and assurances for a trusted eIDAS ecosystem. Furthermore, we welcome amendments to the **relying party registry**, but still see **important omissions** that undermine the purpose of the transparency register. Improvements are still needed to ensure the relying party registry can be a means for transparency and effective enforcement, which are both preconditions for trust. The main outstanding issue in the implementing act on relying party registration is the **lack of clarity in the distinction between KYC[6] and non-KYC use cases**. It is vital for the enforcement of eIDAS that any legal obligation of a relying party to identify their users shall be registered as such, so as to allow the EUDI Wallet to adhere to the right to use pseudonyms as enshrined in the eIDAS regulation.

Lastly, we have to revisit the implementing act on identity matching. There is a **blatant overreach** by extending the scope of the controversial Article 11a about unique identifiers and also allowing the private sector access to centralised systems for identity matching. The Commission proposal **contradicts the eIDAS regulation and the agreement with the European Parliament**.

11a: Identity Matching.....2
Blatant over-reach by illegal inclusion of the private sector.....2
Do not add last-minute unregulated person identifiers.....3
5b: Relying Party Registration.....5
Make the Register of Relying Parties usable.....5
Clearly distinguish KYC from non-KYC use cases in their registration.....6
Protection against illegal information requests.....8
Suspension based on proportional harms.....8
Record keeping.....8
Handling of Contact Information.....8



Privacy and the EUDIF



EPICENTER .WORKS for digital rights

Privacy Analysis eIDAS

European Digital Identity Wallets Implementing Acts

9. September 2024

SUMMARY

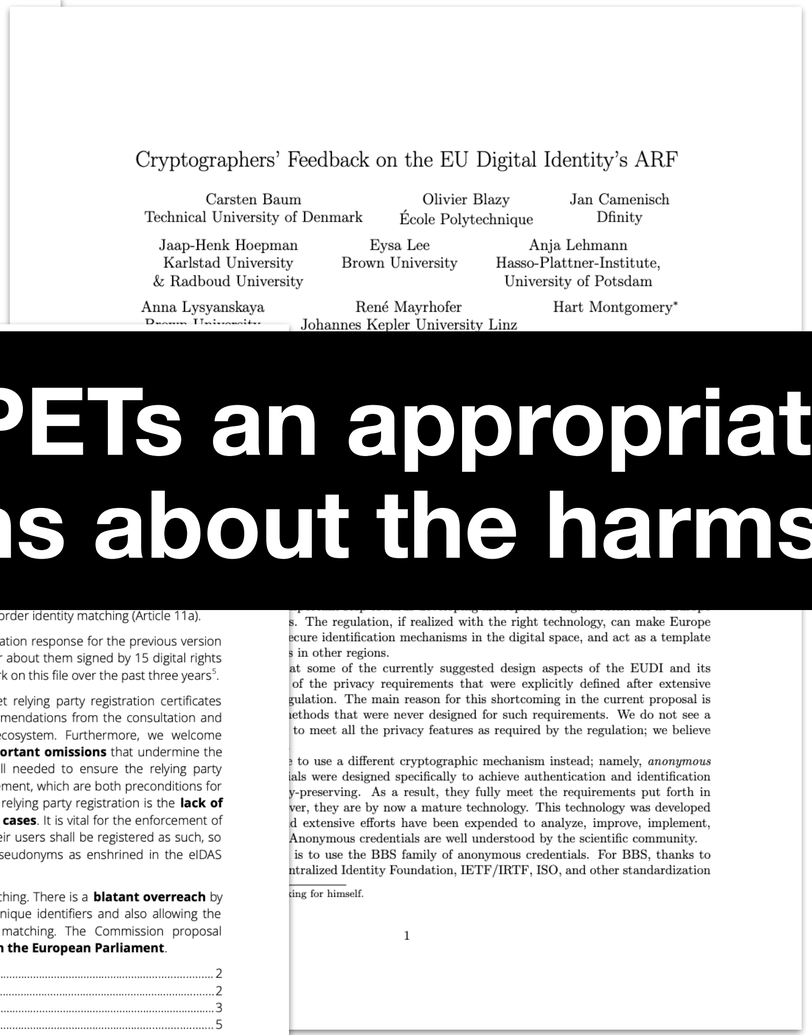
This document is an analysis of the five draft implementing acts for the eIDAS European Digital Identity (EUDI) Wallet[1] from a human rights and data protection perspective. We are building on our extensive work on this dossier over the past three years[2] and our most recent analysis of the ARF 1.4[3], which also forms the basis of these implementing acts. We welcome that our suggestions to ensure pseudonyms can't be trace… the privacy cockpit have b… analysis contained many … unresolved. Hence, we … recommendations based o…

We found an alarming lac… safeguards being required … **if the implementing act… June 2021, while turning … Council.** Importantly, a the… the current consultation[4]. T… the protection of users by … Article 6b of eIDAS. Withou… relying party is allowed … information is unmitigated …

Since the success of the EU… robust protections against th… have lead to this draft. We u… their negotiations leading up… 2024.

Table of Content…

1 https://ec.europa.eu/info/law/b… 9%20Wallets&feedbackOpenDa…
2 https://epicenter.works/en/do…
3 https://epicenter.works/en/con…
4 This implementing act would h… certificates" are defined in the … fields, this seems to be intentic…



Cryptographers' Feedback on the EU Digital Identity's ARF

Carsten Baum, Technical University of Denmark
Olivier Blazy, École Polytechnique
Jan Camenisch, Dfinity
Jaap-Henk Hoepman, Karlstad University & Radboud University
Eysa Lee, Brown University
Anja Lehmann, Hasso-Plattner-Institute, University of Potsdam
Anna Lysyanskaya, Brown University
René Mayrhofer, Johannes Kepler University Linz
Hart Montgomery*

registration of wallet-relying parties (Article 5b) and the cross-border identity matching (Article 11a).

To better understand these proposals, we refer to our consultation response for the previous version of the implementing acts in the second batch[3], the open letter about them signed by 15 digital rights and consumer protection organisations[4] and our extensive work on this file over the past three years[5].

We welcome the decision of the Commission to make wallet relying party registration certificates mandatory. These and other improvements follow our recommendations from the consultation and provide the clarity and assurances for a trusted eIDAS ecosystem. Furthermore, we welcome amendments to the **relying party registry**, but still see **important omissions** that undermine the purpose of the transparency register. Improvements are still needed to ensure the relying party registry can be a means for transparency and effective enforcement, which are both preconditions for trust. The main outstanding issue in the implementing act on relying party registration is the **lack of clarity in the distinction between KYC[6] and non-KYC use cases**. It is vital for the enforcement of eIDAS that any legal obligation of a relying party to identify their users shall be registered as such, so as to allow the EUDI Wallet to adhere to the right to use pseudonyms as enshrined in the eIDAS regulation.

Lastly, we have to revisit the implementing act on identity matching. There is a **blatant overreach** by extending the scope of the controversial Article 11a about unique identifiers and also allowing the private sector access to centralised systems for identity matching. The Commission proposal **contradicts the eIDAS regulation and the agreement with the European Parliament**.

…s. The regulation, if realized with the right technology, can make Europe …ecure identification mechanisms in the digital space, and act as a template …s in other regions.
…at some of the currently suggested design aspects of the EUDI and its … of the privacy requirements that were explicitly defined after extensive …gulation. The main reason for this shortcoming in the current proposal is …ethods that were never designed for such requirements. We do not see a … to meet all the privacy features as required by the regulation; we believe …

… to use a different cryptographic mechanism instead; namely, *anonymous* …ials were designed specifically to achieve authentication and identification …y-preserving. As a result, they fully meet the requirements put forth in …ver, they are by now a mature technology. This technology was developed …d extensive efforts have been expended to analyze, improve, implement, … Anonymous credentials are well understood by the scientific community. … is to use the BBS family of anonymous credentials. For BBS, thanks to …ntralized Identity Foundation, IETF/IRTF, ISO, and other standardization …

…ting for himself.

1

4

Security & privacy
at the heart of EU Digital Identity Wallets

You are in control

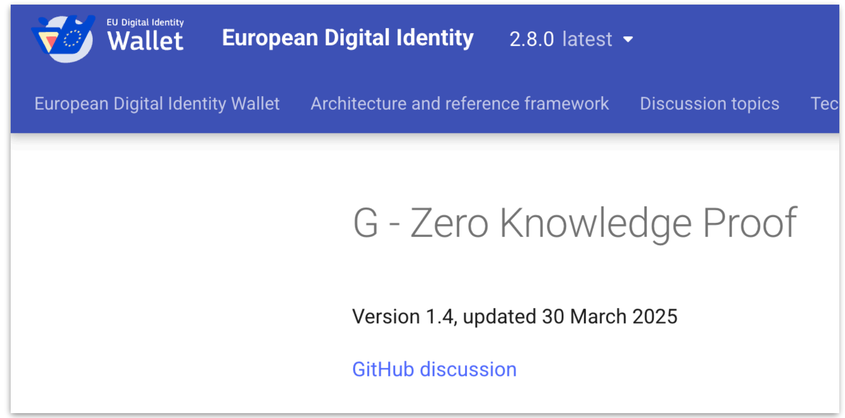
You will have complete control of what data you disclose from your wallet.

Your data will be stored locally on your wallet app. It stays in your control.

Stored in your Wallet

…not be …data, …d

Are PETs an appropriate answer to concerns about the harms of the EUDIF?



EU Digital Identity Wallet | European Digital Identity | 2.8.0 latest

European Digital Identity Wallet | Architecture and reference framework | Discussion topics | Tec…

G - Zero Knowledge Proof

Version 1.4, updated 30 March 2025

GitHub discussion

Modeling the EUDIF

Modeling the EUDIF

- Consolidated EU regulation of the base regulation No 910/2014 (eIDAS) amended by Regulation 2024/1183;
- Implementing EU Reg. 2024/2977: personal identification data and electronic attestations of attributes;
- Implementing EU Reg. 2024/2979: integrity and core functionalities of European Digital Identity Wallets;
- Implementing EU Reg. 2024/2980: trust framework;
- Implementing EU Reg. 2024/2981: certification of European Digital Identity Wallets;
- Implementing EU Reg. 2024/2982: protocols and interfaces to be supported;
- Implementing EU Reg. 2025/846: cross-border identity matching for natural persons;
- Implementing EU Reg. 2025/847: reactions to security breaches of wallets;
- Implementing EU Reg. 2025/848: registration of wallet-relying parties;
- Implementing EU Reg. 2025/849: submitting information to the Commission for the list of certified wallets;
- Implementing EU Reg. 2025/1566: verification of identity and attributes when issuing qualified certificates or qualified electronic attestations.
- Implementing EU Reg. 2025/1567: management of remote qualified electronic signature/seal creation devices;
- Implementing EU Reg. 2025/1568: procedural arrangements for peer reviews of eID schemes;
- Implementing EU Reg. 2025/1569: electronic attestations of attributes (EAA/QEAA specifications);
- Implementing EU Reg. 2025/1570: notifications of certified qualified signature/seal creation devices;
- Implementing EU Reg. 2025/1572: initiation of qualified trust services (notification of intention);
- Implementing EU Reg. 2025/1571: annual reports from supervisory bodies;
- Supporting governance/technical documentation (EUDI Architecture and Reference Framework (ARF);

+ Adjacent EU laws (e.g., GDPR, accreditation, cybersecurity).

Modeling the EUDIF

Parties. Issuers $\mathcal{I}_{\text{iid}}$, Users $\mathcal{U}_{\text{uid}}$, Relying Parties $\mathcal{R}_{\text{rid}}$.

State. The functionality maintains the following global state:

- Wallet : uid, attr_descr $\mapsto$ (iid, attr) mapping a user ID and an attribute description to an issuer ID and an issued attribute.
- Policies : uid, iid $\mapsto$ policy mapping uid and iid to a disclosure policy for the attributes issued by $\mathcal{I}_{\text{iid}}$ to $\mathcal{U}_{\text{uid}}$.
- RPInfo : rid $\mapsto$ rp_info mapping an RP ID rid to registered RP information rp_info (name, address, etc.) for $\mathcal{R}_{\text{rid}}$;
- RPReg : rid $\mapsto$ [(intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$)] mapping rid to a list of registered intended uses, function over attributes, and attribute descriptions for a relying party $\mathcal{R}_{\text{rid}}$.

RP registration. On input (RegisterRPInfo, rp_info) from $\mathcal{R}_{\text{rid}}$: set RPInfo[rid] := rp_info.

RP usage registration. On input (RegisterRP, intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$) from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; append (intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$) to RPReg[rid].

RP usage deregistration. On input (DeregisterRP, intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$) from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; remove (intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$) from RPReg[rid]. RPs can update their registration by deregistering and re-registering.

RP deregistration. On input DeregisterRPInfo from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; set RPInfo[rid] := $\perp$.

Issuance. On input (Issue, iid, $(\text{attr\_descr}_i)_{i\in[m]}$) from a user $\mathcal{U}_{\text{uid}}$:

(1) Get the output of the issuance oracle $o \leftarrow O_{\text{Issue}}(\text{iid}, \text{uid}, (\text{attr\_descr}_i)_{i\in[m]})$. If $o = \perp$, output $\perp$ to issuer and user. Otherwise, parse o as attributes $(\text{attr})_{i\in[m]}$ and a disclosure policy policy.

(2) Set Policies[uid][iid] := policy. For $i \in [m]$: Set Wallet[uid][attr_descr$_i$] := (iid, attr$_i$). Send $((\text{attr\_descr}_i)_{i\in[m]}, \text{policy})$ to the user $\mathcal{U}_{\text{uid}}$ and issuer $\mathcal{I}_{\text{iid}}$.

Presentation. On input (Request, rid, service) from $\mathcal{U}_{\text{uid}}$: Require RPInfo[rid] to be set.

(1) Send (Request, service) to $\mathcal{R}_{\text{rid}}$, and receive $(\text{aux}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$.

(2) If aux = (rp_info, intended_use), store these values; Otherwise, if aux = $i_{\text{intended\_use}}$, set rp_info := RPInfo[rid] and intended_use := RPReg[rid][$i_{\text{intended\_use}}$]; Otherwise, abort. Send $(\text{rp\_info}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]}))$ to $\mathcal{U}_{\text{uid}}$.

(3) Receive $b_{\text{check}} \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$ that indicates if the user optionally wants to check the RP's registration. If $b_{\text{check}} = 1$:
Set $b_{\text{registration}} := 1$ if $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ is stored in RPReg[rid], and $b_{\text{registered}} := 0$ otherwise.
Send $b_{\text{registration}}$ to $\mathcal{U}_{\text{uid}}$, and receive $\text{consent}_1 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If $\text{consent}_1 = 0$, output $\perp$ to rid. Otherwise, continue.

(4) For $i \in [n]$: retrieve $(\text{iid}_i, \text{attr}_i) := \text{Wallet}[\text{uid}][\text{attr\_descr}_i]$ and $\text{policy}_i := \text{Policies}[\text{uid}][\text{iid}_i]$.
Send $b_{\text{policy}} := \bigwedge_{j\in[n]} \text{policy}_j(\text{rid}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ to $\mathcal{U}_{\text{uid}}$ and receive $\text{consent}_2 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If $\text{consent}_2 = 0$, output $\perp$ to $\mathcal{R}_{\text{rid}}$. Otherwise, compute the output $v := f(\text{attr}_1, \ldots, \text{attr}_n)$ and send v to $\mathcal{R}_{\text{rid}}$.

Modeling the EUDIF

Parties. Issuers $\mathcal{I}_{\text{iid}}$, Users $\mathcal{U}_{\text{uid}}$, Relying Parties $\mathcal{R}_{\text{rid}}$.
State. The functionality maintains the following global state:

- Wallet : uid, attr_descr $\mapsto$ (iid, attr) mapping a user ID and an attribute description to an issuer ID and an issued attribute.
- Policies : uid, iid $\mapsto$ policy mapping uid and iid to a disclosure policy for the attributes issued by $\mathcal{I}_{\text{iid}}$ to $\mathcal{U}_{\text{uid}}$.
- RPInfo : rid $\mapsto$ rp_info mapping an RP ID rid to registered RP information rp_info (name, address, etc.) for $\mathcal{R}_{\text{rid}}$;
- RPReg : rid $\mapsto$ $[(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})]$ mapping rid to a list of registered intended uses, function over attributes, and attribute descriptions for a relying party $\mathcal{R}_{\text{rid}}$.

RP registration. On input (RegisterRPInfo, rp_info) from $\mathcal{R}_{\text{rid}}$: set RPInfo[rid] := rp_info.
RP usage registration. On input $(\text{RegisterRP}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; append $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ to RPReg[rid].
RP usage deregistration. On input $(\text{DeregisterRP}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; remove $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from RPReg[rid]. RPs can update their registration by deregistering and re-registering.
RP deregistration. On input DeregisterRPInfo from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; set RPInfo[rid] := $\perp$.
Issuance. On input $(\text{Issue}, \text{iid}, (\text{attr\_descr}_i)_{i\in[m]})$ from a user $\mathcal{U}_{\text{uid}}$:

(1) Get the output of the issuance oracle $o \leftarrow O_{\text{Issue}}(\text{iid}, \text{uid}, (\text{attr\_descr}_i)_{i\in[m]})$. If $o = \perp$, output $\perp$ to issuer and user. Otherwise, parse o as attributes $(\text{attr})_{i\in[m]}$ and a disclosure policy policy.
(2) Set Policies[uid][iid] := policy. For $i \in [m]$: Set Wallet[uid][attr_descr$_i$] := $(\text{iid}, \text{attr}_i)$. Send $((\text{attr\_descr}_i)_{i\in[m]}, \text{policy})$ to the user $\mathcal{U}_{\text{uid}}$ and issuer $\mathcal{I}_{\text{iid}}$.

Presentation. On input (Request, rid, service) from $\mathcal{U}_{\text{uid}}$: Require RPInfo[rid] to be set.

(1) Send (Request, service) to $\mathcal{R}_{\text{rid}}$, and receive $(\text{aux}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$.
(2) If aux = $(\text{rp\_info}, \text{intended\_use})$, store these values; Otherwise, if aux = $i_{\text{intended\_use}}$, set rp_info := RPInfo[rid] and intended_use := RPReg[rid][$i_{\text{intended\_use}}$]; Otherwise, abort. Send $(\text{rp\_info}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]}))$ to $\mathcal{U}_{\text{uid}}$.
(3) Receive $b_{\text{check}} \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$ that indicates if the user optionally wants to check the RP's registration. If $b_{\text{check}} = 1$:
    Set $b_{\text{registration}} := 1$ if $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ is stored in RPReg[rid], and $b_{\text{registered}} := 0$ otherwise.
    Send $b_{\text{registration}}$ to $\mathcal{U}_{\text{uid}}$, and receive $\text{consent}_1 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If $\text{consent}_1 = 0$, output $\perp$ to rid. Otherwise, continue.
(4) For $i \in [n]$: retrieve $(\text{iid}_i, \text{attr}_i) := \text{Wallet}[\text{uid}][\text{attr\_descr}_i]$ and $\text{policy}_i := \text{Policies}[\text{uid}][\text{iid}_i]$.
    Send $b_{\text{policy}} := \bigwedge_{j\in[n]} \text{policy}_j(\text{rid}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ to $\mathcal{U}_{\text{uid}}$ and receive $\text{consent}_2 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If $\text{consent}_2 = 0$, output $\perp$ to $\mathcal{R}_{\text{rid}}$. Otherwise, compute the output $v := f(\text{attr}_1, \ldots, \text{attr}_n)$ and send v to $\mathcal{R}_{\text{rid}}$.

Modeling the EUDIF

Parties. Issuers $\mathcal{I}_{\text{iid}}$, Users $\mathcal{U}_{\text{uid}}$, Relying Parties $\mathcal{R}_{\text{rid}}$.
State. The functionality maintains the following global state:

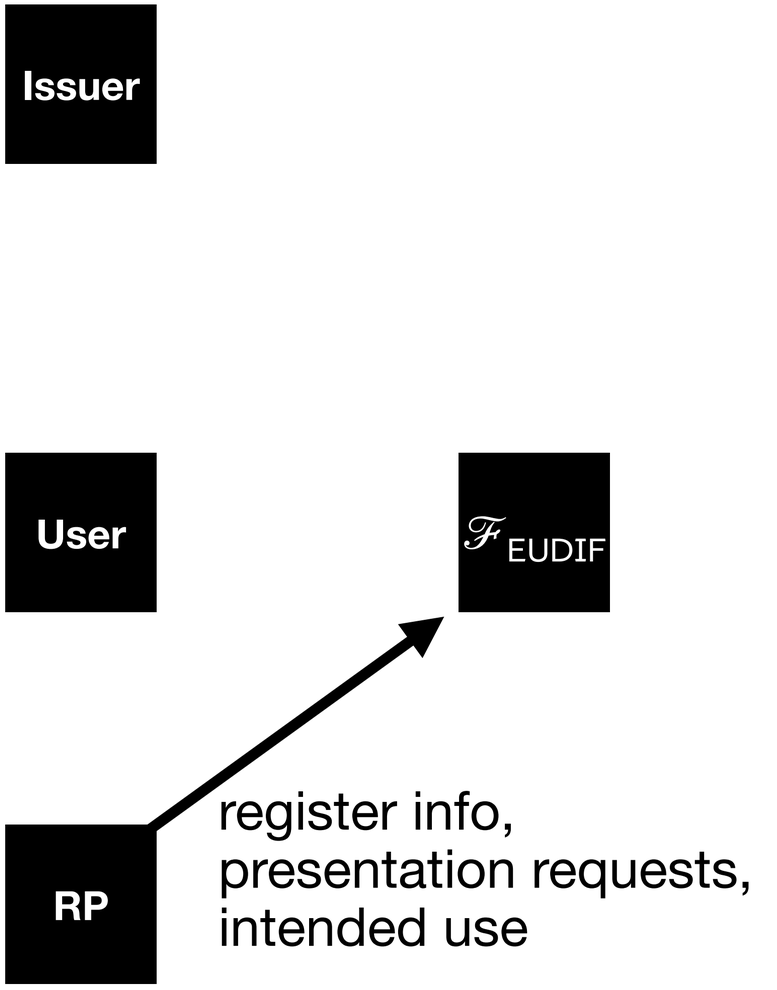
- Wallet : uid, attr_descr $\mapsto$ (iid, attr) mapping a user ID and an attribute description to an issuer ID and an issued attribute.
- Policies : uid, iid $\mapsto$ policy mapping uid and iid to a disclosure policy for the attributes issued by $\mathcal{I}_{\text{iid}}$ to $\mathcal{U}_{\text{uid}}$.
- RPInfo : rid $\mapsto$ rp_info mapping an RP ID rid to registered RP information rp_info (name, address, etc.) for $\mathcal{R}_{\text{rid}}$;
- RPReg : rid $\mapsto$ $[(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})]$ mapping rid to a list of registered intended uses, function over attributes, and attribute descriptions for a relying party $\mathcal{R}_{\text{rid}}$.

RP registration. On input (RegisterRPInfo, rp_info) from $\mathcal{R}_{\text{rid}}$: set RPInfo[rid] := rp_info.
RP usage registration. On input $(\text{RegisterRP}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; append $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ to RPReg[rid].
RP usage deregistration. On input $(\text{DeregisterRP}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; remove $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from RPReg[rid]. RPs can update their registration by deregistering and re-registering.
RP deregistration. On input DeregisterRPInfo from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; set RPInfo[rid] := $\perp$.
Issuance. On input $(\text{Issue}, \text{iid}, (\text{attr\_descr}_i)_{i\in[m]})$ from a user $\mathcal{U}_{\text{uid}}$:

(1) Get the output of the issuance oracle $o \leftarrow O_{\text{Issue}}(\text{iid}, \text{uid}, (\text{attr\_descr}_i)_{i\in[m]})$. If $o = \perp$, output $\perp$ to issuer and user. Otherwise, parse o as attributes $(\text{attr})_{i\in[m]}$ and a disclosure policy policy.
(2) Set Policies[uid][iid] := policy. For $i \in [m]$: Set Wallet[uid][attr_descr$_i$] := $(\text{iid}, \text{attr}_i)$. Send $((\text{attr\_descr}_i)_{i\in[m]}, \text{policy})$ to the user $\mathcal{U}_{\text{uid}}$ and issuer $\mathcal{I}_{\text{iid}}$.

Presentation. On input (Request, rid, service) from $\mathcal{U}_{\text{uid}}$: Require RPInfo[rid] to be set.

(1) Send (Request, service) to $\mathcal{R}_{\text{rid}}$, and receive $(\text{aux}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$.
(2) If aux = $(\text{rp\_info}, \text{intended\_use})$, store these values; Otherwise, if aux = $i_{\text{intended\_use}}$, set rp_info := RPInfo[rid] and intended_use := RPReg[rid][$i_{\text{intended\_use}}$]; Otherwise, abort. Send $(\text{rp\_info}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]}))$ to $\mathcal{U}_{\text{uid}}$.
(3) Receive $b_{\text{check}} \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$ that indicates if the user optionally wants to check the RP's registration. If $b_{\text{check}} = 1$:
    Set $b_{\text{registration}} := 1$ if $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ is stored in RPReg[rid], and $b_{\text{registered}} := 0$ otherwise.
    Send $b_{\text{registration}}$ to $\mathcal{U}_{\text{uid}}$, and receive $\text{consent}_1 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If $\text{consent}_1 = 0$, output $\perp$ to rid. Otherwise, continue.
(4) For $i \in [n]$: retrieve $(\text{iid}_i, \text{attr}_i) := \text{Wallet}[\text{uid}][\text{attr\_descr}_i]$ and $\text{policy}_i := \text{Policies}[\text{uid}][\text{iid}_i]$.
    Send $b_{\text{policy}} := \bigwedge_{j\in[n]} \text{policy}_j(\text{rid}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ to $\mathcal{U}_{\text{uid}}$ and receive $\text{consent}_2 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If $\text{consent}_2 = 0$, output $\perp$ to $\mathcal{R}_{\text{rid}}$. Otherwise, compute the output $v := f(\text{attr}_1, \ldots, \text{attr}_n)$ and send v to $\mathcal{R}_{\text{rid}}$.



Modeling the EUDIF

Parties. Issuers $\mathcal{I}_{\text{iid}}$, Users $\mathcal{U}_{\text{uid}}$, Relying Parties $\mathcal{R}_{\text{rid}}$.

State. The functionality maintains the following global state:

- Wallet : uid, attr_descr $\mapsto$ (iid, attr) mapping a user ID and an attribute description to an issuer ID and an issued attribute.
- Policies : uid, iid $\mapsto$ policy mapping uid and iid to a disclosure policy for the attributes issued by $\mathcal{I}_{\text{iid}}$ to $\mathcal{U}_{\text{uid}}$.
- RPInfo : rid $\mapsto$ rp_info mapping an RP ID rid to registered RP information rp_info (name, address, etc.) for $\mathcal{R}_{\text{rid}}$;
- RPReg : rid $\mapsto$ $[(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})]$ mapping rid to a list of registered intended uses, function over attributes, and attribute descriptions for a relying party $\mathcal{R}_{\text{rid}}$.

RP registration. On input (RegisterRPInfo, rp_info) from $\mathcal{R}_{\text{rid}}$: set RPInfo[rid] := rp_info.

RP usage registration. On input (RegisterRP, intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$) from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; append (intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$) to RPReg[rid].

RP usage deregistration. On input (DeregisterRP, intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$) from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; remove (intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$) from RPReg[rid]. RPs can update their registration by deregistering and re-registering.

RP deregistration. On input DeregisterRPInfo from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; set RPInfo[rid] := $\perp$.

Issuance. On input $(\text{Issue}, \text{iid}, (\text{attr\_descr}_i)_{i\in[m]})$ from a user $\mathcal{U}_{\text{uid}}$:

(1) Get the output of the issuance oracle $o \leftarrow O_{\text{Issue}}(\text{iid}, \text{uid}, (\text{attr\_descr}_i)_{i\in[m]})$. If $o = \perp$, output $\perp$ to issuer and user. Otherwise, parse o as attributes $(\text{attr})_{i\in[m]}$ and a disclosure policy policy.

(2) Set Policies[uid][iid] := policy. For $i \in [m]$: Set Wallet[uid][attr_descr_i] := $(\text{iid}, \text{attr}_i)$. Send $((\text{attr\_descr}_i)_{i\in[m]}, \text{policy})$ to the user $\mathcal{U}_{\text{uid}}$ and issuer $\mathcal{I}_{\text{iid}}$.

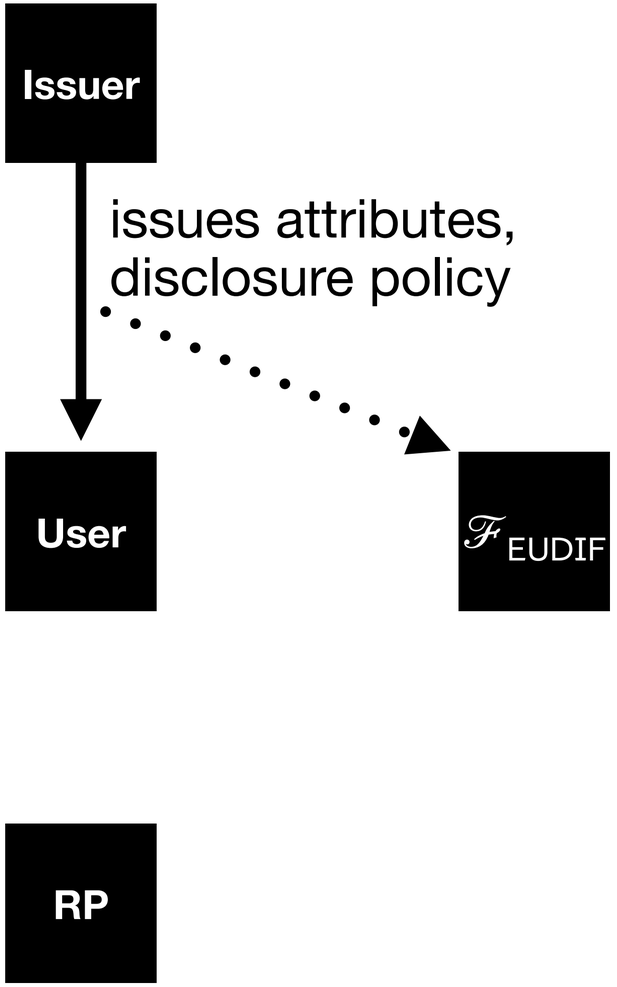
Presentation. On input (Request, rid, service) from $\mathcal{U}_{\text{uid}}$: Require RPInfo[rid] to be set.

(1) Send (Request, service) to $\mathcal{R}_{\text{rid}}$, and receive $(\text{aux}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$.

(2) If aux = (rp_info, intended_use), store these values; Otherwise, if aux = $i_{\text{intended\_use}}$, set rp_info := RPInfo[rid] and intended_use := RPReg[rid][$i_{\text{intended\_use}}$]; Otherwise, abort. Send $(\text{rp\_info}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]}))$ to $\mathcal{U}_{\text{uid}}$.

(3) Receive $b_{\text{check}} \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$ that indicates if the user optionally wants to check the RP's registration. If $b_{\text{check}} = 1$:
Set $b_{\text{registration}} := 1$ if $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ is stored in RPReg[rid], and $b_{\text{registered}} := 0$ otherwise.
Send $b_{\text{registration}}$ to $\mathcal{U}_{\text{uid}}$, and receive $\text{consent}_1 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If $\text{consent}_1 = 0$, output $\perp$ to rid. Otherwise, continue.

(4) For $i \in [n]$: retrieve $(\text{iid}_i, \text{attr}_i) := \text{Wallet}[\text{uid}][\text{attr\_descr}_i]$ and $\text{policy}_i := \text{Policies}[\text{uid}][\text{iid}_i]$.
Send $b_{\text{policy}} := \bigwedge_{j\in[n]} \text{policy}_j(\text{rid}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ to $\mathcal{U}_{\text{uid}}$ and receive $\text{consent}_2 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If $\text{consent}_2 = 0$, output $\perp$ to $\mathcal{R}_{\text{rid}}$. Otherwise, compute the output $v := f(\text{attr}_1, \ldots, \text{attr}_n)$ and send v to $\mathcal{R}_{\text{rid}}$.



Modeling the EUDIF

Parties. Issuers $\mathcal{I}_{\text{iid}}$, Users $\mathcal{U}_{\text{uid}}$, Relying Parties $\mathcal{R}_{\text{rid}}$.

State. The functionality maintains the following global state:

- Wallet : uid, attr_descr $\mapsto$ (iid, attr) mapping a user ID and an attribute description to an issuer ID and an issued attribute.
- Policies : uid, iid $\mapsto$ policy mapping uid and iid to a disclosure policy for the attributes issued by $\mathcal{I}_{\text{iid}}$ to $\mathcal{U}_{\text{uid}}$.
- RPInfo : rid $\mapsto$ rp_info mapping an RP ID rid to registered RP information rp_info (name, address, etc.) for $\mathcal{R}_{\text{rid}}$;
- RPReg : rid $\mapsto$ $[(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})]$ mapping rid to a list of registered intended uses, function over attributes, and attribute descriptions for a relying party $\mathcal{R}_{\text{rid}}$.

RP registration. On input (RegisterRPInfo, rp_info) from $\mathcal{R}_{\text{rid}}$: set RPInfo[rid] := rp_info.

RP usage registration. On input $(\text{RegisterRP}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; append $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ to RPReg[rid].

RP usage deregistration. On input $(\text{DeregisterRP}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; remove $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from RPReg[rid]. RPs can update their registration by deregistering and re-registering.

RP deregistration. On input DeregisterRPInfo from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; set RPInfo[rid] := $\perp$.

Issuance. On input $(\text{Issue}, \text{iid}, (\text{attr\_descr}_i)_{i\in[m]})$ from a user $\mathcal{U}_{\text{uid}}$:

(1) Get the output of the issuance oracle $o \leftarrow O_{\text{Issue}}(\text{iid}, \text{uid}, (\text{attr\_descr}_i)_{i\in[m]})$. If $o = \perp$, output $\perp$ to issuer and user. Otherwise, parse o as attributes $(\text{attr})_{i\in[m]}$ and a disclosure policy policy.

(2) Set Policies[uid][iid] := policy. For $i \in [m]$: Set Wallet[uid][attr_descr$_i$] := (iid, attr$_i$). Send $((\text{attr\_descr}_i)_{i\in[m]}, \text{policy})$ to the user $\mathcal{U}_{\text{uid}}$ and issuer $\mathcal{I}_{\text{iid}}$.

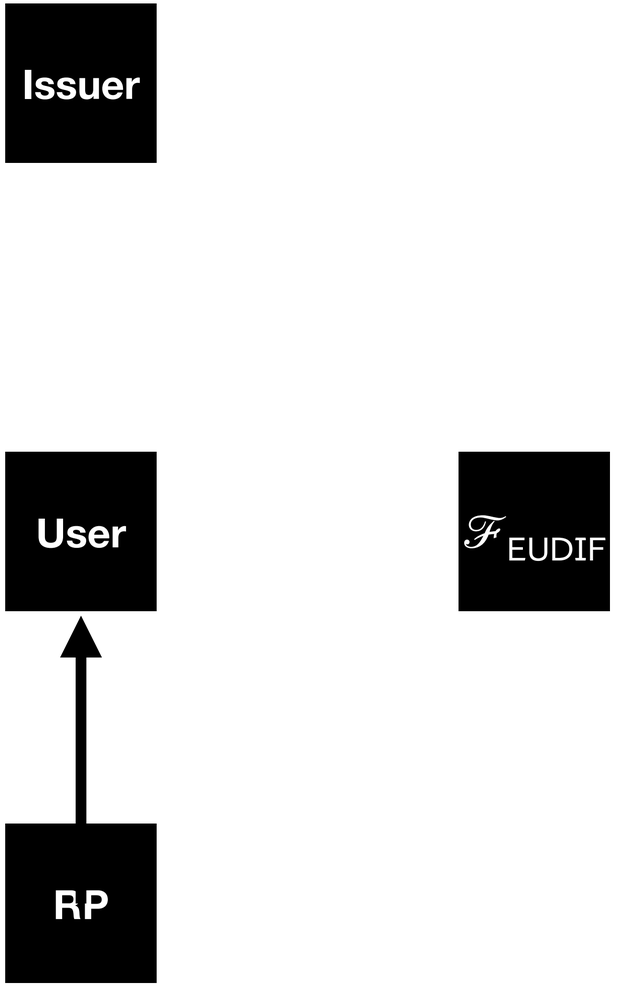
Presentation. On input (Request, rid, service) from $\mathcal{U}_{\text{uid}}$: Require RPInfo[rid] to be set.

(1) Send (Request, service) to $\mathcal{R}_{\text{rid}}$, and receive $(\text{aux}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$.

(2) If aux = (rp_info, intended_use), store these values; Otherwise, if aux = $i_{\text{intended\_use}}$, set rp_info := RPInfo[rid] and intended_use := RPReg[rid][$i_{\text{intended\_use}}$]; Otherwise, abort. Send $(\text{rp\_info}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]}))$ to $\mathcal{U}_{\text{uid}}$.

(3) Receive $b_{\text{check}} \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$ that indicates if the user optionally wants to check the RP's registration. If $b_{\text{check}} = 1$:
Set $b_{\text{registration}} := 1$ if $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ is stored in RPReg[rid], and $b_{\text{registered}} := 0$ otherwise.
Send $b_{\text{registration}}$ to $\mathcal{U}_{\text{uid}}$, and receive consent$_1 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If consent$_1 = 0$, output $\perp$ to rid. Otherwise, continue.

(4) For $i \in [n]$: retrieve $(\text{iid}_i, \text{attr}_i) := \text{Wallet}[\text{uid}][\text{attr\_descr}_i]$ and $\text{policy}_i := \text{Policies}[\text{uid}][\text{iid}_i]$.
Send $b_{\text{policy}} := \bigwedge_{j\in[n]} \text{policy}_j(\text{rid}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ to $\mathcal{U}_{\text{uid}}$ and receive consent$_2 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If consent$_2 = 0$, output $\perp$ to $\mathcal{R}_{\text{rid}}$. Otherwise, compute the output $v := f(\text{attr}_1, \ldots, \text{attr}_n)$ and send v to $\mathcal{R}_{\text{rid}}$.



requests presentation of attributes, shows info, intended use

Modeling the EUDIF

Parties. Issuers $\mathcal{I}_{\text{iid}}$, Users $\mathcal{U}_{\text{uid}}$, Relying Parties $\mathcal{R}_{\text{rid}}$.
State. The functionality maintains the following global state:

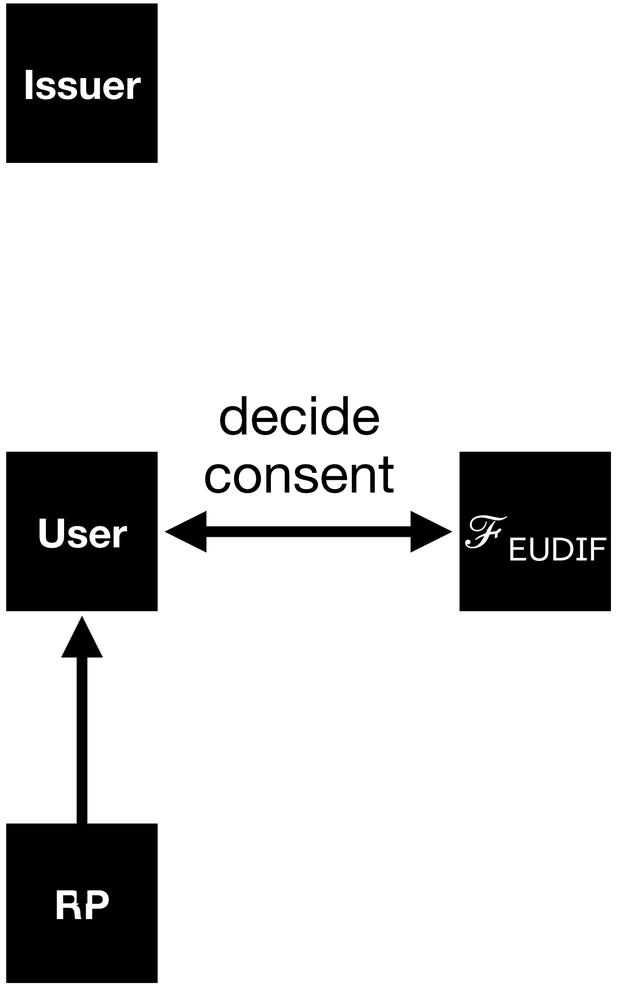
- Wallet : uid, attr_descr $\mapsto$ (iid, attr) mapping a user ID and an attribute description to an issuer ID and an issued attribute.
- Policies : uid, iid $\mapsto$ policy mapping uid and iid to a disclosure policy for the attributes issued by $\mathcal{I}_{\text{iid}}$ to $\mathcal{U}_{\text{uid}}$.
- RPInfo : rid $\mapsto$ rp_info mapping an RP ID rid to registered RP information rp_info (name, address, etc.) for $\mathcal{R}_{\text{rid}}$;
- RPReg : rid $\mapsto$ $[(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})]$ mapping rid to a list of registered intended uses, function over attributes, and attribute descriptions for a relying party $\mathcal{R}_{\text{rid}}$.

RP registration. On input (RegisterRPInfo, rp_info) from $\mathcal{R}_{\text{rid}}$: set RPInfo[rid] := rp_info.
RP usage registration. On input $(\text{RegisterRP}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; append $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ to RPReg[rid].
RP usage deregistration. On input $(\text{DeregisterRP}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; remove $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from RPReg[rid]. RPs can update their registration by deregistering and re-registering.
RP deregistration. On input DeregisterRPInfo from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; set RPInfo[rid] := $\perp$.
Issuance. On input $(\text{Issue}, \text{iid}, (\text{attr\_descr}_i)_{i\in[m]})$ from a user $\mathcal{U}_{\text{uid}}$:

(1) Get the output of the issuance oracle $o \leftarrow O_{\text{Issue}}(\text{iid}, \text{uid}, (\text{attr\_descr}_i)_{i\in[m]})$. If $o = \perp$, output $\perp$ to issuer and user. Otherwise, parse o as attributes $(\text{attr})_{i\in[m]}$ and a disclosure policy policy.
(2) Set Policies[uid][iid] := policy. For $i \in [m]$: Set Wallet[uid][attr_descr$_i$] := (iid, attr$_i$). Send $((\text{attr\_descr}_i)_{i\in[m]}, \text{policy})$ to the user $\mathcal{U}_{\text{uid}}$ and issuer $\mathcal{I}_{\text{iid}}$.

Presentation. On input (Request, rid, service) from $\mathcal{U}_{\text{uid}}$: Require RPInfo[rid] to be set.

(1) Send (Request, service) to $\mathcal{R}_{\text{rid}}$, and receive $(\text{aux}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$.
(2) If aux = (rp_info, intended_use), store these values; Otherwise, if aux = $i_{\text{intended\_use}}$, set rp_info := RPInfo[rid] and intended_use := RPReg[rid][$i_{\text{intended\_use}}$]; Otherwise, abort. Send $(\text{rp\_info}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]}))$ to $\mathcal{U}_{\text{uid}}$.
(3) Receive $b_{\text{check}} \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$ that indicates if the user optionally wants to check the RP's registration. If $b_{\text{check}} = 1$:
    Set $b_{\text{registration}} := 1$ if $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ is stored in RPReg[rid], and $b_{\text{registered}} := 0$ otherwise.
    Send $b_{\text{registration}}$ to $\mathcal{U}_{\text{uid}}$, and receive consent$_1 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If consent$_1 = 0$, output $\perp$ to rid. Otherwise, continue.
(4) For $i \in [n]$: retrieve (iid$_i$, attr$_i$) := Wallet[uid][attr_descr$_i$] and policy$_i$:= Policies[uid][iid$_i$].
    Send $b_{\text{policy}} := \bigwedge_{j\in[n]} \text{policy}_j(\text{rid}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ to $\mathcal{U}_{\text{uid}}$ and receive consent$_2 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If consent$_2 = 0$, output $\perp$ to $\mathcal{R}_{\text{rid}}$. Otherwise, compute the output $v := f(\text{attr}_1, \ldots, \text{attr}_n)$ and send v to $\mathcal{R}_{\text{rid}}$.



requests presentation of attributes, shows info, intended use

Modeling the EUDIF

Parties. Issuers $\mathcal{I}_{\text{iid}}$, Users $\mathcal{U}_{\text{uid}}$, Relying Parties $\mathcal{R}_{\text{rid}}$.

State. The functionality maintains the following global state:

- Wallet : uid, attr_descr $\mapsto$ (iid, attr) mapping a user ID and an attribute description to an issuer ID and an issued attribute.
- Policies : uid, iid $\mapsto$ policy mapping uid and iid to a disclosure policy for the attributes issued by $\mathcal{I}_{\text{iid}}$ to $\mathcal{U}_{\text{uid}}$.
- RPInfo : rid $\mapsto$ rp_info mapping an RP ID rid to registered RP information rp_info (name, address, etc.) for $\mathcal{R}_{\text{rid}}$;
- RPReg : rid $\mapsto$ [(intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$)] mapping rid to a list of registered intended uses, function over attributes, and attribute descriptions for a relying party $\mathcal{R}_{\text{rid}}$.

RP registration. On input (RegisterRPInfo, rp_info) from $\mathcal{R}_{\text{rid}}$: set RPInfo[rid] := rp_info.

RP usage registration. On input (RegisterRP, intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$) from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; append (intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$) to RPReg[rid].

RP usage deregistration. On input (DeregisterRP, intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$) from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; remove (intended_use, f, $(\text{attr\_descr}_i)_{i\in[n]}$) from RPReg[rid]. RPs can update their registration by deregistering and re-registering.

RP deregistration. On input DeregisterRPInfo from $\mathcal{R}_{\text{rid}}$: require RPInfo[rid] to be set; set RPInfo[rid] := $\perp$.

Issuance. On input (Issue, iid, $(\text{attr\_descr}_i)_{i\in[m]}$) from a user $\mathcal{U}_{\text{uid}}$:

(1) Get the output of the issuance oracle $o \leftarrow O_{\text{Issue}}(\text{iid}, \text{uid}, (\text{attr\_descr}_i)_{i\in[m]})$. If $o = \perp$, output $\perp$ to issuer and user. Otherwise, parse o as attributes $(\text{attr})_{i\in[m]}$ and a disclosure policy policy.

(2) Set Policies[uid][iid] := policy. For $i \in [m]$: Set Wallet[uid][attr_descr_i] := $(\text{iid}, \text{attr}_i)$. Send $((\text{attr\_descr}_i)_{i\in[m]}, \text{policy})$ to the user $\mathcal{U}_{\text{uid}}$ and issuer $\mathcal{I}_{\text{iid}}$.

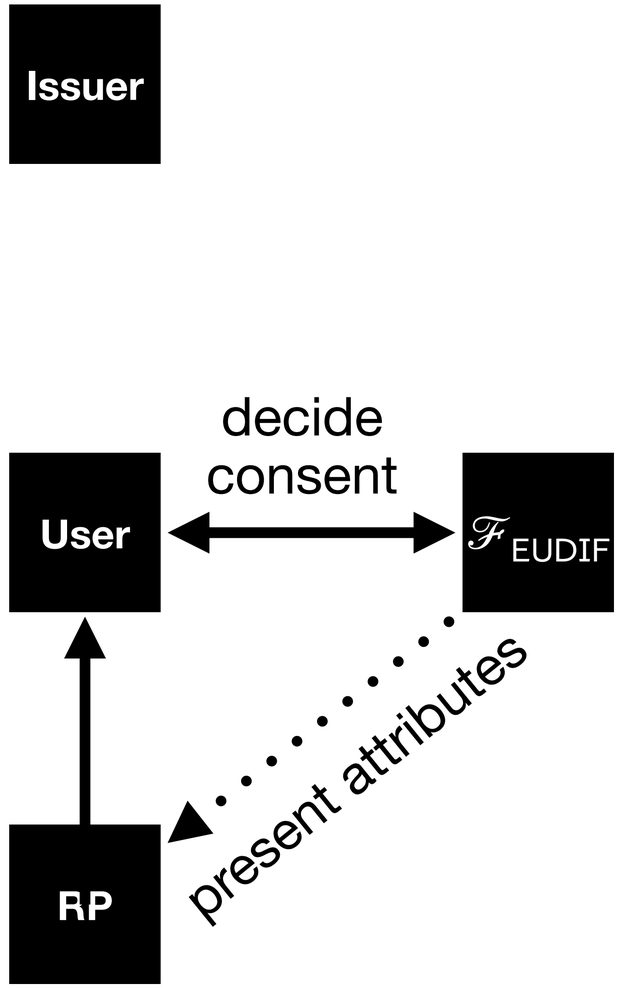
Presentation. On input (Request, rid, service) from $\mathcal{U}_{\text{uid}}$: Require RPInfo[rid] to be set.

(1) Send (Request, service) to $\mathcal{R}_{\text{rid}}$, and receive $(\text{aux}, f, (\text{attr\_descr}_i)_{i\in[n]})$ from $\mathcal{R}_{\text{rid}}$.

(2) If aux = (rp_info, intended_use), store these values; Otherwise, if aux = $i_{\text{intended\_use}}$, set rp_info := RPInfo[rid] and intended_use := RPReg[rid][$i_{\text{intended\_use}}$]; Otherwise, abort. Send $(\text{rp\_info}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]}))$ to $\mathcal{U}_{\text{uid}}$.

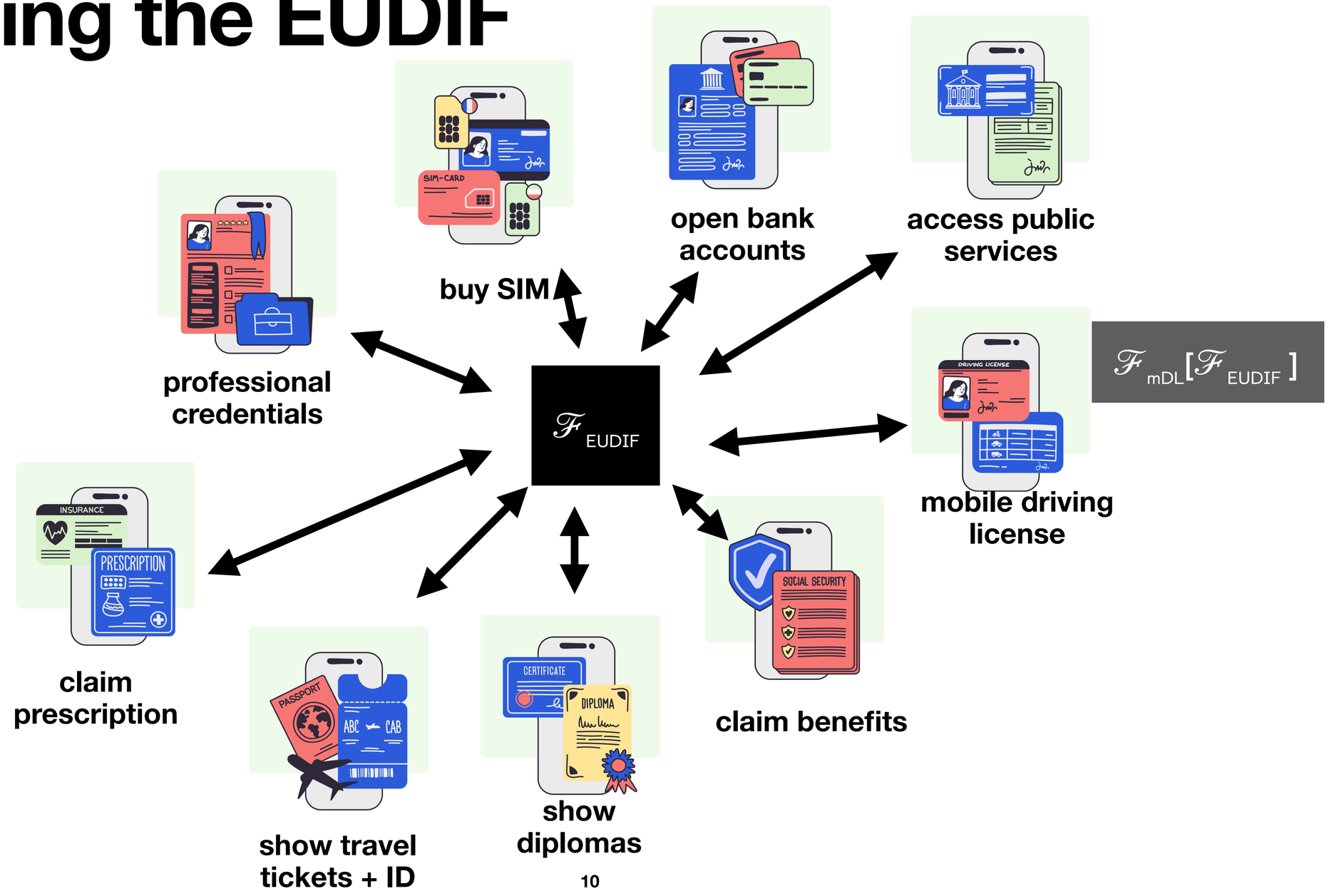
(3) Receive $b_{\text{check}} \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$ that indicates if the user optionally wants to check the RP's registration. If $b_{\text{check}} = 1$:
Set $b_{\text{registration}} := 1$ if $(\text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ is stored in RPReg[rid], and $b_{\text{registered}} := 0$ otherwise.
Send $b_{\text{registration}}$ to $\mathcal{U}_{\text{uid}}$, and receive $\text{consent}_1 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If $\text{consent}_1 = 0$, output $\perp$ to rid. Otherwise, continue.

(4) For $i \in [n]$: retrieve $(\text{iid}_i, \text{attr}_i) := \text{Wallet}[\text{uid}][\text{attr\_descr}_i]$ and $\text{policy}_i := \text{Policies}[\text{uid}][\text{iid}_i]$.
Send $b_{\text{policy}} := \bigwedge_{j\in[n]} \text{policy}_j(\text{rid}, \text{intended\_use}, f, (\text{attr\_descr}_i)_{i\in[n]})$ to $\mathcal{U}_{\text{uid}}$ and receive $\text{consent}_2 \in \{0,1\}$ from $\mathcal{U}_{\text{uid}}$. If $\text{consent}_2 = 0$, output $\perp$ to $\mathcal{R}_{\text{rid}}$. Otherwise, compute the output $v := f(\text{attr}_1, \ldots, \text{attr}_n)$ and send v to $\mathcal{R}_{\text{rid}}$.



requests presentation of attributes, shows info, intended use

Modeling the EUDIF



What are the harms of the EUDIF?

Understanding Pathways to Harm: Age Verification

Understanding Pathways to Harm: Age Verification

Age-restricted online service requests verification that a user is over 18.

Understanding Pathways to Harm: Age Verification

Age-restricted online service requests verification that a user is over 18.

User’s wallet uses credentials issued by a trusted authority to generate proof of age requirement.

Understanding Pathways to Harm: Age Verification

Age-restricted online service requests verification that a user is over 18.

User’s wallet uses credentials issued by a trusted authority to generate proof of age requirement.

Proof is shared with the age-restricted service; user can then access the content.

Understanding Pathways to Harm: Age Verification

ELECTRONIC FRONTIER FOUNDATION

DEEPLINKS BLOG BY JASON KELLEY, ADAM SCHWARTZ | MARCH 10, 2023

Age Verification Mandates Would Undermine Anonymity Online

Age verification systems are surveillance systems. Mandatory age verification, and with it, mandatory identity verification, is the wrong approach to protecting young people online. It would force websites to require visitors to prove their age by submitting information such as government-issued identification. This scheme would lead us further towards an...

Knight–Georgetown Institute

REPORT / JANUARY 29, 2026

Age Assurance Online: A Technical Assessment of Current Systems and their Limitations

Governments around the world are adopting online age assurance requirements of different kinds, reshaping how digital services are accessed by adults and youth. KGI's latest report provides a technical assessment of how age assurance systems work in practice and examines their accuracy, circumvention resistance, availability, and privacy implications.

EN English

Quarterly > Winners & Losers

Online Age-Gating Is the Wrong Answer

Mar 16, 2026 | **JULIA POWLES** and **KATE SIM**

With social-media minimum-age restrictions – or "bans" – gaining traction globally, it is worth considering whether this popular new policy intervention is as effective as its proponents claim. The evidence suggests not, and even raises concerns that age-gating could do more harm than good.

EPICENTER.WORKS for digital rights · AK ÖSTERREICH · EDRi · EFF Electronic Frontier Foundation · Homo Digitalis Protect your rights · EFN Elektronisk Forpost Norge · digitalcourage · SHARE FOUNDATION · DIGITALE GESELLSCHAFT · DEFESA DOS DIREITOS DIGITAIS · yrüschrift.org · DANES JE NOV DAN · ApTI

Open Letter of Digital Rights and Consumer Organisations Concerning eIDAS Implementing Acts

Dear
Christ

The
Comm
adopt
regul
privac

The a
recen
eIDAS
and t
Digita
the W
users
inform
relyin

1 htt
re
2 Se

19

THE ABC OF ABC: AN ANALYSIS OF ATTRIBUTE-BASED CREDENTIALS IN THE LIGHT OF DATA PROTECTION, PRIVACY AND IDENTITY

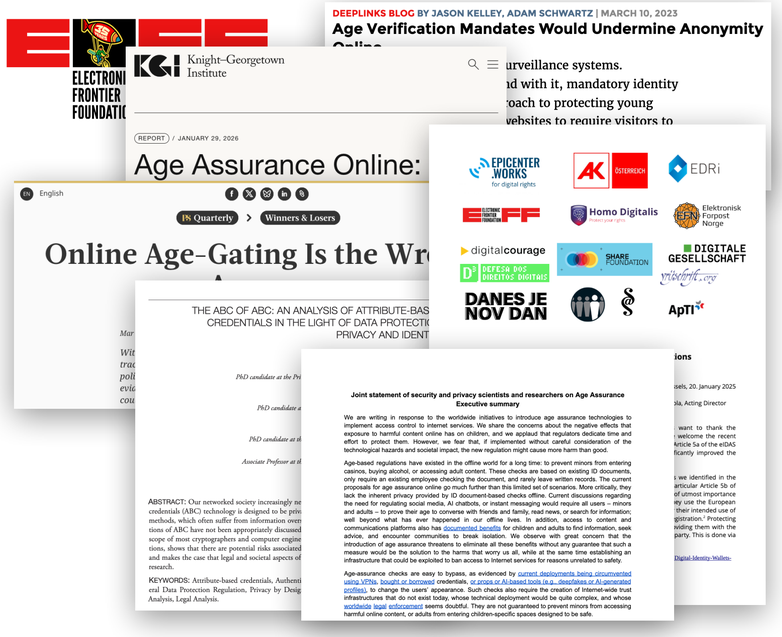
ABSTR
creden
metho
tions c

Joint statement of security and privacy scientists and researchers on Age Assurance
Executive summary

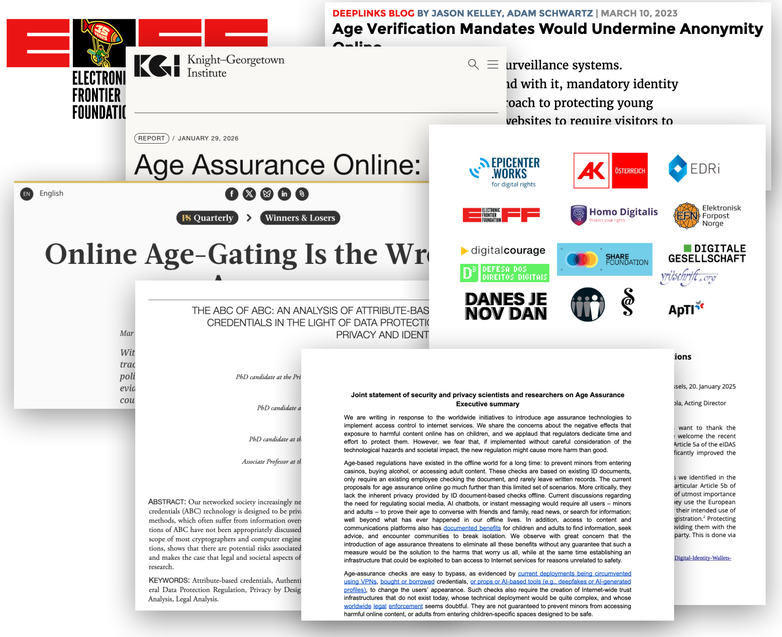
We are writing in response to the worldwide initiatives to introduce age assurance technologies to implement access control to internet services. We share the concerns about the negative effects that exposure to harmful content online has on children, and we applaud that regulators dedicate time and effort to protect them. However, we fear that, if implemented without careful consideration of the technological hazards and societal impact, the new regulation might cause more harm than good.

Age-based regulations have existed in the offline world for a long time: to prevent minors from entering casinos, buying alcohol, or accessing adult content. These checks are based on existing ID documents, only require an existing employee checking the document, and rarely leave written records. The current proposals for age assurance online go much further than this limited set of scenarios. More critically, they lack the inherent privacy provided by ID document-based checks offline. Current discussions regarding the need for regulating social media, AI chatbots, or instant messaging would require all users – minors and adults – to prove their age to converse with friends and family, read news, or search for information; well beyond what has ever happened in our offline lives. In addition, access to content and communications platforms also has documented benefits for children and adults to find information, seek advice, and encounter communities to break isolation. We observe with great concern that the

Understanding Pathways to Harm: Age Verification



Understanding Pathways to Harm: Age Verification



Harms

Increased cybersecurity risks

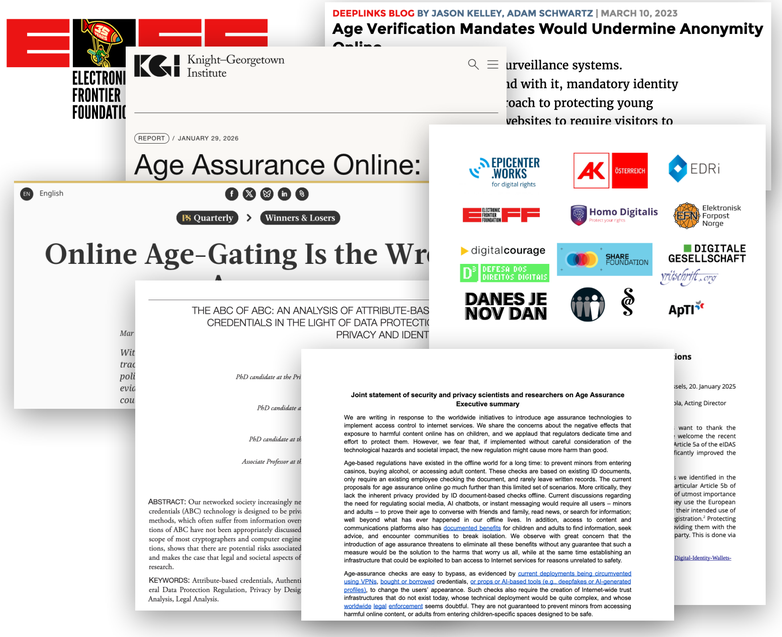
Censorship

Structural exclusion

Discrimination

Targeted ads

Understanding Pathways to Harm: Age Verification

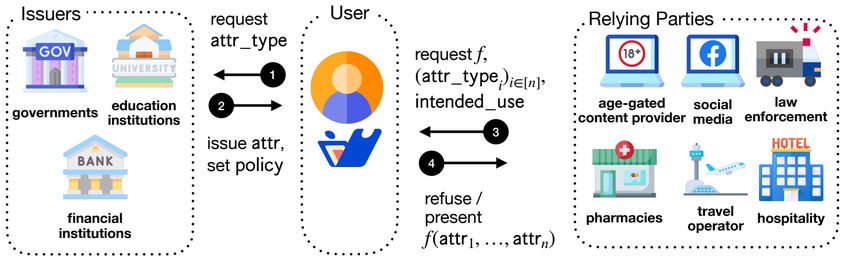


→

Harms

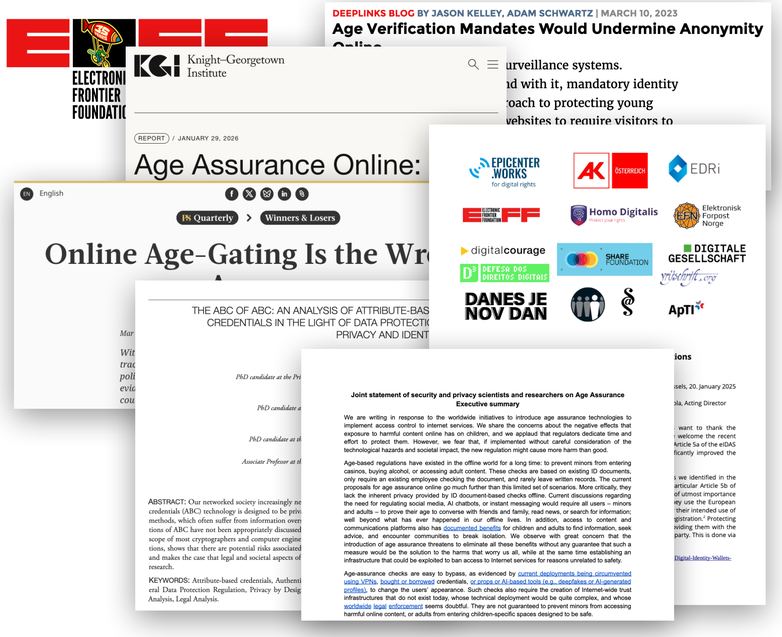
- Increased cybersecurity risks
- Censorship
- Structural exclusion
- Discrimination
- Targeted ads

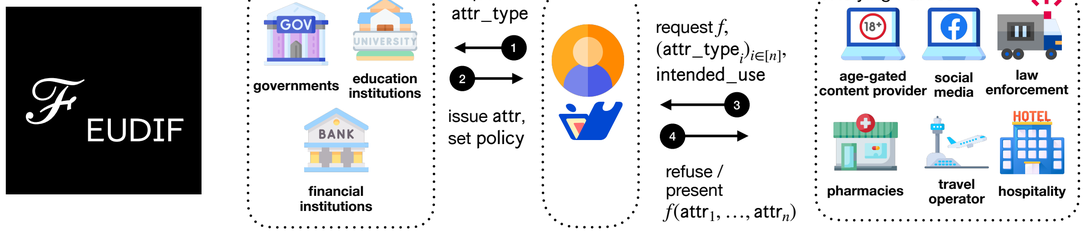
EUDIF



+ “naive” implementation in a digital context

Understanding Pathways to Harm: Age Verification





+ “naive” implementation in a digital context

Harms

- Increased cybersecurity risks
- Censorship
- Structural exclusion
- Discrimination
- Targeted ads

System Components

- Persistent identifier
- Online metadata
- Requires access to device
- Service provider chooses whether to implement age verification
- Repeated observations
- Issuance by a government entity
- Reveal age information

Understanding Pathways to Harm: Age Verification

Discrimination

Structural exclusion

Increased cybersecurity risks

Harm

Targeted ads

Censorship

Repeated observations

Service provider chooses whether to implement age verification

Persistent identifier

Reveal age information

Issuance by a government entity

Requires access to device

Online metadata

System Components

Understanding Pathways to Harm: Age Verification

Discrimination Structural exclusion Increased cybersecurity risks **Harm**

Targeted ads Censorship

Repeated observations Service provider chooses whether to implement age verification Persistent identifier

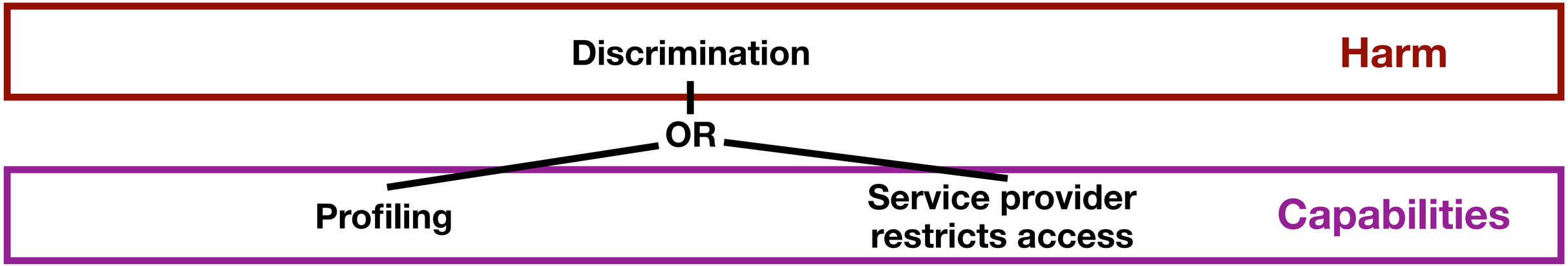
Reveal age information Issuance by a government entity Requires access to device Online metadata **System Components**

Understanding Pathways to Harm: Age Verification

Discrimination	Harm

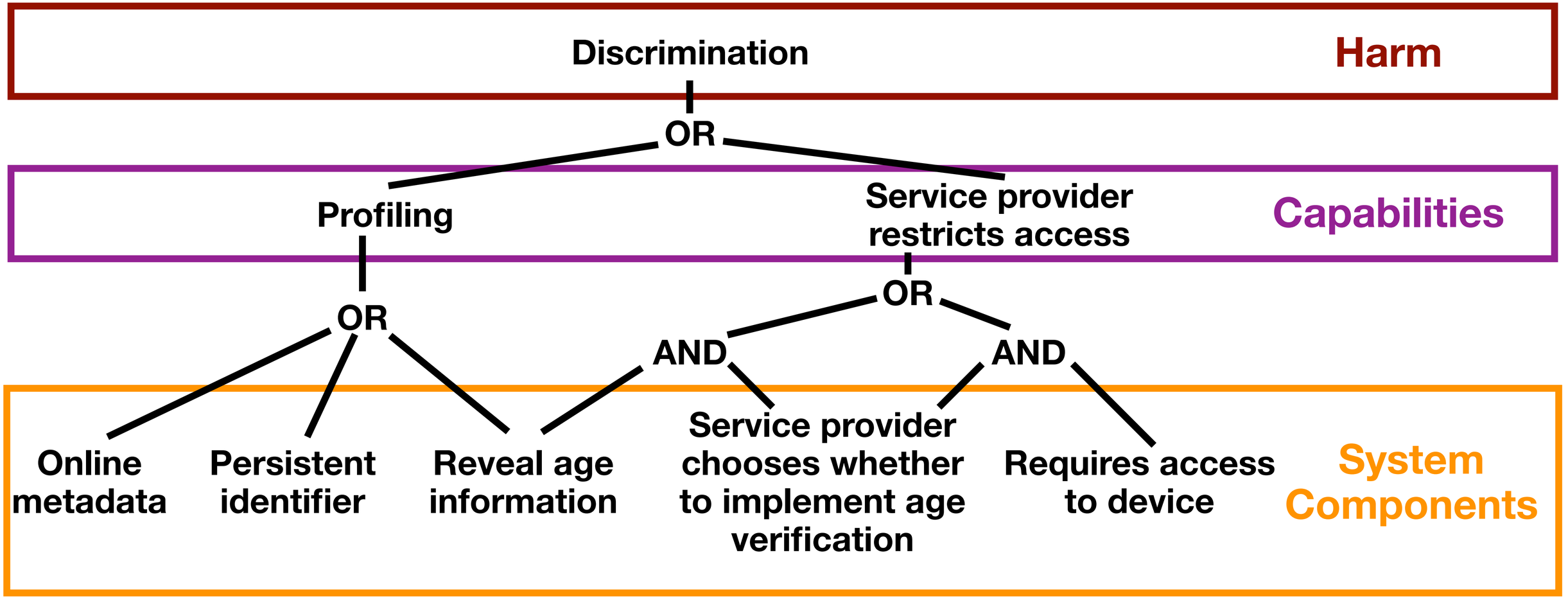
Online metadata	Persistent identifier	Reveal age information	Service provider chooses whether to implement age verification	Requires access to device	System Components

Understanding Pathways to Harm: Age Verification

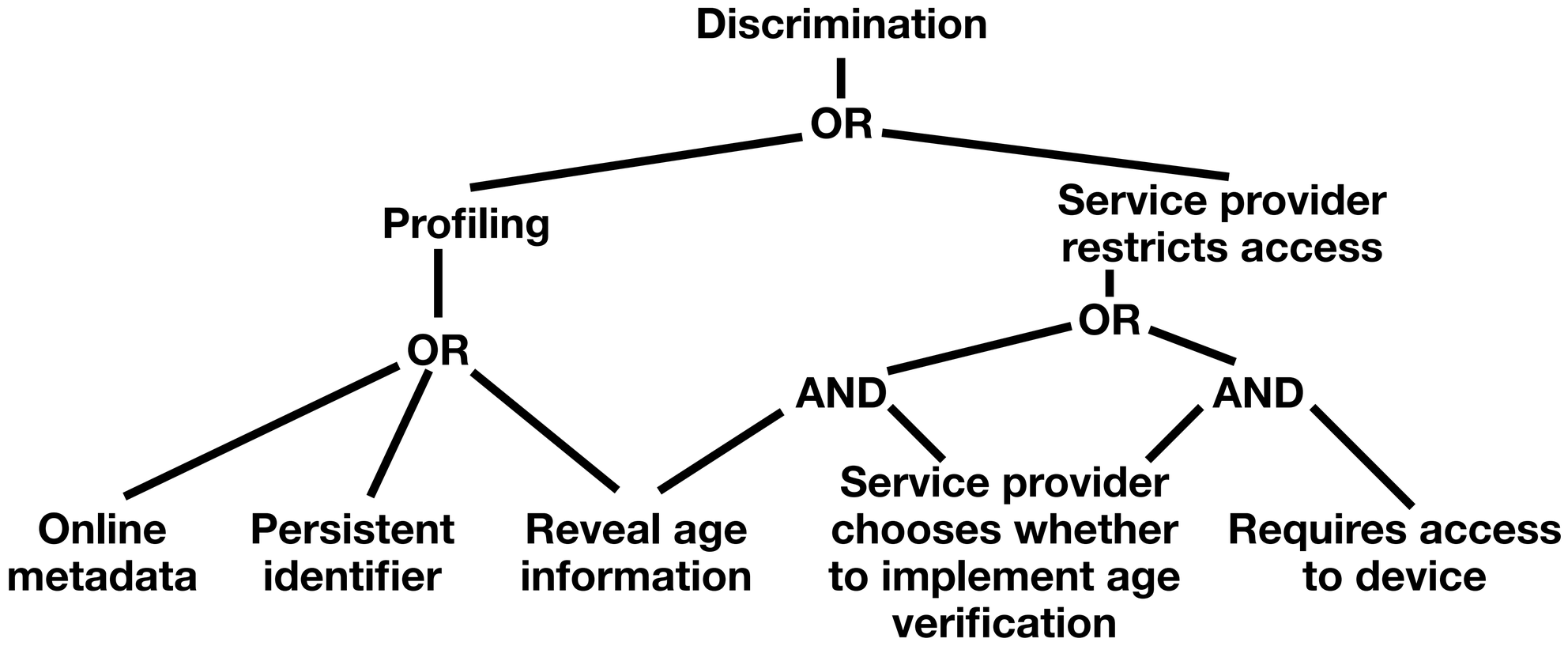


Online metadata	Persistent identifier	Reveal age information	Service provider chooses whether to implement age verification	Requires access to device	System Components

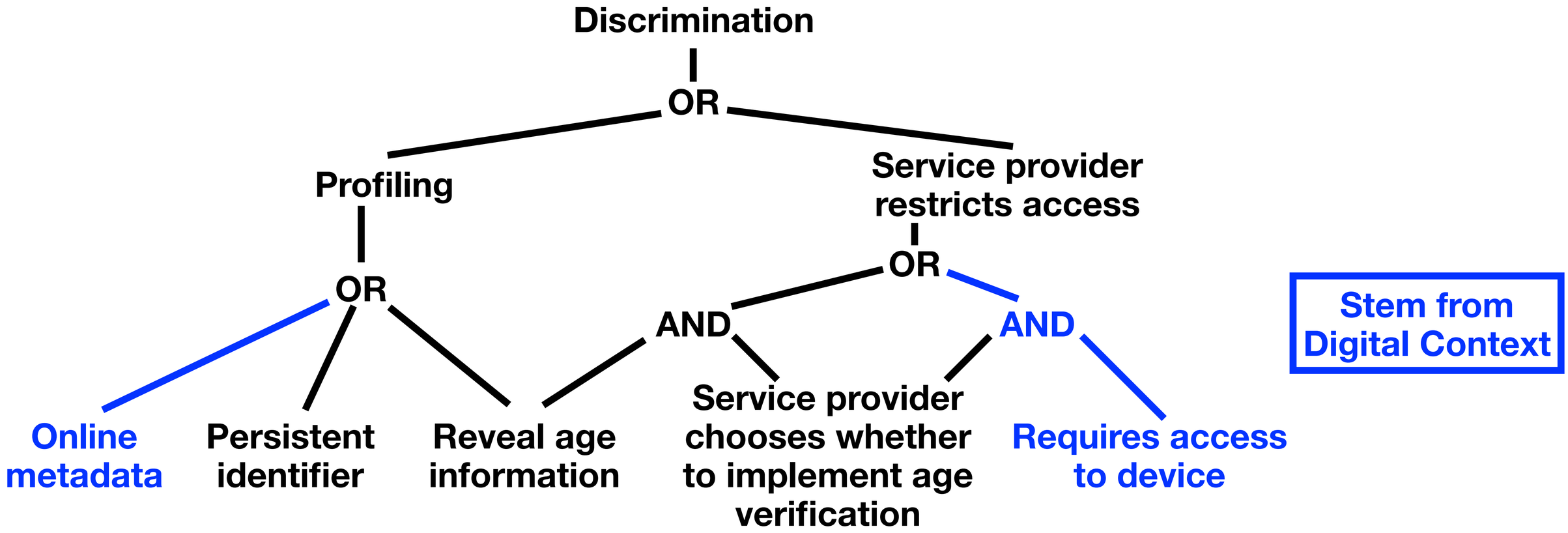
Understanding Pathways to Harm: Age Verification



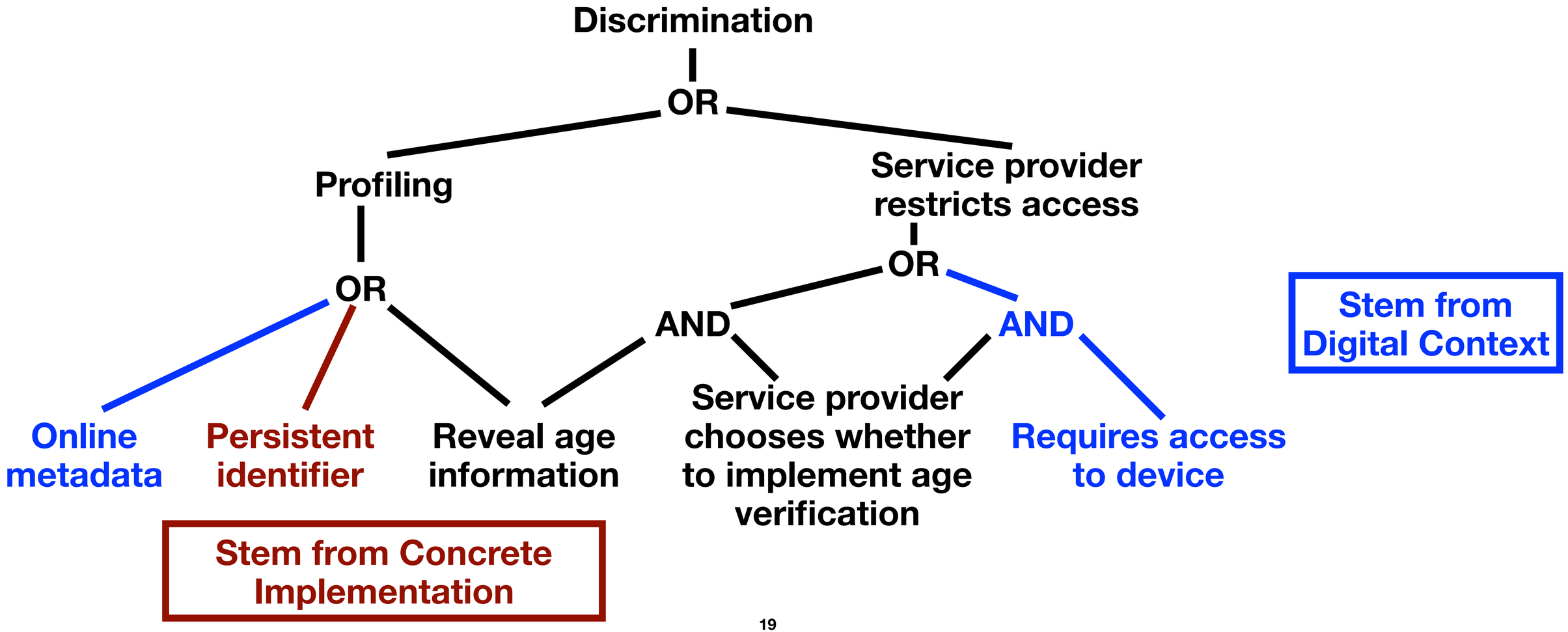
Understanding Pathways to Harm: Age Verification



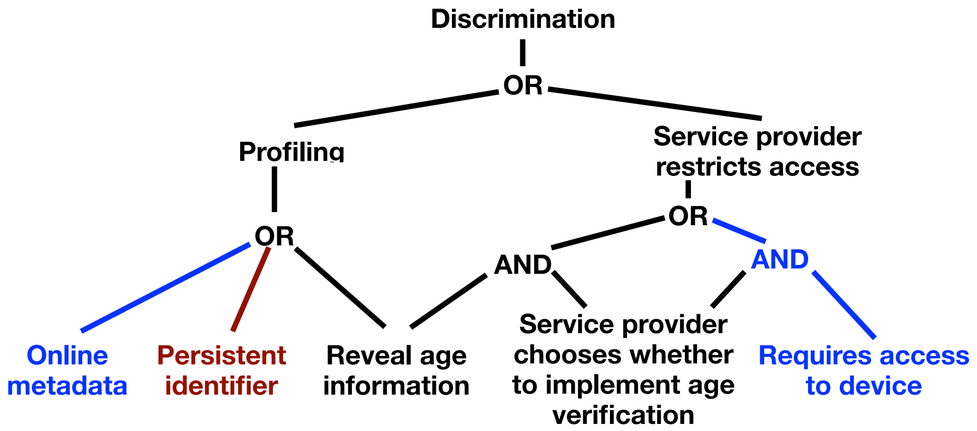
Understanding Pathways to Harm: Age Verification



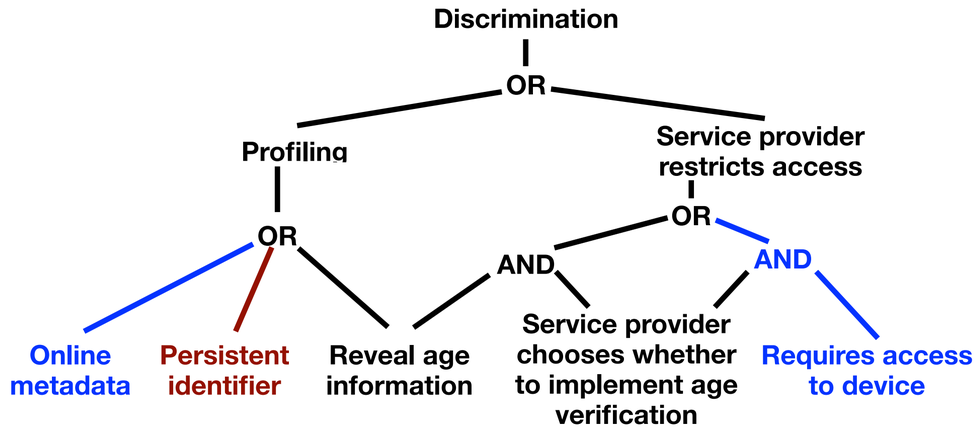
Understanding Pathways to Harm: Age Verification

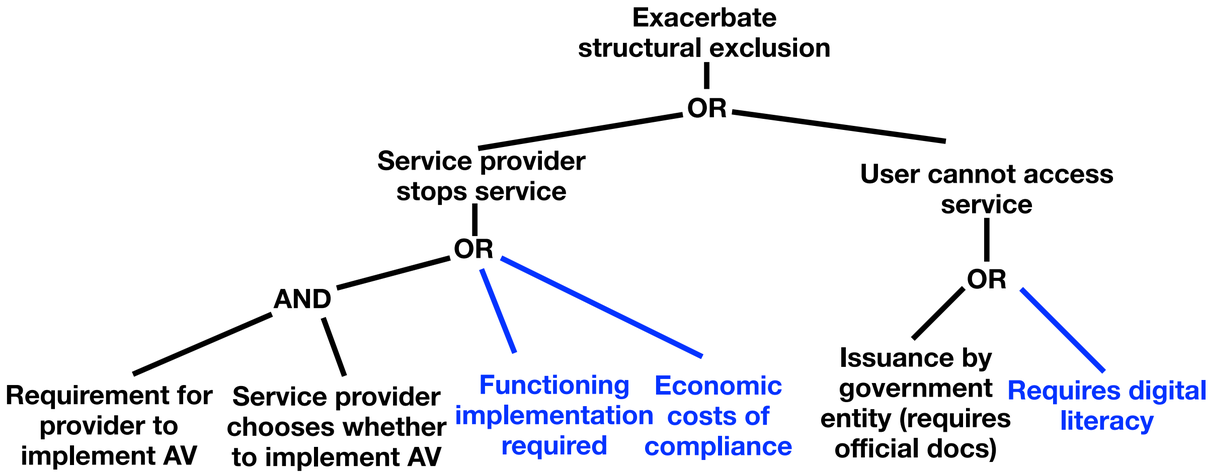


Understanding Pathways to Harm: Age Verification

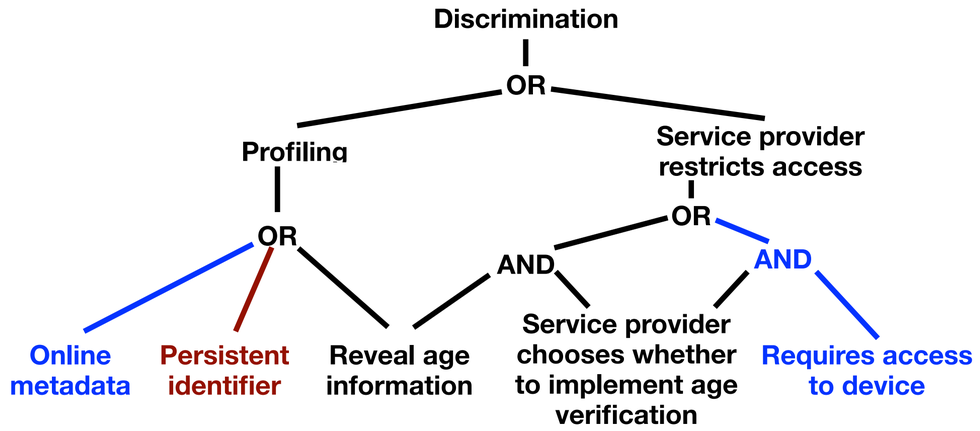


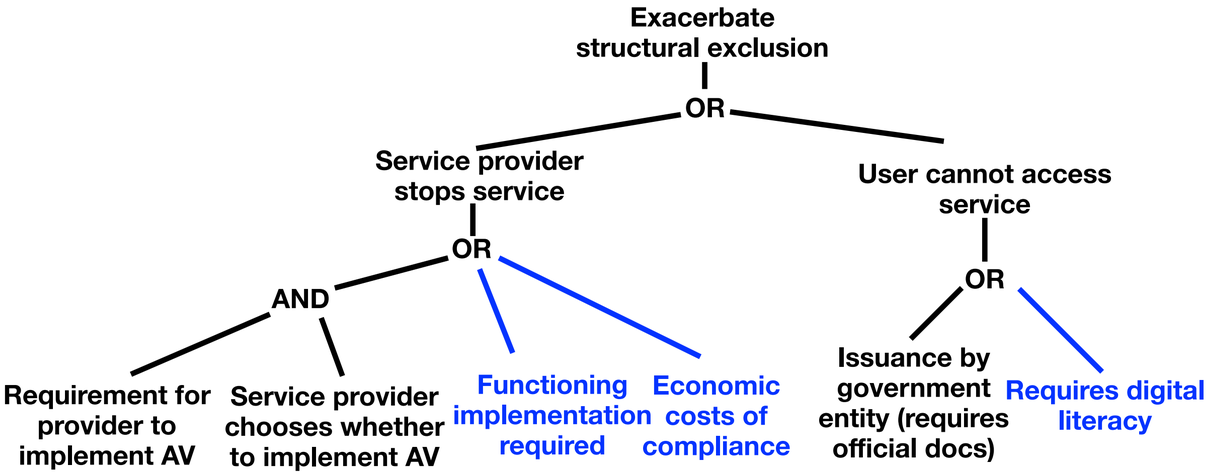
Understanding Pathways to Harm: Age Verification

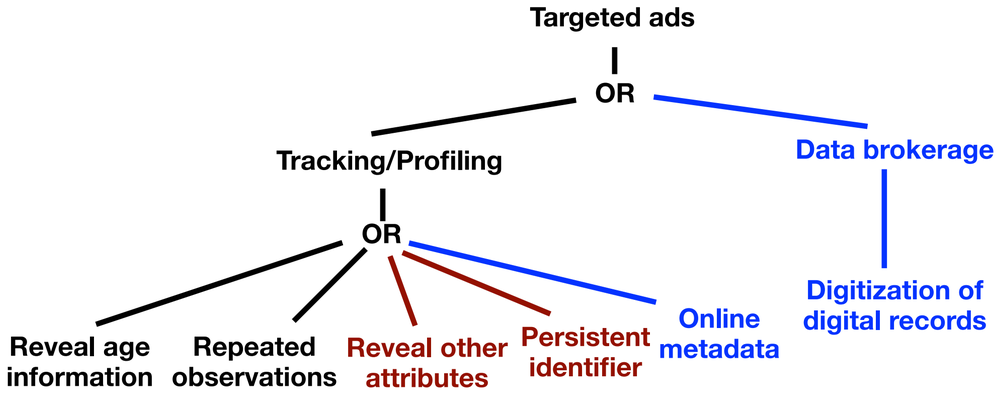




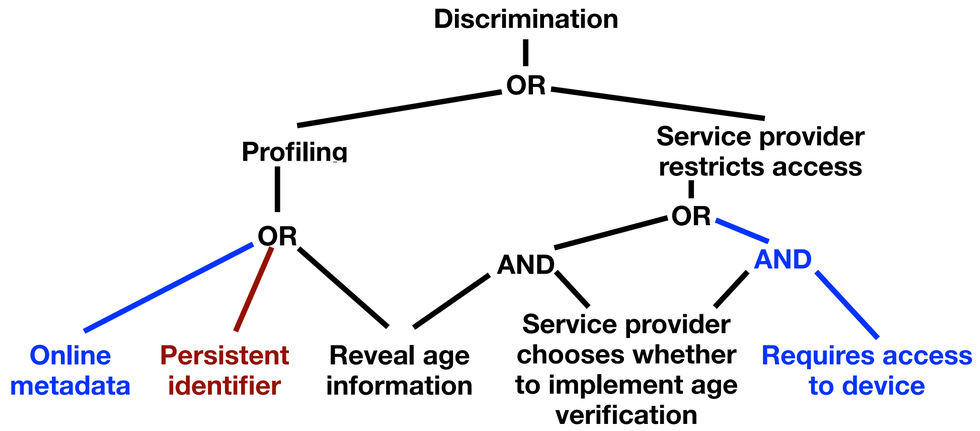
Understanding Pathways to Harm: Age Verification

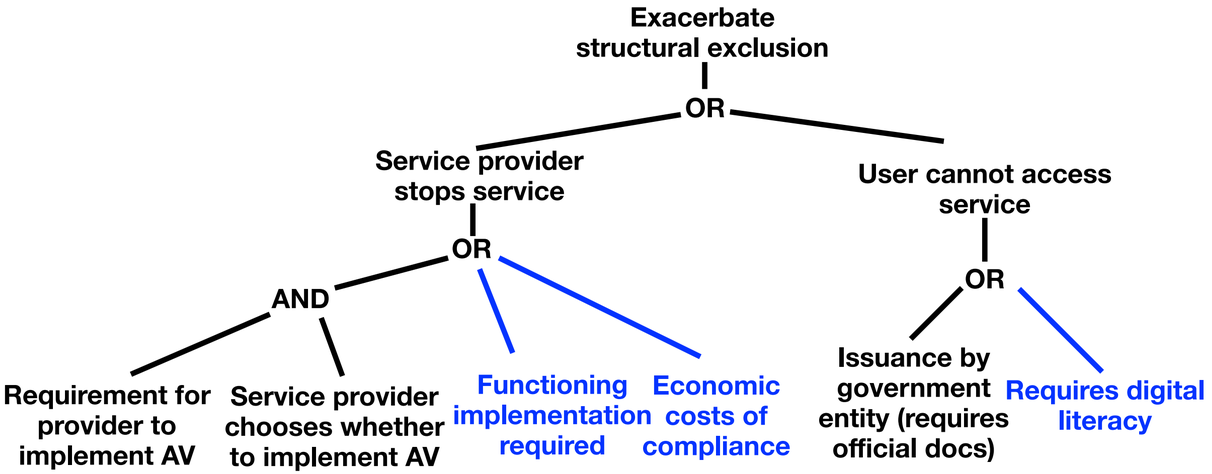


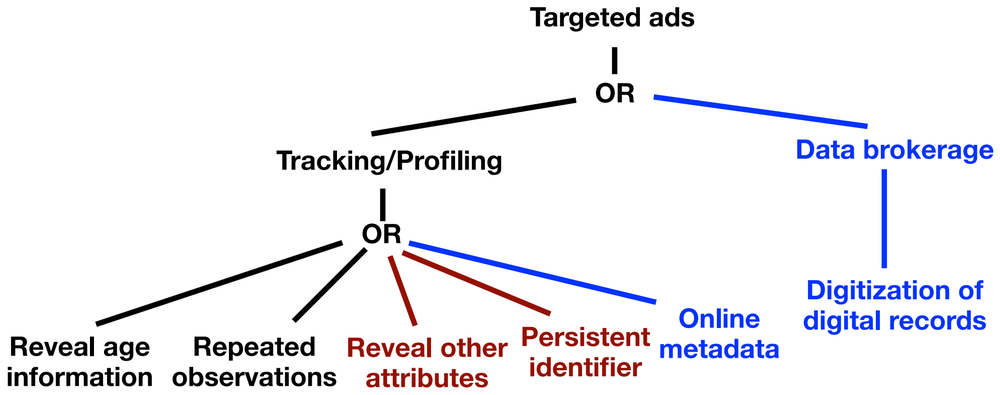




Understanding Pathways to Harm: Age Verification







Black and blue pathways are *inherent* to the functionality and its digital context: these cannot be eliminated by PETs.

Understanding Pathways to Harm: Mobile Driving Licenses

Car rental service requires driving license information.

Understanding Pathways to Harm: Mobile Driving Licenses

Car rental service requires driving license information.

User uploads mobile driving license and identity information to rental website when booking.

Understanding Pathways to Harm: Mobile Driving Licenses

Car rental service requires driving license information.

User uploads mobile driving license and identity information to rental website when booking.

In person: Clerk checks user's wallet information from their device, and verifies user's identity.

Understanding Pathways to Harm: Mobile Driving Licenses

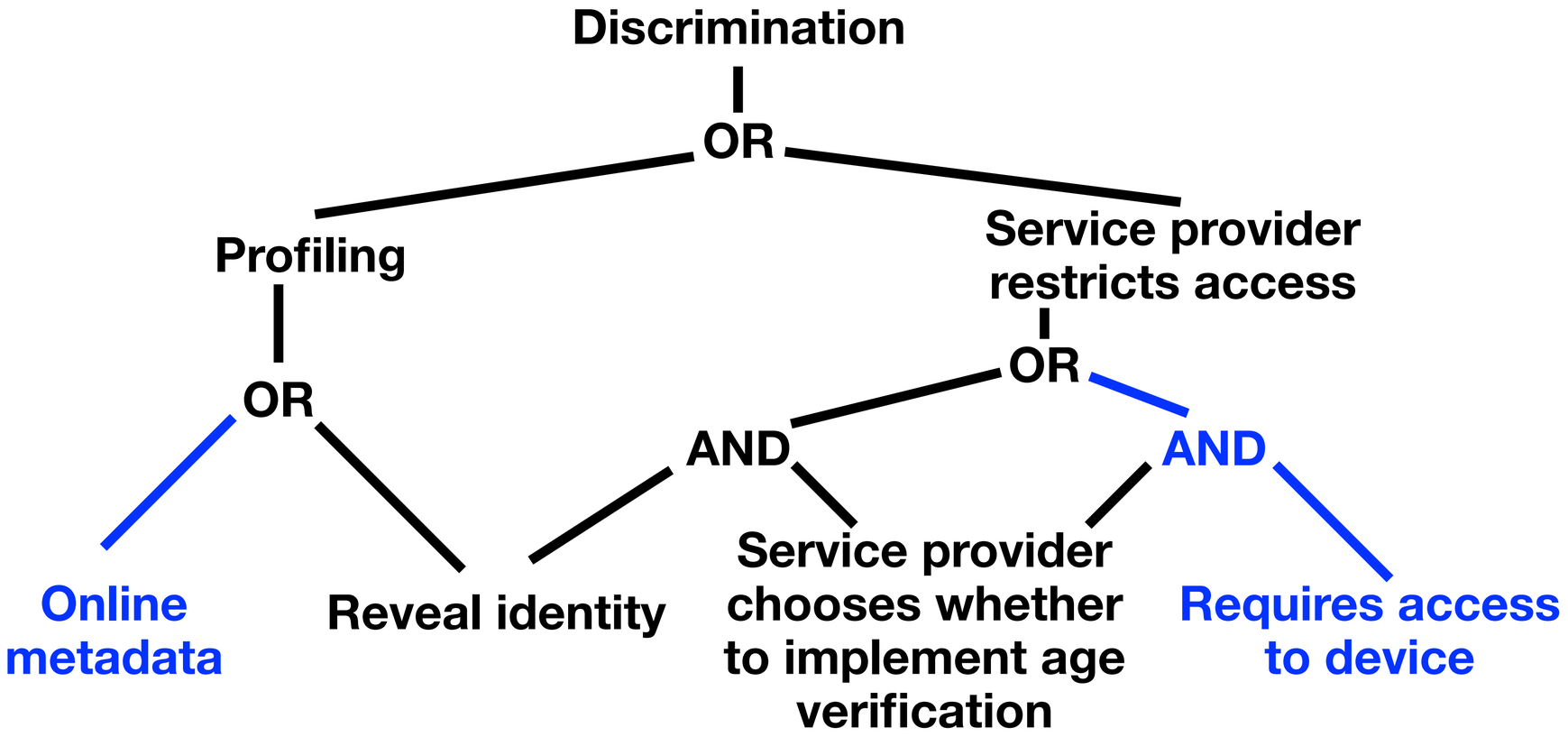
Car rental service requires driving license information.

User uploads mobile driving license and identity information to rental website when booking.

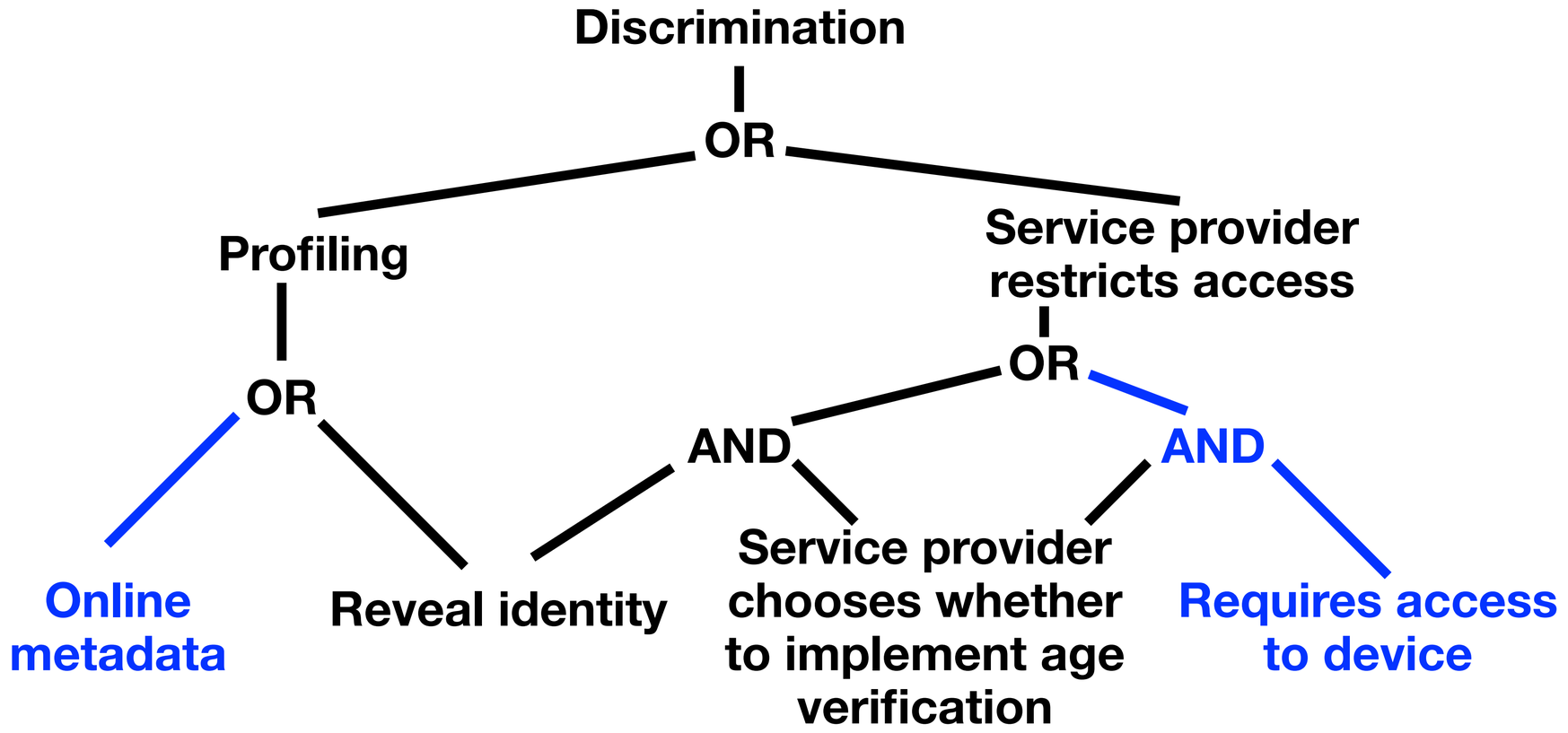
In person: Clerk checks user’s wallet information from their device, and verifies user’s identity.

User obtains car rental.

Understanding Pathways to Harm: Mobile Driving Licenses

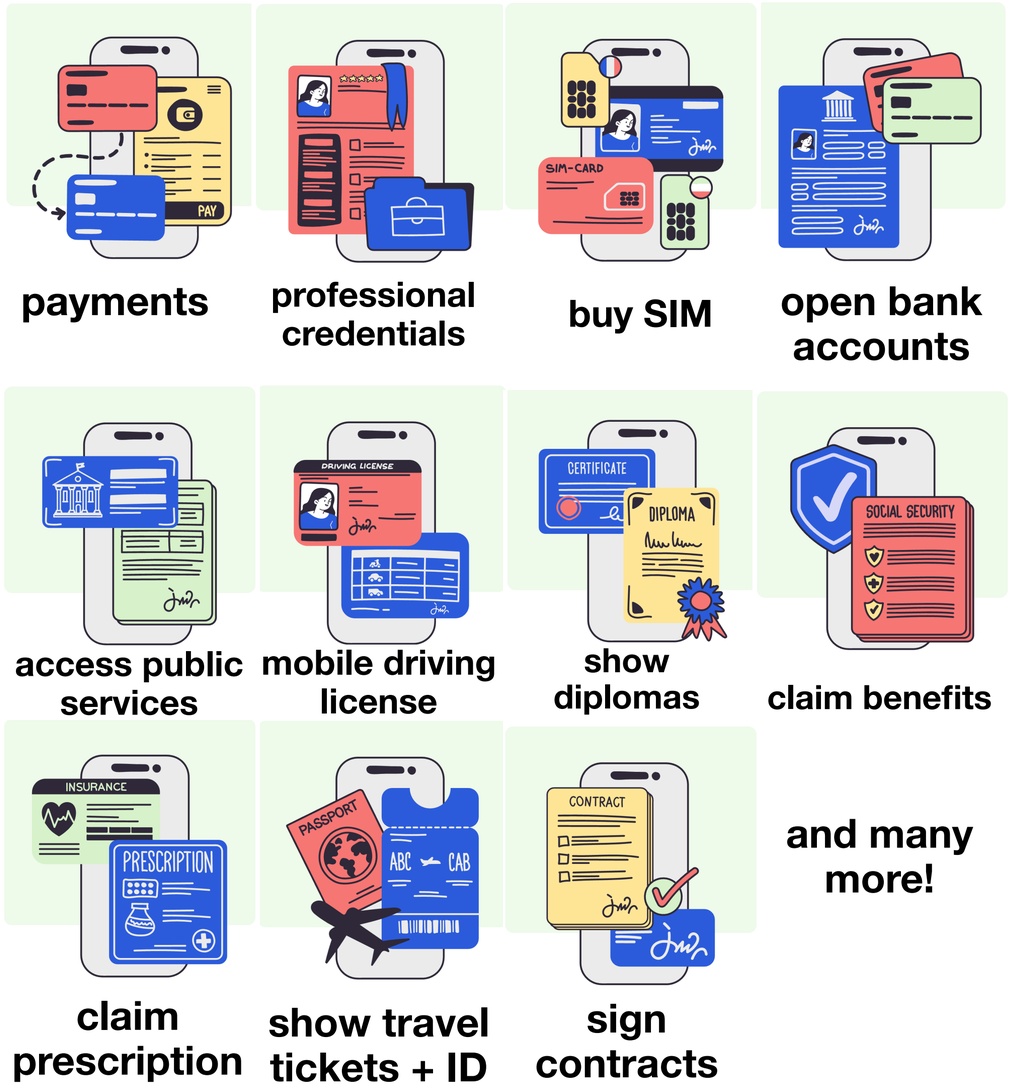


Understanding Pathways to Harm: Mobile Driving Licenses

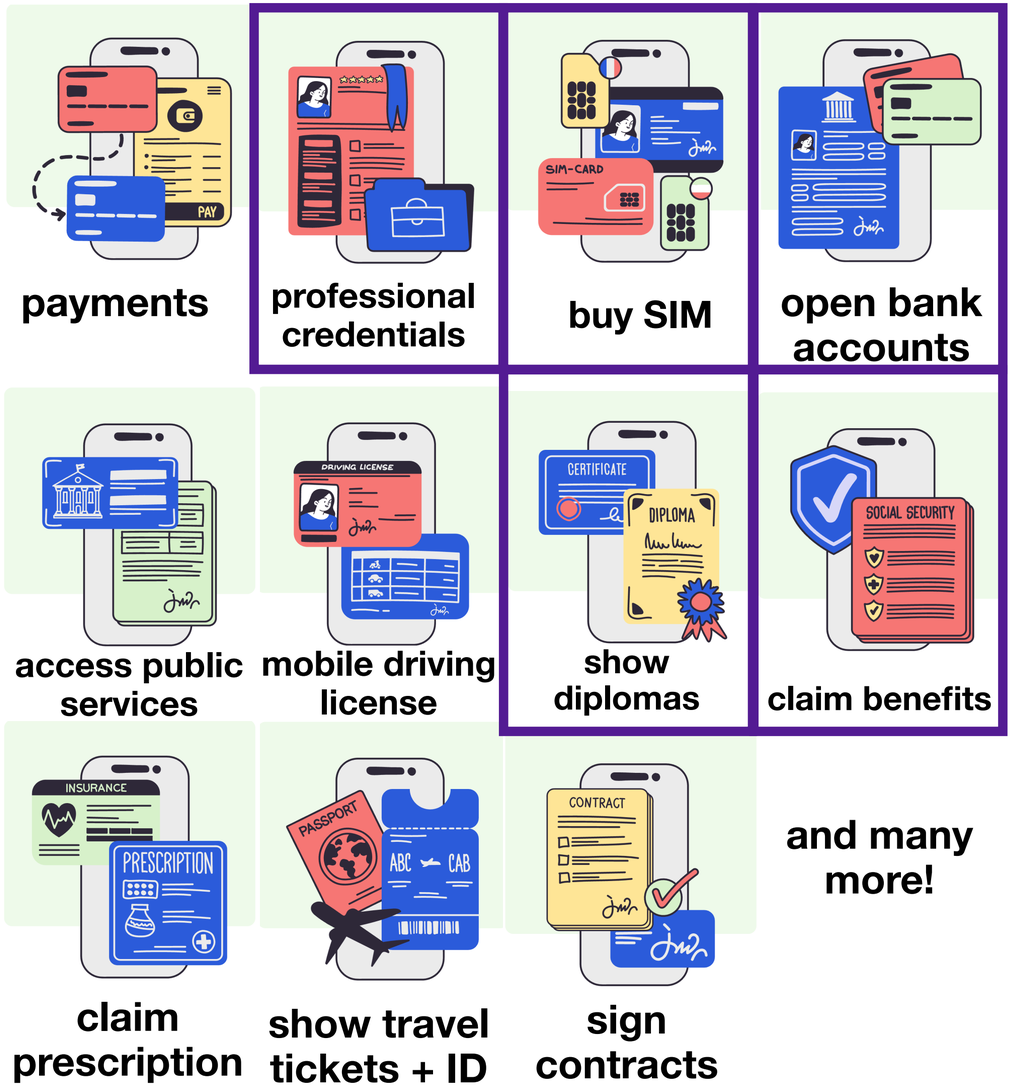


PETs cannot eliminate harms in identity use cases.

PETs cannot eliminate harms from identity use cases:

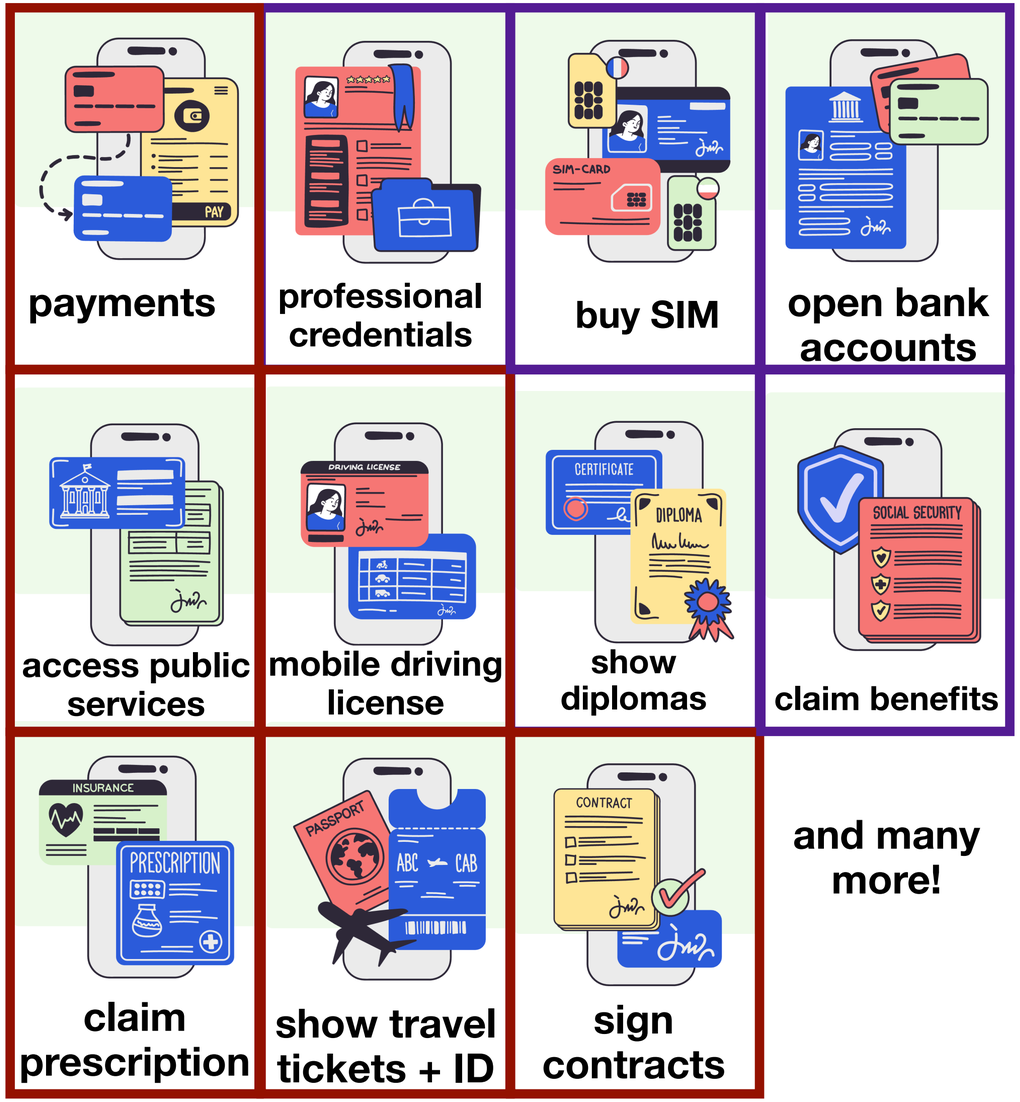


PETs cannot eliminate harms from identity use cases:



Use cases are not yet specified (but likely need identity)…

PETs cannot eliminate harms from identity use cases:

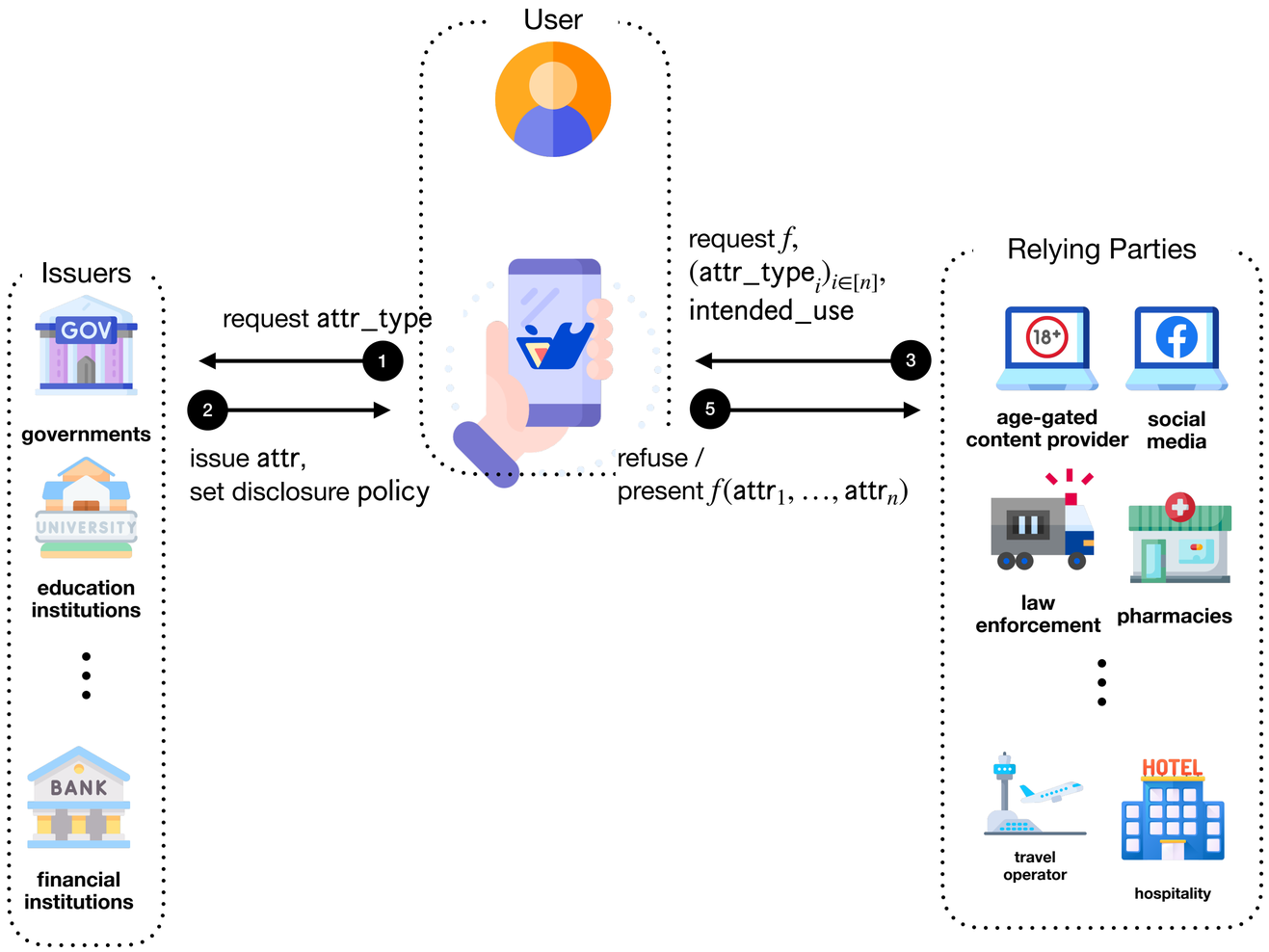


Use cases are not yet specified (but likely need identity)…

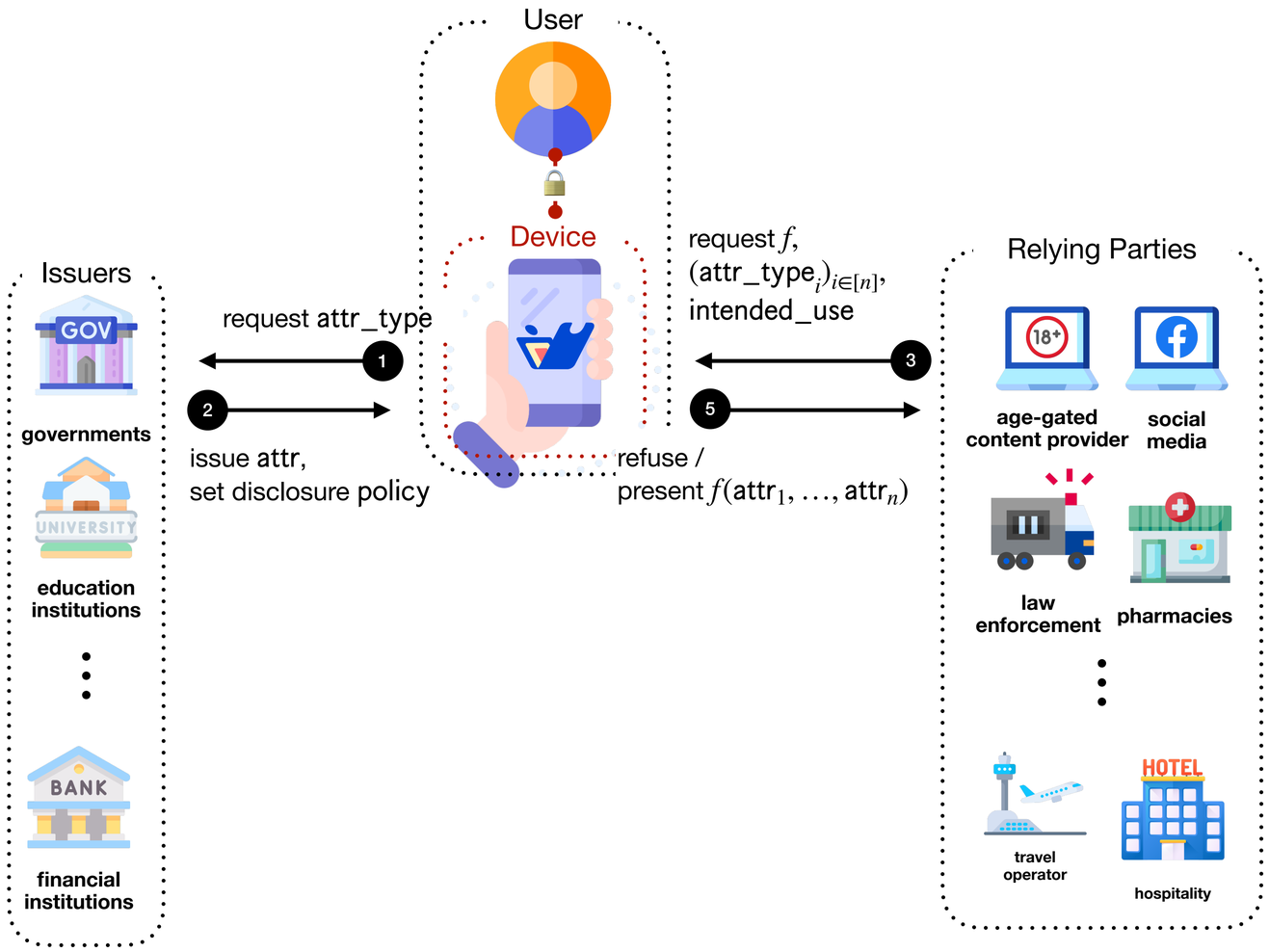
and many more!

…or already require identity

Deployment requirements introduce new pathways

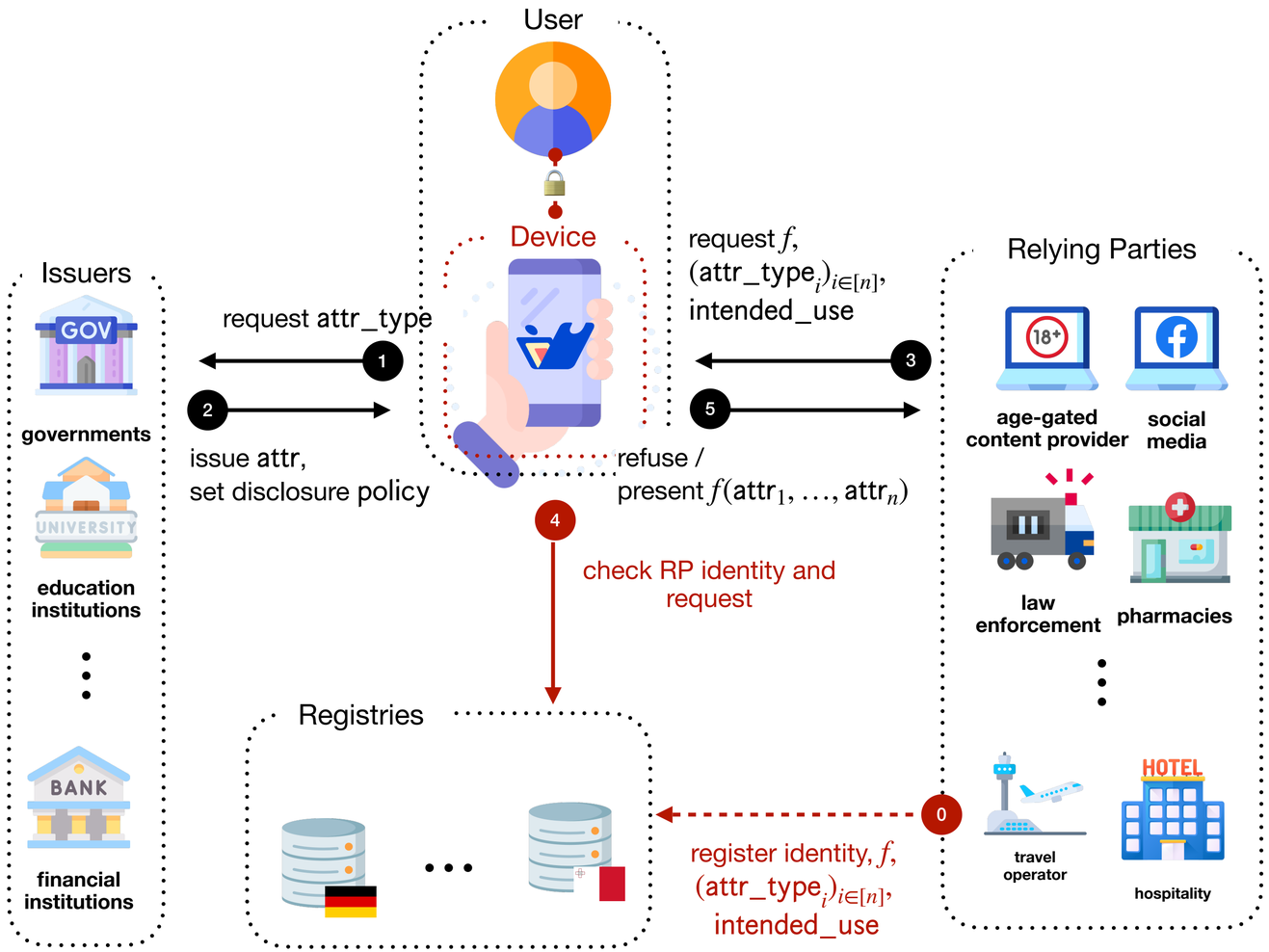


Deployment requirements introduce new pathways



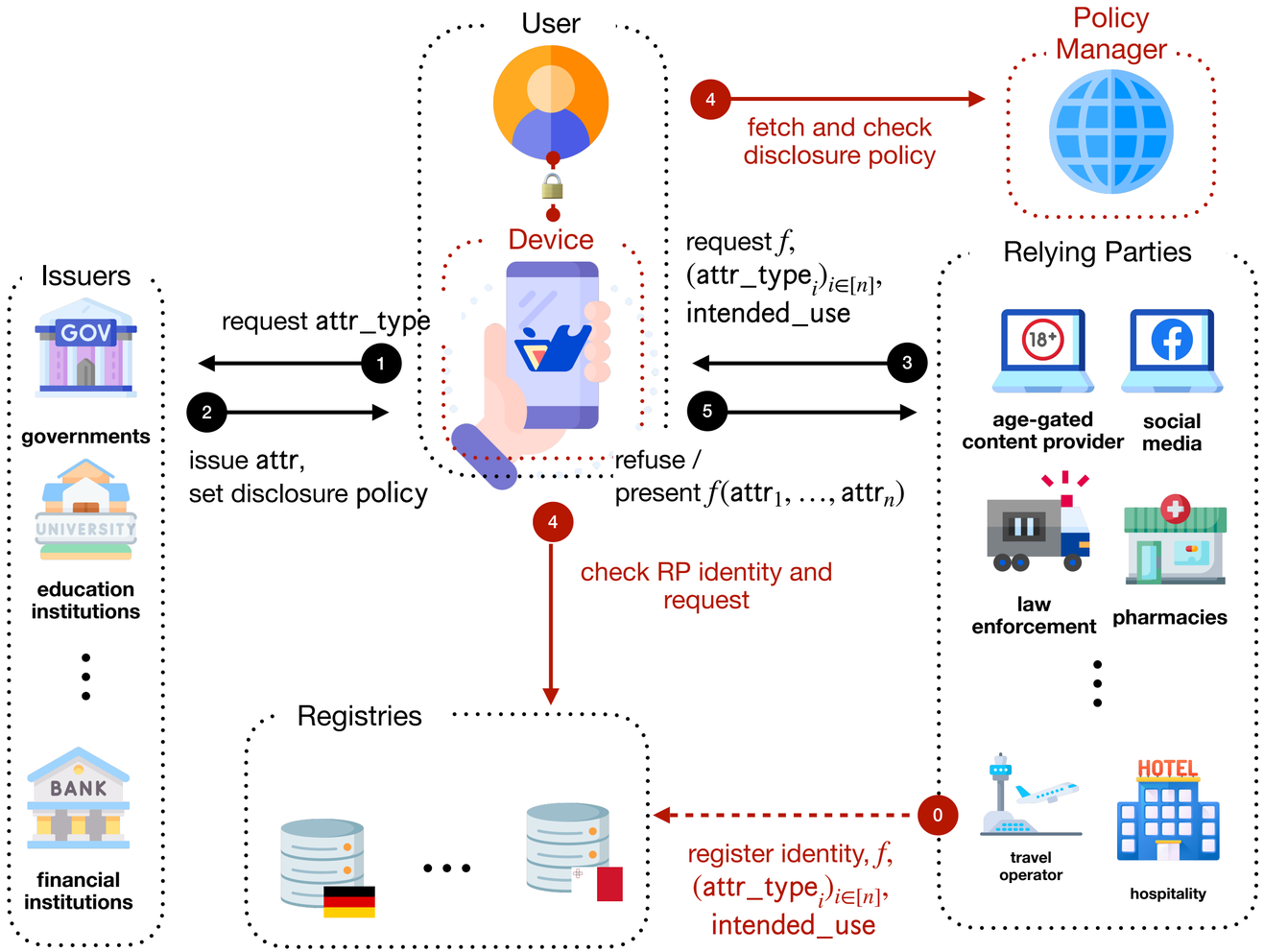
- Device requirements (incl. binding) increase exclusion risk.

Deployment requirements introduce new pathways



- Device requirements (incl. binding) increase exclusion risk.

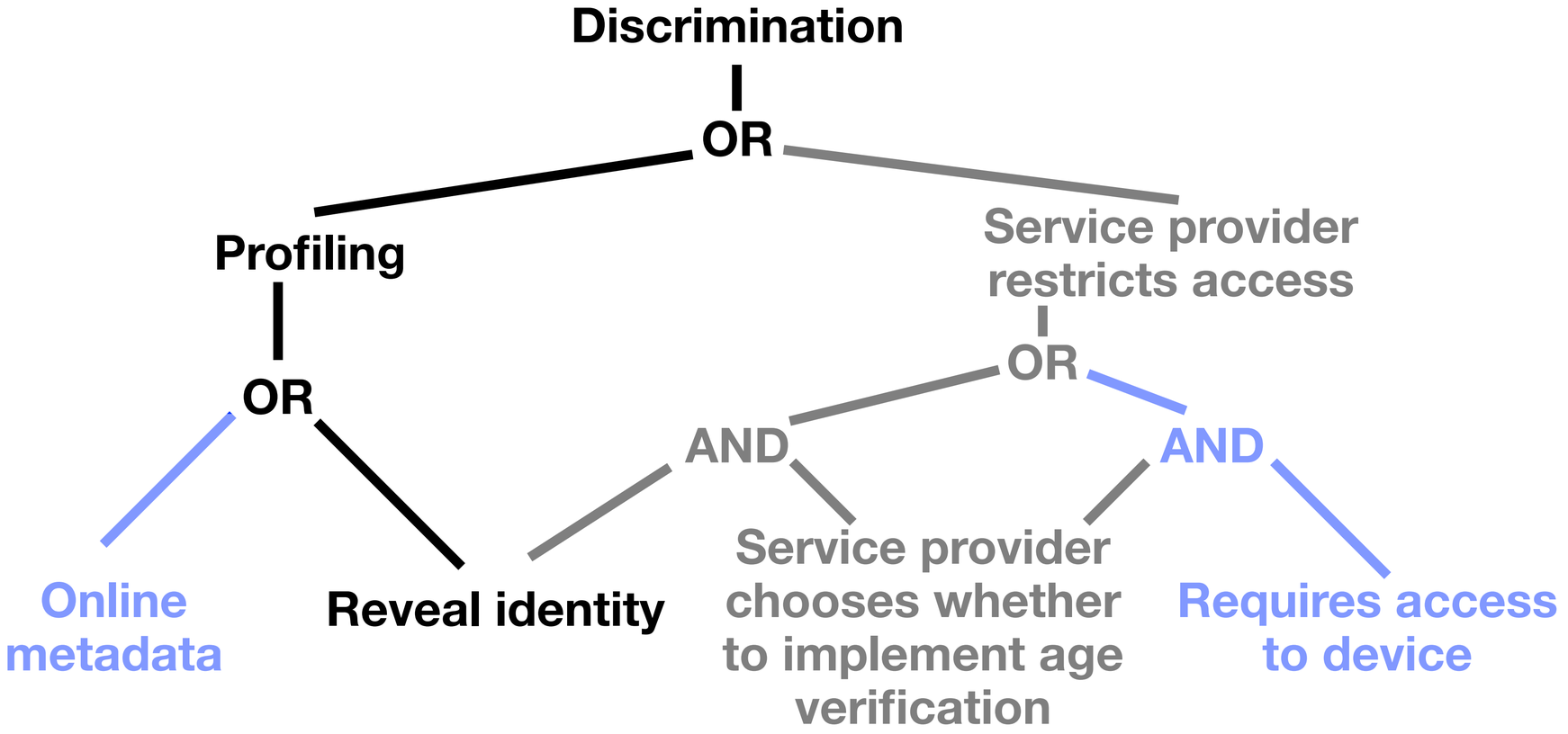
Deployment requirements introduce new pathways



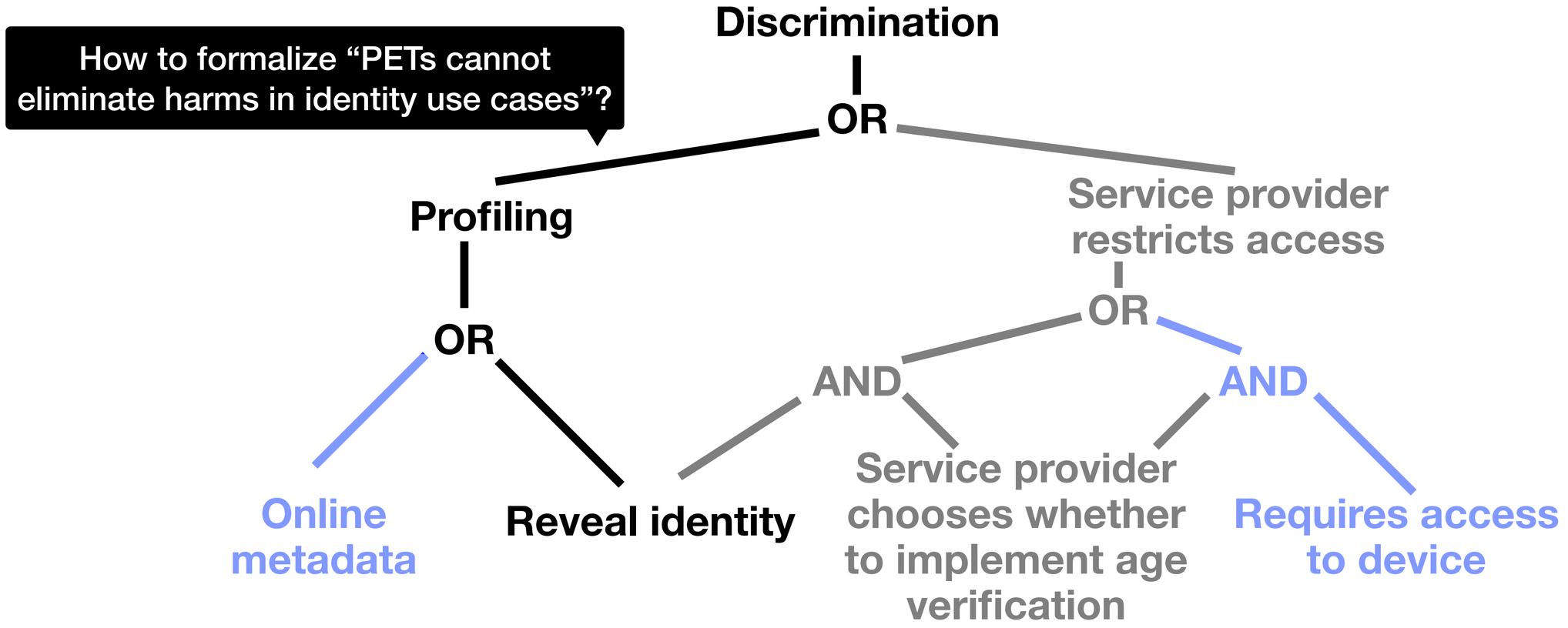
- Device requirements (incl. binding) increase exclusion risk.
- RP registration authorities, and policy managers introduce new entities that learn information.

Fundamental harm-enabling capabilities

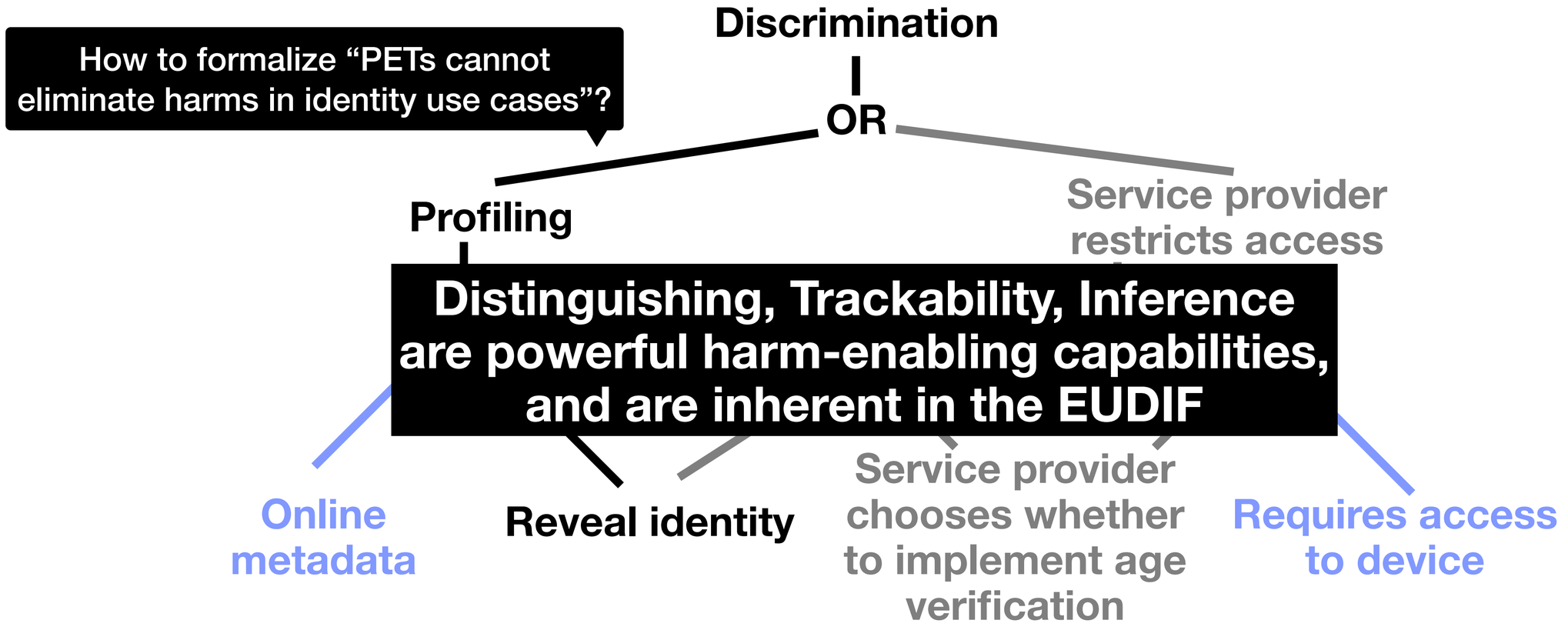
Fundamental harm-enabling capabilities



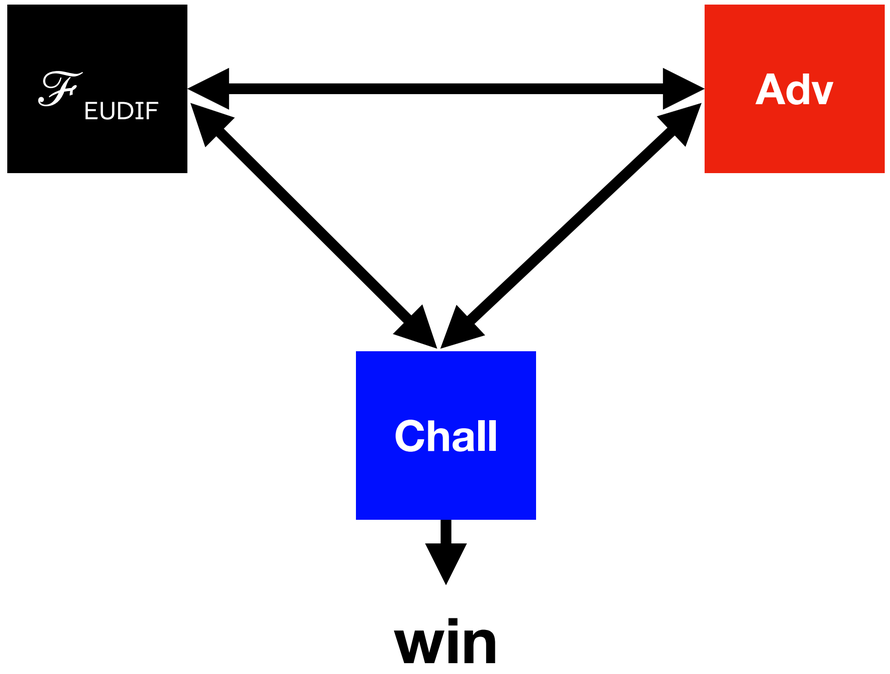
Fundamental harm-enabling capabilities



Fundamental harm-enabling capabilities

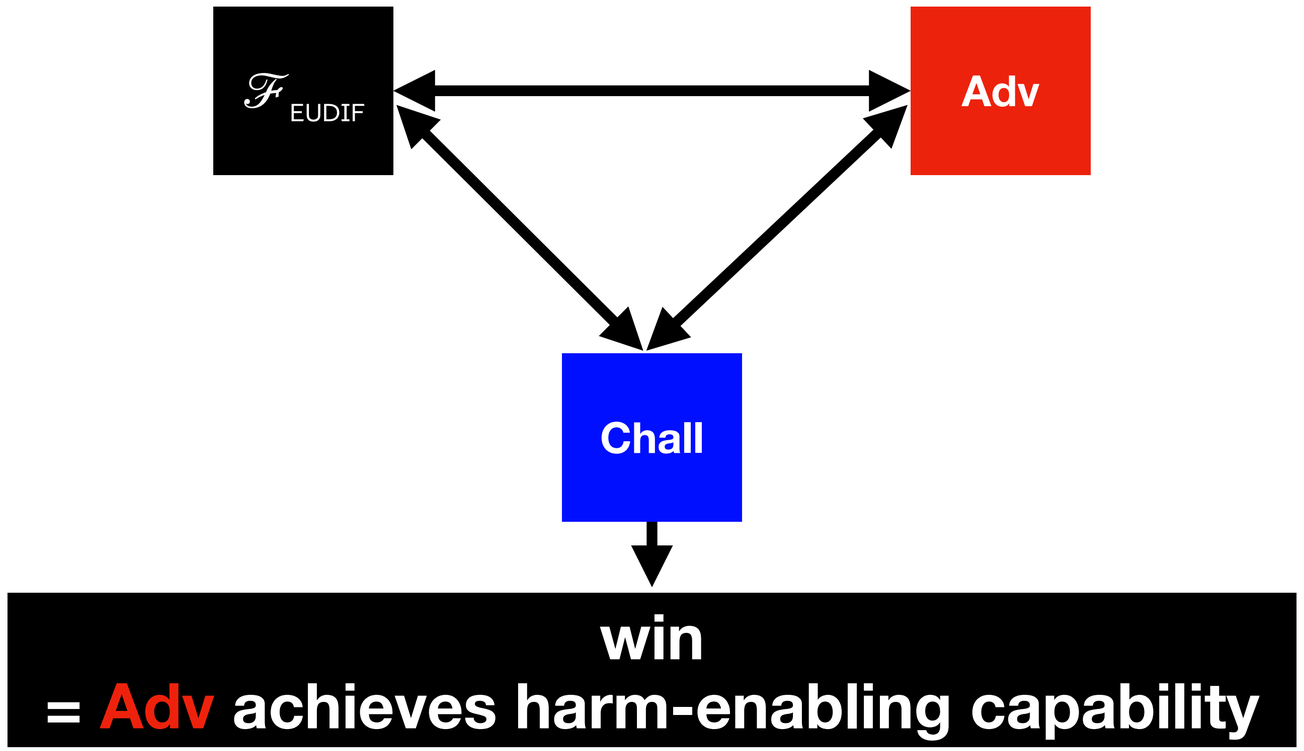


Fundamental harm-enabling capabilities



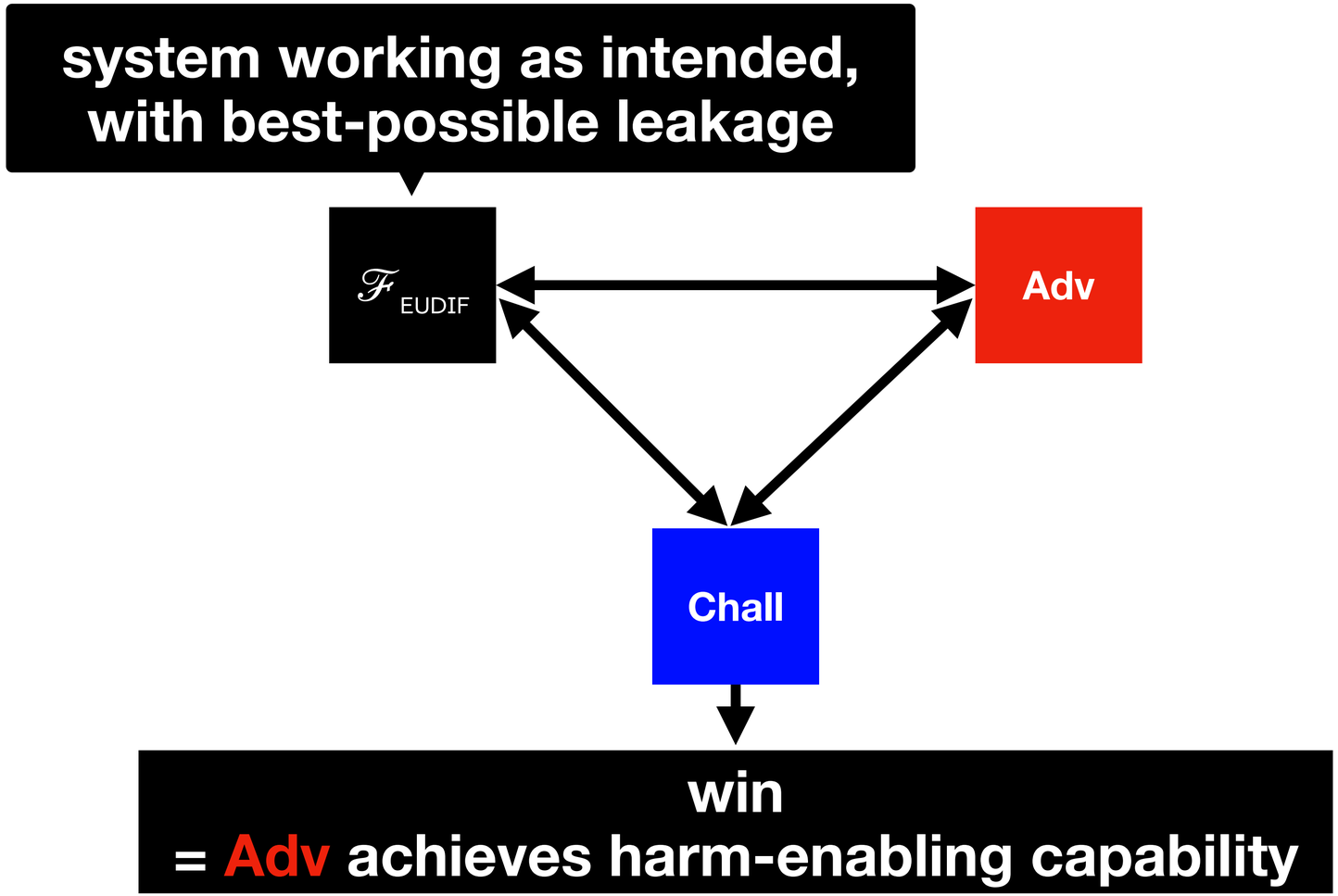
[ACdGJK26] “Are ideal functionalities really ideal?”, Arapinis, Cortier, de Groote, Jacomme, Kremer, CSF’26

Fundamental harm-enabling capabilities



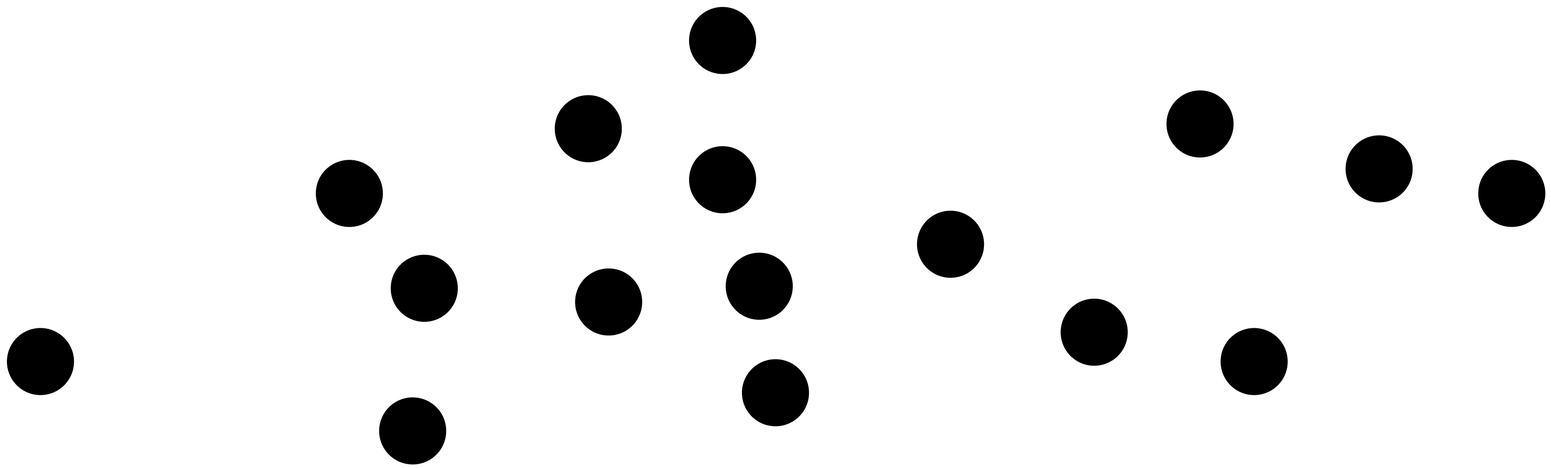
[ACdGJK26] “Are ideal functionalities really ideal?”, Arapinis, Cortier, de Groote, Jacomme, Kremer, CSF’26

Fundamental harm-enabling capabilities

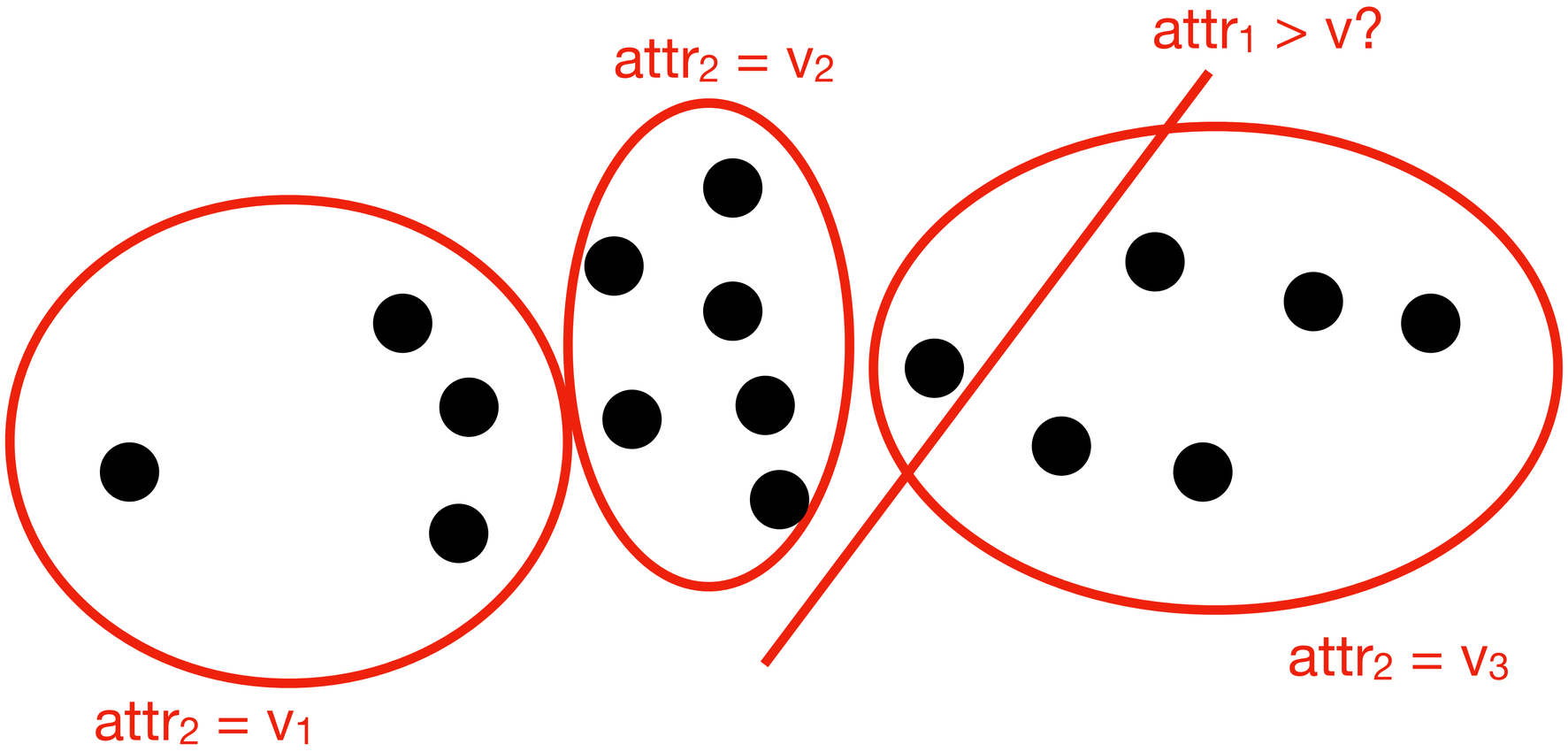


[ACdGJK26] “Are ideal functionalities really ideal?”, Arapinis, Cortier, de Groote, Jacomme, Kremer, CSF’26

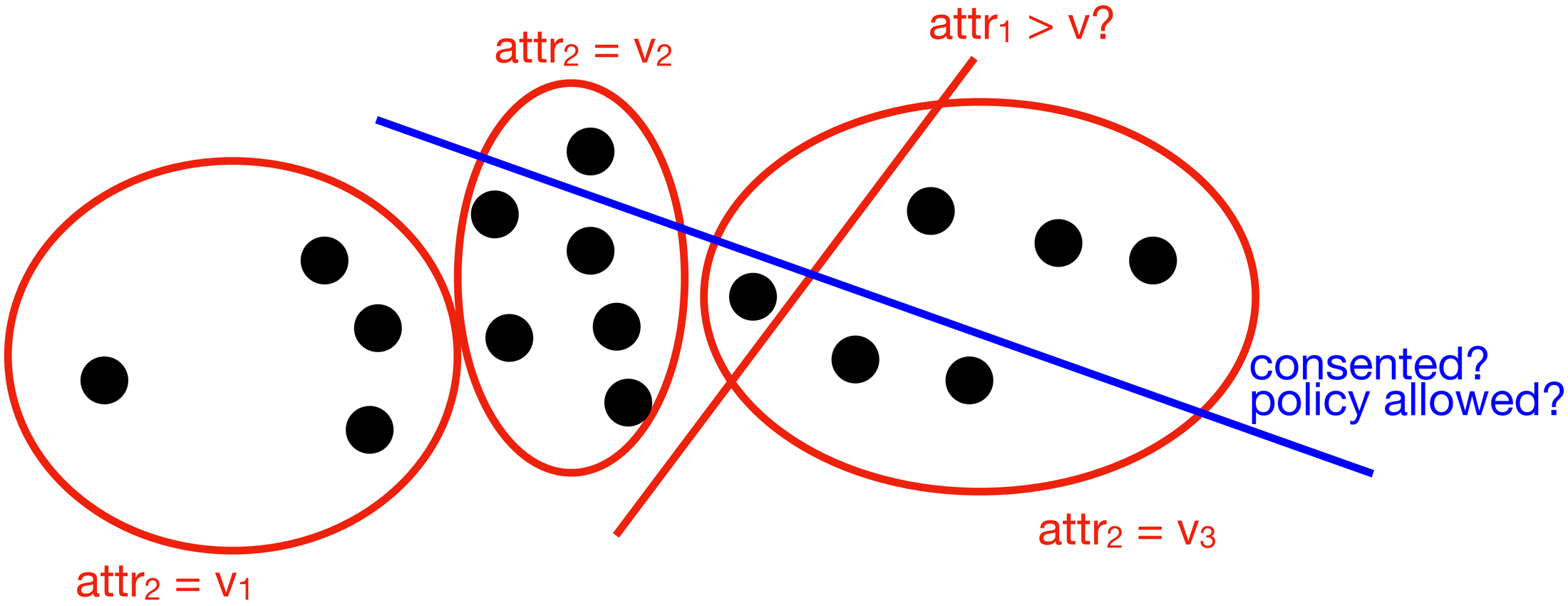
Distinguishing / Tracking



Distinguishing / Tracking

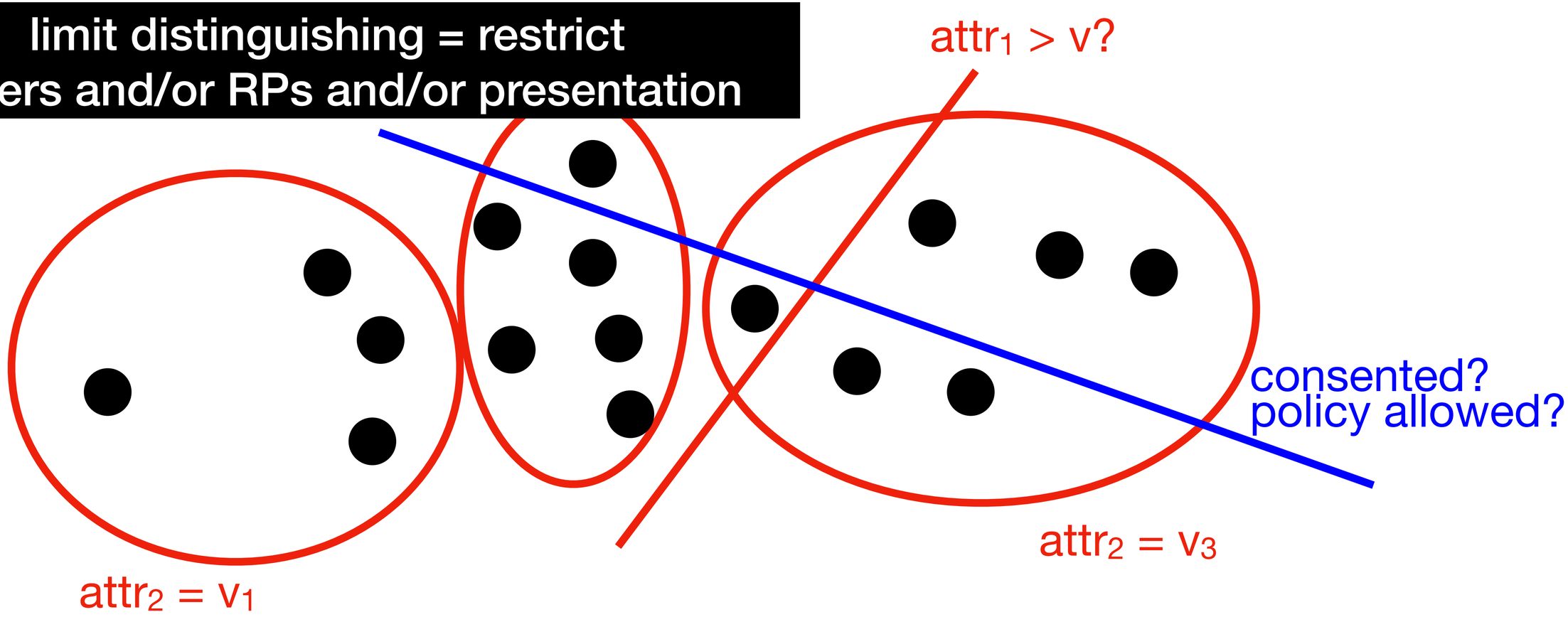


Distinguishing / Tracking



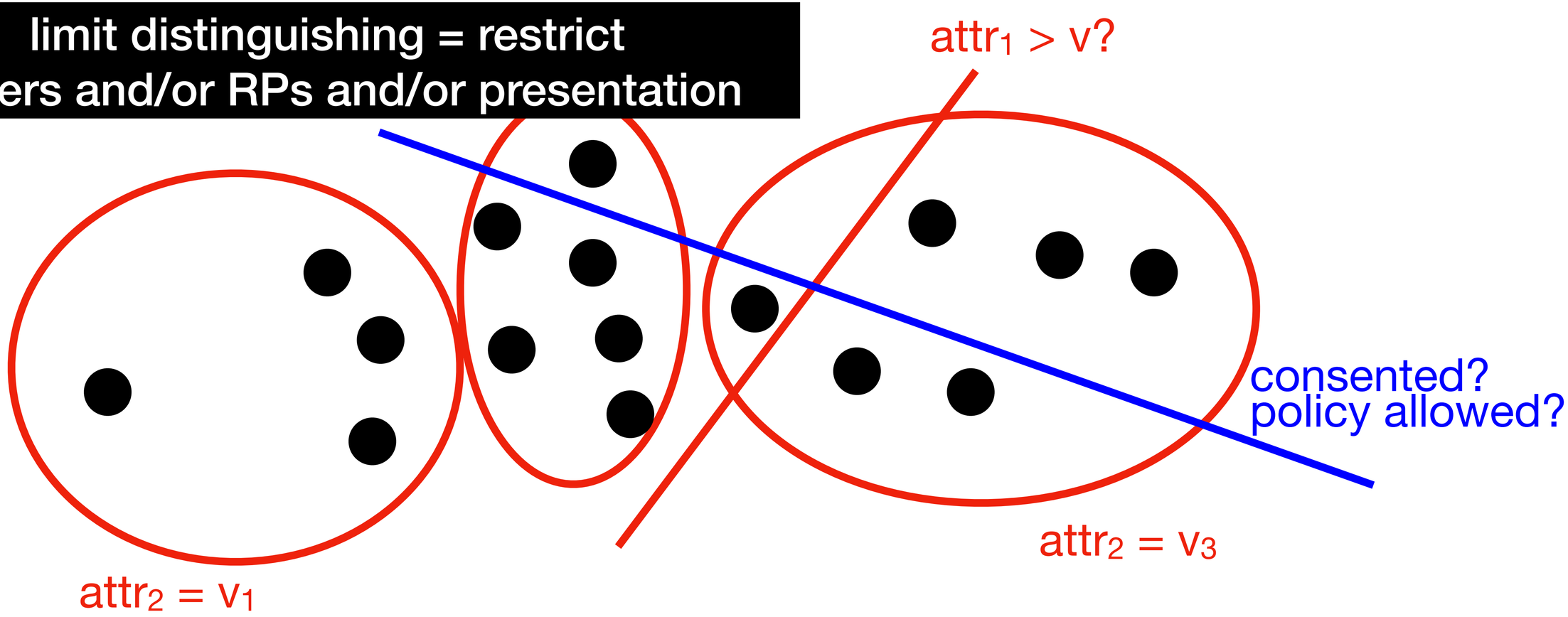
Distinguishing / Tracking

limit distinguishing = restrict issuers and/or RPs and/or presentation



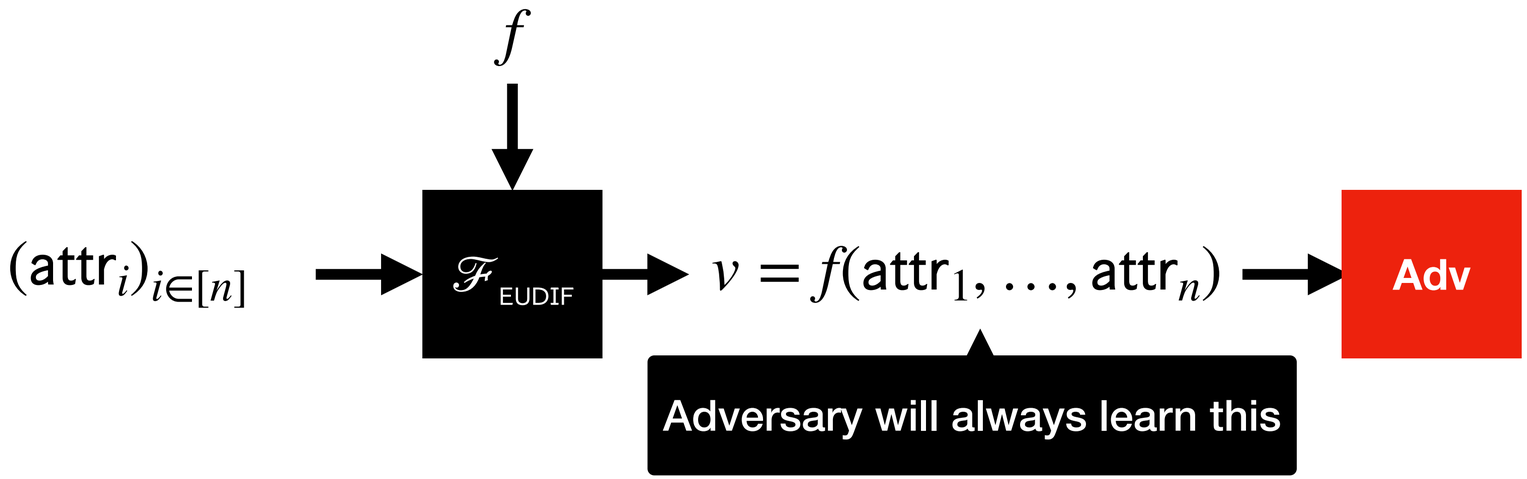
Distinguishing / Tracking

limit distinguishing = restrict issuers and/or RPs and/or presentation

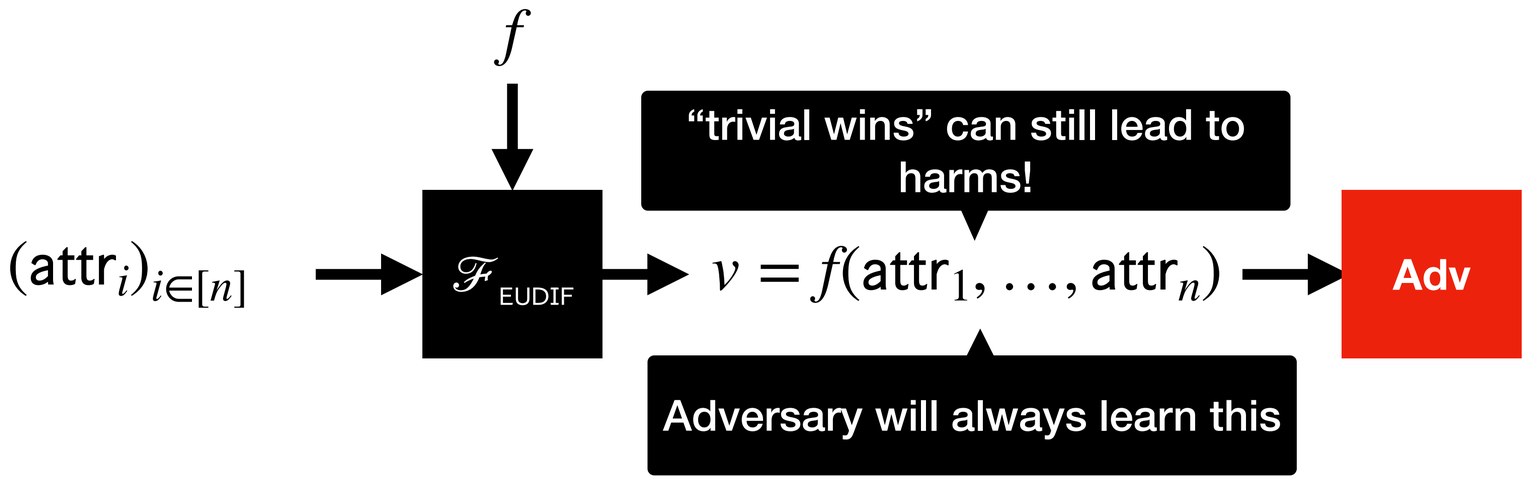


limit trackability = enforce k-anonymity / l-diversity / t-closeness

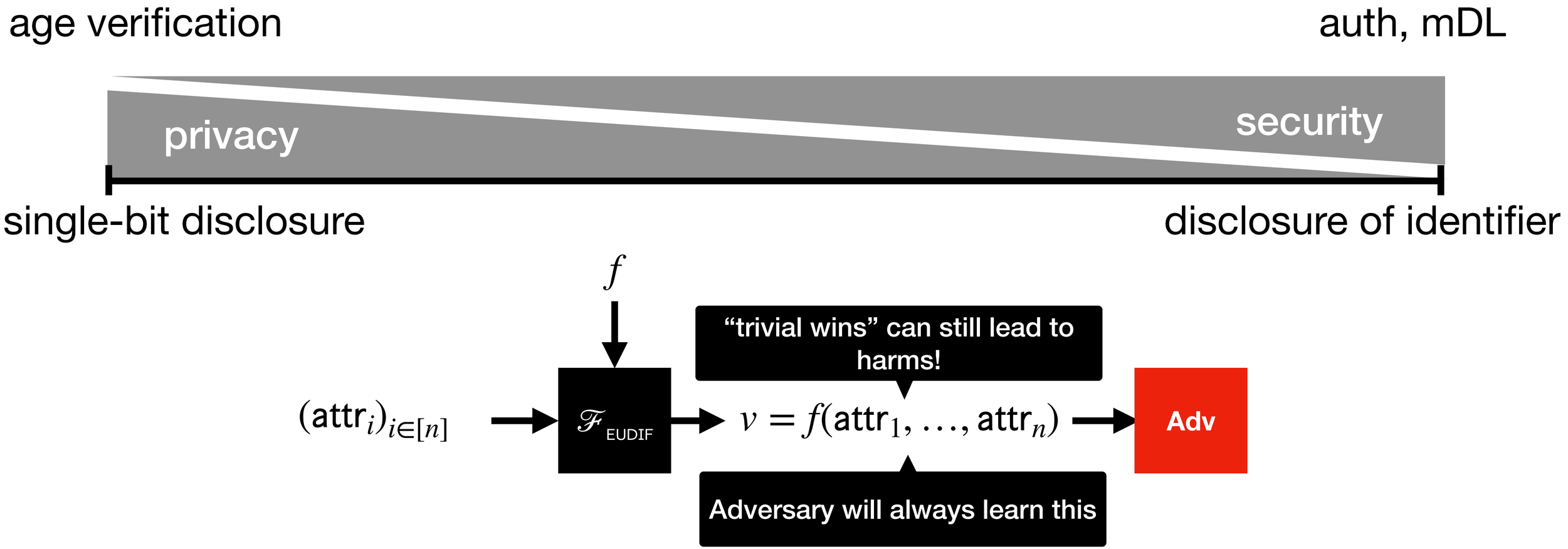
Inference


f
$(\text{attr}_i)_{i\in[n]}$
$\mathcal{F}_{\text{EUDIF}}$
$v = f(\text{attr}_1, \ldots, \text{attr}_n)$
Adv
Adversary will always learn this

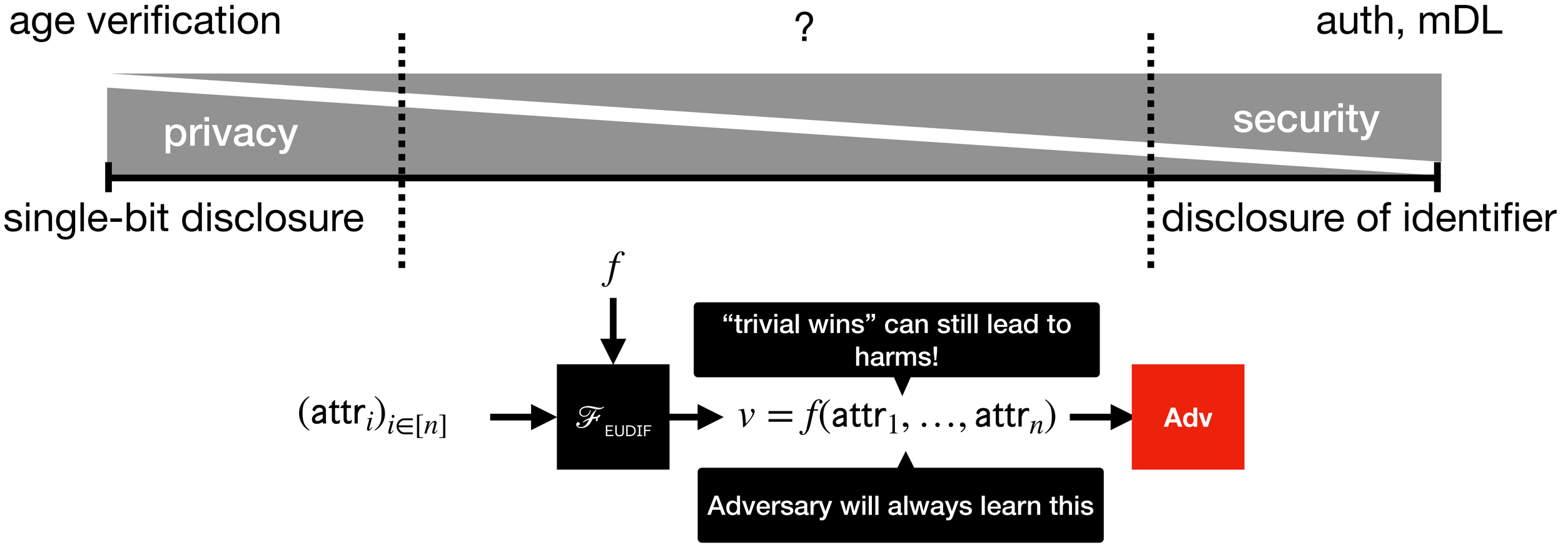
Inference


f
“trivial wins” can still lead to harms!
$(\text{attr}_i)_{i\in[n]}$
$\mathscr{F}_{\text{EUDIF}}$
$v = f(\text{attr}_1, \ldots, \text{attr}_n)$
Adv
Adversary will always learn this

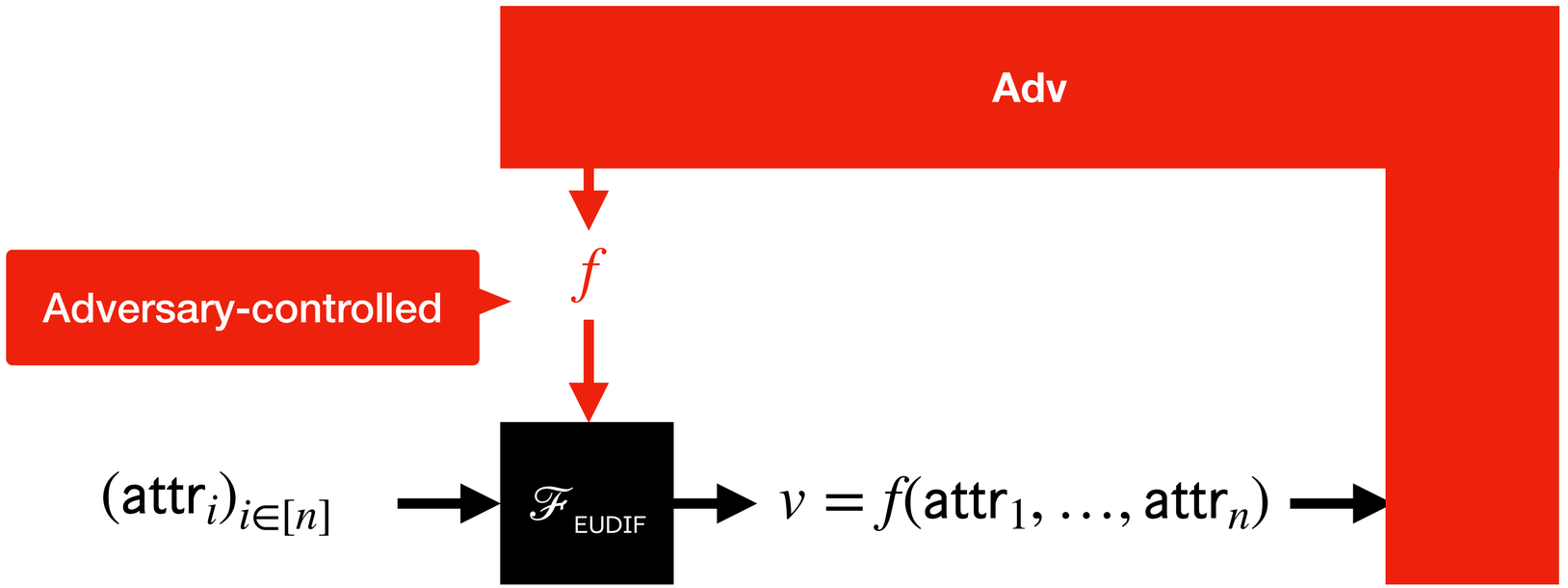
Inference


age verification
auth, mDL
privacy
security
single-bit disclosure
disclosure of identifier
f
“trivial wins” can still lead to harms!
$(\text{attr}_i)_{i\in[n]}$
$\mathcal{F}_{\text{EUDIF}}$
$v = f(\text{attr}_1, \ldots, \text{attr}_n)$
Adv
Adversary will always learn this

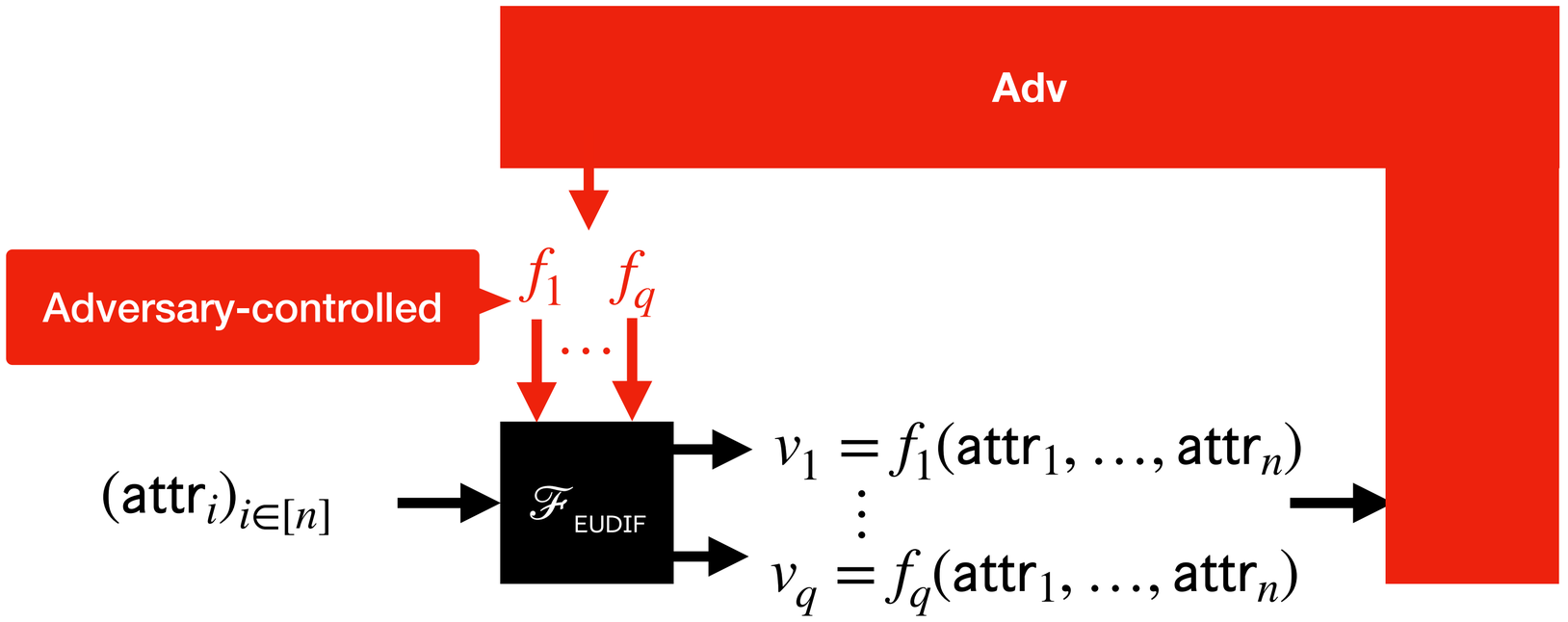
Inference


age verification
?
auth, mDL
privacy
security
single-bit disclosure
disclosure of identifier
f
“trivial wins” can still lead to harms!
$(\text{attr}_i)_{i\in[n]}$
$\mathscr{F}_{\text{EUDIF}}$
$v = f(\text{attr}_1, \ldots, \text{attr}_n)$
Adv
Adversary will always learn this

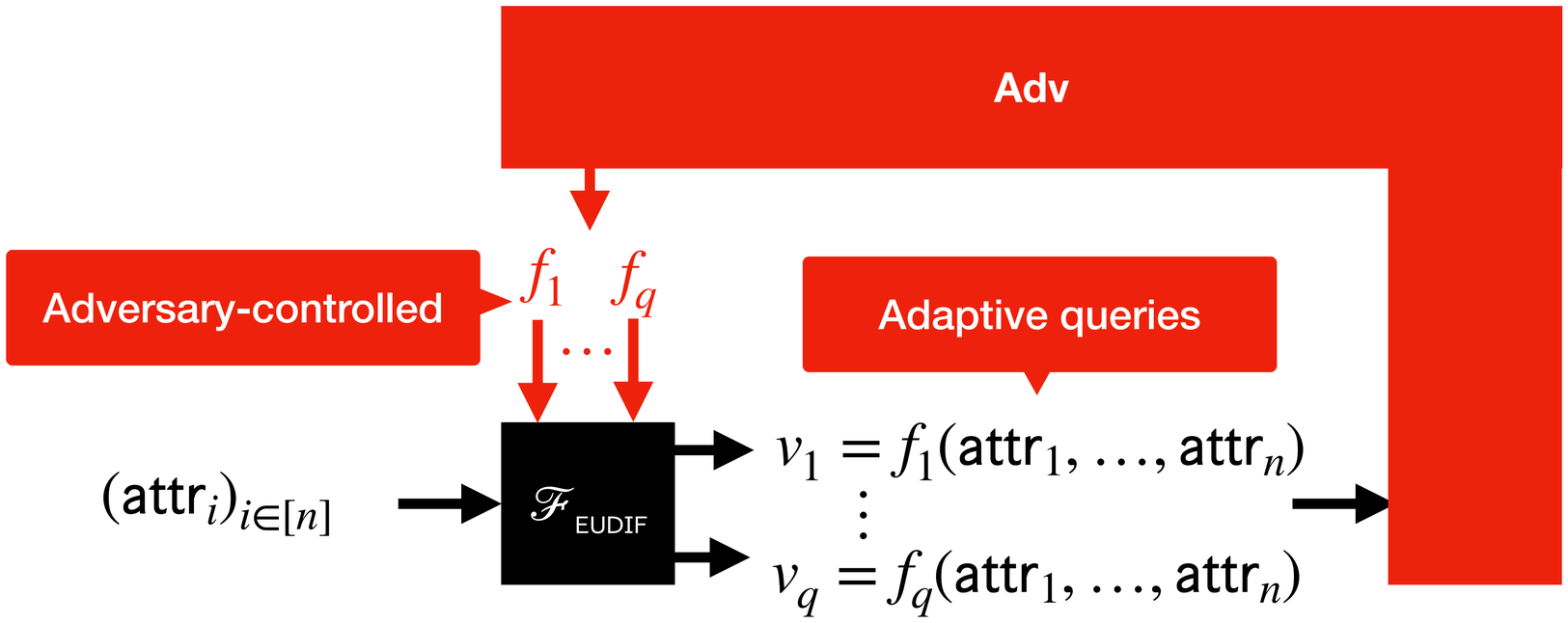
Inference


Adv
Adversary-controlled
f
$(\text{attr}_i)_{i\in[n]}$
$\mathcal{F}_{\text{EUDIF}}$
$v = f(\text{attr}_1, \ldots, \text{attr}_n)$

Inference

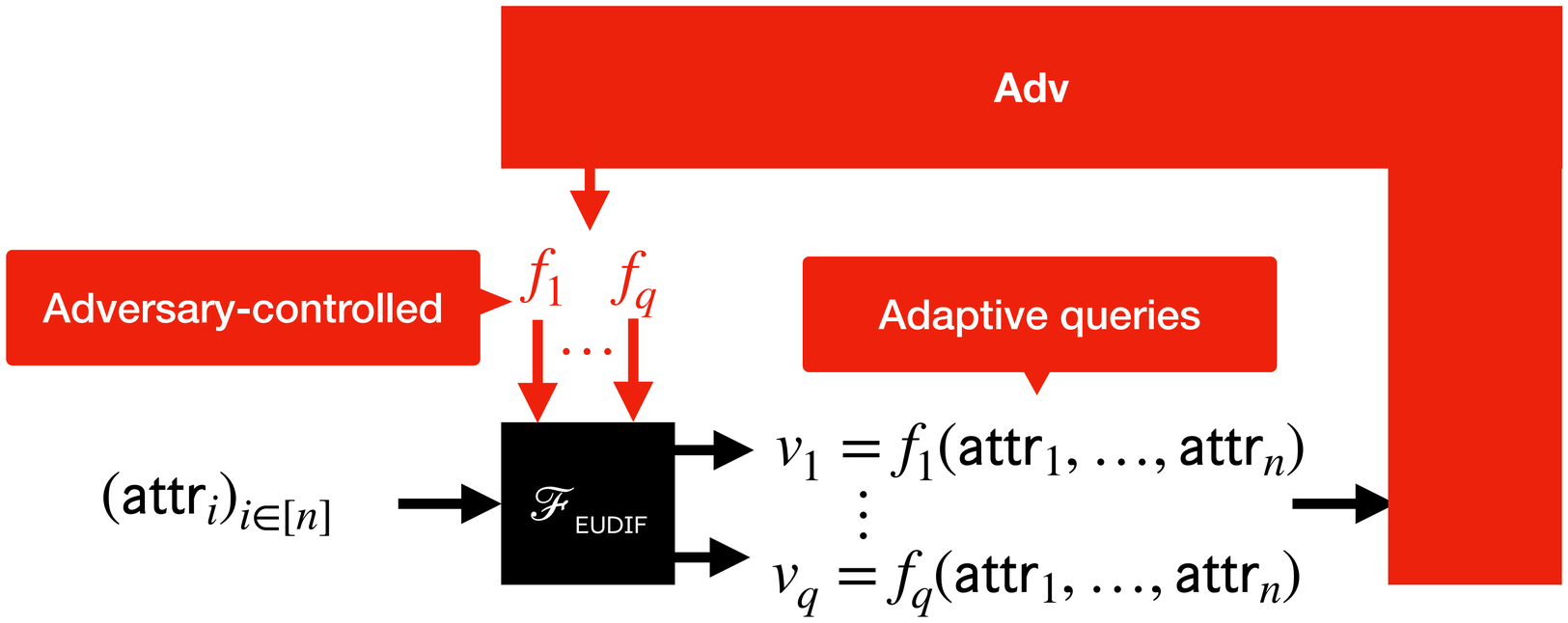

Adv
Adversary-controlled
f_1 f_q
$\cdots$
$(\mathsf{attr}_i)_{i\in[n]}$
$\mathcal{F}_{\text{EUDIF}}$
$v_1 = f_1(\mathsf{attr}_1, \ldots, \mathsf{attr}_n)$
$\vdots$
$v_q = f_q(\mathsf{attr}_1, \ldots, \mathsf{attr}_n)$

Inference


Adv
Adversary-controlled
f_1 f_q
Adaptive queries
$(\mathsf{attr}_i)_{i\in[n]}$
$\mathcal{F}_{\text{EUDIF}}$
$v_1 = f_1(\mathsf{attr}_1, \ldots, \mathsf{attr}_n)$
$v_q = f_q(\mathsf{attr}_1, \ldots, \mathsf{attr}_n)$

Inference

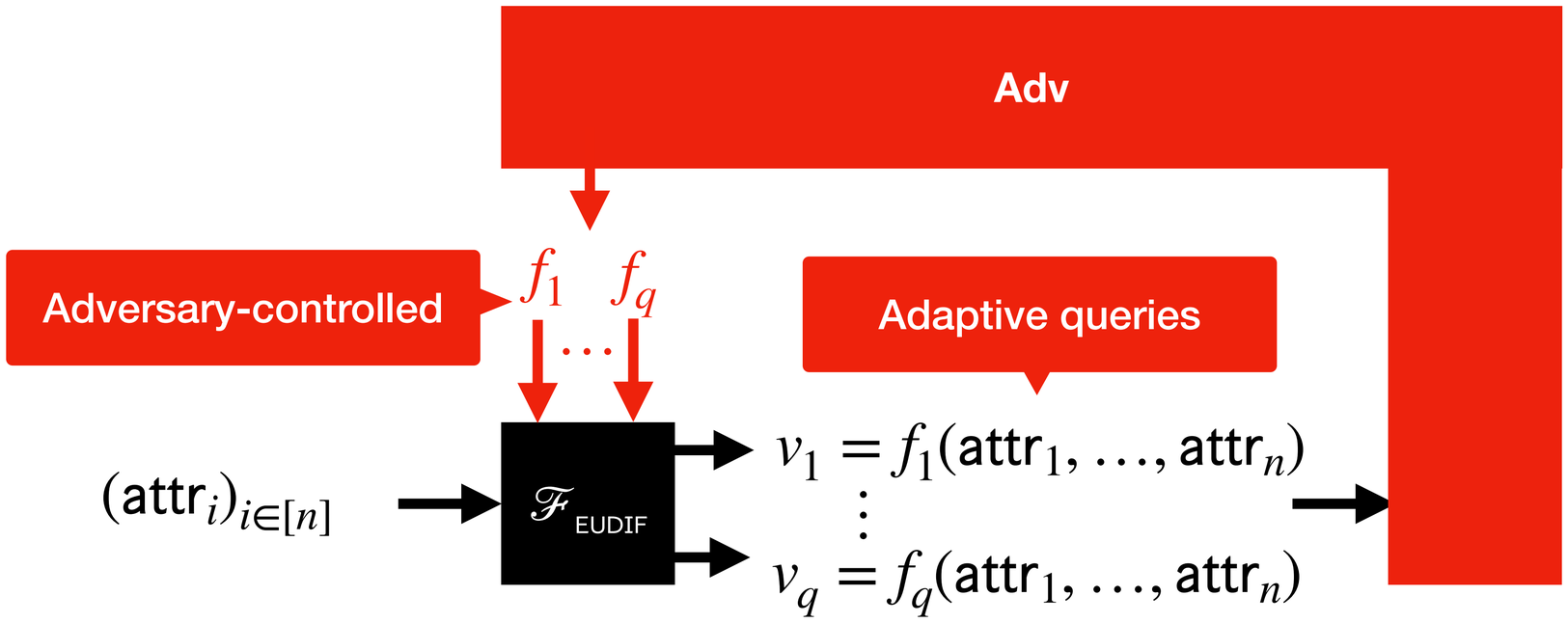
In the EUDIF, adversarial RP can perform arbitrary inference of user's attributes



Inference

In the EUDIF, adversarial RP can perform arbitrary inference of user's attributes

limiting function creep or overidentification = restrict which attributes can be issued/requested



Takeaways

- **PETs are necessary, but not sufficient** to reduce harms from DIFs
- **Properties of PETs can be easily voided by system design**; privacy doesn’t compose nicely, and harms arise at the system-level
- **Policy is the only tool that can reduce harm-enabling capabilities inherent to DIFs**

Takeaways

- **PETs are necessary, but not sufficient** to reduce harms from DIFs
- **Properties of PETs can be easily voided by system design**; privacy doesn't compose nicely, and harms arise at the system-level
- **Policy is the only tool that can reduce harm-enabling capabilities inherent to DIFs**
- **Beware of privacy-washing**: don't let your nice crypto work mislead people into building/using a harmful system

Takeaways

- **PETs are necessary, but not sufficient** to reduce harms from DIFs
- **Properties of PETs can be easily voided by system design**; privacy doesn't compose nicely, and harms arise at the system-level
- **Policy is the only tool that can reduce harm-enabling capabilities inherent to DIFs**
- **Beware of privacy-washing**: don't let your nice crypto work mislead people into building/using a harmful system
- **DIFs with PETs are much better than DIFs without PETs**
- **Anonymous credentials have value in a DIF beyond their privacy properties**
- **Still plenty of opportunities for PETs development**: efficient/non-SE-dependent device binding, PIR, anonymous communication

Takeaways

- **PETs are necessary, but not sufficient** to reduce harms from DIFs
- **Properties of PETs can be easily voided by system design**; privacy doesn't compose nicely, and harms arise at the system-level
- **Policy is the only tool that can reduce harm-enabling capabilities inherent to DIFs**
- **Beware of privacy-washing**: don't let your nice crypto work mislead people into building/using a harmful system
- **DIFs with PETs are much better than DIFs without PETs**
- **Anonymous credentials have value in a DIF beyond their privacy properties**
- **Still plenty of opportunities for PETs development**: efficient/non-SE-dependent device binding, PIR, anonymous communication

ia.cr/2026/867

Thanks! Questions?

christian.knabenhans@epfl.ch
sveitch@ethz.ch