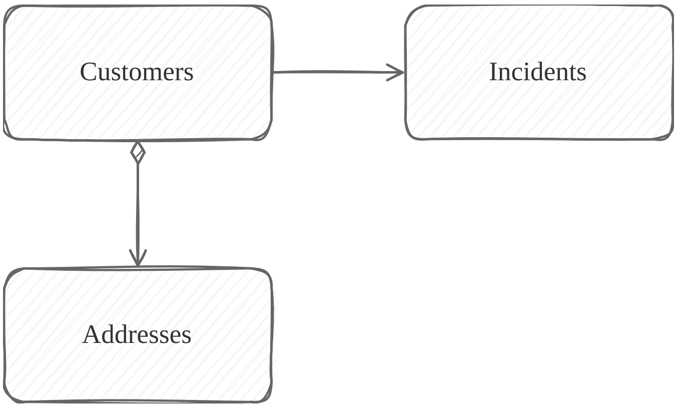
Annotating Personal Data

In order to automate audit logging, personal data management, and data retention management as much as possible, the first and frequently only task to do as an application developer is to identify entities and elements (potentially) holding personal data using `@PersonalData` annotations.

Table of Contents

Reference App Sample

In the remainder of this guide, we use the Incidents Management reference sample app as the base to add data privacy and audit logging to.



So, let's annotate the data model to identify personal data. In essence, in all our entities we search for elements which carry personal data, such as person names, birth dates, etc., and tag them accordingly. All found entities are classified as either *Data Subjects*, *Subject Details* or *Related Data Objects*.

Following the best practice of separation of concerns, we annotate our domain model in a separate file *srv/data-privacy.cds*, which we add to our project and fill it with the following content:

> For the time beeing also replace the data in *data/sap.capire.incidents-Customers.csv*.

db/data-privacy.cds   data/sap.capire.incidents-Customers.csv

cds

```cds
using { sap.capire.incidents as my } from '../db/schema';

extend my.Customers with {
    dateOfBirth : Date;
};

annotate my.Customers with @PersonalData : {
  DataSubjectRole : 'Customer',
  EntitySemantics : 'DataSubject'
} {
  ID           @PersonalData.FieldSemantics: 'DataSubjectID';
  firstName    @PersonalData.IsPotentiallyPersonal;
  lastName     @PersonalData.IsPotentiallyPersonal;
  email        @PersonalData.IsPotentiallyPersonal;
  phone        @PersonalData.IsPotentiallyPersonal;
  dateOfBirth  @PersonalData.IsPotentiallyPersonal;
  creditCardNo @PersonalData.IsPotentiallySensitive;
};

annotate my.Addresses with @PersonalData: {
  EntitySemantics : 'DataSubjectDetails'
} {
  customer      @PersonalData.FieldSemantics: 'DataSubjectID';
  city          @PersonalData.IsPotentiallyPersonal;
  postCode      @PersonalData.IsPotentiallyPersonal;
  streetAddress @PersonalData.IsPotentiallyPersonal;
};

annotate my.Incidents with @PersonalData : {
  EntitySemantics : 'Other'
} {
  customer @PersonalData.FieldSemantics: 'DataSubjectID';
};
```

@PersonalData...

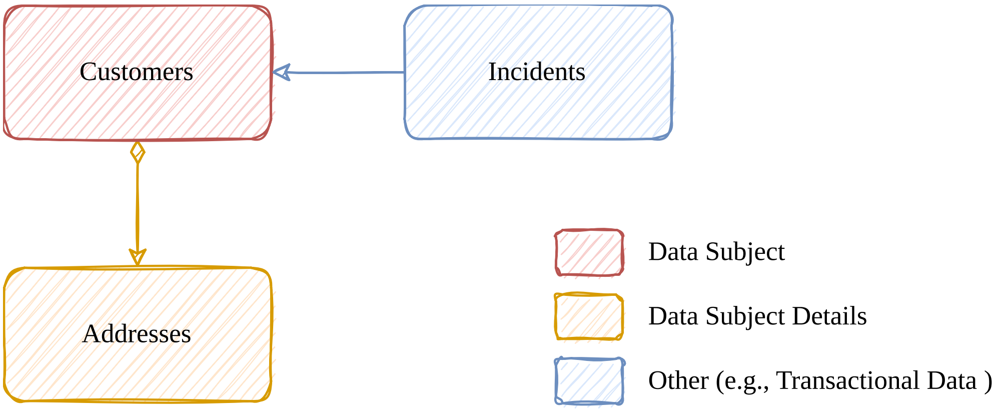
Let's break down the annotations to identify personal data, shown in the sample above. These annotations fall into three categories:

- **Entity-level annotations** signify relevant entities as *Data Subjects*, *Data Subject Details*, or *Related Data Objects* in data privacy terms, as depicted in the graphic below.
- **Key-level annotations** signify object primary keys, as well as references to data subjects (which have to be present on each object).
- **Field-level annotations** identify elements containing personal data.

↳ *Learn more about these annotations in the @PersonalData OData vocabulary .*

@PersonalData .EntitySemantics

The entity-level annotation `@PersonalData.EntitySemantics` signifies relevant entities as *Data Subject*, *Data Subject Details*, or *Other* in data privacy terms, as depicted in the following graphic.



The following table provides some further details.

Annotation	Description
`DataSubject`	The entities of this set describe a data subject (an identified or identifiable natural person), for example, Customer or Vendor.
`DataSubjectDetails`	The entities of this set contain details of a data subject (an identified or identifiable natural person) but do not by themselves identify/describe a data subject, for example, Addresses.
`Other`	Entities containing personal data or references to data subjects, but not representing data subjects or data subject details by themselves. For example, customer quote, customer order, or purchase order with involved business partners. These entities are relevant for audit logging. There are no restrictions on their structure. The properties should be annotated suitably with `FieldSemantics` .

Hence, we annotate our model as follows:

cds

```cds
annotate my.Customers with @PersonalData: {
  EntitySemantics: 'DataSubject'
};

annotate my.Addresses with @PersonalData: {
  EntitySemantics: 'DataSubjectDetails'
};

annotate my.Incidents with @PersonalData: {
  EntitySemantics: 'Other'
};
```

@PersonalData .DataSubjectRole

Can be added to *`@PersonalData.EntitySemantics: 'DataSubject'`*. It's a user-chosen string specifying the role name to use. If omitted, the default is the entity name. Use case is similar to providing user-friendly labels for the UI, although in this case there's no i18n.

In our model, we can add the *`DataSubjectRole`* as follows:

```cds
annotate my.Customers with @PersonalData: {
  EntitySemantics: 'DataSubject',
  DataSubjectRole: 'Customer'
};
```

@PersonalData .FieldSemantics: DataSubjectID

Use this annotation to identify data subject's unique key, or a reference to it. References are commonly associations or foreign keys in subject details entities, or related ones, referring to a subject entity.

- Each *`@PersonalData`* entity needs to identify a *`DataSubjectID`* element.
- For entities with *`DataSubject`* semantics, this is typically the primary key.
- For entities with *`DataSubjectDetails`* or *`Other`* semantics, this is usually an association to the data subject.
- Fields marked as *`DataSubjectID`* should use *`not null`* to guarantee a value is present at all times.

Hence, we annotate our model as follows:

```cds
annotate my.Customers with {
  ID @PersonalData.FieldSemantics: 'DataSubjectID'
};

annotate my.Addresses with {
  customer @PersonalData.FieldSemantics: 'DataSubjectID'
};

annotate my.Incidents with {
  customer @PersonalData.FieldSemantics: 'DataSubjectID'
};
```

@PersonalData .IsPotentiallyPersonal

`@PersonalData.IsPotentiallyPersonal` tags which fields are personal and, for example, require audit logs if modified.

cds

```
annotate my.Customers with {
  firstName @PersonalData.IsPotentiallyPersonal;
  lastName  @PersonalData.IsPotentiallyPersonal;
  email     @PersonalData.IsPotentiallyPersonal;
  phone     @PersonalData.IsPotentiallyPersonal;
};
```

@PersonalData .IsPotentiallySensitive

`@PersonalData.IsPotentiallySensitive` tags which fields are sensitive and, for example, require audit logs in case of access.

cds

```
annotate my.Customers with {
  creditCardNo @PersonalData.IsPotentiallySensitive;
};
```

Next Steps...

Having annotated your data model with *`@PersonalData`* annotations, you can now go on to the respective tasks that leverage these annotations to automate as much as possible:

- *Automated Audit Logging*
- *Personal Data Management*
- *Data Retention Management*

Edit this page

Last updated: 09/10/2025, 07:18

Previous page
Data Privacy

Next page
Automatic Audit Logging

Was this page helpful?

👍 👎