The MonIKA-Framework - A Trail Balloon of a Cooperative Monitoring Framework for Anomaly Detection

Arnold Sykosch <sykosch@cs.uni-bonn.de>

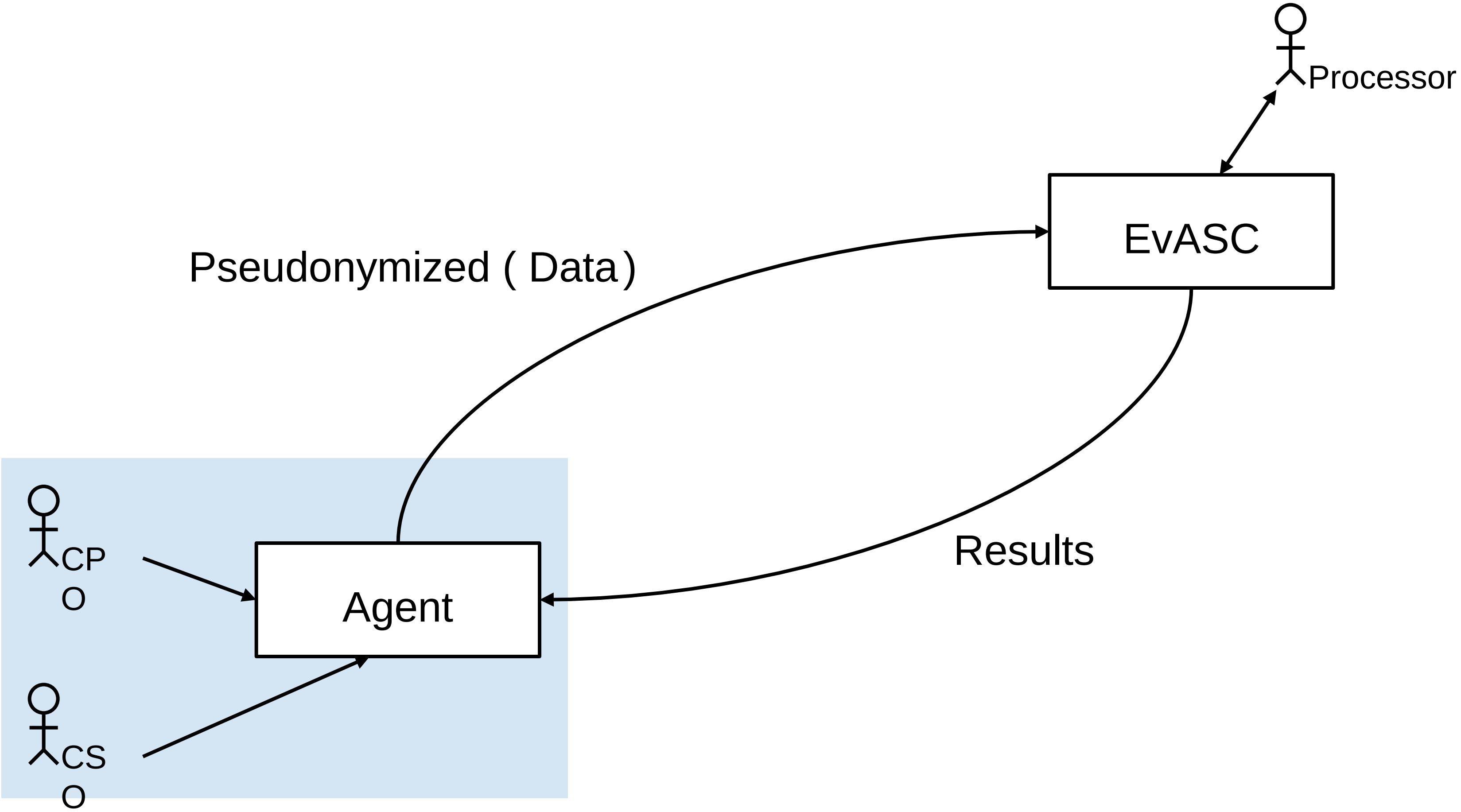
MonIKA

Fraunhofer
FKIE

1. Key Requirements

- Information Fusion
  - Gathering of information to one place
  - A global data schema
- Privacy Protection
  - Pseudonymization
  - Purpose Limitation
- Anomaly Detection
  - Access for clasification algorithms
  - Result management

2. The Basic Architecture



1. Key Requirements

- Information Fusion
  - Gathering of information to one place
  - A global data schema

- Privacy Protection
  - Pseudonymization
  - Purpose Limitation

- Anomaly Detection
  - Access for clasification algorithms
  - Result management

3. Pseudonymization by Policy
Confidentiality and Availability Requirements

- Requirements against the data

- Availability Requirements
  - *Laid down by:* the Processor.
  - *If not met:* The classification algorithm can not work.

- Confidentiality Requirements
  - *Laid down by:* the CPO
  - *If not met:* No agreement from CPO, therefore no data from one Partic.

3. Pseudonymization by Policy
Parts & Pieces

`<pseudonym>`

What should the output in the global schema be called?

`<data>`

What data fields is the input?

`<link>*`

How should the generated pseudonym be linkable?

`<revocation>*`

Should pseudonymety be revocable?

3. Pseudonymization by Policy
Parts & Pieces

`<pseudonym>`

What should the output in the global schema be called?

`<data>`

What is the input?

`<link>*`

How should the generated pseudonym be linkable?

`<revocation>*`

Should pseudonymety be revocable?

3. Pseudonymization by Policy
An Example

```
<pseudonym name=“ipaddress” applcation=“app” sensor=“snort”>
  <data>
    tokenize(replace(ip, '(.)', '$1&#xE0F1;'), '&#xE0F1;')
  </data>
</pseudonym>
```

3. Pseudonymization by Policy

An Example

```
<pseudonym name=“ipaddress” applcation=“app” sensor=“snort”>
  <data>
    tokenize(replace(ip, '(.)', '$1&#xE0F1;'), '&#xE0F1;')
  </data>
  <link>







  </link>
</pseudonym>
```

3. Pseudonymization by Policy
An Example

```
<pseudonym name=“ipaddress” application=“app” sensor=“sens”>
  <data>
    tokenize(replace(ip, '(.)', '$1&#xE0F1;'), '&#xE0F1;')
  </data>
  <link>
    <type>prefix</type>



  </link>
</pseudonym>
```

3. Pseudonymization by Policy
An Example

```
<pseudonym name=“ipaddress” application=“app” sensor=“sens”>
  <data>
    tokenize(replace(ip, '(.)', '$1&#xE0F1;'), '&#xE0F1;')
  </data>
  <link>
    <type>prefix</type>
    <relation>app.snort.ipaddress</relation>


  </link>
</pseudonym>
```

3. Pseudonymization by Policy
An Example

```
<pseudonym name=“ipaddress” application=“app” sensor=“sens”>
  <data>
    tokenize(replace(ip, '(.)', '$1&#xE0F1;'), '&#xE0F1;')
  </data>
  <link>
    <type>prefix</type>
    <relation>app.snort.ipaddress</relation>
    <condition>alert==“ICMP-Redirect”</condition>

  </link>
</pseudonym>
```

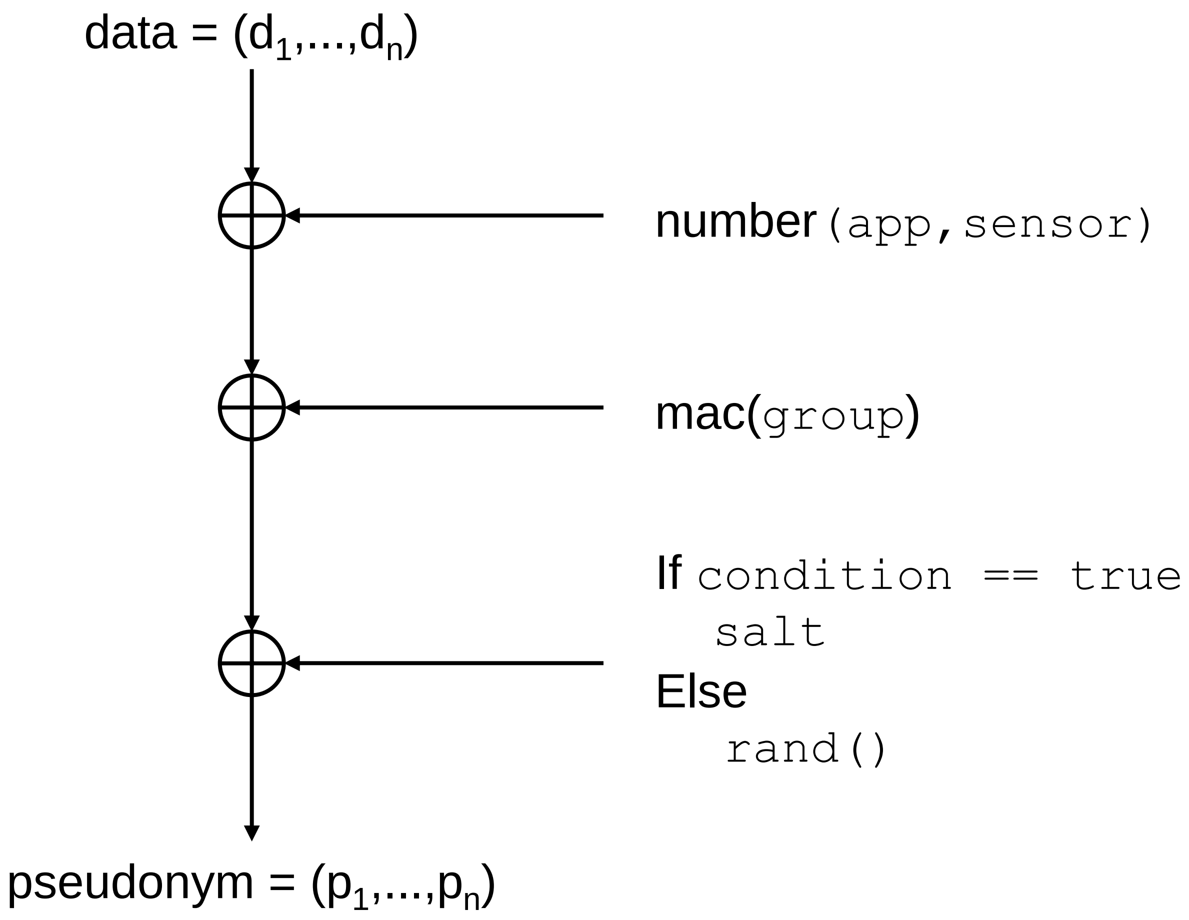
3. Pseudonymization by Policy

An Example

```
<pseudonym name=“ipaddress” application=“app” sensor=“sens”>
  <data>
    tokenize(replace(ip, '(.)', '$1&#xE0F1;'), '&#xE0F1;')
  </data>
  <link>
    <type>prefix</type>
    <relation>app.snort.ipaddress</relation>
    <condition salt=“...”>alert==“ICMP-Redirect”</condition>
    <group>receiver</group>
  </link>
</pseudonym>
```

3. Pseudonym Generation

What happens then?



data = $(d_1,...,d_n)$

number(app, sensor)

mac(group)

```
If condition == true
   salt
Else
   rand()
```

pseudonym = $(p_1,...,p_n)$

Q & A