ETH zürich

ESORICS

Enforcing the GDPR

ESORICS 2023 | The Hague, September 25, 2023

François Hublet 

`francois.hublet@inf.ethz.ch`

David Basin

`basin@inf.ethz.ch`

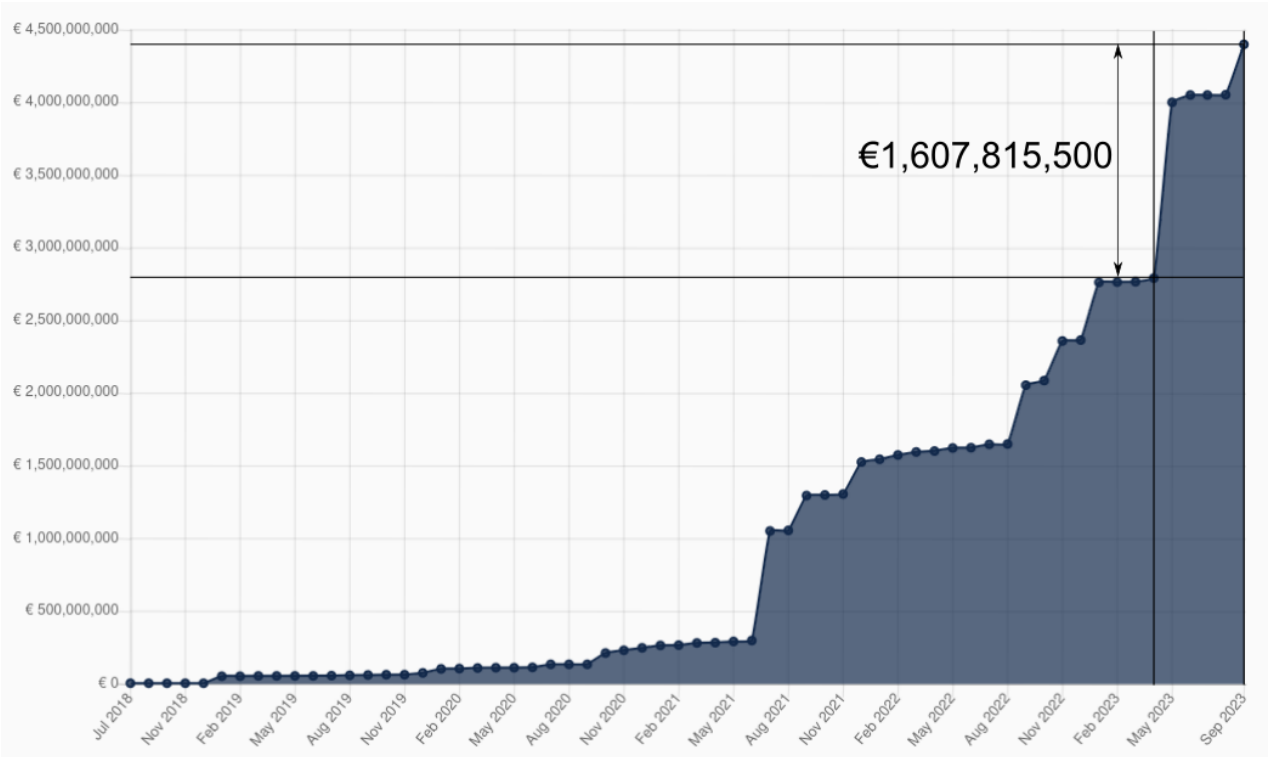
Srđan Krstić

`srdan.krstic@inf.ethz.ch`

Institute of Information Security, Department of Computer Science, ETH Zürich

More privacy thanks to the GDPR?

Since the time our paper was written, over €**1.5 billion** in fines have been imposed on entities violating the GDPR.



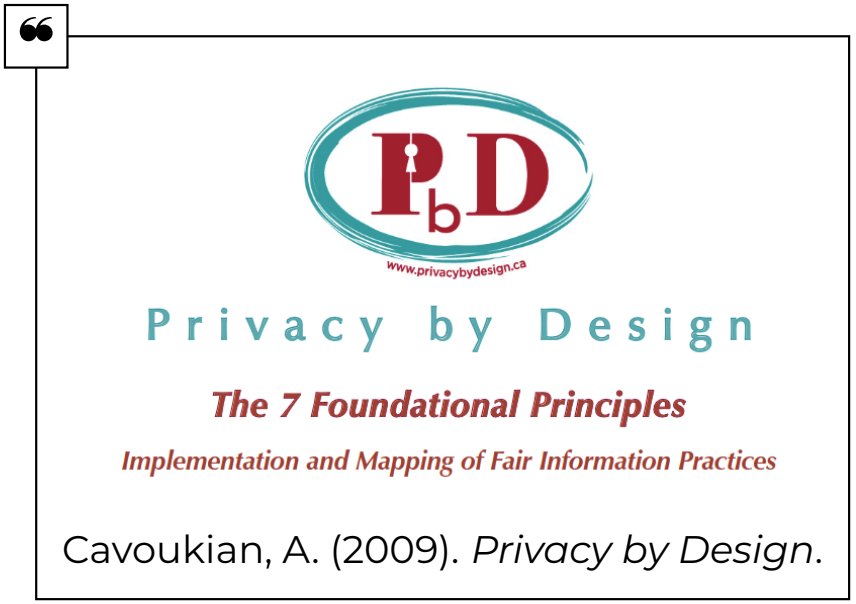
Source: `www.enforcementtracker.com`

How can we do better?

Be proactive!

How can we do better?

Be proactive!



Cavoukian, A. (2009). *Privacy by Design*.

How can we do better?

Be proactive!

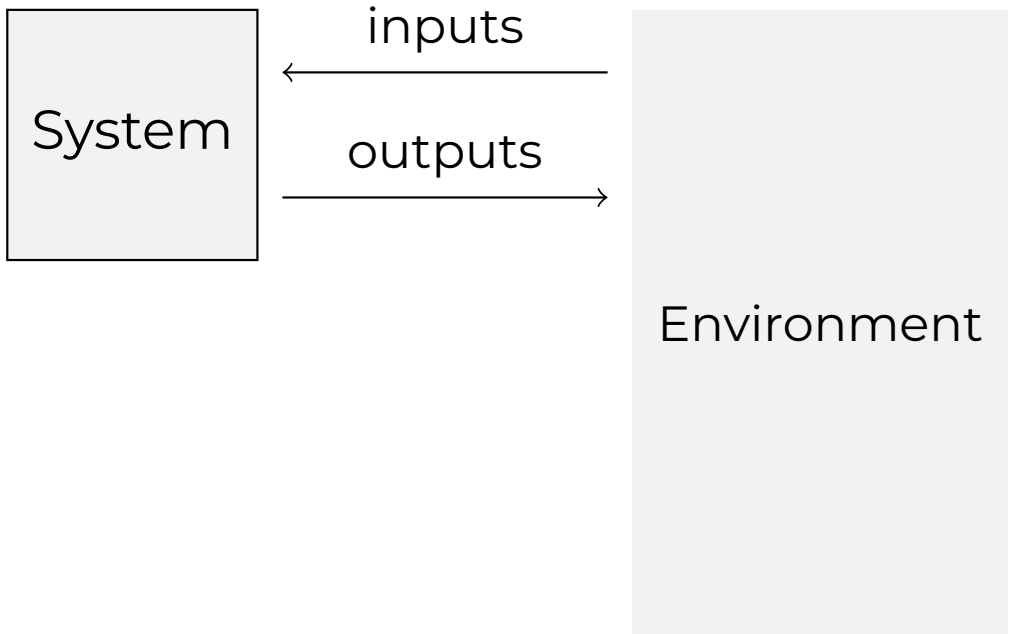
Cavoukian, A. (2009). *Privacy by Design*.

Data protection by design and by default

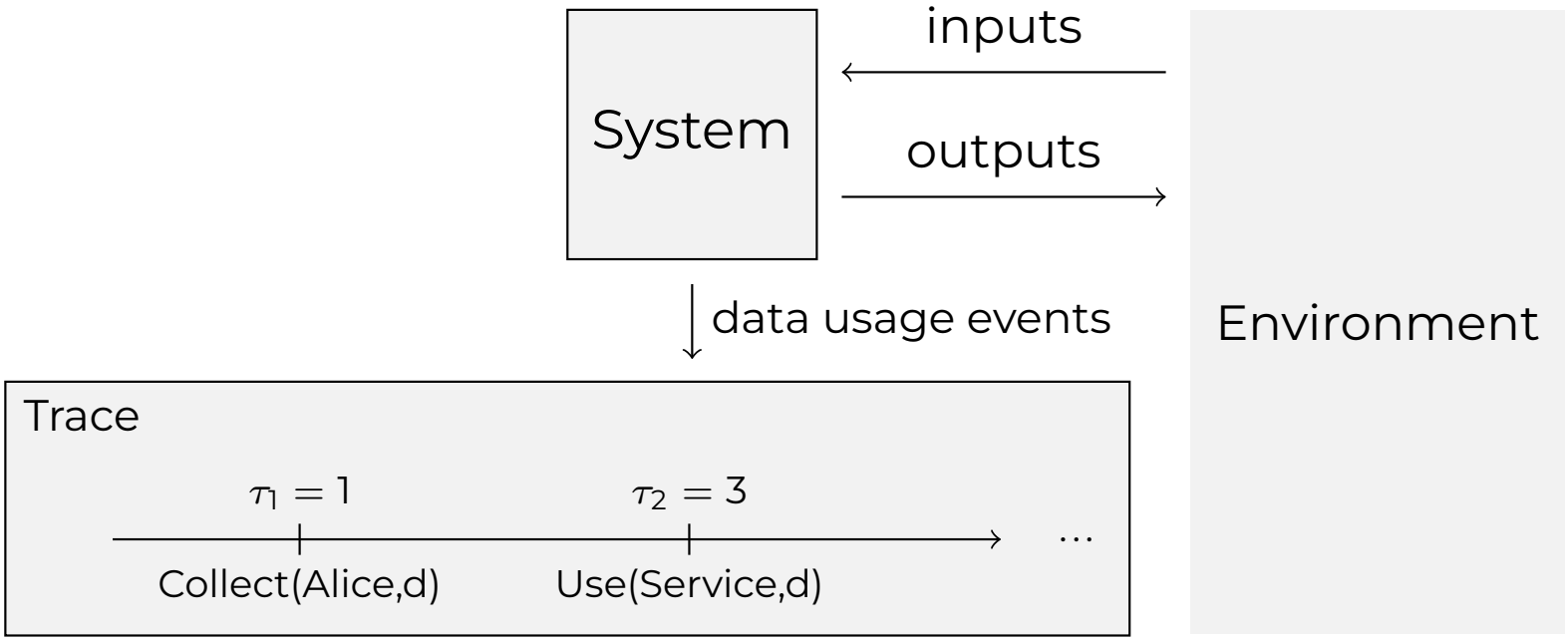
Taking into account the state of the art, the cost of implementation and the nature, scope, context and purposes of processing [...] ***the controller shall [...] implement appropriate technical and organisational measures [...] in order to meet the requirements of this Regulation*** *and protect the rights of data subjects.*

— GDPR, art. 25(1)

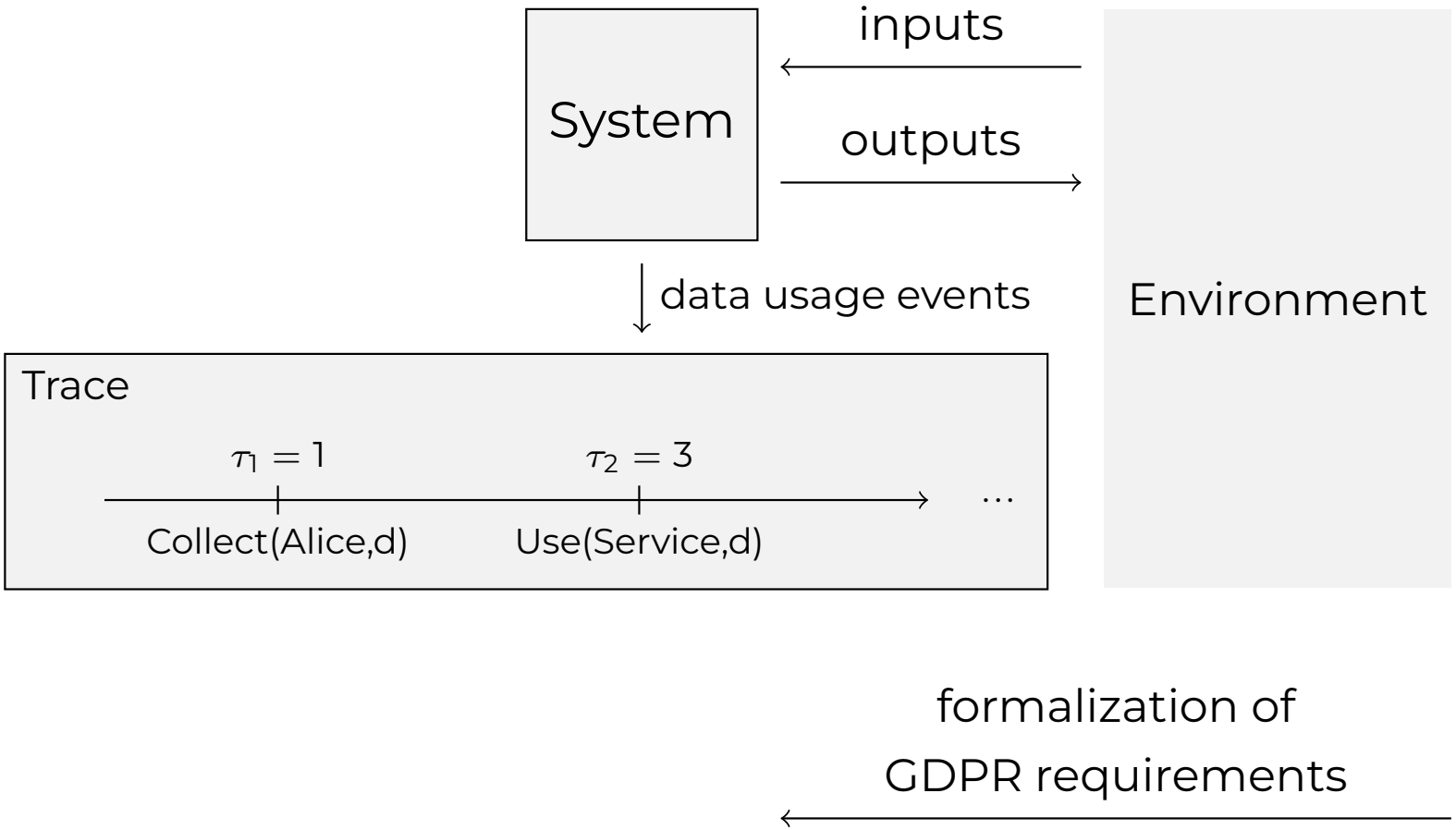
This paper: PbD using Runtime Enforcement



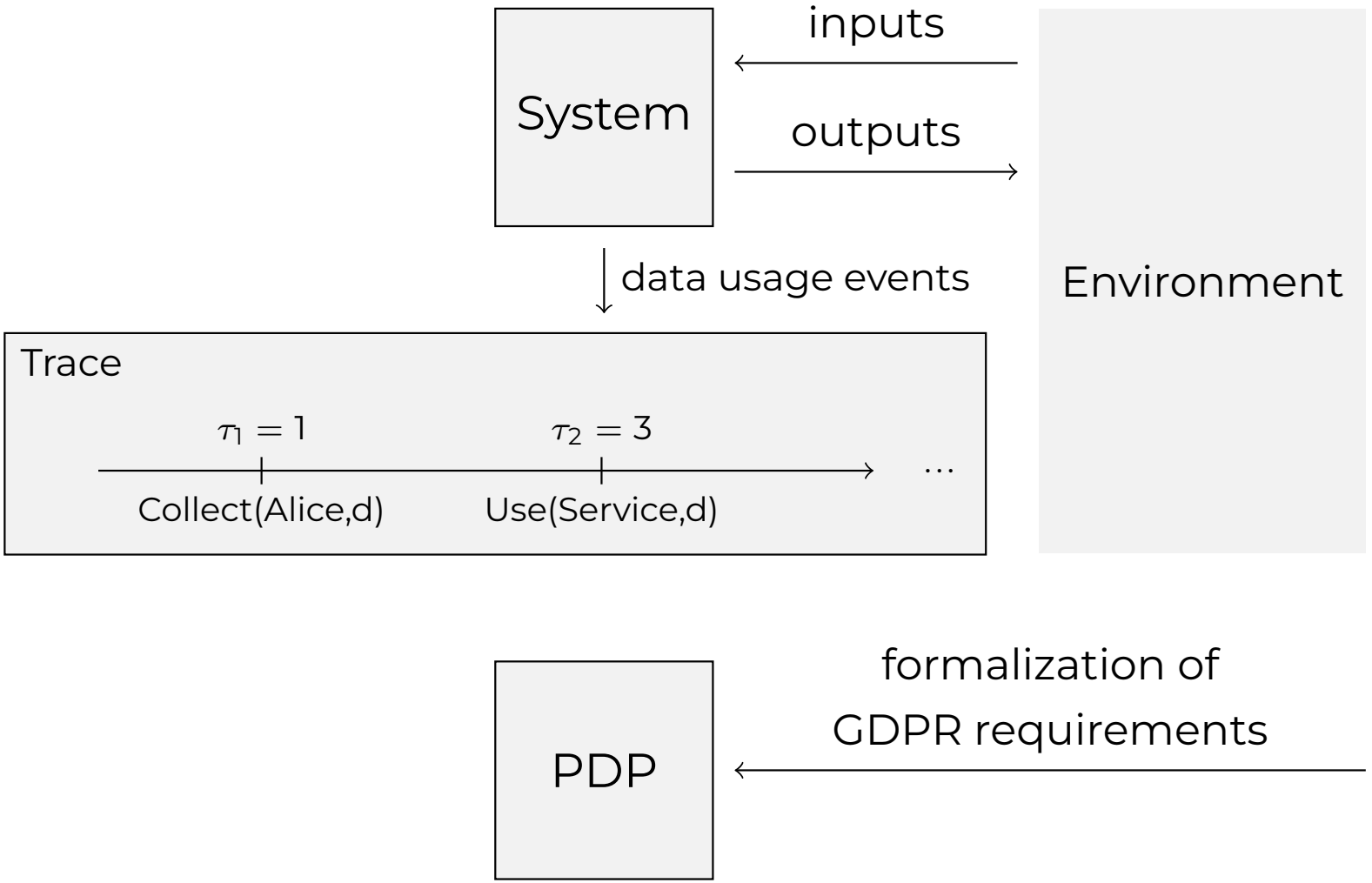
This paper: PbD using Runtime Enforcement



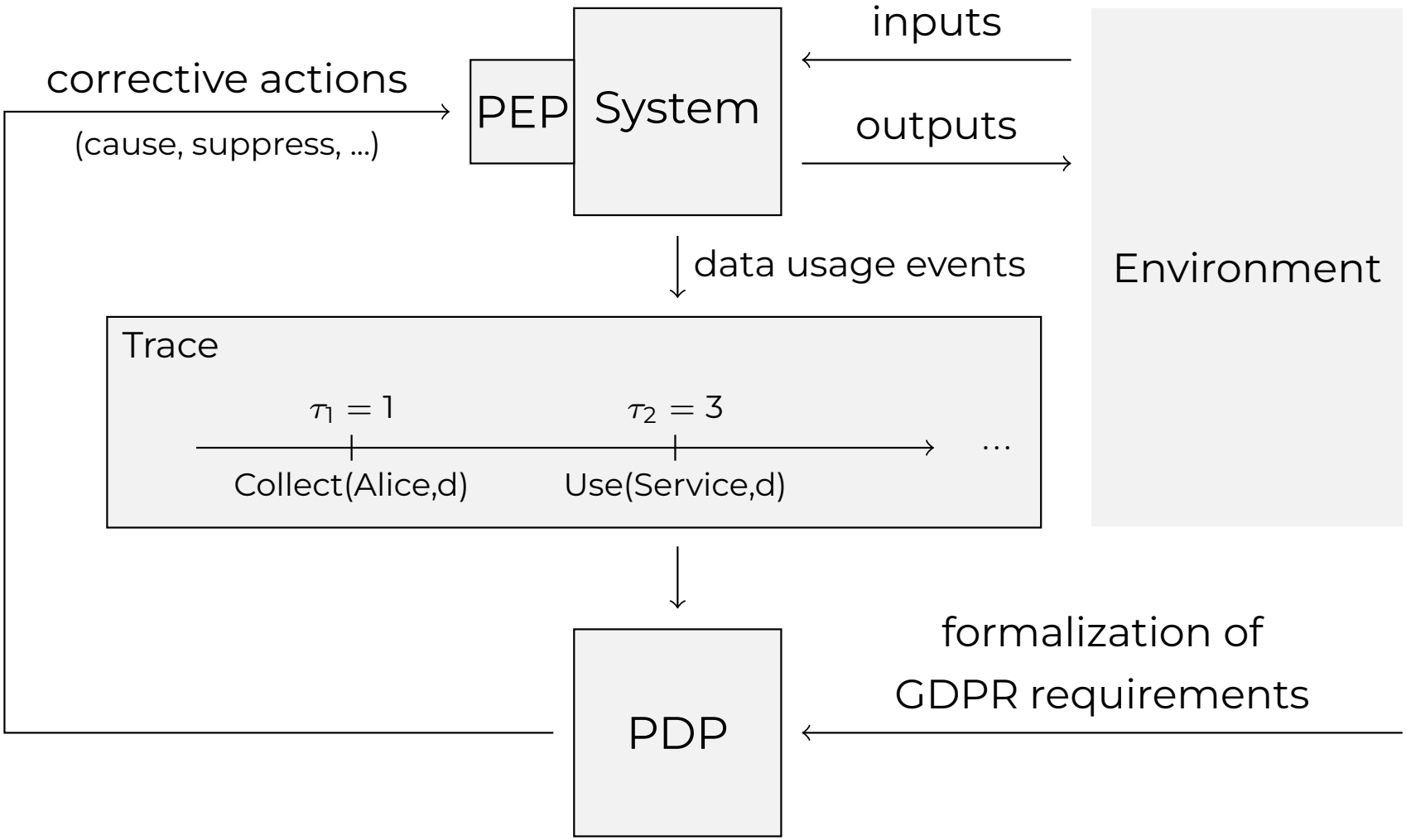
This paper: PbD using Runtime Enforcement



This paper: PbD using Runtime Enforcement



This paper: PbD using Runtime Enforcement



This paper: Runtime Enforcement of the GDPR

Challenges

Contributions

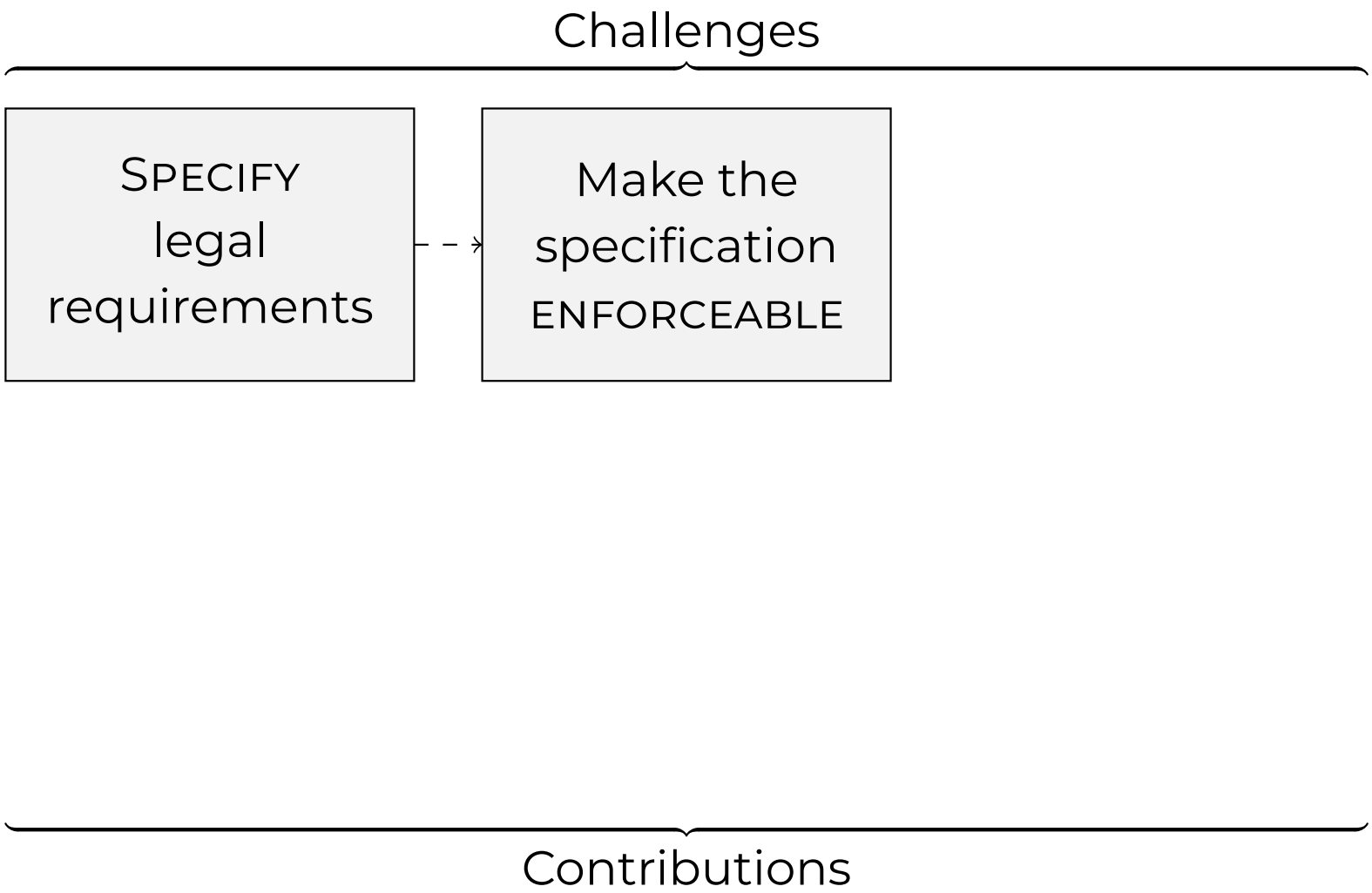
This paper: Runtime Enforcement of the GDPR

Challenges

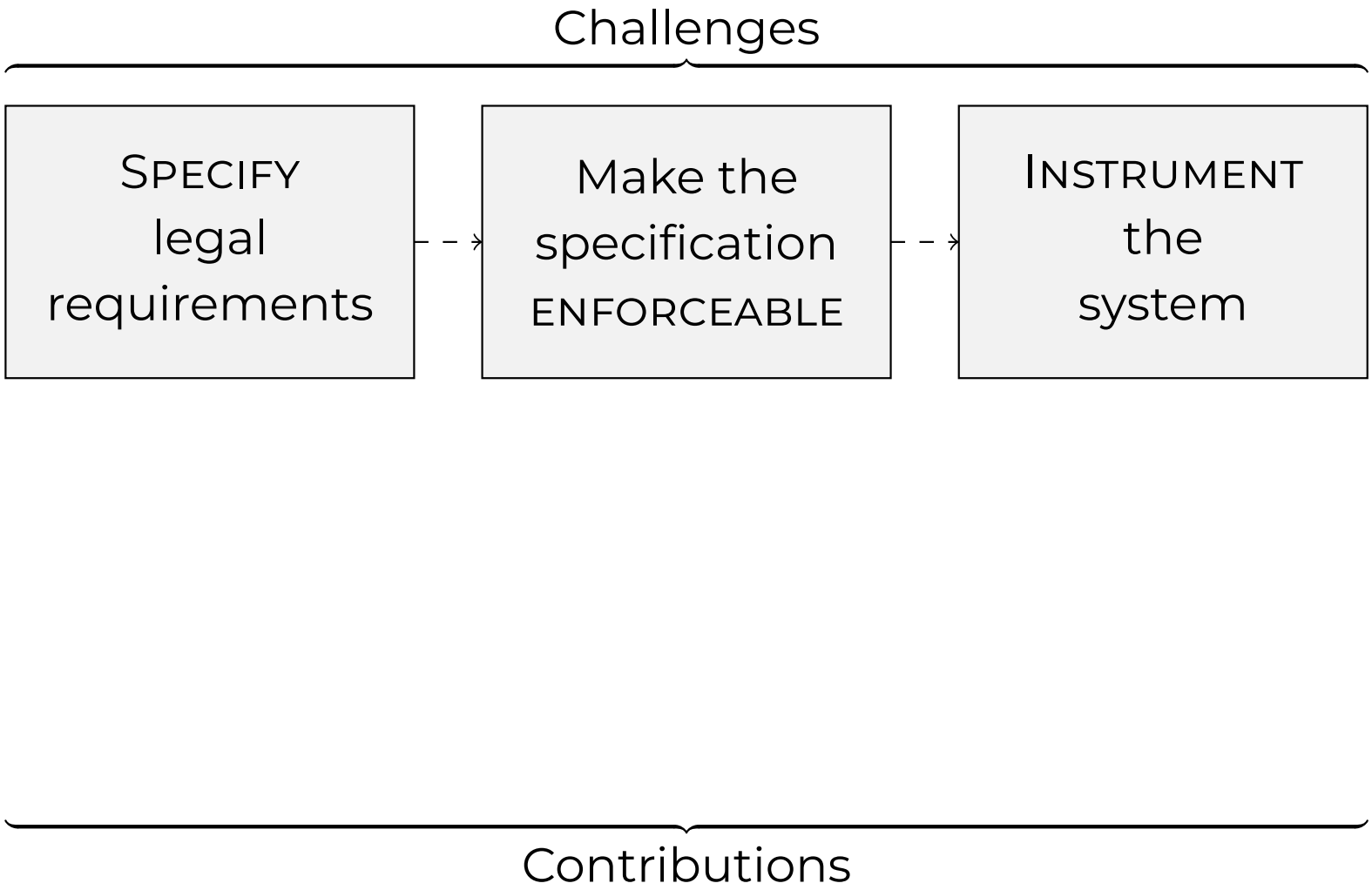
SPECIFY legal requirements

Contributions

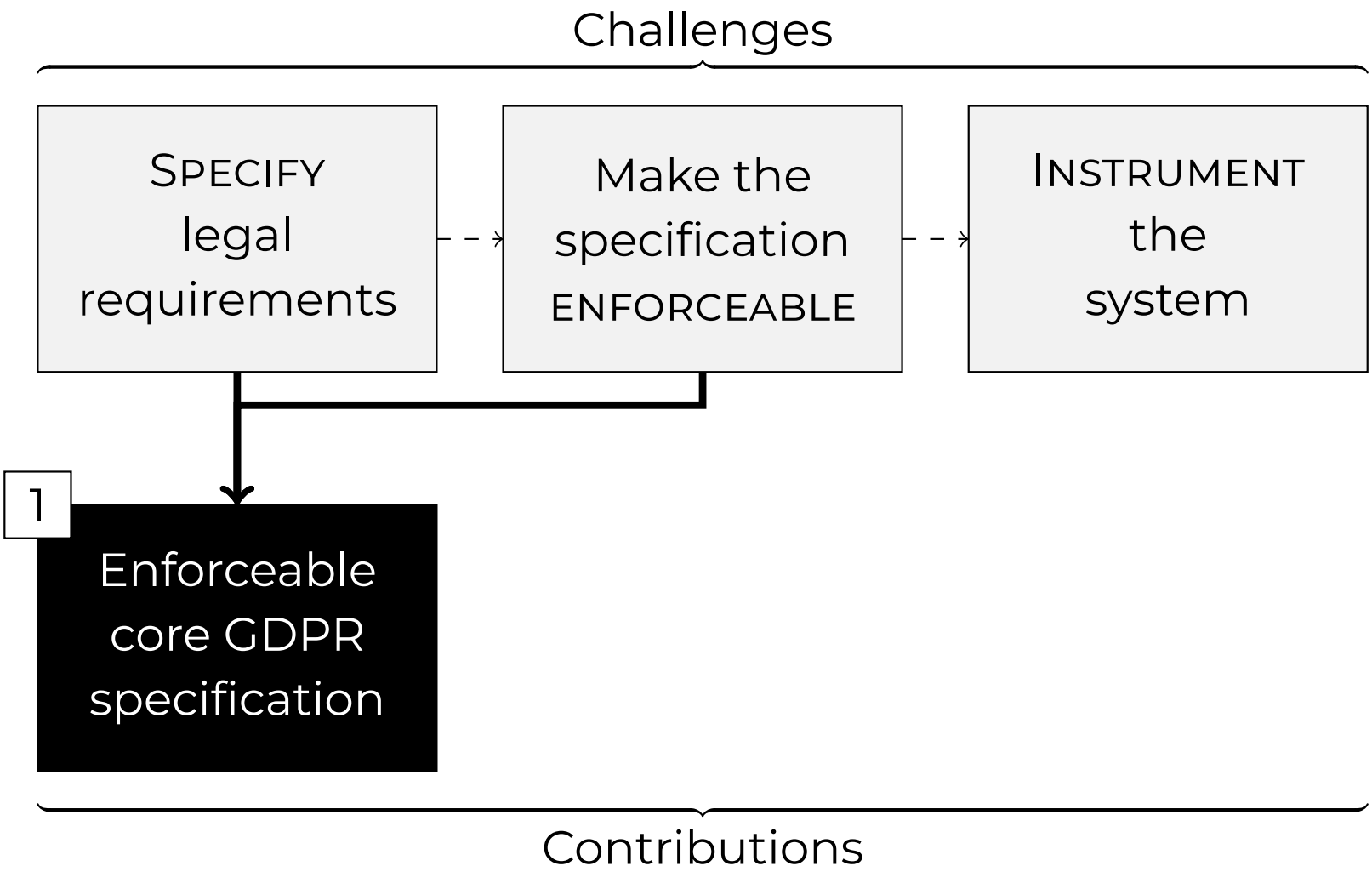
This paper: Runtime Enforcement of the GDPR



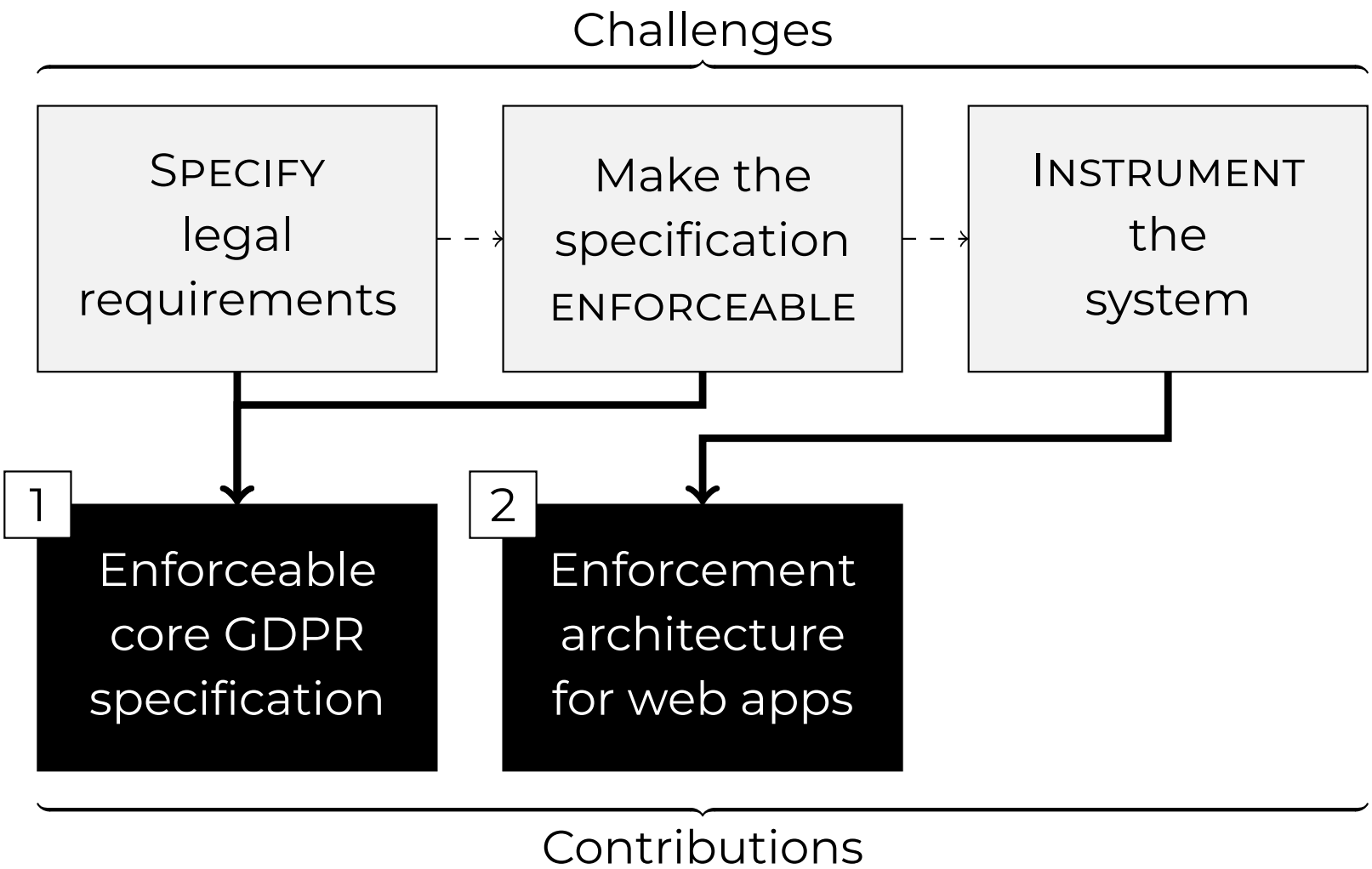
This paper: Runtime Enforcement of the GDPR



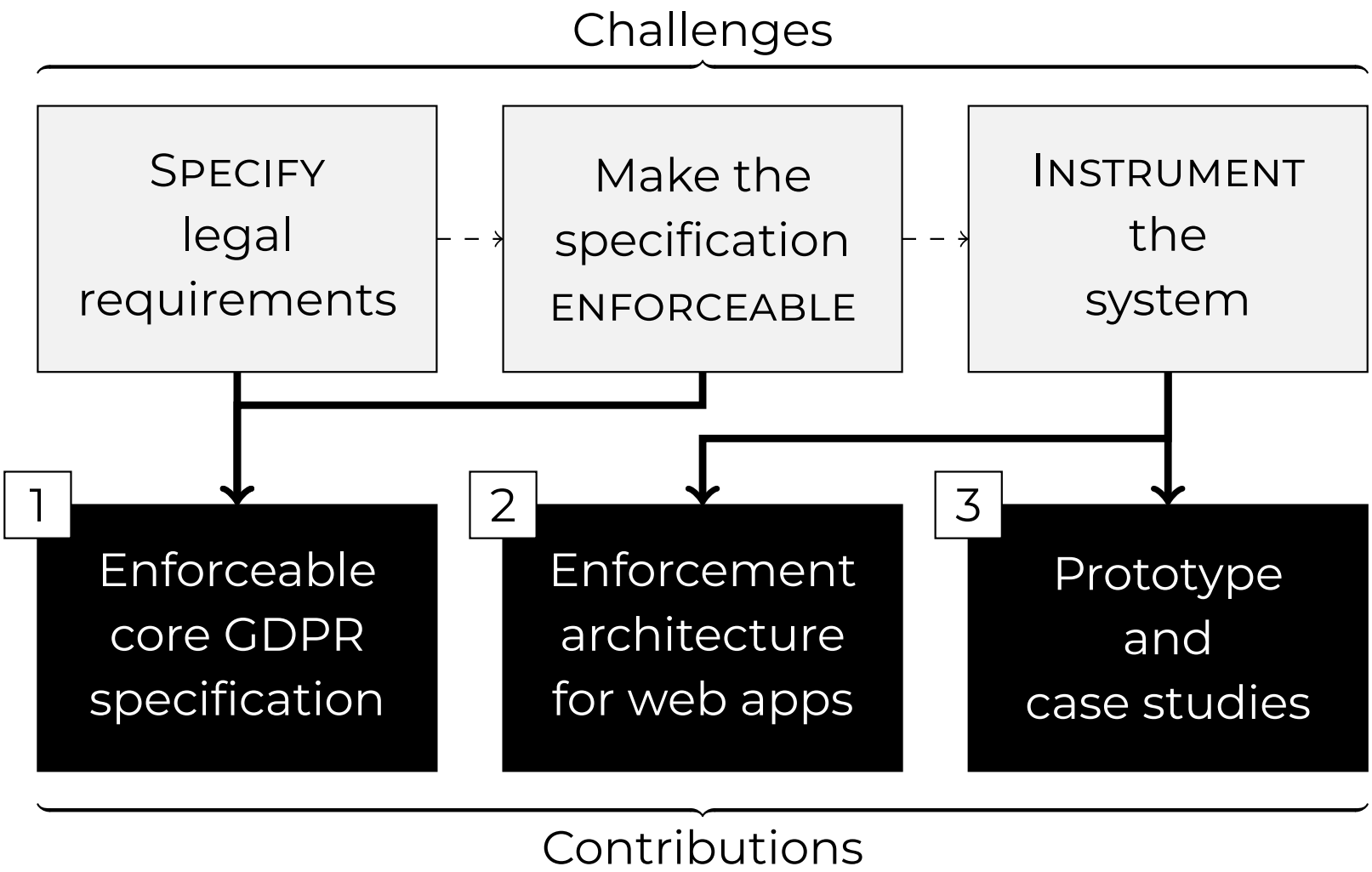
This paper: Runtime Enforcement of the GDPR



This paper: Runtime Enforcement of the GDPR



This paper: Runtime Enforcement of the GDPR



1

Enforceable core GDPR specification

Previous work: Monitorable GDPR specification

> | Article | MFOTL Formula |
> |---|---|
> | 5(1)(c)
Data minimisation | collect (*data*, *dataid*, *dsid*) IMPLIES
EVENTUALLY use(*data*, *dataid*, *dsid*) |
> | 5(1)(e)
Storage limitation | collect (*data*, *dataid*, *dsid*) IMPLIES
EVENTUALLY delete(*data*, *dataid*, *dsid*) |
> | 6(1)
Lawful processing | use(*data*, *dataid*, *dsid*) IMPLIES ONCE
(ds_consent(*dsid*, *data*) OR legal_grounds(*dsid*, *data*)) |
> | 7(3)
Consent | use(*data*, *dataid*, *dsid*) IMPLIES
(ONCE legal_grounds(*dsid*, *data*)) OR
(NOT ds_revoke(*dsid*, *data*) SINCE ds_consent(*dsid*, *data*)) |
>
> Arfelt, E. et al. (2019). Monitoring the GDPR. *ESORICS'19*.

- Focuses on a simple specification of **data subject rights**
- Uses **Metric First-Order Temporal Logic** (MFOTL)
- **Monitors** traces for GDPR violations

Previous work: MFOTL enforcement

Algorithm 1. Function enf

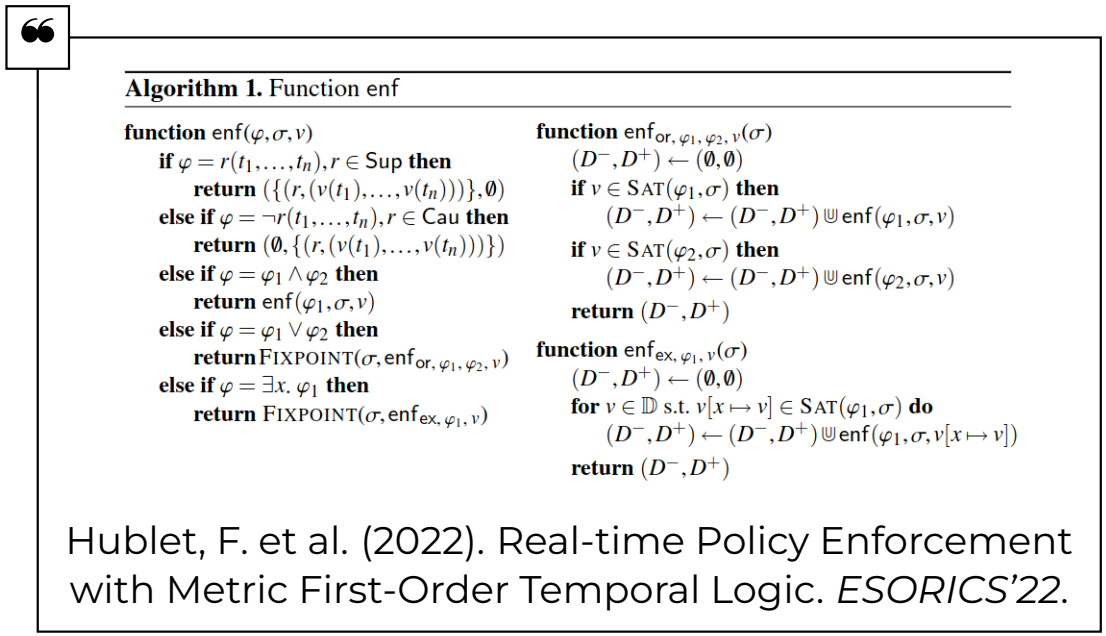
function $\mathsf{enf}(\varphi, \sigma, v)$
  if $\varphi = r(t_1, \ldots, t_n), r \in \mathsf{Sup}$ **then**
    return $(\{(r, (v(t_1), \ldots, v(t_n)))\}, \emptyset)$
  else if $\varphi = \neg r(t_1, \ldots, t_n), r \in \mathsf{Cau}$ **then**
    return $(\emptyset, \{(r, (v(t_1), \ldots, v(t_n)))\})$
  else if $\varphi = \varphi_1 \wedge \varphi_2$ **then**
    return $\mathsf{enf}(\varphi_1, \sigma, v)$
  else if $\varphi = \varphi_1 \vee \varphi_2$ **then**
    return $\textsc{Fixpoint}(\sigma, \mathsf{enf}_{\mathsf{or}, \varphi_1, \varphi_2, v})$
  else if $\varphi = \exists x.\ \varphi_1$ **then**
    return $\textsc{Fixpoint}(\sigma, \mathsf{enf}_{\mathsf{ex}, \varphi_1, v})$

function $\mathsf{enf}_{\mathsf{or}, \varphi_1, \varphi_2, v}(\sigma)$
  $(D^-, D^+) \leftarrow (\emptyset, \emptyset)$
  if $v \in \textsc{Sat}(\varphi_1, \sigma)$ **then**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_1, \sigma, v)$
  if $v \in \textsc{Sat}(\varphi_2, \sigma)$ **then**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_2, \sigma, v)$
  return (D^-, D^+)

function $\mathsf{enf}_{\mathsf{ex}, \varphi_1, v}(\sigma)$
  $(D^-, D^+) \leftarrow (\emptyset, \emptyset)$
  for $v \in \mathbb{D}$ s.t. $v[x \mapsto v] \in \textsc{Sat}(\varphi_1, \sigma)$ **do**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_1, \sigma, v[x \mapsto v])$
  return (D^-, D^+)

Hublet, F. et al. (2022). Real-time Policy Enforcement with Metric First-Order Temporal Logic. *ESORICS'22*.

Previous work: MFOTL enforcement



Algorithm 1. Function enf

function $\mathsf{enf}(\varphi, \sigma, v)$
  if $\varphi = r(t_1, \ldots, t_n), r \in \mathsf{Sup}$ **then**
    return $(\{(r, (v(t_1), \ldots, v(t_n)))\}, \emptyset)$
  else if $\varphi = \neg r(t_1, \ldots, t_n), r \in \mathsf{Cau}$ **then**
    return $(\emptyset, \{(r, (v(t_1), \ldots, v(t_n)))\})$
  else if $\varphi = \varphi_1 \wedge \varphi_2$ **then**
    return $\mathsf{enf}(\varphi_1, \sigma, v)$
  else if $\varphi = \varphi_1 \vee \varphi_2$ **then**
    return $\textsc{Fixpoint}(\sigma, \mathsf{enf}_{\mathsf{or}, \varphi_1, \varphi_2, v})$
  else if $\varphi = \exists x.\ \varphi_1$ **then**
    return $\textsc{Fixpoint}(\sigma, \mathsf{enf}_{\mathsf{ex}, \varphi_1, v})$

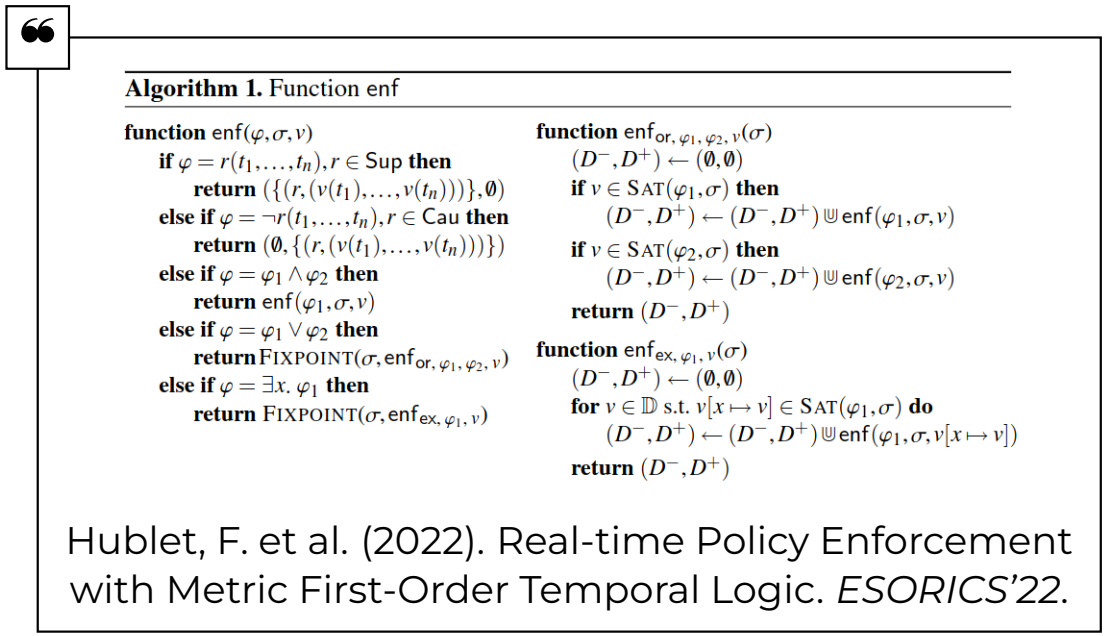
function $\mathsf{enf}_{\mathsf{or}, \varphi_1, \varphi_2, v}(\sigma)$
  $(D^-, D^+) \leftarrow (\emptyset, \emptyset)$
  if $v \in \textsc{Sat}(\varphi_1, \sigma)$ **then**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_1, \sigma, v)$
  if $v \in \textsc{Sat}(\varphi_2, \sigma)$ **then**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_2, \sigma, v)$
  return (D^-, D^+)

function $\mathsf{enf}_{\mathsf{ex}, \varphi_1, v}(\sigma)$
  $(D^-, D^+) \leftarrow (\emptyset, \emptyset)$
  for $v \in \mathbb{D}$ s.t. $v[x \mapsto v] \in \textsc{Sat}(\varphi_1, \sigma)$ **do**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_1, \sigma, v[x \mapsto v])$
  return (D^-, D^+)

Hublet, F. et al. (2022). Real-time Policy Enforcement with Metric First-Order Temporal Logic. *ESORICS'22*.

- Supports **suppressable** and **causable** events

Previous work: MFOTL enforcement



Algorithm 1. Function enf

function $\mathsf{enf}(\varphi, \sigma, v)$
  if $\varphi = r(t_1, \ldots, t_n), r \in \mathsf{Sup}$ **then**
    return $(\{(r, (v(t_1), \ldots, v(t_n)))\}, \emptyset)$
  else if $\varphi = \neg r(t_1, \ldots, t_n), r \in \mathsf{Cau}$ **then**
    return $(\emptyset, \{(r, (v(t_1), \ldots, v(t_n)))\})$
  else if $\varphi = \varphi_1 \wedge \varphi_2$ **then**
    return $\mathsf{enf}(\varphi_1, \sigma, v)$
  else if $\varphi = \varphi_1 \vee \varphi_2$ **then**
    return $\textsc{Fixpoint}(\sigma, \mathsf{enf}_{\mathsf{or}, \varphi_1, \varphi_2, v})$
  else if $\varphi = \exists x.\ \varphi_1$ **then**
    return $\textsc{Fixpoint}(\sigma, \mathsf{enf}_{\mathsf{ex}, \varphi_1, v})$

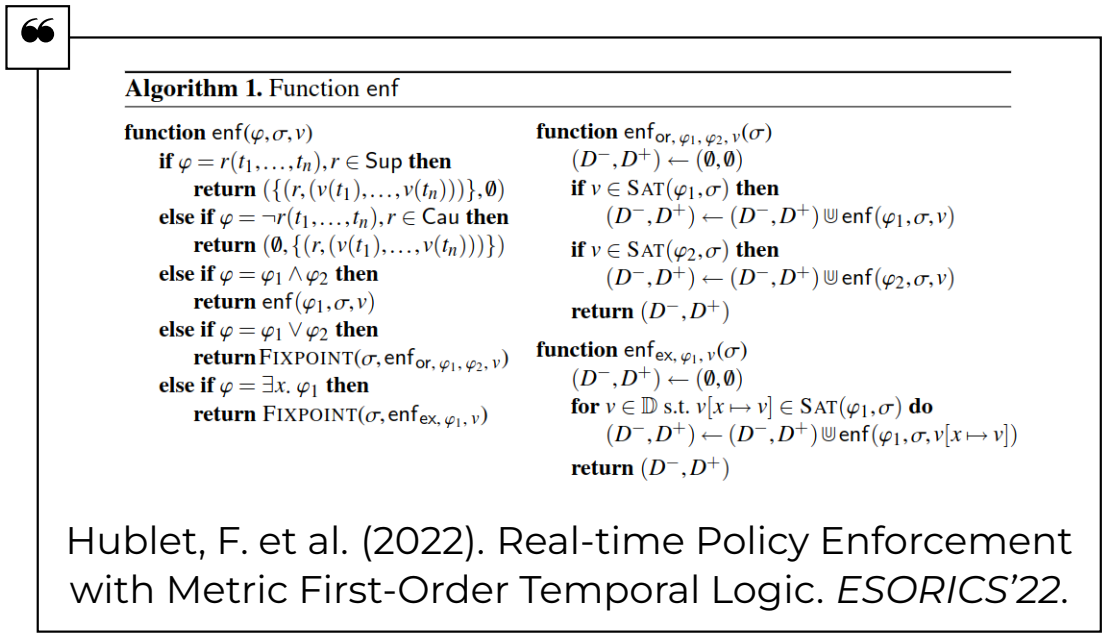
function $\mathsf{enf}_{\mathsf{or}, \varphi_1, \varphi_2, v}(\sigma)$
  $(D^-, D^+) \leftarrow (\emptyset, \emptyset)$
  if $v \in \textsc{Sat}(\varphi_1, \sigma)$ **then**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_1, \sigma, v)$
  if $v \in \textsc{Sat}(\varphi_2, \sigma)$ **then**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_2, \sigma, v)$
  return (D^-, D^+)

function $\mathsf{enf}_{\mathsf{ex}, \varphi_1, v}(\sigma)$
  $(D^-, D^+) \leftarrow (\emptyset, \emptyset)$
  for $v \in \mathbb{D}$ s.t. $v[x \mapsto v] \in \textsc{Sat}(\varphi_1, \sigma)$ **do**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_1, \sigma, v[x \mapsto v])$
  return (D^-, D^+)

Hublet, F. et al. (2022). Real-time Policy Enforcement with Metric First-Order Temporal Logic. *ESORICS'22*.

- Supports **suppressable** and **causable** events
- Characterizes an **enforceable fragment** of MFOTL

Previous work: MFOTL enforcement



Algorithm 1. Function enf

function $\mathsf{enf}(\varphi, \sigma, v)$
  if $\varphi = r(t_1, \ldots, t_n), r \in \mathsf{Sup}$ **then**
    return $(\{(r, (v(t_1), \ldots, v(t_n)))\}, \emptyset)$
  else if $\varphi = \neg r(t_1, \ldots, t_n), r \in \mathsf{Cau}$ **then**
    return $(\emptyset, \{(r, (v(t_1), \ldots, v(t_n)))\})$
  else if $\varphi = \varphi_1 \wedge \varphi_2$ **then**
    return $\mathsf{enf}(\varphi_1, \sigma, v)$
  else if $\varphi = \varphi_1 \vee \varphi_2$ **then**
    return $\textsc{Fixpoint}(\sigma, \mathsf{enf}_{\mathsf{or}, \varphi_1, \varphi_2, v})$
  else if $\varphi = \exists x.\ \varphi_1$ **then**
    return $\textsc{Fixpoint}(\sigma, \mathsf{enf}_{\mathsf{ex}, \varphi_1, v})$

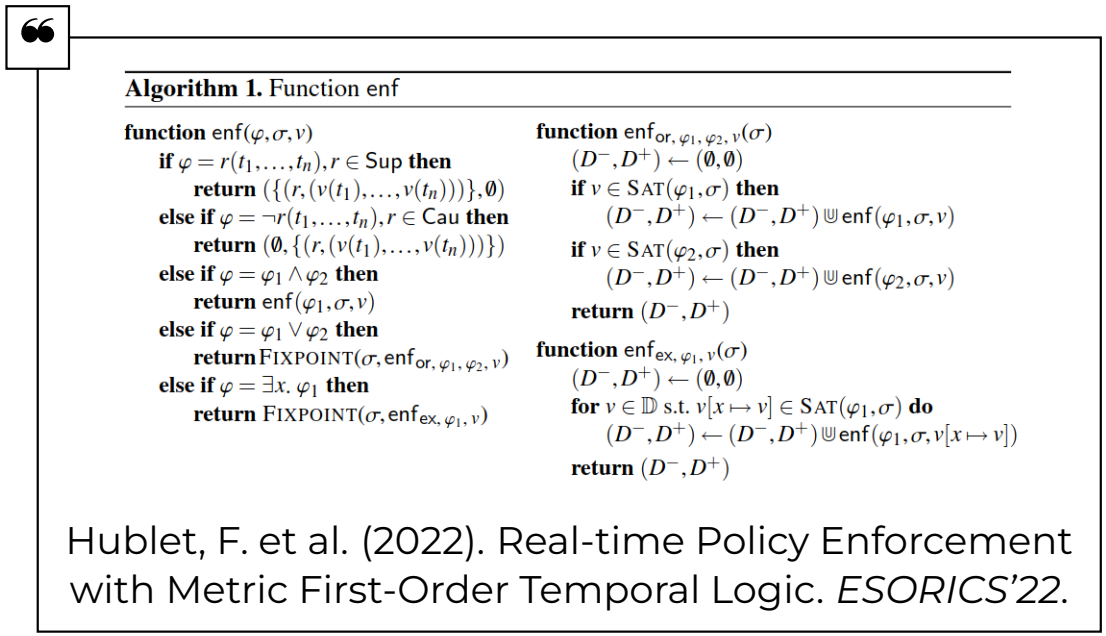
function $\mathsf{enf}_{\mathsf{or}, \varphi_1, \varphi_2, v}(\sigma)$
  $(D^-, D^+) \leftarrow (\emptyset, \emptyset)$
  if $v \in \textsc{Sat}(\varphi_1, \sigma)$ **then**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_1, \sigma, v)$
  if $v \in \textsc{Sat}(\varphi_2, \sigma)$ **then**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_2, \sigma, v)$
  return (D^-, D^+)

function $\mathsf{enf}_{\mathsf{ex}, \varphi_1, v}(\sigma)$
  $(D^-, D^+) \leftarrow (\emptyset, \emptyset)$
  for $v \in \mathbb{D}$ s.t. $v[x \mapsto v] \in \textsc{Sat}(\varphi_1, \sigma)$ **do**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_1, \sigma, v[x \mapsto v])$
  return (D^-, D^+)

Hublet, F. et al. (2022). Real-time Policy Enforcement with Metric First-Order Temporal Logic. *ESORICS'22*.

- Supports **suppressable** and **causable** events
- Characterizes an **enforceable fragment** of MFOTL
- Provides a PDP (**EnfPoly** tool)

Previous work: MFOTL enforcement



Algorithm 1. Function enf

function $\mathsf{enf}(\varphi, \sigma, v)$
  if $\varphi = r(t_1, \ldots, t_n), r \in \mathsf{Sup}$ **then**
    return $(\{(r, (v(t_1), \ldots, v(t_n)))\}, \emptyset)$
  else if $\varphi = \neg r(t_1, \ldots, t_n), r \in \mathsf{Cau}$ **then**
    return $(\emptyset, \{(r, (v(t_1), \ldots, v(t_n)))\})$
  else if $\varphi = \varphi_1 \wedge \varphi_2$ **then**
    return $\mathsf{enf}(\varphi_1, \sigma, v)$
  else if $\varphi = \varphi_1 \vee \varphi_2$ **then**
    return $\textsc{Fixpoint}(\sigma, \mathsf{enf}_{\mathsf{or}, \varphi_1, \varphi_2, v})$
  else if $\varphi = \exists x.\ \varphi_1$ **then**
    return $\textsc{Fixpoint}(\sigma, \mathsf{enf}_{\mathsf{ex}, \varphi_1, v})$

function $\mathsf{enf}_{\mathsf{or}, \varphi_1, \varphi_2, v}(\sigma)$
  $(D^-, D^+) \leftarrow (\emptyset, \emptyset)$
  if $v \in \textsc{Sat}(\varphi_1, \sigma)$ **then**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_1, \sigma, v)$
  if $v \in \textsc{Sat}(\varphi_2, \sigma)$ **then**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_2, \sigma, v)$
  return (D^-, D^+)

function $\mathsf{enf}_{\mathsf{ex}, \varphi_1, v}(\sigma)$
  $(D^-, D^+) \leftarrow (\emptyset, \emptyset)$
  for $v \in \mathbb{D}$ s.t. $v[x \mapsto v] \in \textsc{Sat}(\varphi_1, \sigma)$ **do**
    $(D^-, D^+) \leftarrow (D^-, D^+) \Cup \mathsf{enf}(\varphi_1, \sigma, v[x \mapsto v])$
  return (D^-, D^+)

Hublet, F. et al. (2022). Real-time Policy Enforcement with Metric First-Order Temporal Logic. *ESORICS'22*.

- Supports **suppressable** and **causable** events
- Characterizes an **enforceable fragment** of MFOTL
- Provides a PDP (**EnfPoly** tool)
- Two main inputs: **signature** + **MFOTL formula**

GDPR signature for enforcement

Event	Description	Type
	The data subject *ds*:	
`DSConsent`(*ds*, *prp*, *d*)	gives consent to use data *d* for purpose *prp*	
`DSRevoke`(*ds*, *prp*, *d*)	revokes consent given to use data *d* for purpose *prp*	
...	...	...
`DSErase`(*ds*, *d*)	requests erasure of data *d*	
...	...	...
	The application:	
`Collect`(*ds*, *d*)	collects data *d* of data subject *ds*	
`Use`(*prp*, *d*)	uses data *d* for purpose *prp*	Ⓢ
...	...	...
`Erase`(*d*)	erases data *d*	Ⓒ
...	...	...
`LegalGround`(*grd*, *d*)	application claims legal ground *grd* for using data *d*	

Ⓒ = Causable; Ⓢ = Suppressable; unmarked events are only-observable

Formalization of core GDPR requirements

We cover 6 core requirements / data subject rights:

- Purpose-based usage
- Right to access
- Right to rectification
- Right to erasure
- Right to restriction
- Right to object

Formalization of core GDPR requirements

We cover 6 core requirements / data subject rights:

- **Purpose-based usage**
- Right to access
- Right to rectification
- **Right to erasure**
- Right to restriction
- Right to object

Core GDPR requirements using MFOTL

Example: **Purpose-based usage**

$$\alpha \,\hat{\mathsf{S}}\, \beta := \alpha \,\mathsf{S}\, (\alpha \wedge \beta)$$

$$\varphi_{\mathrm{Purp}} = \exists prp, d, ds.\ \texttt{Use}(prp, d) \wedge \blacklozenge\, \texttt{Collect}(ds, d)$$
$$\wedge \neg((\neg \texttt{DSRevoke}(ds, prp, d)\ \hat{\mathsf{S}}\ \texttt{DSConsent}(ds, prp, d))$$
$$\vee\ (\exists grd.\ \blacklozenge\, \texttt{LegalGround}(grd, d)))$$

Purpose-based usage is violated when

Core GDPR requirements using MFOTL

Example: **Purpose-based usage**

$$\alpha \,\hat{\mathsf{S}}\, \beta := \alpha \,\mathsf{S}\, (\alpha \wedge \beta)$$

$$\varphi_{\mathrm{Purp}} = \exists prp, d, ds.\ \mathtt{Use}(prp, d) \wedge \blacklozenge\, \mathtt{Collect}(ds, d)$$
$$\wedge \neg((\neg \mathtt{DSRevoke}(ds, prp, d)\ \hat{\mathsf{S}}\ \mathtt{DSConsent}(ds, prp, d))$$
$$\vee\ (\exists grd.\ \blacklozenge\, \mathtt{LegalGround}(grd, d)))$$

Purpose-based usage is violated when some data *d* is used for purpose *prp*

Core GDPR requirements using MFOTL

Example: **Purpose-based usage**

$$\alpha \,\hat{\mathsf{S}}\, \beta := \alpha \,\mathsf{S}\, (\alpha \wedge \beta)$$

$$\varphi_{\mathrm{Purp}} = \exists prp, d, ds.\ \mathtt{Use}(prp, d) \wedge \blacklozenge\, \mathtt{Collect}(ds, d)$$
$$\wedge \neg((\neg \mathtt{DSRevoke}(ds, prp, d)\ \hat{\mathsf{S}}\ \mathtt{DSConsent}(ds, prp, d))$$
$$\vee\ (\exists grd.\ \blacklozenge\, \mathtt{LegalGround}(grd, d)))$$

Purpose-based usage is violated when some data *d* is used for purpose *prp*, data *d* **was** collected from data subject *ds*

Core GDPR requirements using MFOTL

Example: **Purpose-based usage**

$$\alpha \,\hat{\mathsf{S}}\, \beta := \alpha \,\mathsf{S}\, (\alpha \wedge \beta)$$

$$\varphi_{\mathrm{Purp}} = \exists prp, d, ds.\ \texttt{Use}(prp, d) \wedge \blacklozenge\, \texttt{Collect}(ds, d)$$
$$\wedge \neg((\neg \texttt{DSRevoke}(ds, prp, d)\ \hat{\mathsf{S}}\ \texttt{DSConsent}(ds, prp, d))$$
$$\vee\ (\exists grd.\ \blacklozenge\, \texttt{LegalGround}(grd, d)))$$

Purpose-based usage is violated when some data *d* is used for purpose *prp*, data *d* **was** collected from data subject *ds* and neither

Core GDPR requirements using MFOTL

Example: **Purpose-based usage**

$$\alpha \,\hat{\mathsf{S}}\, \beta := \alpha \,\mathsf{S}\, (\alpha \wedge \beta)$$

$$\varphi_{\mathrm{Purp}} = \exists prp, d, ds.\ \mathtt{Use}(prp, d) \wedge \blacklozenge\, \mathtt{Collect}(ds, d)$$
$$\wedge \neg((\neg \mathtt{DSRevoke}(ds, prp, d)\ \hat{\mathsf{S}}\ \mathtt{DSConsent}(ds, prp, d))$$
$$\vee\ (\exists grd.\ \blacklozenge\, \mathtt{LegalGround}(grd, d)))$$

Purpose-based usage is violated when some data *d* is used for purpose *prp*, data *d* **was** collected from data subject *ds* and neither the data subject *ds* has given (and since then not revoked) consent to process data *d* for purpose *prp*

Core GDPR requirements using MFOTL

Example: **Purpose-based usage**

$$\alpha \,\hat{\mathsf{S}}\, \beta := \alpha \,\mathsf{S}\, (\alpha \wedge \beta)$$

$$\varphi_{\mathrm{Purp}} = \exists prp, d, ds.\ \texttt{Use}(prp, d) \wedge \blacklozenge\, \texttt{Collect}(ds, d)$$
$$\wedge \neg((\neg \texttt{DSRevoke}(ds, prp, d)\ \hat{\mathsf{S}}\ \texttt{DSConsent}(ds, prp, d))$$
$$\vee\ (\exists grd.\ \blacklozenge\, \texttt{LegalGround}(grd, d)))$$

Purpose-based usage is violated when some data *d* is used for purpose *prp*, data *d* **was** collected from data subject *ds* and neither the data subject *ds* has given (and since then not revoked) consent to process data *d* for purpose *prp* nor has the application claimed some legal ground *grd* to use data *d*.

Core GDPR requirements using MFOTL

Example: **Purpose-based usage**

$$\alpha \,\hat{\mathsf{S}}\, \beta := \alpha \,\mathsf{S}\, (\alpha \wedge \beta)$$

$$\begin{aligned}\varphi_{\mathrm{Purp}} = \exists \mathit{prp}, \mathit{d}, \mathit{ds}.\ &\texttt{Use}(\mathit{prp}, \mathit{d}) \wedge \blacklozenge\, \texttt{Collect}(\mathit{ds}, \mathit{d}) \\ &\wedge \neg((\neg \texttt{DSRevoke}(\mathit{ds}, \mathit{prp}, \mathit{d}) \,\hat{\mathsf{S}}\, \texttt{DSConsent}(\mathit{ds}, \mathit{prp}, \mathit{d})) \\ &\quad \vee (\exists \mathit{grd}.\ \blacklozenge\, \texttt{LegalGround}(\mathit{grd}, \mathit{d})))\end{aligned}$$

Purpose-based usage is violated when some data *d* is used for purpose *prp*, data *d* **was** collected from data subject *ds* and neither the data subject *ds* has given (and since then not revoked) consent to process data *d* for purpose *prp* nor has the application claimed some legal ground *grd* to use data *d*.

Core GDPR requirements using MFOTL (II)

Example: **Right to erasure**

$$\varphi_{\text{Erasure}} = \exists ds, ut.\ \texttt{DSErase}(ds, d) \wedge \blacklozenge\, \texttt{Collect}(ds, d) \wedge \neg \texttt{Erase}(d)$$

The right to erasure is violated when

Core GDPR requirements using MFOTL (II)

Example: **Right to erasure**

$$\varphi_{\text{Erasure}} = \exists ds, ut.\ \texttt{DSErase}(ds, d) \wedge \blacklozenge\, \texttt{Collect}(ds, d) \wedge \neg \texttt{Erase}(d)$$

The right to erasure is violated when data subject *ds* has requested the erasure of data *d*

Core GDPR requirements using MFOTL (II)

Example: **Right to erasure**

$$\varphi_{\text{Erasure}} = \exists ds, ut.\ \texttt{DSErase}(ds, d) \wedge \blacklozenge\, \texttt{Collect}(ds, d) \wedge \neg \texttt{Erase}(d)$$

The right to erasure is violated when data subject *ds* has requested the erasure of data *d*, data *d* **was** collected from data subject *ds*

Core GDPR requirements using MFOTL (II)

Example: **Right to erasure**

$$\varphi_{\text{Erasure}} = \exists ds, ut.\ \texttt{DSErase}(ds, d) \wedge \blacklozenge\, \texttt{Collect}(ds, d) \wedge \neg \texttt{Erase}(d)$$

The right to erasure is violated when data subject *ds* has requested the erasure of data *d*, data *d* **was** collected from data subject *ds* but data *d* is not erased.

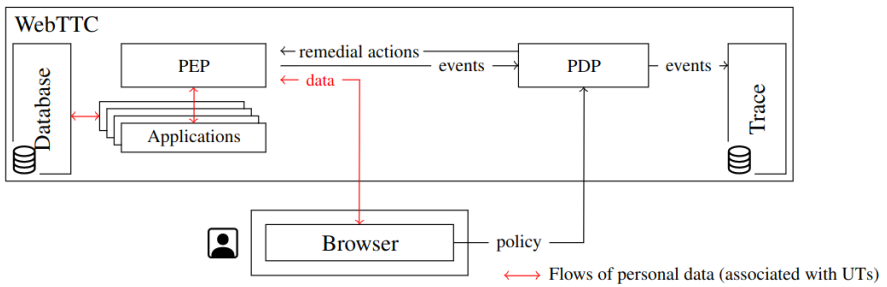
Enforceability

- Our formalization is **enforceable**, being in the enforceable fragment from Hublet et al. (2022)
- Enforcement through suppression / causation
  - Purpose based-usage: suppression of `Use`
  - Right to erasure: causation of `Erase`

2

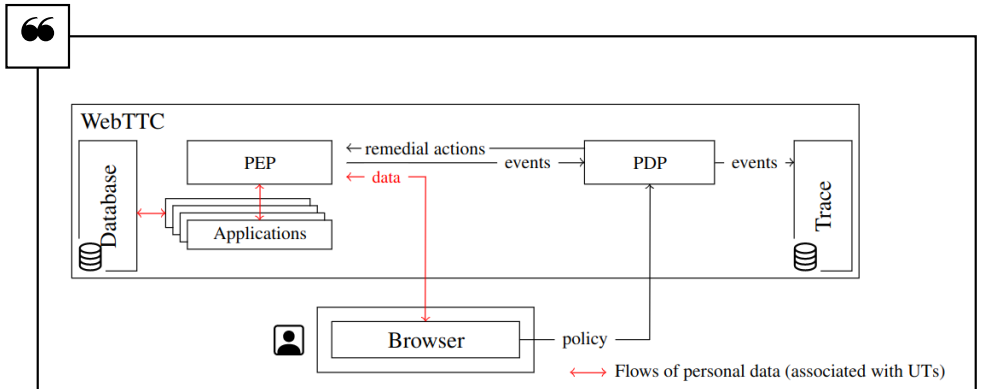
Enforcement architecture for web apps

Previous work: WebTTC



Hublet, F. et al. (2024). User-Controlled Privacy: Taint, Track, and Control. *PETS'24*, to appear.

Previous work: WebTTC



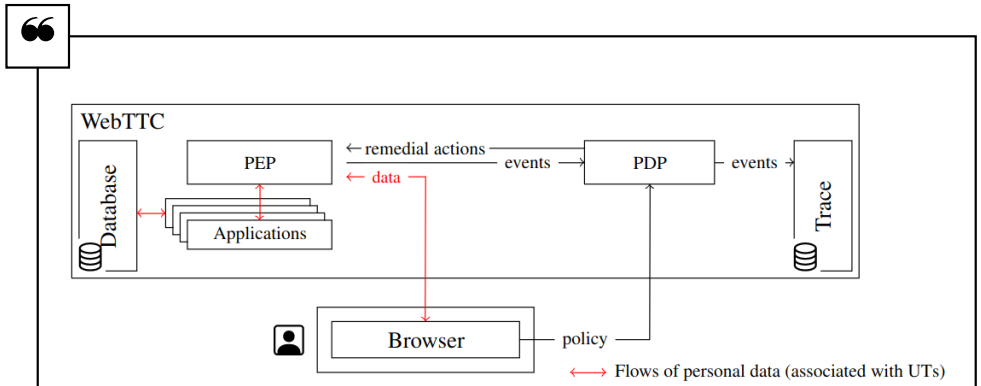
Hublet, F. et al. (2024). User-Controlled Privacy: Taint, Track, and Control. *PETS'24*, to appear.

- Enforces **MFOTL privacy policies** in web applications by observing the following events:

Event	Description	Type
$\texttt{In}(u, d)$	Data subject *ds* inputs a value with UT *d*	
$\texttt{Out}(u, prp, d)$	Value with UT *d* is output to data subject *ds* for purpose *prp*	Ⓢ
$\texttt{Itf}(d, o)$	Input with UT *d* interferes with output *o*	

Ⓢ = Suppressable; unmarked events are only-observable

Previous work: WebTTC



Hublet, F. et al. (2024). User-Controlled Privacy: Taint, Track, and Control. *PETS'24*, to appear.

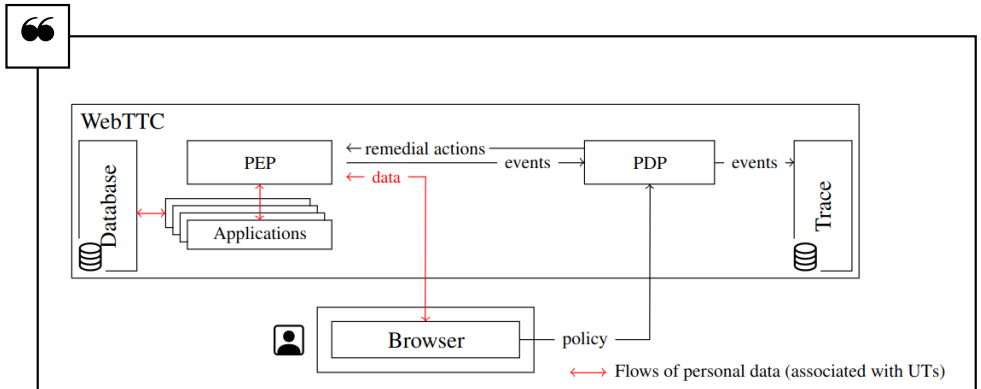
- Enforces **MFOTL privacy policies** in web applications by observing the following events:

Event	Description	Type
$\texttt{In}(u, d)$	Data subject *ds* inputs a value with UT *d*	
$\texttt{Out}(u, prp, d)$	Value with UT *d* is output to data subject *ds* for purpose *prp*	Ⓢ
$\texttt{Itf}(d, o)$	Input with UT *d* interferes with output *o*	

Ⓢ = Suppressable; unmarked events are only-observable

- **Python-like language** with dynamic Information Flow Control

Previous work: WebTTC



Hublet, F. et al. (2024). User-Controlled Privacy: Taint, Track, and Control. *PETS'24*, to appear.

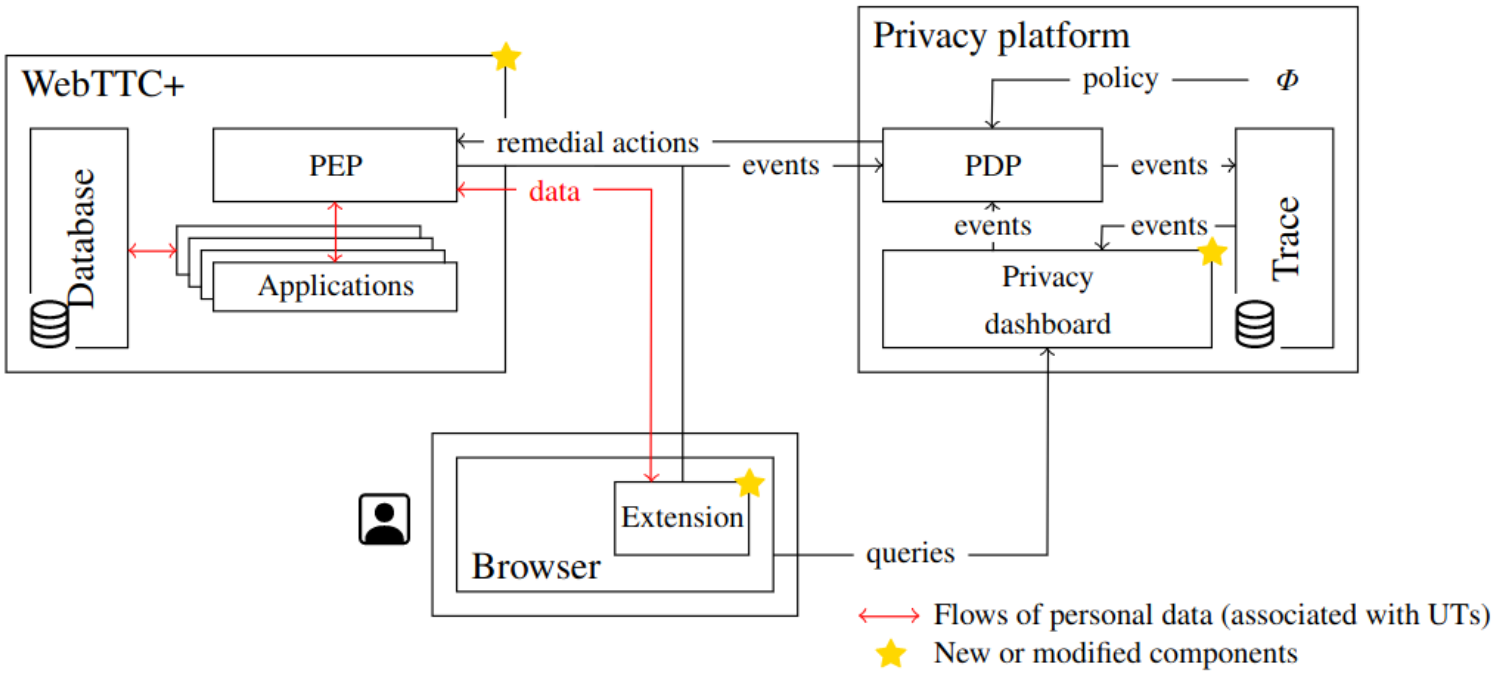
- Enforces **MFOTL privacy policies** in web applications by observing the following events:

Event	Description	Type
$\texttt{In}(u, d)$	Data subject *ds* inputs a value with UT *d*	
$\texttt{Out}(u, prp, d)$	Value with UT *d* is output to data subject *ds* for purpose *prp*	Ⓢ
$\texttt{Itf}(d, o)$	Input with UT *d* interferes with output *o*	

Ⓢ = Suppressable; unmarked events are only-observable

- **Python-like language** with dynamic Information Flow Control
- Associates **unique taints** (UTs) to inputs and propagates them

Our enforcement architecture: WebTTC+

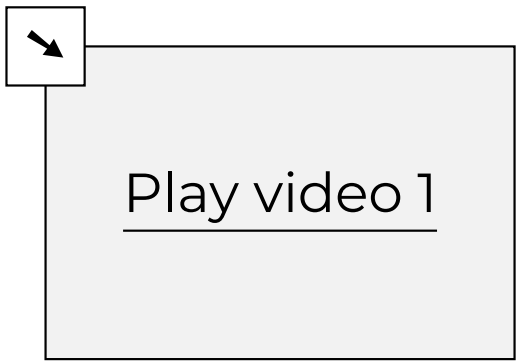


Instrumentation

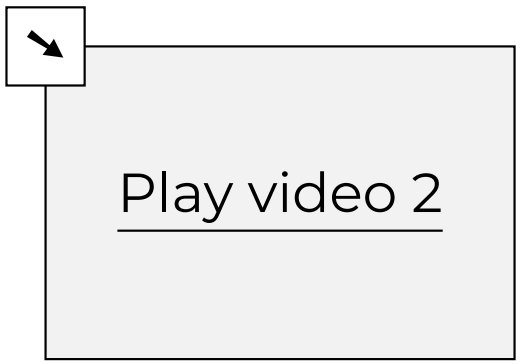
- Events are emitted at input or output time, on user request (`DS`...) or on PDP request (causable events)

Event	Type	When emitted?	By...
`DSConsent`(ds, prp, d)		Input time / on user request	Extension / Dashboard
`DSRevoke`(ds, prp, d)		On user request	Dashboard
`DSErase`(ds, d)		On user request	Dashboard
`Collect`(ds, d, sp)		Input time	Extension
`Use`(prp, d)	Ⓢ	Output time	PEP
`Erase`(d)	Ⓒ	On enforcer request	PEP

Instrumentation: input time



Instrumentation: user requests



3

Prototype and case studies

Prototype

Includes:

- WebTTC+
- Privacy platform including EnfPoly wrapper
- PoC browser extension
- EnfPoly (Hublet et al. 2022)

$\sim$ 3k lines of code on top of WebTTC and EnfPoly

Evaluation

We ported three applications from previous work (Wang et al. 2019, Yang et al. 2016).

Evaluation

We ported three applications from previous work (Wang et al. 2019, Yang et al. 2016).

Research questions:

Evaluation

We ported three applications from previous work (Wang et al. 2019, Yang et al. 2016).

Research questions:

RQ1: Can realistic web applications be developed in WebTTC+? How is the size of the code base impacted?

Evaluation

We ported three applications from previous work (Wang et al. 2019, Yang et al. 2016).

Research questions:

RQ1: Can realistic web applications be developed in WebTTC+? How is the size of the code base impacted?

RQ2: How much runtime overhead does WebTTC+ incur compared to a baseline without enforcement?

Evaluation

We ported three applications from previous work (Wang et al. 2019, Yang et al. 2016).

Research questions:

RQ1: Can realistic web applications be developed in WebTTC+? How is the size of the code base impacted?

RQ2: How much runtime overhead does WebTTC+ incur compared to a baseline without enforcement?

RQ3: What share of the GDPRs provisions does our implementation effectively enforce?

Evaluation results

RQ1: Functionality preserved with low code overhead ~10%

Evaluation results

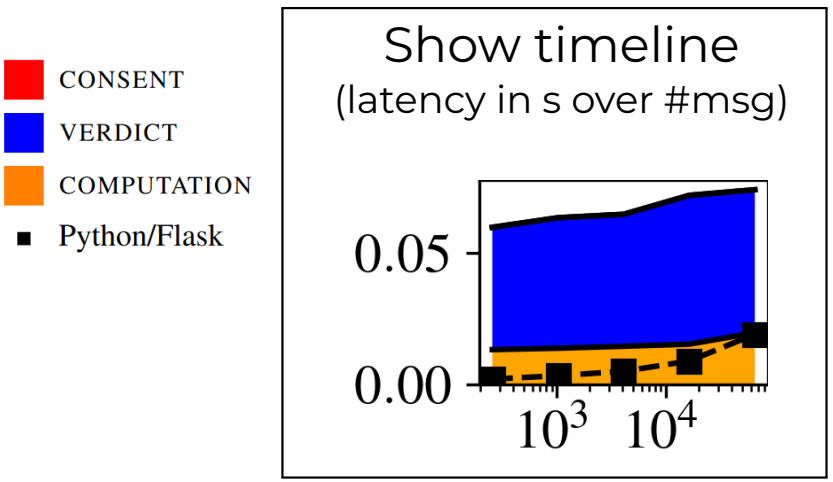
RQ1: Functionality preserved with low code overhead ~10%

RQ2: Runtime overhead within 1 order of magnitude of baseline, latency < 75 ms, usability is preserved

Evaluation results

RQ1: Functionality preserved with low code overhead ~10%

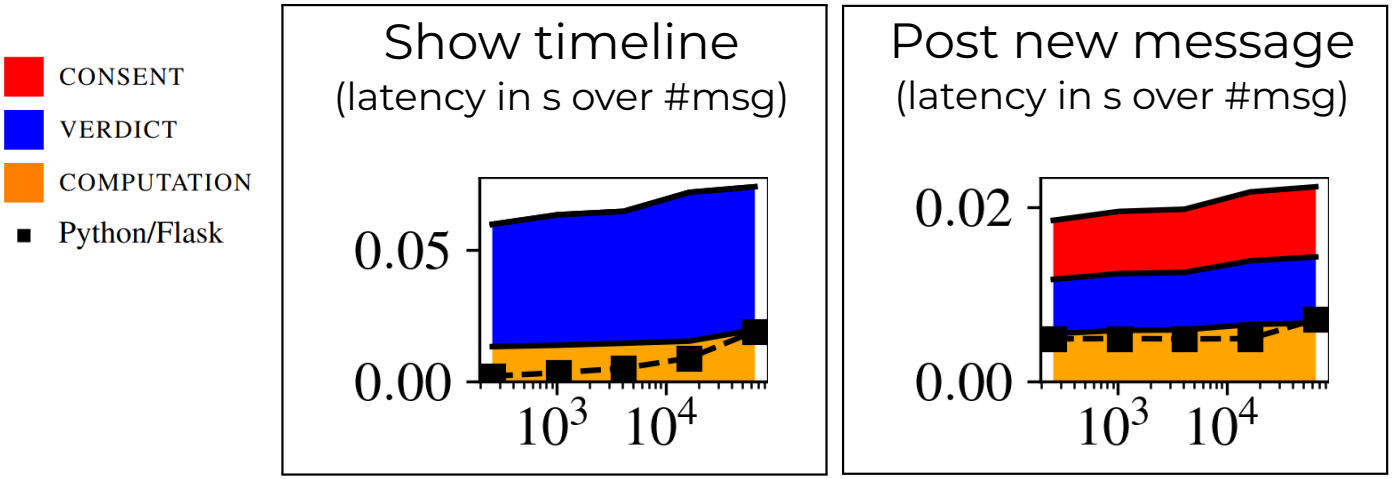
RQ2: Runtime overhead within 1 order of magnitude of baseline, latency < 75 ms, usability is preserved



Evaluation results

RQ1: Functionality preserved with low code overhead ~10%

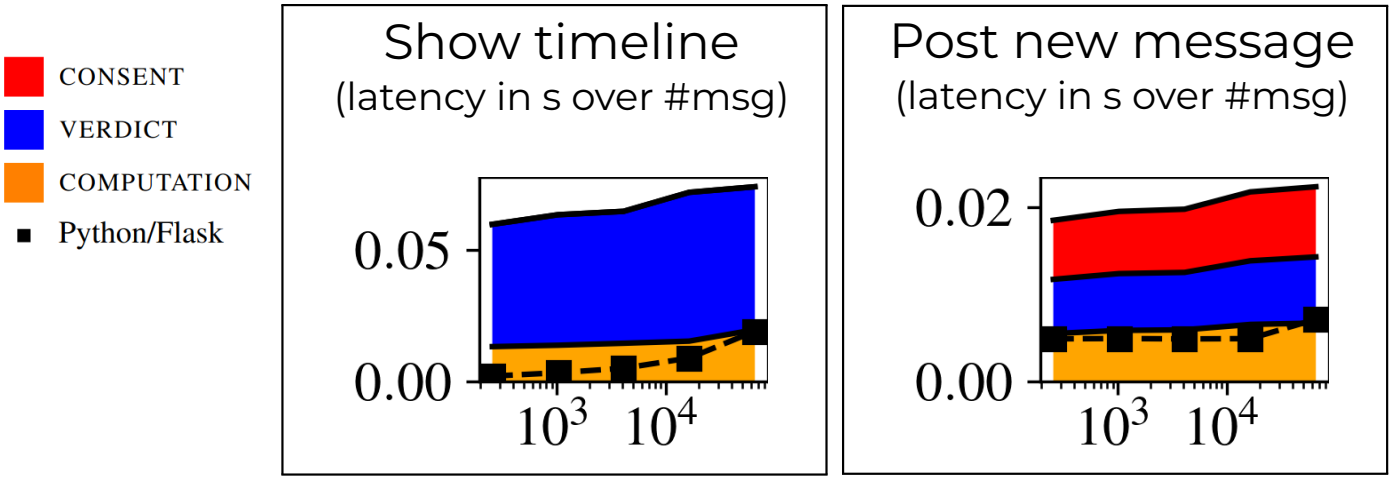
RQ2: Runtime overhead within 1 order of magnitude of baseline, latency < 75 ms, usability is preserved



Evaluation results

RQ1: Functionality preserved with low code overhead ~10%

RQ2: Runtime overhead within 1 order of magnitude of baseline, latency < 75 ms, usability is preserved



RQ3: About 2/3 of the fines imposed until May 2023 are related to at least one of the articles addressed by our approach

Conclusion and future work

- First work to provide an enforceable specification + enforcement architecture for a core of the GDPR

Conclusion and future work

- First work to provide an enforceable specification + enforcement architecture for a core of the GDPR
- New approach to PbD using runtime enforcement

Conclusion and future work

- First work to provide an enforceable specification + enforcement architecture for a core of the GDPR
- New approach to PbD using runtime enforcement

Contributions

Future work

Conclusion and future work

- First work to provide an enforceable specification + enforcement architecture for a core of the GDPR
- New approach to PbD using runtime enforcement

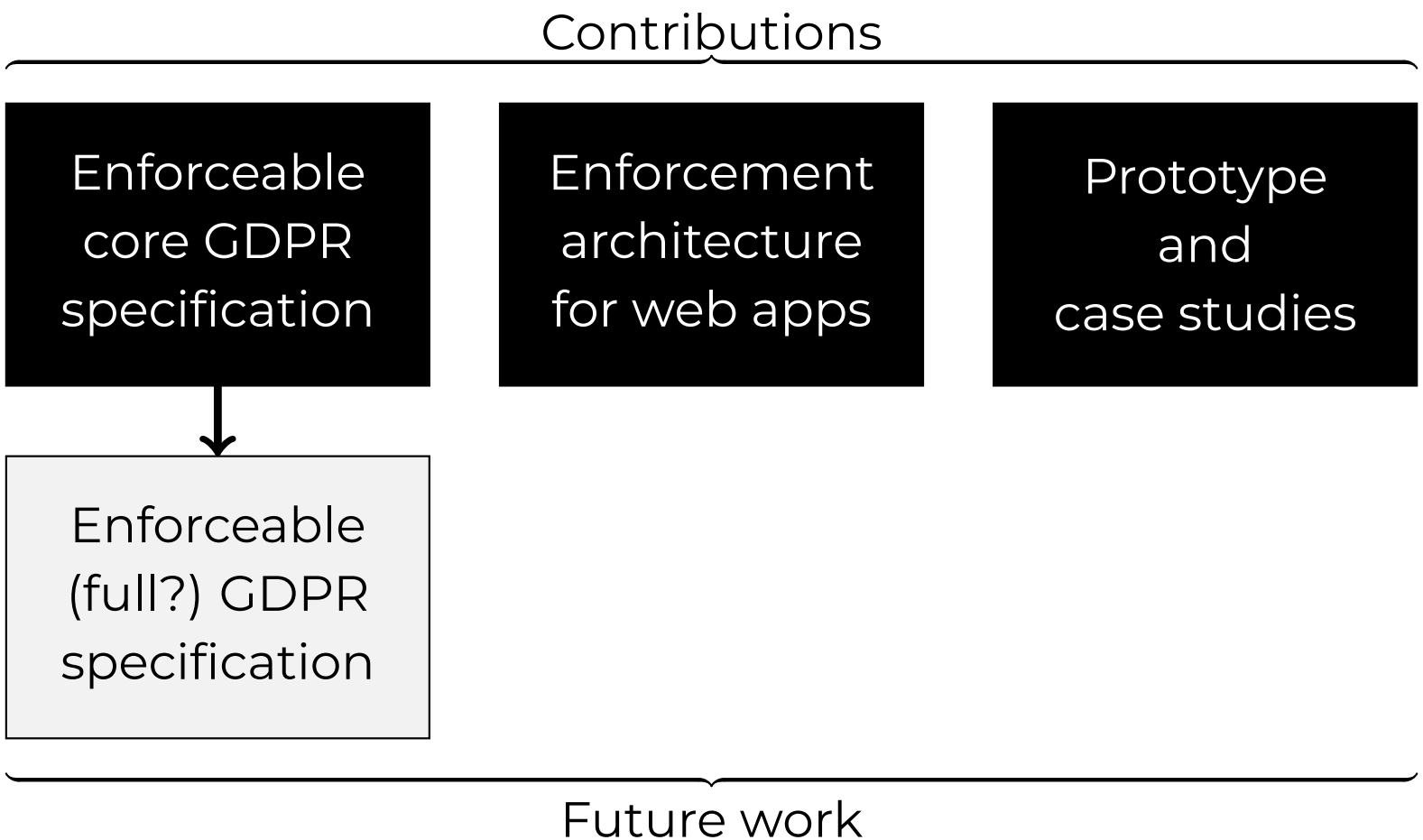
Contributions

Enforceable core GDPR specification	Enforcement architecture for web apps	Prototype and case studies

Future work

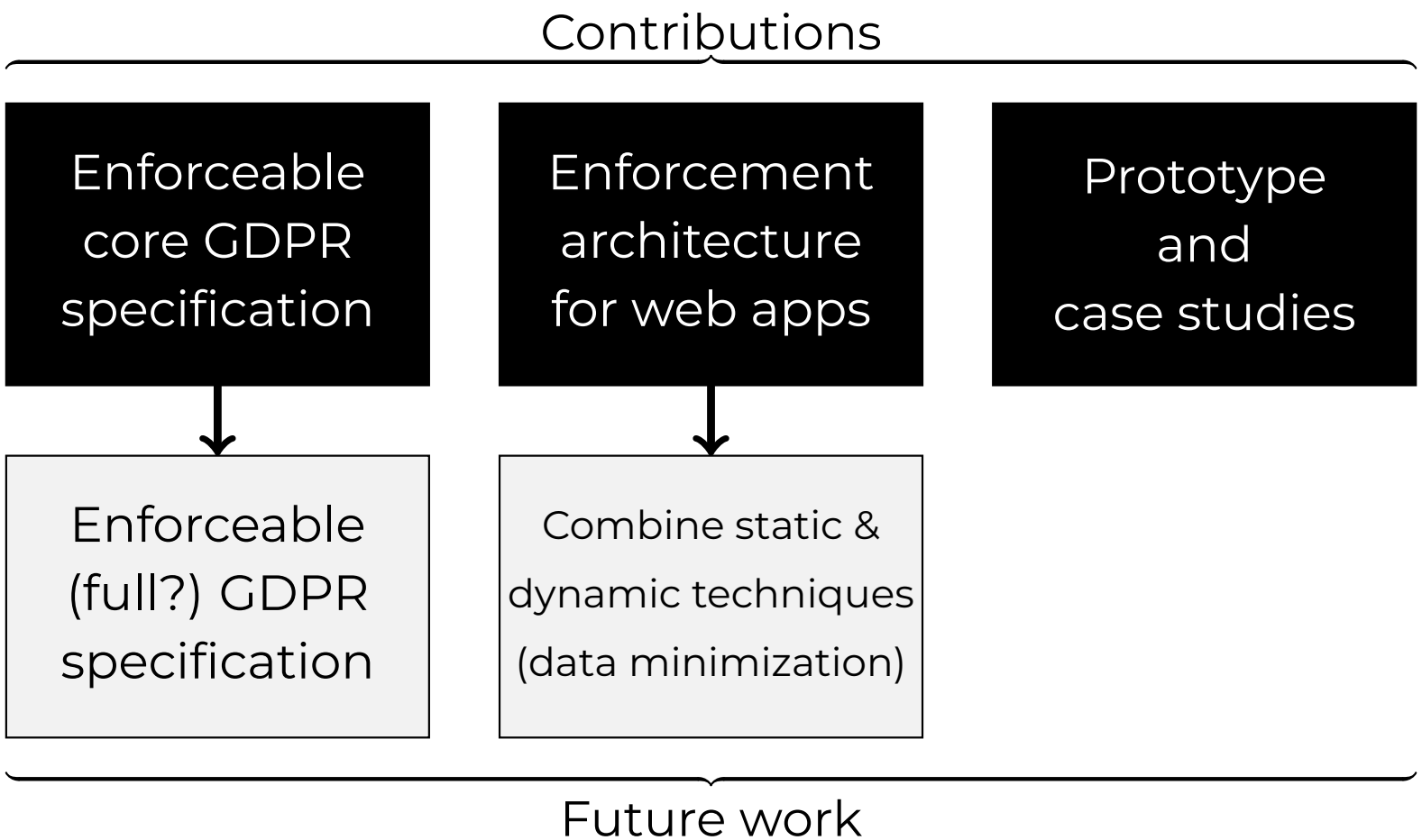
Conclusion and future work

- First work to provide an enforceable specification + enforcement architecture for a core of the GDPR
- New approach to PbD using runtime enforcement



Conclusion and future work

- First work to provide an enforceable specification + enforcement architecture for a core of the GDPR
- New approach to PbD using runtime enforcement



Conclusion and future work

- First work to provide an enforceable specification + enforcement architecture for a core of the GDPR
- New approach to PbD using runtime enforcement