Data Privacy Compliance API

API Framework

Summary

This codebase collects laws relating to data privacy and protection, in order to make compliance risk assessments about the handling of personal information. The fundamental output is an assessment of the regulatory risks generated by a proposed use of someone's personal information. The interface to this knowledge base is a REST web service API.

Basic web service inputs (see the API Definition section below for more details):

* The intended activities related to the data (e.g. publication, processing, transfer, etc.)
* Facts about the person ("data subject") whose personal information is of concern
* Facts about the data processor & controller
* Facts about the data recipient

Basic API outputs:

* Assumptions made
* Overall risk factor:
  - Color code on a risk spectrum (green, yellow, orange, red)
  - Risk percentage
* For each regulatory scheme:
  - Name
  - Whether it applies
  - Overall risk of noncompliance
  - Reasons for possible noncompliance
* List of inputs that are missing + relevant, in order of usefulness

Sources of uncertainty:

* Missing inputs
* Incomplete legal rules

The goal is to keep the API response time under 50ms.

API

Definition

Defines the API. Note: list parameters are passed by repeating the url querystring: https://<url>?IntendedActivities=Transmission&IntendedActivities=Publication&...

```
api = APIFunction[{
    "IntendedActivities" → <|"Interpreter" → AnySubset[listOfActivities],
      "Input" → {}, "Required" → False|>,

    "DateOfIntendedActivity" → <|"Interpreter" → "Date", "Required" → False|>,
    "DateOfDataCollection" → <|"Interpreter" → "Date", "Required" → False|>,

    "SubjectType" → <|"Interpreter" → listOfEntityTypes, "Required" → False|>,
    "ControllerType" → <|"Interpreter" → listOfEntityTypes, "Required" → False|>,
    "ProcessorType" → <|"Interpreter" → listOfEntityTypes, "Required" → False|>,
    "RecipientType" → <|"Interpreter" → listOfEntityTypes, "Required" → False|>,

    "SubjectCountry" → <|Interpreter → countryEntities, "Required" → False|>,
    "SubjectCountryRegion" → <|
      Interpreter → "String", "Input" → "", "Required" → False|>,

    "ControllerCountry" → <|Interpreter → countryEntities, "Required" → False|>,
    "ProcessorCountry" → <|Interpreter → countryEntities, "Required" → False|>,
    "RecipientCountry" → <|Interpreter → countryEntities, "Required" → False|>,

    "ControllerIndustry" → <|
      "Interpreter" → listOfIndustries, "Required" → False|>,
    "ProcessorIndustry" → <|"Interpreter" → listOfIndustries,
      "Required" → False|>,

    "Contents" → <|"Interpreter" → AnySubset[listOfContents],
      "Input" → {}, "Required" → False|>,

    "Source" → <|"Interpreter" → {"Data subject", "Public register",
        "Public domain", "Third party"}, "Input" → "", "Required" → False|>,

    "ProcessingPurposes" → <|"Interpreter" → AnySubset[listOfPurposes],
      "Input" → {}, "Required" → False|>,
    "CollectionPurposes" → <|"Interpreter" → AnySubset[listOfPurposes],
      "Input" → {}, "Required" → False|>,

    "Consents" → <|"Interpreter" → AnySubset[listOfConsentTypes],
      "Input" → {}, "Required" → False|>,
    "SubjectAge" → <|"Interpreter" → "Number", "Input" → 0, "Required" → False|>,
    "StudentQ" → <|"Interpreter" → {True, False, missing},
      "Input" → missing, "Required" → False|>,

    "Anonymization" → <|"Interpreter" → listOfAnonymizationTypes,
      "Input" → "Neither", "Required" → False|>
   },
   initialize[#] &, "JSON"
  ];
```

Lists used in the API definition:

```
listOfActivities = {"Collection", "Deletion",
   "Disclosure", "Processing", "Publication", "Transfer", "Storage"};
listOfEntityTypes = {"Natural person", "Corporation",
   "Nonprofit organization", "National government",
   "Subnational government", "Intergovernmental organization"};
listOfIndustries = {"Public communications network",
   "Video rental or distribution"};
listOfContents = {"Address", "Age", "Audio", "Authenticating information",
   "Biometric", "Civil legal judgment",  "Consumer credit information",
   "Criminal history", "Cultural data", "Customer purchase history",
   "Date of birth", "Economic data",  "Educational information",
   "Family information", "Financial data", "Financial messaging data",
   "Gender", "Genetic", "Health data", "IP address", "Name",
   "National identifier", "Occupation", "Online identifier",
   "Organizational title", "Passenger name record", "Photograph",
   "Political opinion", "Racial or ethnic origin", "Religious or other belief",
   "Sex life information", "Social data", "Telephone number", "Location",
   "Trade union membership", "Unique identifier", "Vehicle data", "Video"};
listOfPurposes = {"Artistic or literary expression", "Authentication",
   "Religious membership", "Clinical trial", "Compliance with legal obligation",
   "Consumer credit evaluation", "Contract performance", "Criminal procedure",
   "Employment law or relations", "Exercise of official authority",
   "Financial transaction", "Historical", "Insurance claim processing",
   "Journalism", "Law enforcement", "Legal proceeding or claim", "Marketing",
   "National security", "Necessary legitimate interests of controller",
   "Necessary legitimate interests of third party",
   "Nonprofit membership processing", "Official statistics",
   "Payment processing", "Payment collection", "Personal or household activity",
   "Profiling", "Provision of healthcare", "Public health",
   "Public interest", "Public security", "Scientific",
   "Statistical", "Surveillance", "Telecommunications",
   "Vital interests of the data subject", "Vital interests of third party"};
listOfConsentTypes = {"Data subject consent", "Parental consent",
   "Data subject incapable of giving consent",
   "Data made public by data subject"};
listOfAnonymizationTypes = {"Anonymized", "Pseudonymized", "Neither"};

countryEntities = Union[CountryData["Countries", "Name"], {"European Union"}];
```

Deployment

Deploys the API:

```
CloudDeploy[api, "privacy_API", Permissions → "Public"]

CloudDeploy[api, "privacy_API_test", Permissions → "Public"]
```

Global variables

Ordinarily, global variables are to be avoided in favor of a functional programming approach. However, in this application, there are a limited number of input variables and defining them globally has the

benefit of terse, easily understood code. The alternative would be to use expressions like `inputs[["subjectCountry"]]` everywhere, which seems unnecessarily verbose. Because the global variables are read-only, there are no concerns about state or side-effects.

Globals:

```
intendedActivities = {};
activityDate = Today;
collectionDate = Today;
subjectType = "";
controllerType = "";
processorType = "";
recipientType = "";
subjectCountry = "";
subjectCountryRegion = "";
controllerCountry = "";
processorCountry = "";
recipientCountry = "";
controllerIndustry = "";
processorIndustry = "";
contents = {};
source = "";
processingPurposes = {};
collectionPurposes = {};
consents = {};
subjectAge = 0;
studentQ = False;
```

Assumption regarding missing information

Assumptions made:

```
assumptions = {};
```

Sets the global variables, make any necessary assumptions about missing information, and process the API request:

```
initialize[rawInputs_] := Block[{},

  intendedActivities = rawInputs[["IntendedActivities"]];
  If[intendedActivities == {}, AppendTo[assumptions,
    "IntendedActivities = {Processing}"]; intendedActivities = "Processing";];

  activityDate = Today; (* inputs[["DateOfIntendedActivity"]]; *)
  collectionDate = Today; (* inputs[["DateOfDataCollection"]]; *)

  subjectType = rawInputs[["SubjectType"]];
  If[subjectType == Missing["NoInput"], AppendTo[assumptions,
    "SubjectType = Natural person"]; subjectType = "Natural person";];

  controllerType = rawInputs[["ControllerType"]];
  If[controllerType == Missing["NoInput"], AppendTo[assumptions,
    "ControllerType = Corporation"]; controllerType = "Corporation";];

  processorType = rawInputs[["ProcessorType"]];
```

```
If[processorType == Missing["NoInput"], AppendTo[assumptions,
  "ProcessorType = Corporation"]; processorType = "Corporation";];

recipientType = rawInputs[["RecipientType"]];
If[recipientType == Missing["NoInput"], AppendTo[assumptions,
  "RecipientType = Corporation"]; recipientType = "Corporation";];

subjectCountry := rawInputs[["SubjectCountry"]];
If[subjectCountry == Missing["NoInput"], AppendTo[assumptions,
  "SubjectCountry = United States"]; subjectCountry = "United States";];

subjectCountryRegion = rawInputs[["SubjectCountryRegion"]];
If[subjectCountryRegion == Missing["NoInput"], AppendTo[assumptions,
  "SubjectCountryRegion = ''"]; subjectCountryRegion = "";];

controllerCountry = rawInputs[["ControllerCountry"]];
If[controllerCountry == Missing["NoInput"],
 AppendTo[assumptions, "ControllerCountry = United States"];
 controllerCountry = "United States";];

processorCountry = rawInputs[["ProcessorCountry"]];
If[processorCountry == Missing["NoInput"], AppendTo[assumptions,
  "ProcessorCountry = United States"]; processorCountry = "United States";];

recipientCountry = rawInputs[["RecipientCountry"]];
If[recipientCountry == Missing["NoInput"], AppendTo[assumptions,
  "RecipientCountry = United States"]; recipientCountry = "United States";];

controllerIndustry = rawInputs[["ControllerIndustry"]];
If[controllerIndustry == Missing["NoInput"],
 AppendTo[assumptions, "ControllerIndustry = ''"]; controllerIndustry = "";];

processorIndustry = rawInputs[["ProcessorIndustry"]];
If[processorIndustry == Missing["NoInput"],
 AppendTo[assumptions, "ProcessorIndustry = ''"]; processorIndustry = "";];

contents = rawInputs[["Contents"]];
If[contents == {}, AppendTo[assumptions, "Contents = {}"]; contents = {};];

source = rawInputs[["Source"]];
If[source == "",
 AppendTo[assumptions, "Source = Third party"]; source = "Third party";];

processingPurposes = rawInputs[["ProcessingPurposes"]];
If[processingPurposes == {},
 AppendTo[assumptions, "ProcessingPurposes = {}"]; processingPurposes = {};];

collectionPurposes = rawInputs[["CollectionPurposes"]];
If[collectionPurposes == {},
 AppendTo[assumptions, "CollectionPurposes = {}"]; collectionPurposes = {};];

consents = rawInputs[["Consents"]];
If[consents == {}, AppendTo[assumptions, "Consents = {}"]; consents = {};];

subjectAge = rawInputs[["SubjectAge"]];
```

```
 If[subjectAge == 0,
  AppendTo[assumptions, "SubjectAge = 26"]; subjectAge = 26;];

 studentQ = rawInputs[["StudentQ"]];
 If[studentQ == missing,
  AppendTo[assumptions, "StudentQ = False"]; studentQ = False;];

 Return[assess];
]
```

Risk Assessment

High-level control

Determines the main output of the API:

```
assess := Block[{result, risks, reasonList, text, start,
    end, color, regsChecked, risk, overallRisk, colorObject, echo},

   start = AbsoluteTime[];

   regsChecked =
    {riskReportEUGDPR , riskReportCOPPA, riskReportVPPA, riskReportCalCiv1798 };

   risk = cumulativeRiskSimple[Map[#[["riskPercentage"]] &, regsChecked]];

   colorObject = colorCode[risk];

   overallRisk = {
     "riskPercentage" → risk,
     "colorCode" → colorObject[[1]],
     "text" → colorObject[[2]]
    };

   end = AbsoluteTime[];

   result = <|
     "assumptions" → assumptions,
     "overallRisk" → overallRisk,
     "regulationsChecked" → regsChecked,
     "processingTimeInMS" → (end - start) * 1000
     |>;

   Return[result];
  ];
```

Accumulate evidence of compliance risk

Data structure (example):

```
riskFactorsExample = {{0.1, "A"}, {0.2, "B"}, {0.15, "C"}};
```

Function that determines the cumulative risk based on all risk factors. It uses the formula R = 1 - (1 – r_1)(1 - r $_2$)(1 - r $_3$) ..., representing the compound risk of independent events.

```
cumulativeRisk[riskFactors_List] :=
 1 - Apply[Times, Map[1 - # &, Map[First[#] &, riskFactors]]]
```

```
cumulativeRisk[riskFactorsExample]
```

```
0.388
```

Same as above, but it operates on a list of numbers.

```
cumulativeRiskSimple[riskFactors_List] := 1 - Apply[Times, Map[1 - # &, riskFactors]]
```

Generate a list of the risk factors in order of most to least risky:

```
riskFactorsInOrder[riskFactors_List] := Map[#[[2]] &, Reverse[Sort[riskFactors]]]
```

```
riskFactorsInOrder[riskFactorsExample]
```

Generate output data structure for each regulation

Returns a data structure like: {“name”→ “EU General Data Protection Directive (proposed)”, “applicable”→ True,”riskPercentage”→ 0.17, “riskFactors”→ {“a”,”b”,”c”}}

```
riskReport[riskFactors_, regName_, citation_, applicable_ ] := Block[{factors},

  If[applicable, factors = ReleaseHold[riskFactors]];

  Return[<|
    "name" → regName,
    "citation" → citation,
    "applicable" → applicable,
    "riskPercentage" → If[! applicable, 0, cumulativeRisk[factors]],
    "riskFactors" → If[! applicable, {}, riskFactorsInOrder[factors]]
    |>]]
```

```
SetAttributes[riskReport, HoldFirst]
```

Assign a color code

Determine the applicable color code (green, yellow, orange, red) and corresponding text.

```
colorCode[risk_] := Which[
  risk == 1, {"Red", "Out of compliance"},
  risk ≥ 0.8, {"Red", "Extreme risk"},
  risk ≥ 0.5, {"Orange", "Significant risk"},
  risk ≥ 0.1, {"Yellow", "Moderate risk"},
  risk > 0, {"Green", "Low risk"},
  True, {"Green", "No risks identified"}
 ]
```

Common Inferences

General

```
controllerIsGovt := IntersectingQ[{controllerType}, {"National government" ,
   "Subnational government", "Intergovernmental organization"}]
```

Consent-related:

```
parentalConsent := MemberQ[consents, "Parental consent"];
subjectConsent := MemberQ[consents, "Data subject consent"];
subjectIncapableOfConsent :=
  MemberQ[consents, "Data subject incapable of giving consent"];
madePublicBySubject := MemberQ[consents, "Data made public by data subject"];
```

Plumbing

Intended activities:

```
activityIs[activities_] := MemberQ[intendedActivities, activities];
activitiesAreAny[activities_List] := IntersectingQ[intendedActivities, activities];
```

Processing purpose:

```
processingFor[purpose_] := MemberQ[processingPurposes, purpose];
processingForAny[purposes_List] := IntersectingQ[processingPurposes, purposes];
```

Data content:

```
contentIs[purpose_] := MemberQ[contents, purpose];
contentIsAny[purposes_List] := IntersectingQ[contents, purposes];
```

Industries:

```
industryIs[industry_] :=
 MemberQ[processorIndustry, industry] || MemberQ[controllerIndustry, industry]
```

Applicable Law

Data privacy and protection laws from around the world are encoded below. The list currently includes only countries with a large number of English speakers. Feel free to add other countries as necessary.

European Union

Common functions

EU member countries

```
euCountryQ[c_] := MemberQ[currentEUMembers, c];

currentEUMembers = CountryData["EuropeanUnion", "Name"];

currentEUMembers
```

```
{Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic,
 Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland,
 Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Poland,
 Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, United Kingdom}
```

General Data Protection Regulation (GDPR)

The following rules model the proposed EU GDPR (Jan. 25, 2012 version), available at http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf.

Art. 1 - Subject matter

```
applicableSubjectMatterEUGDPR := subjectType == "Natural person"
```

Art. 2 - Material scope

```
withinMaterialScopeEUGDPR :=
 (* Art. 2(2)(a) *)
 !processingFor["National security"] &&

  (* Art. 2(2)(b) *)
  !controllerCountry == "European Union" &&

  (* Art. 2(2)(c), referring to TEU ch. 2
   (Specific provisions on the common foreign and security policy) *)
  !(euCountryQ[controllerCountry] && processingFor["National security"]) &&

  (* Art. 2(2)(d) *)
  !(processorType == "Natural person" && controllerType == "Natural person" &&
     processingFor["Personal or household activity"]) &&

  !(controllerIsGovt && processingForAny[
      "Criminal procedure", "Law enforcement", "Public security"])
```

Art. 3 - Territorial scope

```
withinTerritorialScopeGDPR :=
 (* Art. 3(1) *)
 euCountryQ[controllerCountry] ||
  euCountryQ[processorCountry ] ||

(* Art. 3(2) *)
(euCountryQ[subjectCountry ] && ! euCountryQ[controllerCountry] &&
    processingForAny[{"Marketing", "Surveillance"}] )

(* Art. 3(3) *)
(* euMemberExtraterritorialJurisdictionQ[controllerCountry] *)
```

Art. 4 - Definitions

```
(* Art. 4(18) *)
subjectIsChildUnderEUGDPR := subjectAge < 18
```

Art. 6 - Lawfulness of processing

Certain provisions are omitted because they are covered elsewhere or otherwise irrelevant.

```
compliesEUGDPRart6 :=
 (* Art. 6(1) *)
 ! activityIs["Processing"] ||

  (* Art. 6(1)(a) *)
  subjectConsent ||

  (* Art. 6(1)(b)-(d) *)
  processingForAny[{"Contract performance", "Compliance with legal obligation",
    "Vital interests of the data subject"}] ||

  (* Art. 6(1)(e) *)
  processingFor["Public interest"]
```

Art. 8 - Data of a child

Article 8(1):

```
compliesEUGDPRart8 := subjectAge ≥ 13 || parentalConsent
```

Art. 9 - Special categories of personal data

```
compliesEUGDPRart9 :=
 (* Art. 9(1) *)! contentIsAny[
    {"Racial or ethnic origin", "Political opinion", "Religious or other belief",
     "Trade union membership", "Genetic", "Health data", "Criminal history"}] ||

  (* Art. 9(2)(a) *)
  subjectConsent ||

  (* Art. 9(2)(b) *)
  processingFor["Employment law or relations"] ||

  (* Art. 9(2)(c) *)
  (processingForAny[{"Vital interests of data subject",
      "Vital interests of third party"}] && ! subjectIncapableOfConsent) ||

  (* TODO: Art. 9(2)(d) *)

  (* Art. 9(2)(e) *)
  source == "Public domain" ||

  (* Art. 9(2)(f) *)
  (processingFor["Legal proceeding or claim"] &&
    processingForAny[{"Necessary legitimate interests of controller",
      "Necessary legitimate interests of third party"}] ) ||

  (* Art. 9(2)(g) *)
  processingFor["Public interest"] ||

  (* Art. 9(2)(h) *)
  (contentIs["Health data"] && processingForAny[
     {"Provision of healthcare", "Public health"}] && compliesEUGDPRart81) ||

  (* Art. 9(2)(i) *)
  (processingForAny[{"Historical", "Statistical", "Scientific"}] &&
    compliesEUGDPRart83) ||

  (* Art. 9(2)(j) *)
  (contentIs["Criminal history"] && (controllerIsGovt || processingForAny[
       "Compliance with legal obligation", "Exercise of official authority",
       "Legal proceeding or claim", "Public interest"]))
```

Art. 20 - Profiling

```
compliesEUGDPRart20 :=
 (* Art. 20(1) *)
 !processingFor["Profiling"] ||

  (* Art. 20(2)(a) *)
  processingFor["Contract performance"] ||

  (* Art. 20(2)(b) *)
  processingFor["Compliance with legal obligation"] ||

  (* Art. 20(2)(c) *)
  subjectConsent ||

  (* Art. 20(3) *)
  Complement[listOfPurposes,
   {"Racial or ethnic origin", "Political opinion", "Religious or other belief",
    "Trade union membership", "Genetic", "Health data", "Criminal history"}]
```

Art. 33 - Data protection impact statement

```
dataProtectionImpactAssessmentRequiredEUGDPR :=
 (* Art. 33(1) *)
 activityIs["Processing"] &&

  (
   (* Art. 33(2)(a) (see Art. 20) *)
   processingFor["Profiling"] ||

    (* Art. 33(2)(b) *)
    contentIsAny[{"Biometric", "Genetic", "Health data",
      "Racial or ethnic origin", "Sex life information"}] ||
    processingForAny[{"Clinical trial", "Provision of healthcare",
      "Public health", "Public interest", "Statistical"}] ||

    (* Art. 33(2)(a) (see Art. 20) *)
    processingFor["Surveillance"] ||

    (* Art. 33(2)(d) *)
    subjectIsChildUnderEUGDPR || contentIsAny[{"Biometric", "Genetic"}]

  ) &&

  (* Art. 33(5) *)
  !(controllerIsGovt &&
     processingForAny[{"Compliance with legal obligation", "Public interest"}])
```

Art. 41 - Transfers with an adequacy decision

EU regulations on international transfers of personal data are summarized at http://ec.europa.eu/justice/-data-protection/document/international-transfers/adequacy/index_en.htm.

```
compliesEUGDPRart41 :=
 (* Art. 41(1) *)
 ! activityIs["Transfer"] ||

  (* Art. 1(3) *)
  euCountryQ[recipientCountry] ||

  (* Art. 41(1) *)
  MemberQ[{"Andorra", "Argentina", "Faroe Islands",
    "Guernsey", "Israel", "Isle of Man", "Jersey", "New Zealand",
    "Switzerland", "Uruguay", "United States"}, recipientCountry] ||
  (recipientCountry == "Canada" && recipientType == "Corporation") ||

  (* Art. 44(1) *)
  derogationPermittedUnderEUGDPRart44
```

Art. 42 - Transfers with appropriate safeguards

```
riskOfTransferWithoutSafeguardsUnderEUGDPRart42 := activityIs["Transfer"] &&
  ! compliesEUGDPRart41 && ! derogationPermittedUnderEUGDPRart44
```

Art. 43 - Transfers with corporate rules

```
riskOfTransferUnderEUGDPRart43 := activityIs["Transfer"] && ! compliesEUGDPRart41
```

Art. 44 - Transfers, derogations

```
derogationPermittedUnderEUGDPRart44 :=
 (* Art. 44(1)(a) *)
 subjectConsent ||

   (* Art. 44(1)(b)-(c) *)
   processingFor["Contract performance"] ||

   (* Art. 44(1)(d)-(e) *)
   processingForAny[{"Public interest", "Legal proceeding or claim"}] ||

   (* Art. 44(1)(f) *)
   (processingForAny[{"Vital interests of the data subject",
       "Vital interests of third party"}] && subjectIncapableOfConsent) ||

   (* Art. 44(1)(g) *)
   source = "Public register" ||

   (* Art. 44(1)(h) *)
   processingFor["Necessary legitimate interests of controller"]
```

Art. 80 - Freedom of expression

```
euGDPRPossibleRightsConflict :=
 activitiesAreAny[{"Publication", "Processing", "Transmission"}] &&
  processingForAny[{"Artistic or literary expression",
    "Journalism", "Personal or household activity"}]
```

Art. 81 - Data concerning health

```
compliesEUGDPRart81 :=
 (* Art. 81(1) *)
 !contentIs["Health data"] ||
  processingForAny[
   {"Provision of healthcare", "Public health", "Public interest"}] ||

  (* Art. 81(2) *)
  (processingForAny[{"Historical", "Statistical", "Scientific"}] &&
    compliesEUGDPRart83)
```

Art. 83 - Historical, statistical, and scientific purposes

```
compliesEUGDPRart83 :=
 (* Art. 83(1) *)
 (!processingForAny[{"Historical", "Statistical", "Scientific"}] ||
    processingFor["Necessary legitimate interests of controller"]) &&

  (* Art. 83(2) *)
  (!(processingForAny[{"Historical", "Statistical", "Scientific"}] &&
       activityIs["Publication"]) || (subjectConsent ||
      processingFor["Necessary legitimate interests of controller"] ||
      source == "Public domain"))
```

Art. 89 - Relationship to Directive 2002/58/EC

Art. 91 - Entry into force

Temporarily omitted.

Risk analysis

```
riskAnalysisEUGDPR := DeleteCases[{
    (* Outright violations *)
    If[! compliesEUGDPRart6, {1, "Art. 6 - Lawfulness of processing"}],
    If[! compliesEUGDPRart8,
     {1, "Art. 8 - Processing of personal data of a child"}],
    If[! compliesEUGDPRart9,
     {1, "Art. 9 - Processing of special categories of personal data"}],
    If[! compliesEUGDPRart20, {1, "Art. 20 - Measures based on profiling"}],
    If[! compliesEUGDPRart41,
     {1, "Art. 41 - Transfers (of personal data) with an adequacy decision "}],
    If[! compliesEUGDPRart81,
     {1, "Art. 81 - Processing of personal data concerning health"}],
    If[! compliesEUGDPRart83, {1, "Art. 83 - Processing for historical,
        statistical, and scientific research purposes"}],

    (* Mere risks *)
    If[dataProtectionImpactAssessmentRequiredEUGDPR,
     {0.5, "Art. 33 - Data protection impact assessment"}],
    If[riskOfTransferWithoutSafeguardsUnderEUGDPRart42, {0.4,
      "Art. 42 - Transfers (of personal data) by way of adequate safeguards"}],
    If[riskOfTransferUnderEUGDPRart43, {0.4, "Art. 43 - Transfers (of
        personal data) by way of binding corporate rules"}],
    If[euGDPRPossibleRightsConflict, {0.4,
      "Art. 80 - Processing of personal data and freedom of expression"}]

  }, Null]


riskReportEUGDPR := riskReport[
  riskAnalysisEUGDPR ,
  "General Data Protection Directive (EU, proposed)",
  "http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_
    11_en.pdf",
  applicableSubjectMatterEUGDPR &&
   withinMaterialScopeEUGDPR && withinTerritorialScopeGDPR
 ]
```

Australia

Canada

India

Nigeria

Pakistan

United Kingdom

United States

Common functions

U.S. territories:

```
usTerritoryQ[c_] := MemberQ[{"Puerto Rico", "Guam", "Northern Mariana Islands",
    "United States Virgin Islands", "American Samoa"}, c];
usOrUSTerritoryQ[c_] := c == "United States" || usTerritoryQ[c];
```

Children's Online Privacy Protection Act, 16 C.F.R. Part 312

See also 15 U.S.C. 6501 et seq.

Applicability

See en.wikipedia.org/wiki/Children%27s_Online_Privacy_Protection_Act:

```
appliesCOPPA := activityDate ≥ DateObject[{2000, 4, 21}] && coppaRequirementsApply
```

Sec. 312.2 - Definitions

"Child":

```
coppaChild := subjectAge < 13
```

From the definitions of "collection" and "disclosure," and from 312.1:

```
coppaActivities := activitiesAreAny[{"Collection", "Publication", "Transfer"}]
```

"Operator," construed broadly:

```
coppaOperator :=
 (usOrUSTerritoryQ[processorCountry] || usOrUSTerritoryQ[controllerCountry] ||
    usOrUSTerritoryQ[subjectCountry]) && processorType == "Corporation"
```

"Personal information":

```
coppaPII :=
 contentIsAny[{"Address", "Audio", "Authenticating information", "Biometric",
   "Cultural data", "Date of birth", "Gender", "Health data", "IP address",
   "Name", "National identifier", "Online identifier", "Photograph",
   "Racial or ethnic origin", "Religious or other belief", "Social data",
   "Telephone number", "Location", "Unique identifier", "Video"}]
```

Sec. 312.3 - Collection, use, and/or disclosure of personal information from and about children on the Internet

```
coppaRequirementsApply :=
 coppaChild && coppaActivities && coppaOperator  && coppaPII
```

Sec. 312.4 - Notice

```
mustNotifyUnderCOPPAsec4 = True;
```

Sec. 312.5 - Parental consent

```
compliesCOPPAsec5 :=
 parentalConsent || processingForAny[{"Criminal procedure", "Law enforcement",
    "National security", "Public security"}]
```

Sec. 312.6 - Right of parent to review personal information provided by a child

```
mustAllowReviewUnderCOPPAsec6 = True;
```

Sec. 312.10 - Data retention and deletion requirements

```
mustDeleteUnderCOPPAsec10 := activityIs["Storage"]
```

Risk analysis

```
riskAnalysisCOPPA := DeleteCases[{
    (* Outright violations *)
    If[! compliesCOPPAsec5,
     {1, "Sec. 312.5 - Parental consent to collection, use, or
        disclosure of child's personal information"}],

    (* Mere risks *)
    If[mustNotifyUnderCOPPAsec4,
     {0.3, "Sec. 312.4 - Obligation to provide notice to parent"}],
    If[mustAllowReviewUnderCOPPAsec6,
     {0.3, "Sec. 312.6 - Right of parent to review personal
        information provided by a child"}],
    If[mustDeleteUnderCOPPAsec10 , {0.1, "Sec. 312.10 - Obligation to
        retain data for only as long as reasonably necessary"}]

  }, Null]


riskReportCOPPA := riskReport[
  riskAnalysisCOPPA ,
  "Children's Online Privacy Protection Act (U.S.)",
  "16 C.F.R. Part 312",
  appliesCOPPA
 ]
```

Video Privacy Protection Act of 1988, 18 U.S.C. 2710

Applicability

Assuming a broad territorial scope:

```
appliesVPPA :=
 activityDate ≥ DateObject[{1988, 11, 5}] && (usOrUSTerritoryQ[processorCountry] ||
    usOrUSTerritoryQ[controllerCountry] || usOrUSTerritoryQ[subjectCountry]) &&
  industryIs["Video rental or distribution"]
```

Sec. 2710(b)(1) - Non-disclosure requirement

```
violatesVPPAb1 := activitiesAreAny[{"Disclosure", "Publication", "Transfer"}] &&
  contentIsAny[{"Address", "Authenticating information",
    "Customer purchase history", "Date of birth", "Gender", "IP address", "Name",
    "Online identifier", "Telephone number", "Location", "Unique identifier"}] &&
  ! disclosureExceptionAppliesVPPA
```

Sec. 2710(b)(2) - Exceptions to non-disclosure requirement

```
disclosureExceptionAppliesVPPA :=
 (* (b)(2)(B) *)
 subjectConsent ||

  (* (b)(2)(C), (b)(3) *)
  processingForAny[
   {"Criminal procedure", "Law enforcement", "Public security"}] ||

  (* (b)(2)(D) *)
  (contentIsAny[{"Name", "Address"}] &&
    (! contentIs["Customer purchase history"] || processingFor["Marketing"])) ||

  (* (b)(2)(F) *)
  processingForAny[
   {"Compliance with legal obligation", "Legal proceeding or claim"}]
```

Sec. 2710(e) - Destruction of old records

```
riskOfViolatingVPPAsec2710e := activityIs["Storage"]
```

Risk analysis

```
riskAnalysisVPPA := DeleteCases[{
    (* Outright violations *)
    If[violatesVPPAb1,
     {1, "18 U.S.C. 2710(b) - Wrongful disclosure of video tape rental
        or sale records"}],

    (* Mere risks *)
    If[riskOfViolatingVPPAsec2710e ,
     {0.2, "18 U.S.C. 2710(e) - Destruction of old records"}]

  }, Null]


riskReportVPPA := riskReport[
  riskAnalysisVPPA ,
  "Video Privacy Protection Act of 1988 (U.S.)",
  "18 U.S.C. 2710",
  appliesVPPA
 ]
```

California "Shine the Light" Law, Cal. Civil Code. § 1798.83

This law requires post-disclosure notification for transfers of a person's information from one corporation to another for direct marketing purposes.

Applicability

See https://en.wikipedia.org/wiki/California_Shine_the_Light_law:

```
appliesCalCiv1798 :=
 (* Wikipedia *)
 activityDate ≥ DateObject[{2005, 1, 1}] &&

  (* 1798.83(a), (e)(2) *)
  (controllerType == "Corporation" || processorType == "Corporation") &&

  (* 1798.83 *)
  subjectCountry == "United States" && subjectCountryRegion == "California"
```

Sec. 1798.83(a) - Disclosure requirement

```
riskOfViolatingCalCiv179883a :=
 (* 1798.83(a), (d)(3) *)
 activitiesAreAny[{{"Disclosure", "Publication", "Transfer"}}] &&
  processingFor["Marketing"] &&
  ! exceptionAppliesCalCiv179883d && personalInfoCalCiv179883d6
```

Sec. 1798.83(d) - Exceptions to disclosure requirement

```
exceptionAppliesCalCiv179883d :=
 (* 1798.83(d)(1)(C) *)
 processingForAny[{"Payment processing", "Insurance claim processing"}] ||

  (* 1798.83(d)(2) *)
  processingFor["Consumer credit evaluation"] ||

  (* 1798.83(d)(3) *)
  processingFor["Payment collection"]
```

Sec. 1798.83(d)(6) - Categories of personal information

```
personalInfoCalCiv179883d6 :=
 contentIsAny[{"Name", "Address", "Online identifier", "Date of birth",
   "Age", "Family information", "Biometric", "Racial or ethnic origin",
   "Religious or other belief", "Occupation", "Telephone number",
   "Educational information", "Political opinion", "Health data",
   "Genetic", "Customer purchase history", "National identifier",
   "Financial data", "Consumer credit information"}]
```

Risk analysis

```
riskAnalysisCalCiv1798 := DeleteCases[{
   (* Risk *)
   If[riskOfViolatingCalCiv179883a ,
    {0.25, "Cal. Civil Code. 1798.83 - Disclosure of
       personal information for direct marketing purposes"}]

  }, Null]


riskReportCalCiv1798 := riskReport[
  riskAnalysisCalCiv1798 ,
  "California 'Shine the Light' Law (U.S.)",
  "Cal. Civil Code. 1798.83",
  appliesCalCiv1798
 ]
```