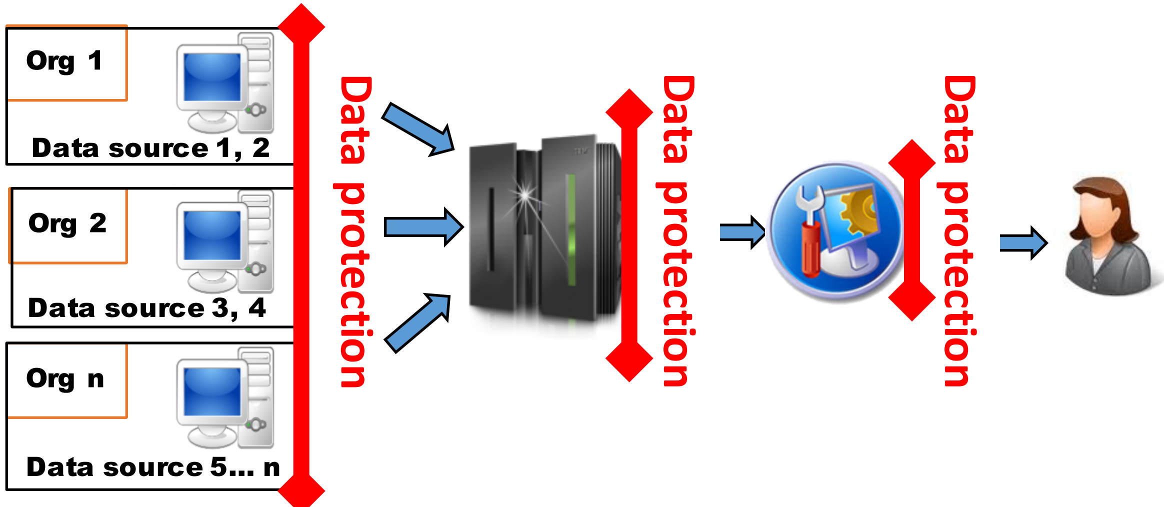
Pull It Together

Enabling Interoperability of Digital Forensic Systems Using a Standard Representation and Supporting API

Sean Barnum, FireEye

Ryan Griffith, DC3



Motivations

- Exchange cyber-investigation information in standardized form
- Interoperability between systems and tools
- Maintain provenance at all phases of cyber-investigation lifecycle
- Provide structured/linked data to support intelligent analysis
- Enhance tool testing and validation of results
- Control access to privileged, proprietary, and personal information
- Restrict use of data covered under license agreements

Evolution of Standard Language

- **CybOX** (Cyber Observable eXpression)
  - Former open-source standard for representing digital objects and interrelationships
  - Subsumed by STIX specifically for cyber threat intelligence
- **DFAX** (Digital Forensic Analysis eXpression) [1]
  - Utilized CybOX for representing digital forensic information
  - Provided specification for representing provenance and forensic actions
- **UCO** (Unified Cyber Ontology)
  - Provide an abstract layer and express constructs that are common across the cyber domain (Action Lifecycle)
- **CASE** (Cyber-investigation Analysis Standard Expression)
  - Representing the broadest possible range of cyber-investigation domains, including digital forensics, incident response, and counter terrorism.

[1] Devised by Eoghan Casey and Sean Barnum at MITRE.org

Community of current contributors

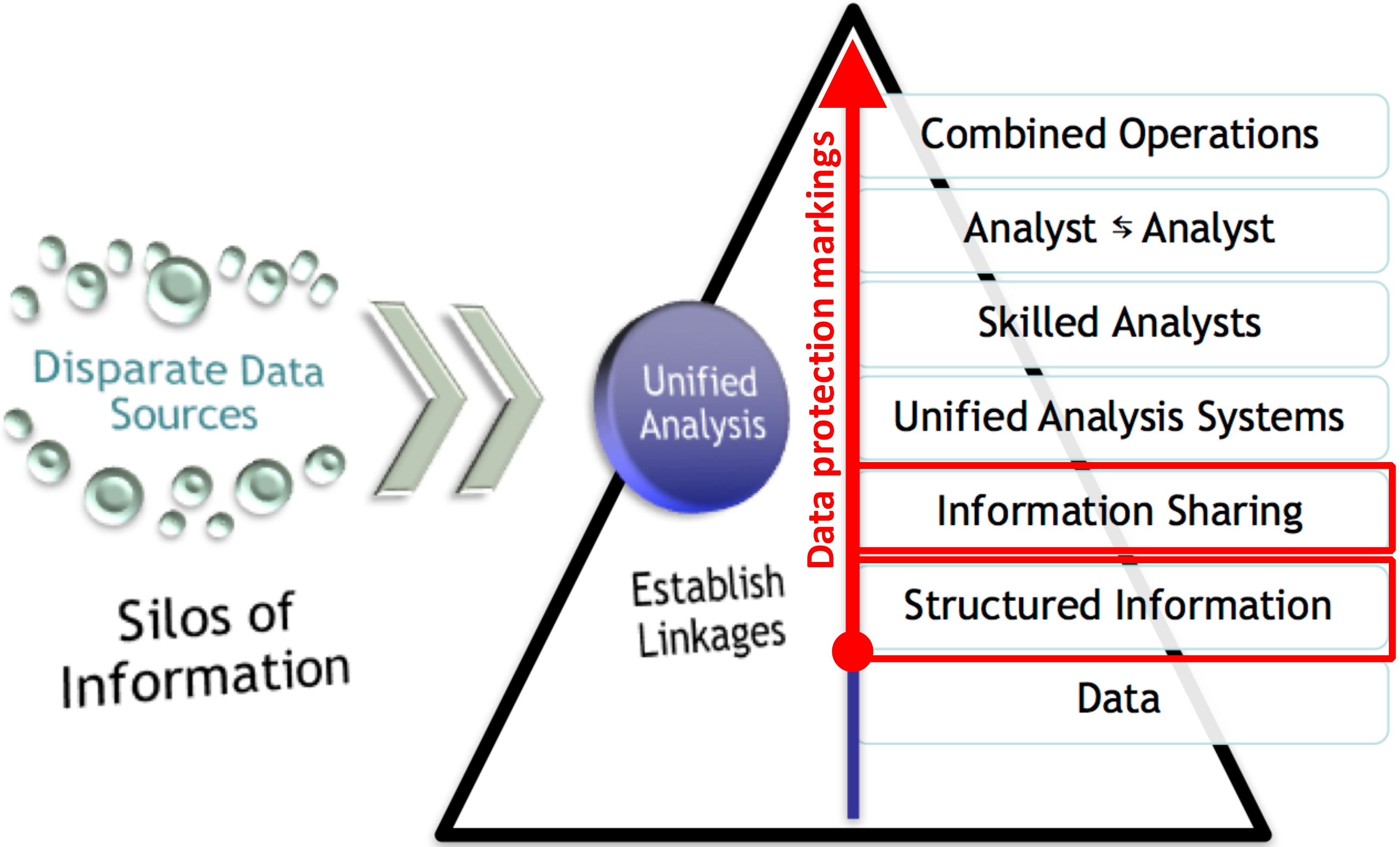
- U.S. Department of Defense Cyber Crime Center (DC3)
- U.S. National Institute of Standards Technology (NIST)
- MITRE
- Netherlands Forensic Institute (NFI)
- University of Lausanne (UNIL)
- EU Evidence project
- Various commercial entities getting involved…

Commercial entities getting involved

- Access Data
- Basis Technology (Autopsy)
- Cellebrite
- FireEye
- Guidance Software
- I2 – IBM
- Magnet Forensics
- Mercure
- Mobile Edit
- Network Miner
- Nuix
- Oxygen
- Volatility
- XRY

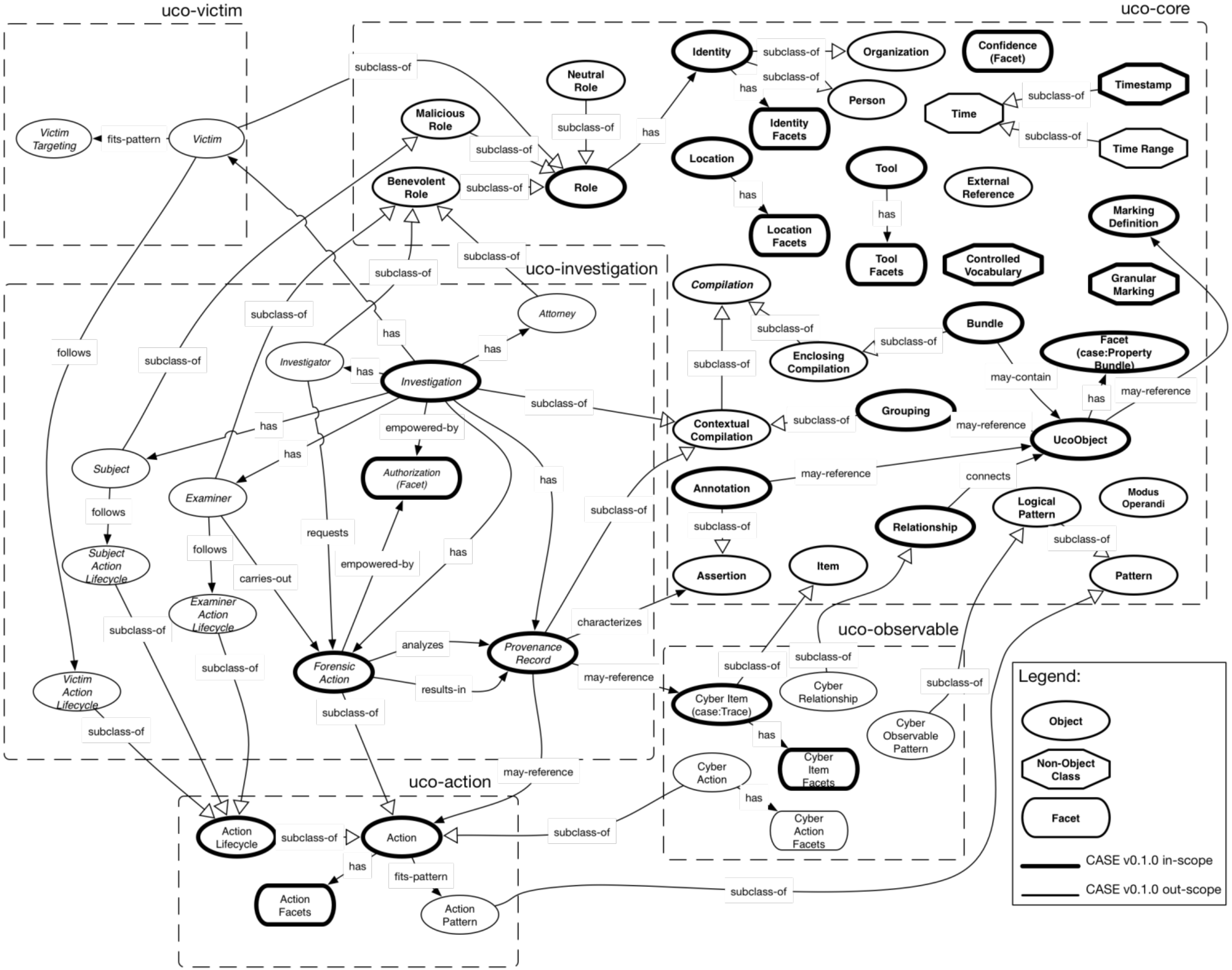
UCO & CASE – The big picture

- Pull together information across investigations, orgs, and domains



UCO

- Digital forensics
- Incident response
- Cyber security

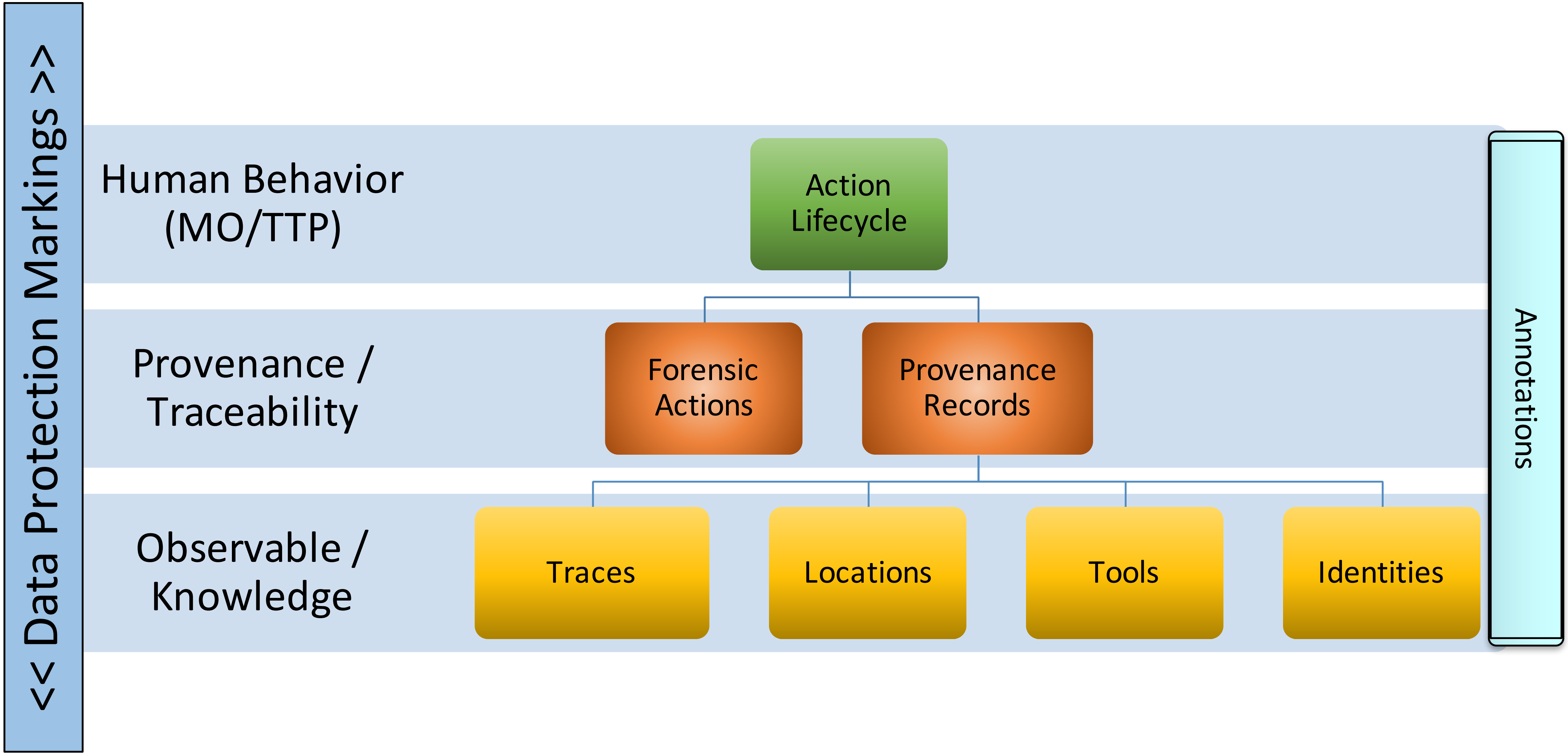


What CASE covers

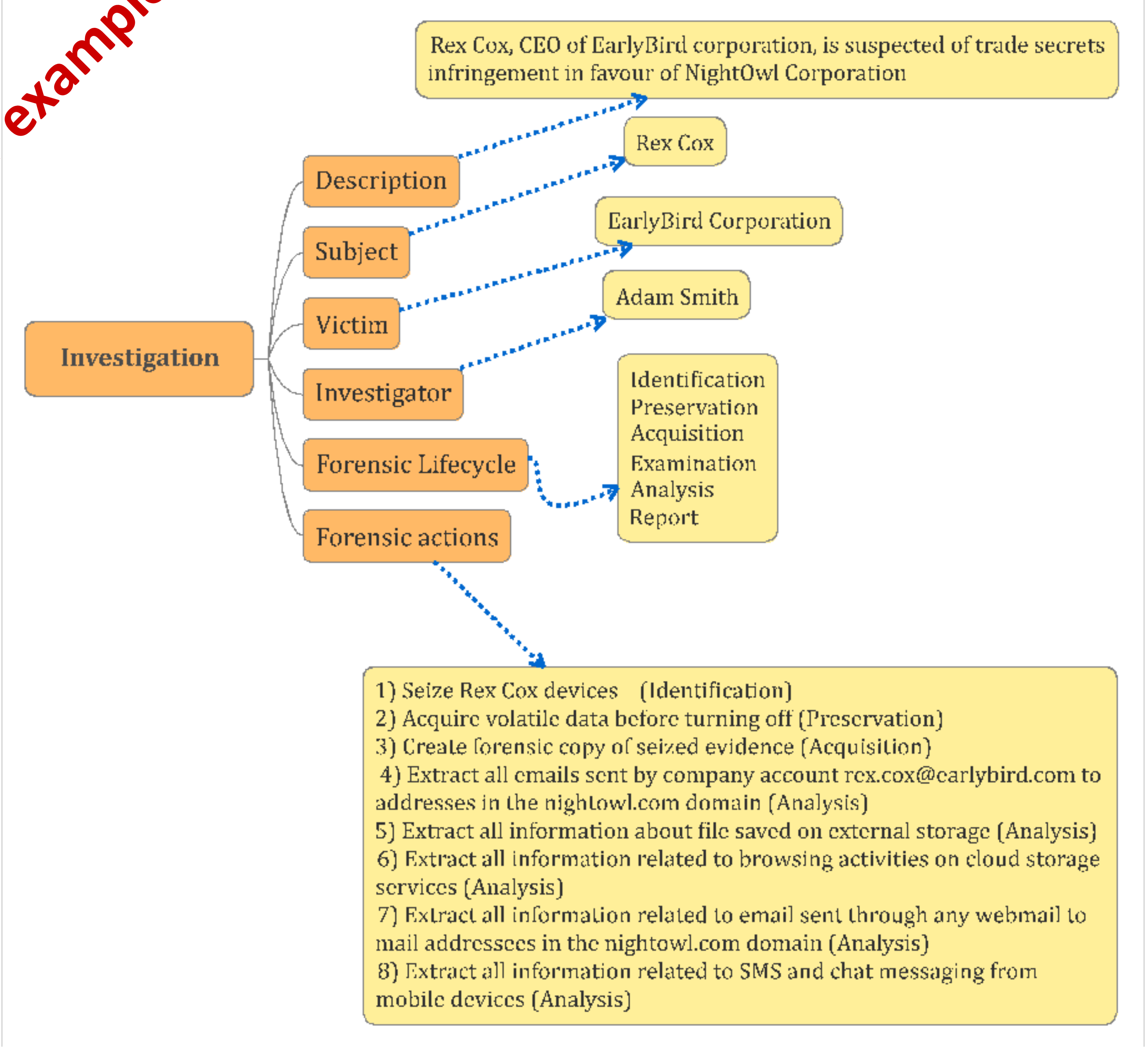
<< Data Protection Markings >>

- Digital traces and their context
  - Digital evidence details, including references, locations
  - Tool output, forensic findings, analysis results
- Provenance & Forensic Actions
  - Case details
  - Who did what to evidence where, when, and how?
- Processes
  - Tool & method details to support reproducibility / transparency
  - Kill chain phases of cyber attacks or grooming phases in sex offenses
- Offender & Victim Behaviors
  - Actions associated with known behaviors (e.g., evidence destruction or concealment)

CASE overview



CASE – investigative example



Investigation

Description

Rex Cox, CEO of EarlyBird corporation, is suspected of trade secrets infringement in favour of NightOwl Corporation

Subject

Rex Cox

Victim

EarlyBird Corporation

Investigator

Adam Smith

Forensic Lifecycle

Identification
Preservation
Acquisition
Examination
Analysis
Report

Forensic actions

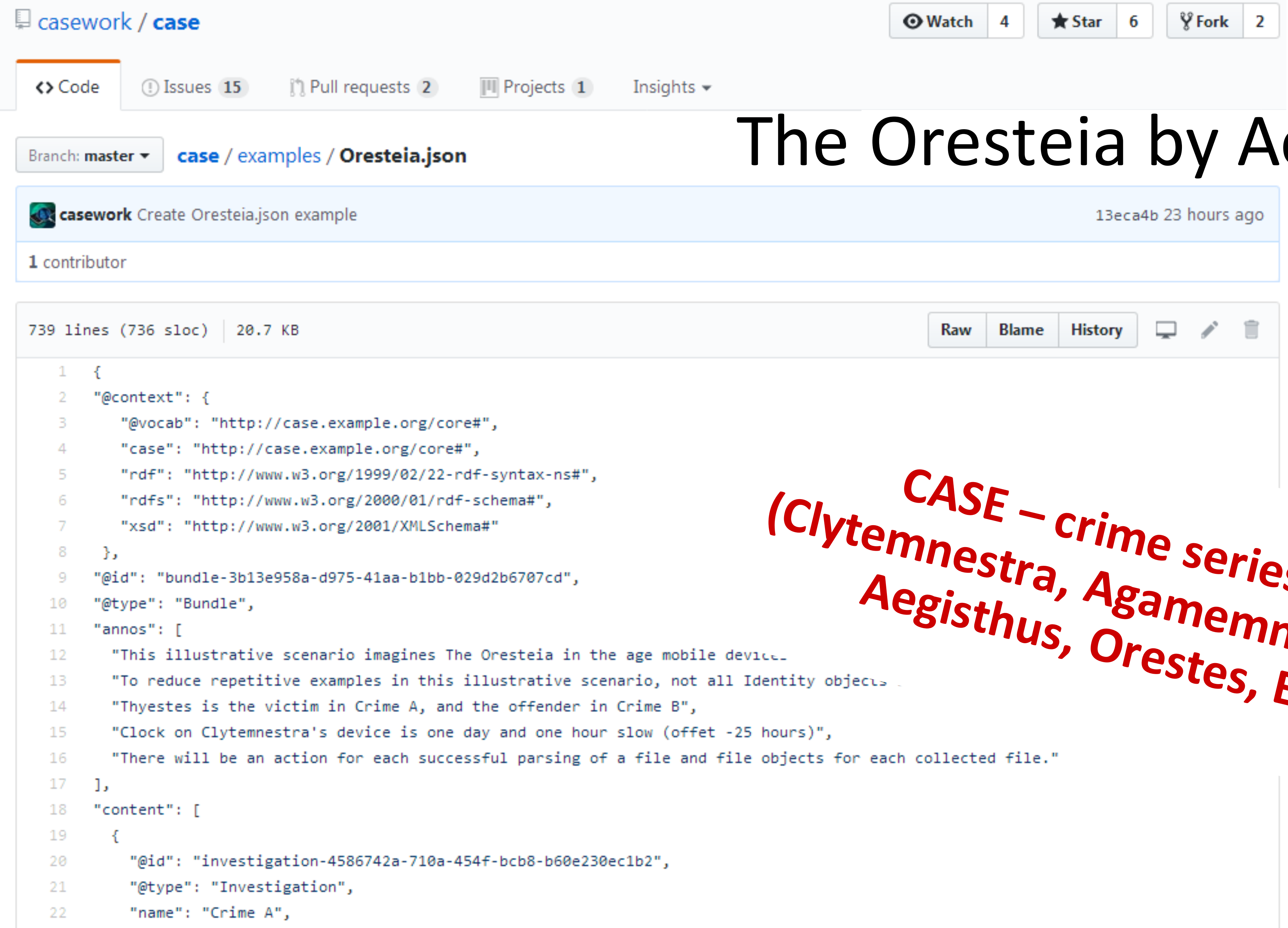
1) Seize Rex Cox devices (Identification)
2) Acquire volatile data before turning off (Preservation)
3) Create forensic copy of seized evidence (Acquisition)
4) Extract all emails sent by company account rex.cox@earlybird.com to addresses in the nightowl.com domain (Analysis)
5) Extract all information about file saved on external storage (Analysis)
6) Extract all information related to browsing activities on cloud storage services (Analysis)
7) Extract all information related to email sent through any webmail to mail addressees in the nightowl.com domain (Analysis)
8) Extract all information related to SMS and chat messaging from mobile devices (Analysis)

CASE default serialisation is JSON-LD

```
{
  "@context": {
   "@vocab": "https://github.com/casework",
     },
  "@graph": [
     {
     "@id": "digital_photograph1",
     "@type": ”Trace",
     "propertyBundle":  [
      {
        "@type": “File",
        "magicNumber": "/9j/4AAQSkZ",
        "mimeType": "image/jpg"
      },
      {
        "@type": “ContentData",
        "hash“: [
          "@type": "Hash",
          "hashMethod":  "MD5",
          "hashValue": "1D6EBB5A789ABD108FF578263E1F40F3" }
         ]
       },
```

```
       {
   "@type": "RasterPicture",
   "pictureType": "jpg",
   "pictureheight": 12345,
   "picturewidth": 12345,
   "bitsPerPixel": 2
  },
    {
     "@type": “EXIF",
     "exifData": [
           "key":  "Make“,
           "value": "Canon"
           ...
   ]
  },
```

The Oresteia by Aeschylus



Branch: master

case / examples / Oresteia.json

casework Create Oresteia.json example 13eca4b 23 hours ago

1 contributor

739 lines (736 sloc) | 20.7 KB

```
{
"@context": {
   "@vocab": "http://case.example.org/core#",
   "case": "http://case.example.org/core#",
   "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
   "rdfs": "http://www.w3.org/2000/01/rdf-schema#",
   "xsd": "http://www.w3.org/2001/XMLSchema#"
 },
"@id": "bundle-3b13e958a-d975-41aa-b1bb-029d2b6707cd",
"@type": "Bundle",
"annos": [
  "This illustrative scenario imagines The Oresteia in the age mobile devic
  "To reduce repetitive examples in this illustrative scenario, not all Identity objects
  "Thyestes is the victim in Crime A, and the offender in Crime B",
  "Clock on Clytemnestra's device is one day and one hour slow (offet -25 hours)",
  "There will be an action for each successful parsing of a file and file objects for each collected file."
],
"content": [
  {
    "@id": "investigation-4586742a-710a-454f-bcb8-b60e230ec1b2",
    "@type": "Investigation",
    "name": "Crime A",
```

CASE – crime series example (Clytemnestra, Agamemnon, Cassandra, Aegisthus, Orestes, Electra...)

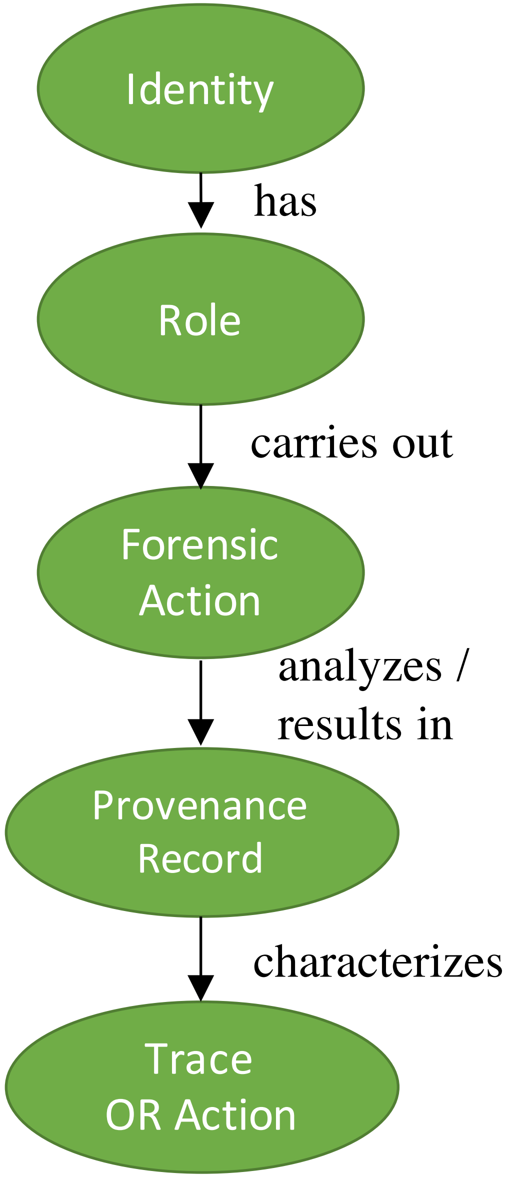
CASE “baked-in” data markings

```
{
  "@id": "marking-01bc435348294d558d520a0790df9",
  "@type": "MarkingDefinition",
  "name": "FIRST.org Mailing List IEP",
  "definitionType": "IEPMarking",
  "definition": [
    {
      "@type": "IEPMarking",
      "version": 1,
      "reference": "https://www.first.org/mailinglistiep",
      "start-date": "2016-06-09 10:09:00",
      "end-date": "2016-12-31 10:09:00",
"encrypt-in-transit": "MAY",
      "encrypt-at-rest": "MAY",
      "permitted-actions": "EXTERNALLY VISIBLE DIRECT ACTS",
      "affected-party-notifications": "MAY",
      "tlp": "AMBER",
      "attribution": "MUST NOT",
      "obfuscate-affected-parties": "MUST",
      "unmodified-resale": "MUST NOT",
      "external-reference":
"www.first.org/about/policies/bylaws"
    }
  ]
}
```

CASE
assert data markings
on bundle (comprehensive)
OR object (granular)

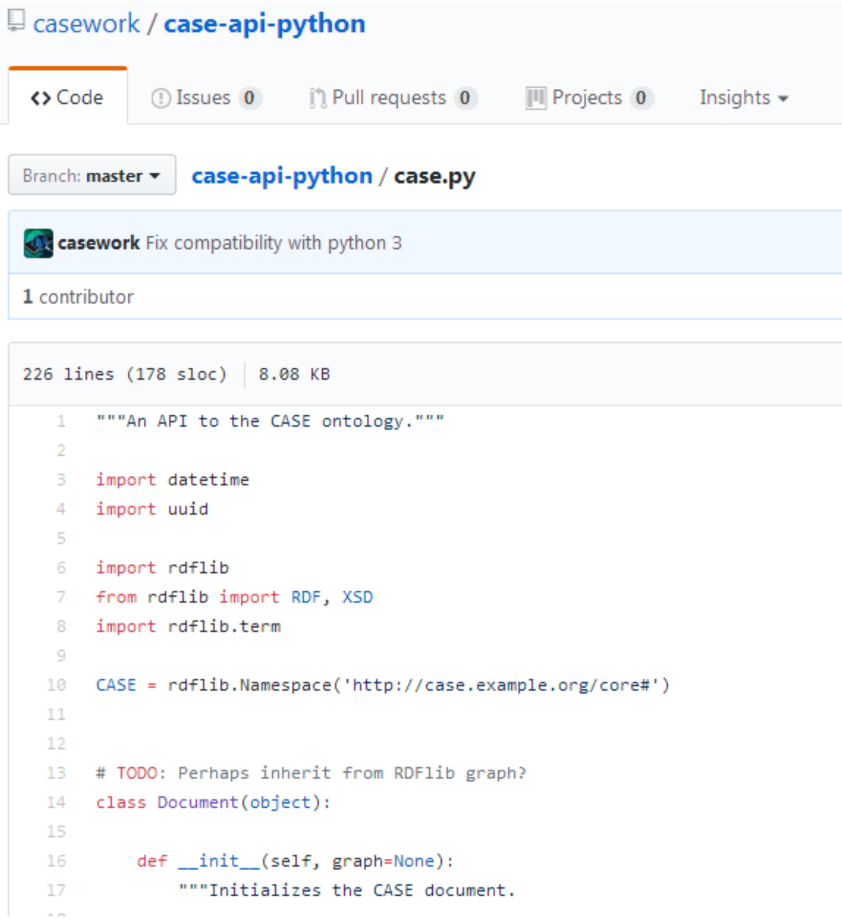
```
{
  "@id": "email-59e9cf76-08c3-4f0b-a319-2a3b55b54f03",
  "@type": "Trace",
  "objectMarking": ["marking-01bc435348294d558d520ab7e0790df9"],
  "propertyBundle": [
   {
     "@type": "EmailMessage",
     "to": ["EmailAccount-bb704188-de16-4743-92fc-b4cba6f9f464"],
     "cc": ["EmailAccount-6c0e2c89-05c2-4713-8a2e-51126725c783"],
     "bcc": ["EmailAccount-a41737ad-558c-44a4-8031-40c623b3f07b"],
     "from": "EmailAccount-bcc67257-331c-4151-8818-1196eb91e7e0",
     "subject": "Example email message",
     "sender": "EmailAccount-bcc67257-331c-4151-8818-1196eb91e7e0",
     "receivedTime": "2017-03-28T13:44:23.40Z",
     "sentTime": "2017-03-28T13:44:22.19Z",
     "messageID": "CAKBqNfyKo+ZXtkz6DUjW-pvHkJy6kwO82jTbkNA@mail.gmail.com"
   }
  ]
},
```

Provenance Records & Forensic Actions



- Evidence handling
  - Who obtained digital traces, when, where, how…
- Evidence processing
  - What tool/method was used, parameters, results…
- Evidence analysis (automated and human)
  - What is the meaning of specific digital traces…

Proof-of-concept API



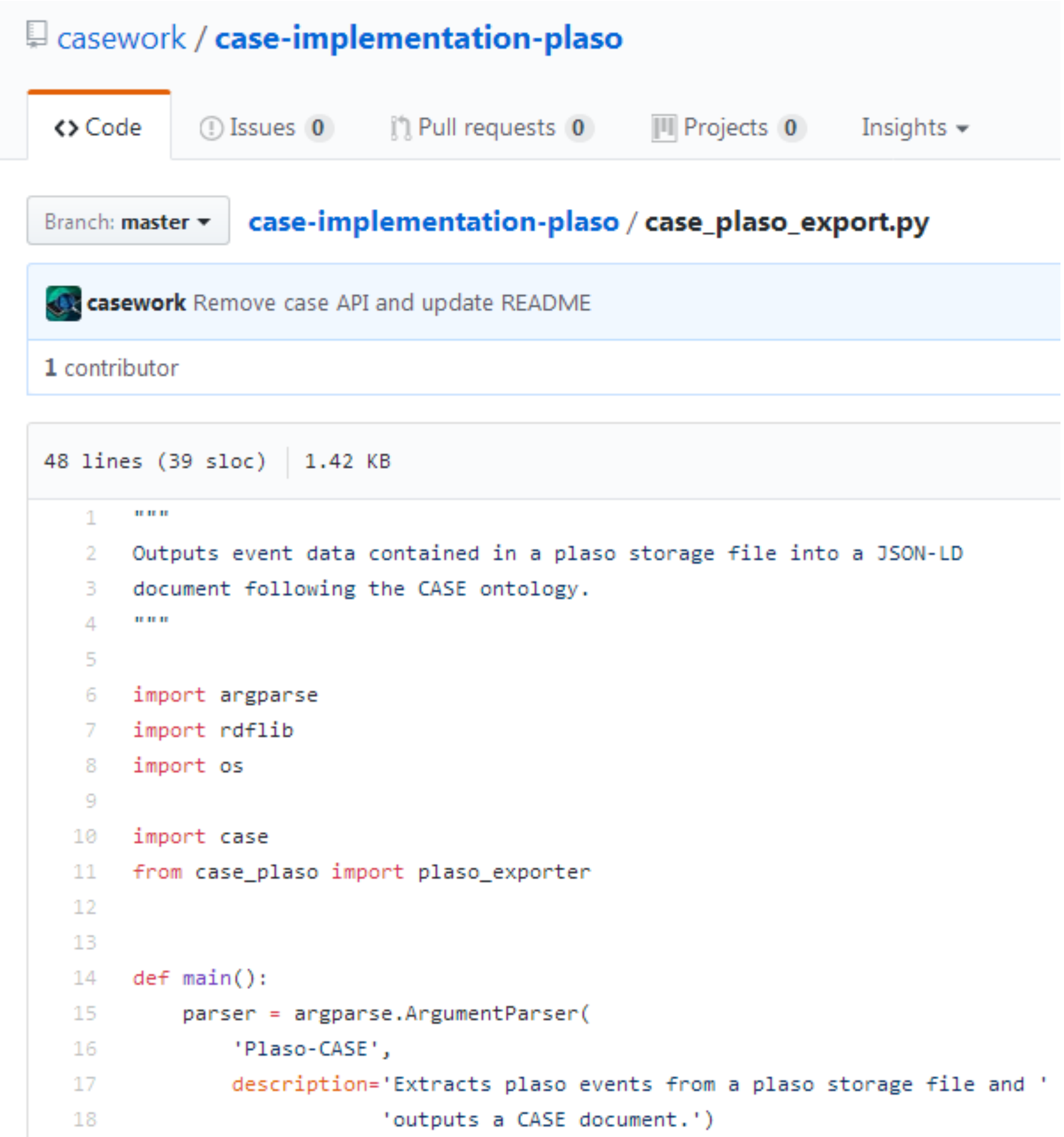
case-api-python / case.py

casework Fix compatibility with python 3

1 contributor

226 lines (178 sloc) | 8.08 KB

```python
"""An API to the CASE ontology."""

import datetime
import uuid

import rdflib
from rdflib import RDF, XSD
import rdflib.term

CASE = rdflib.Namespace('http://case.example.org/core#')


# TODO: Perhaps inherit from RDFlib graph?
class Document(object):

    def __init__(self, graph=None):
        """Initializes the CASE document.
```



case-implementation-plaso / case_plaso_export.py

casework Remove case API and update README

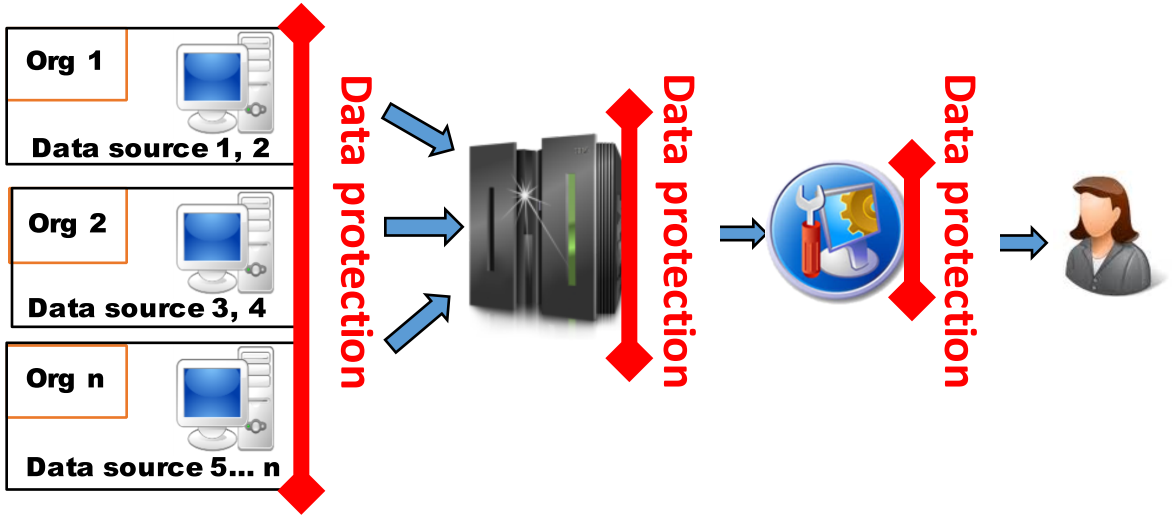
1 contributor

48 lines (39 sloc) | 1.42 KB

```python
"""
Outputs event data contained in a plaso storage file into a JSON-LD
document following the CASE ontology.
"""

import argparse
import rdflib
import os

import case
from case_plaso import plaso_exporter


def main():
    parser = argparse.ArgumentParser(
        'Plaso-CASE',
        description='Extracts plaso events from a plaso storage file and '
                    'outputs a CASE document.')
```

Benefits of standard format

- Less time extracting and combining data
- More time analyzing information
- Breakdown data silos
- Visibility across all sources
- Tool testing and validation
- Find links and patterns
- Data protection



Future work

2017:

- Community review of CASE
- Map current tools and fill gaps (GitHub)
- Implement in current tools

2018:

- Tools exporting CASE packages
- Systems importing CASE packages
- Systems automatically exchanging CASE packages

The Future: Systems fusing CASE packages from all tools… Fusion!

Join us

- CASE GitHub

  https://github.com/casework

- Regular developer calls

- Coordination POC

  cyberinvestigationexpress@gmail.com