A Data Modelling Framework to Unify Cyber Security Knowledge

OmnibusCyber

Authors:

Dr. Paolo Di Prodi

About Me

Paolo Di Prodi

Phd in Machine Learning

Software and Automation Engineer

Mostly Data Science in Cyber Security.

Hands on malware reversing.

Problem statement

Internal representation of cyber data

External: Threat Intelligence Exchange

Real example of streaming infrastructure

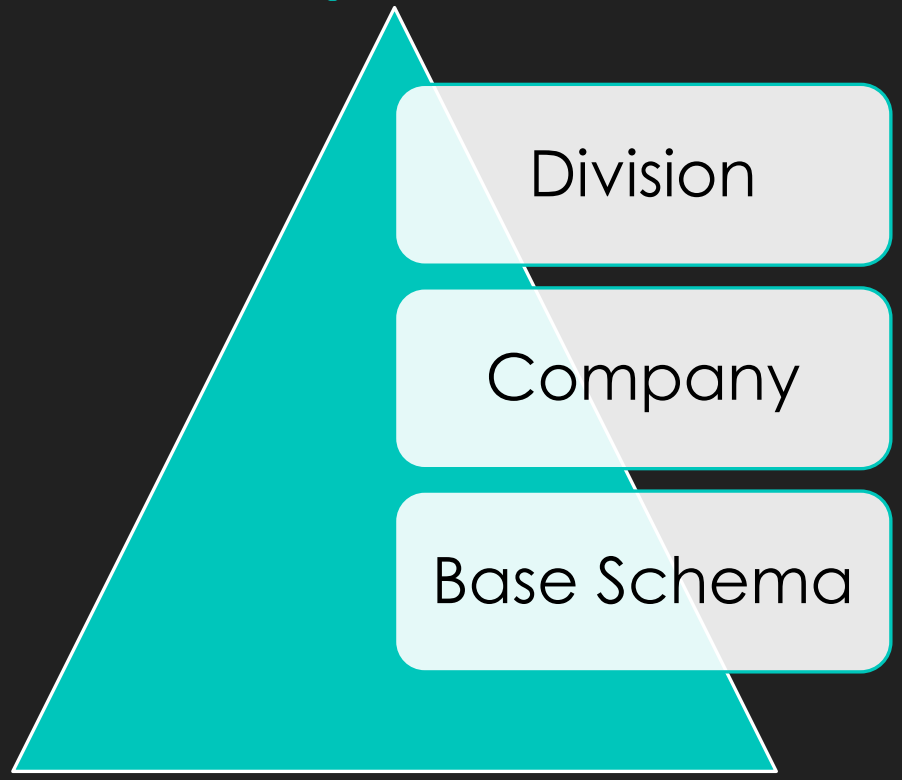
IPS/IDS/AV/EPP/EDR

ProtoBuf/JSON/Avro/MQTT

Central or Distributed

We need a unified data model to avoid silos between sources/products.

Omnibus Goal: flexibility



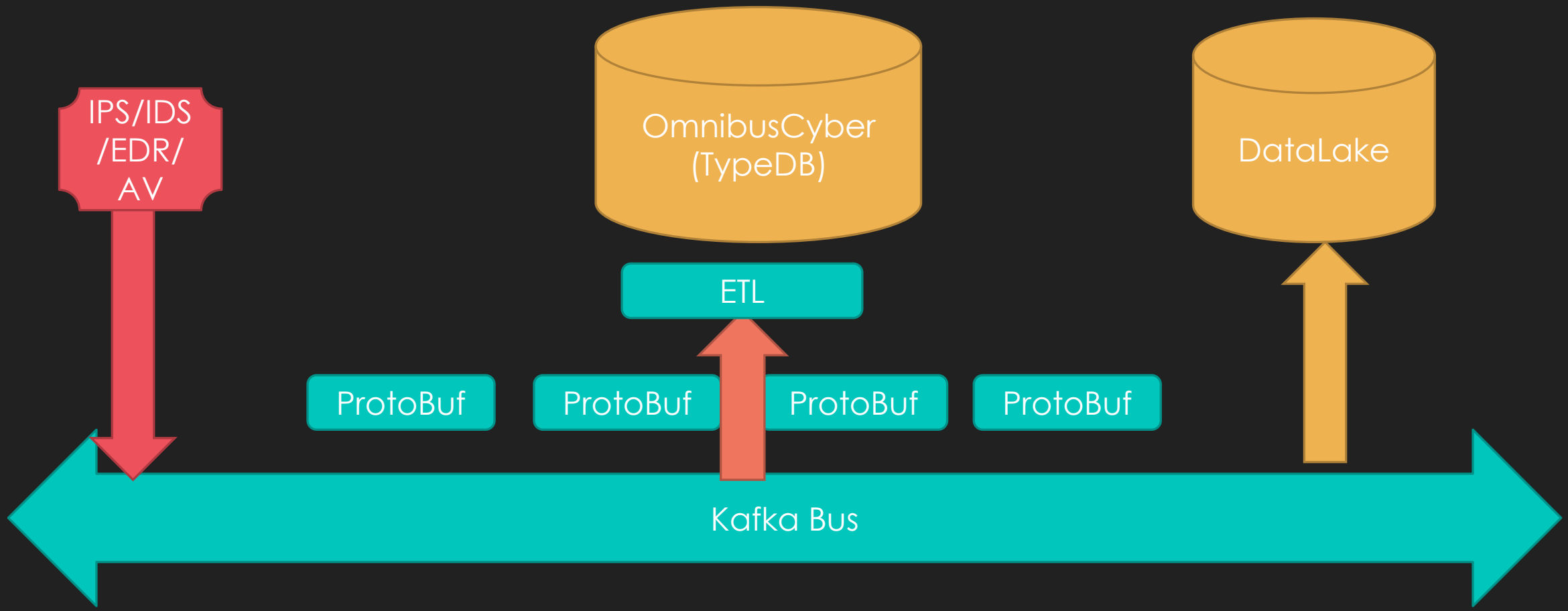
- Base Schema:
  - CVE/CVSS/CWE/CAPEC
  - MAEC
  - COCOA
  - ATT&CK, DEFEND, ATTCK FLOW
  - VERIZON/VERIS
- Basic pattern: inherit and extend

OCSF: https://github.com/ocsf/

Released: August 2022
@BlackHat

Live Demo

Demo video: https://youtu.be/R0fyiBZCEyg



Configuration mapping internals

Inherit and expand

- Example here is to derive CVE entity:
  - Add relation to device object
  - Add relation to volume count
  - Add N-ary relation

device	device	vuln	count
desk1234	2022-10-01	CVE-2022-123	1000

Specific schema

```
# version=1.0
# namespace=mycompany

define

vuln sub cve_record_4,
    plays has_detected:cve;

sensor sub device,
    plays has_detected:device;

volume sub metric,
    plays has_detected:metric;

has_detected sub relation,
    owns timestamp,
    relates cve,
    relates metric,
    relates device;
```

Simple YAML config

```yaml
version: 1.0
type: protobuf
entities:
  hostname:
    entity: "sensor"
    attribute: "hostname"
  cve:
    entity: "vuln"
    attribute: "id"
  count:
    entity: "volume"
    attribute : "data"
relations:
  timestamp:
    relation: "has_detected"
    attribute: "timestamp"
    roles:
      vuln: "cve"
      volume: "metric"
      sensor: "device"
```

Going forward

Auto Load

- Each entity should have an authoritative source
- This means auto enrichment in real time if required.

ML

- Community Detection
- Link Prediction
- Graph Embeddings

OCSF

- Need to build a JSON schema to TypeDB transformer
- Need to avoid shortcomings of JSON schema and TypeDB

Project: https://github.com/robomotic/omnicyberdb/tree/experimental

TypeDB limitations

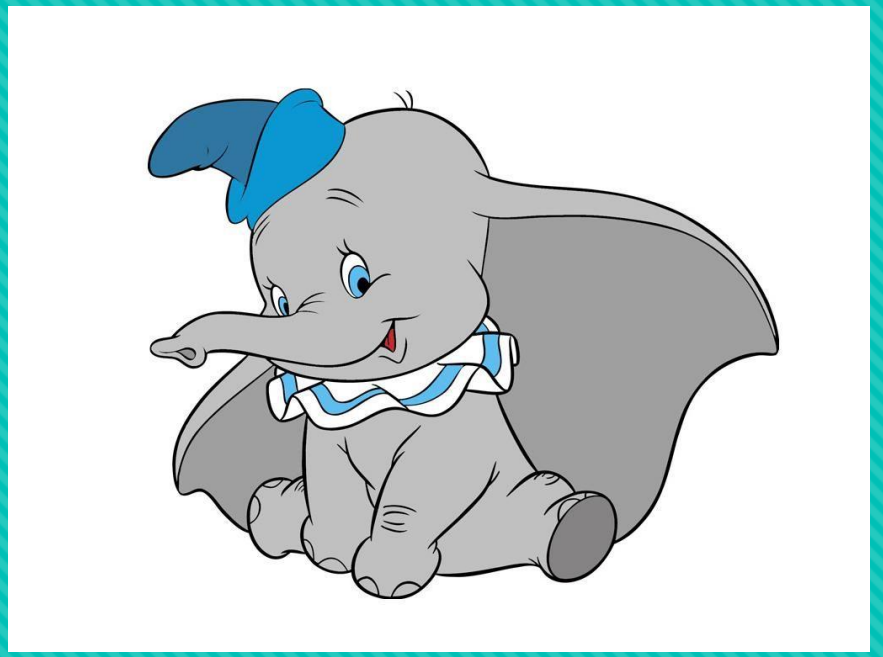
Schema	Scope	Data	Queries
• Dependencies • Annotations • Keyword escaping	• Namespaces • Versions • Multiple Inheritance	• Validation • Array/Vector Type • Orphan attributes handling • Upsert!!!	• Materialized Views? • More aggregation operators!

Time for Q&A

Email: paolo.research@fortinet.com OR paolo@robomotic.com

FortiGuard labs: https://www.fortinet.com/fortiguard/labs

Appendix



Meet the elephants

UCO ad OCSF

UCO

Unified Cyber Ontology (UCO)

- **A foundation for standardized information representation across the cyber security domain/ecosystem**
- Last version: 0.9.0 on 16 June 2022
- First Version: 01.0 on 5 Jan 2017
- Based on:
  - OWL
  - Java 11
- Key stats:
  - 418 Classes
  - 707 Properties
  - 11812 Triples

RDF Adoption

Focus on Observables

Triple Store DB

SparQL for query

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX type:
<http://dbpedia.org/class/yago/> PREFIX prop: <http://dbpedia.org/property/> SELECT
?country_name ?population WHERE { ?country a type:LandlockedCountries ; rdfs:label
?country_name ; prop:populationEstimate ?population . FILTER (?population > 15000000) . }
```

UCO continued

Schema Visualization

- https://ontology.unifiedcyberontology.org/uco/documentation/entities-az.html
- WebVOWL:
- https://service.tib.eu/webvowl/#iri=https://ontology.unifiedcyberontology.org/uco/observable

Example IPv4 Address

Implementation

```
@prefix observable: <https://ontology.unifiedcyberontology.org/uco/observable/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .

observable:IPv4Address a owl:Class,
        sh:NodeShape ;
    rdfs:label "IPv4Address"@en ;
    rdfs:comment "An IPv4 (Internet Protocol version 4) address is an IPv4 standards conformant
    rdfs:subClassOf observable:IPAddress ;
    sh:targetClass observable:IPv4Address .
```

Open Cybersecurity Schema Framework (OCSF)

- The Open Cybersecurity Schema Framework is an open-source project, delivering an extensible framework for developing schemas, along with a vendor-agnostic core security schema. Vendors and other data producers can adopt and extend the schema for their specific domains.
- OCSF is intended to be used by both products and devices which produce log events, analytic systems, and logging systems which retain log events.
- First Version: 14 July 2022
- Schema: JSON

Based on JSON Schema

There is no reference database implementation.

JSON-Schema supports schema *composition* but not inheritance

- Schema Browser
- https://schema.ocsf.io/


```
"ip_t": {
  "caption": "IP Address",
  "description": "Internet Protocol address (IP address), in either IPv4 or IPv6 format.",
  "max_len": 40,
  "observable": 2,
  "regex": "/^(?>(?>([a-f0-9]{1,4})(?>:(?1)){7}|(?!(?:.*[a-f0-9](?>:|$)){8,})((?1)(?>:
  "type": "string_t",
  "type_name": "String"
},
```


IP Address	ip_t	String	Max length: 40 /^(?>(?>([a-f0-9]{1,4})(?>:(?1)){7}|(?!(?:.*[a-f0-9](?>:|$)){8,})((?1)(?>:(?1)){0,6})?::(?2)?)\|(?>(?>(?1)(?>:(?1)){5}:\|(?!(?:.*[a-f0-9]:){6,})(?3)?::(?>((?1)(?>:(?1)){0,4}):)?)?(25[0-5]\|2[0-4][0-9]\|1[0-9]{2}\|[1-9]?[0-9])(?>\.(?4)){3}))$/iD	Internet Protocol address (IP address), in either IPv4 or IPv6 format.

Our advantages

Extensibility	Reference implementation	ER	Sharing
• Base Schema • Inheritance	• TypeDB • Toolbox	• Entity-Relationships • URI	• Native STIX import/export

Why not everything STIX?

Examples


```
{
 "type": "vulnerability",
 "spec_version": "2.1",
 "id": "vulnerability--0c7b5b88-8ff7-4a4d-aa9d-feb398cd0061",
 "created": "2016-05-12T08:17:27.000Z",
 "modified": "2016-05-12T08:17:27.000Z",
 "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
 "name": "CVE-2016-1234",
 "external_references": [
  {
   "source_name": "cve",
   "external_id": "CVE-2016-1234"
  }
 ]
}
```


Not Applicable Common Properties
defanged, **extensions**
Vulnerability Specific Properties
name, **description**

Property Name	Type	Description
type (required)	string	The value of this property MUST be vulnerability.
external_references (optional)	list of type external-reference	A list of external references which refer to non-STIX information. This property **MAY** be used to provide one or more Vulnerability identifiers, such as a CVE ID [CVE]. When specifying a CVE ID, the **source_name** property of the external reference **MUST** be set to cve and the **external_id** property **MUST** be the exact CVE identifier.
name (required)	string	A name used to identify the Vulnerability.
description (optional)	string	A description that provides more details and context about the Vulnerability, potentially including its purpose and its key characteristics.

What about CWE,CAPEC, ATTCK?

Source	Relationship Type	Target	Description
—	—	—	—
Reverse Relationships			
attack-pattern, campaign, intrusion-set, malware, threat-actor, tool	targets	vulnerability	See forward relationship for definition.
malware	exploits	vulnerability	See forward relationship for definition.
course-of-action	mitigates, remediates	vulnerability	See forward relationship for definition.
infrastructure	has	vulnerability	See forward relationship for definition.

STIX Databases and Extensions

Section 7.3

- Extension Definition Policy
- JSON schema

Section 11

- Custom Object Extensions
- Deprecated

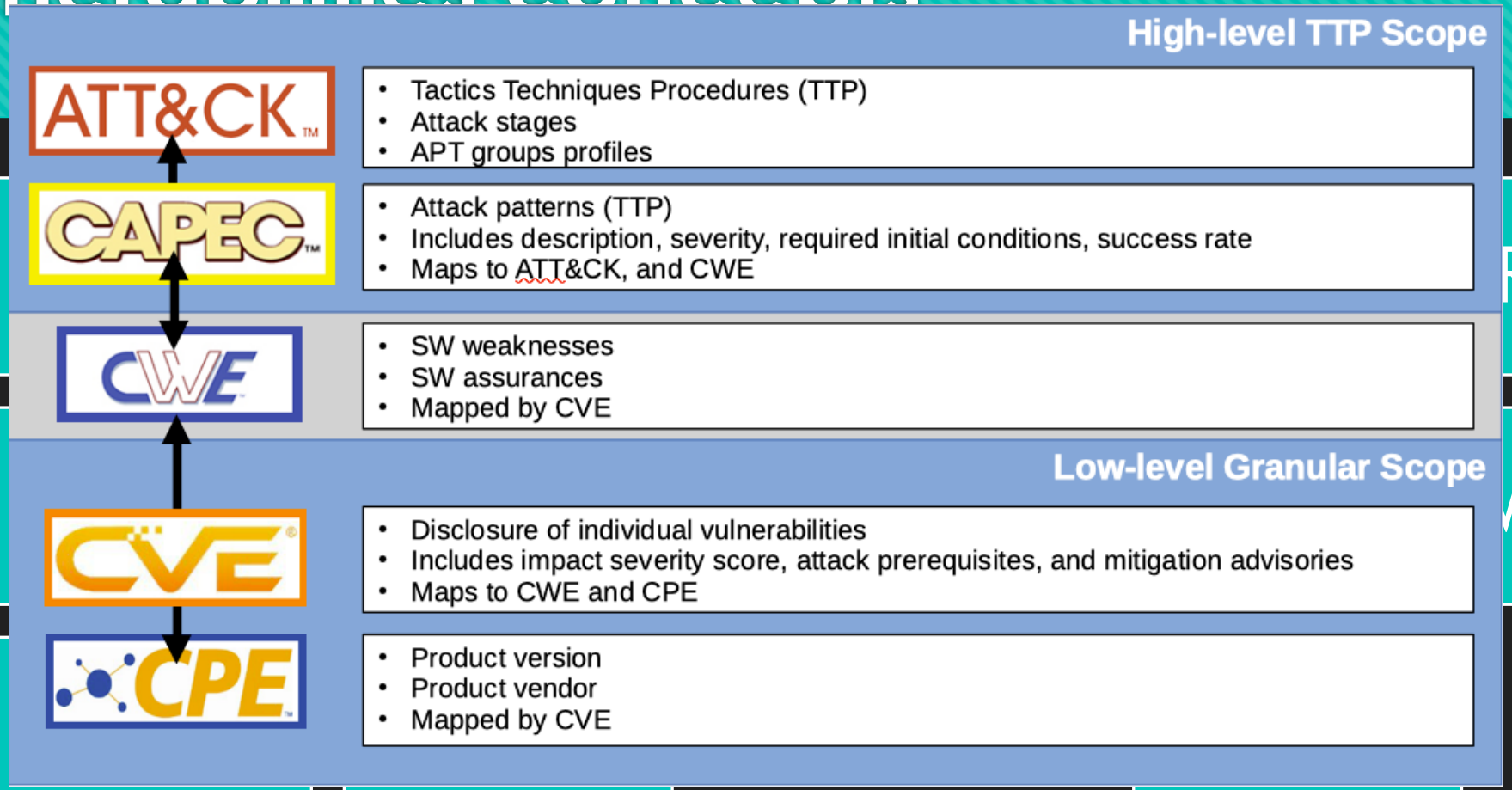
- This would be a lot of work!

Spoiler Alert: a fully OASIS compliant datastore with TypeDB.
STIX version 2.1

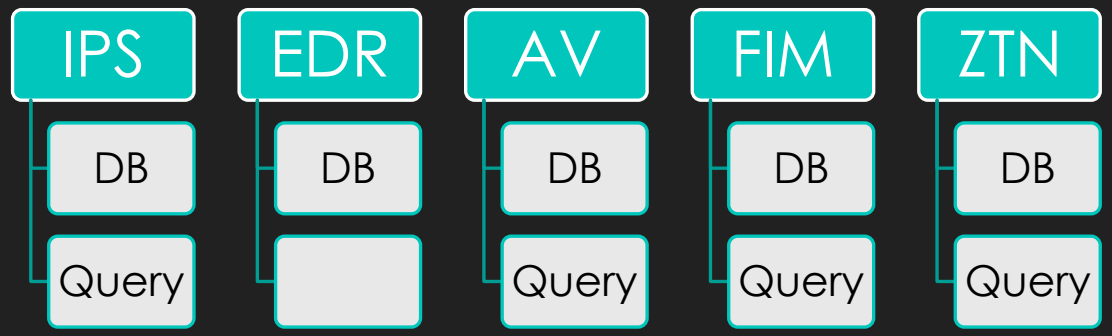
The Sheriff of data modelling

- Classical drama buy vs build vs reuse
- Buy is not an option
- Build is usually the option
- How can we avoid typical mistakes?
- Can provide a basic structure?
- With the ability to extend to each company?

Vulnerabilities concepts



God created silos in the last day



- Each product have their own syntax, taxonomies and ontologies
- Building a federated DB is a big challenge
- I mean just even look at the SIEM vendor space….

Omnibus Cyber

Base Schema

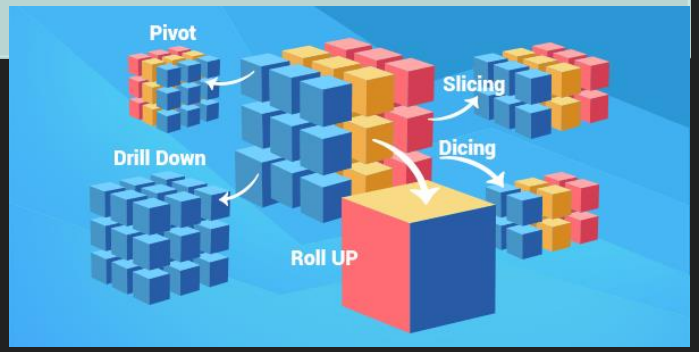
- Entity
- Relations
- Rules
- URI

ETL

- External reference
- Loaders

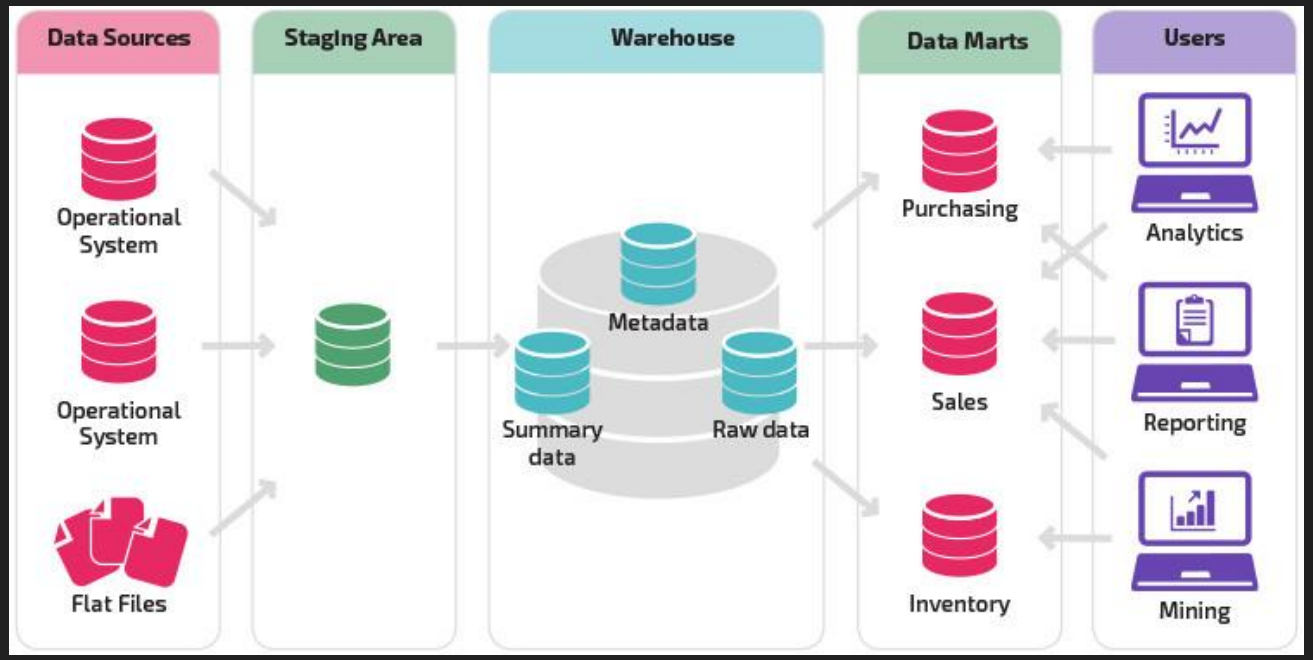
OLAP

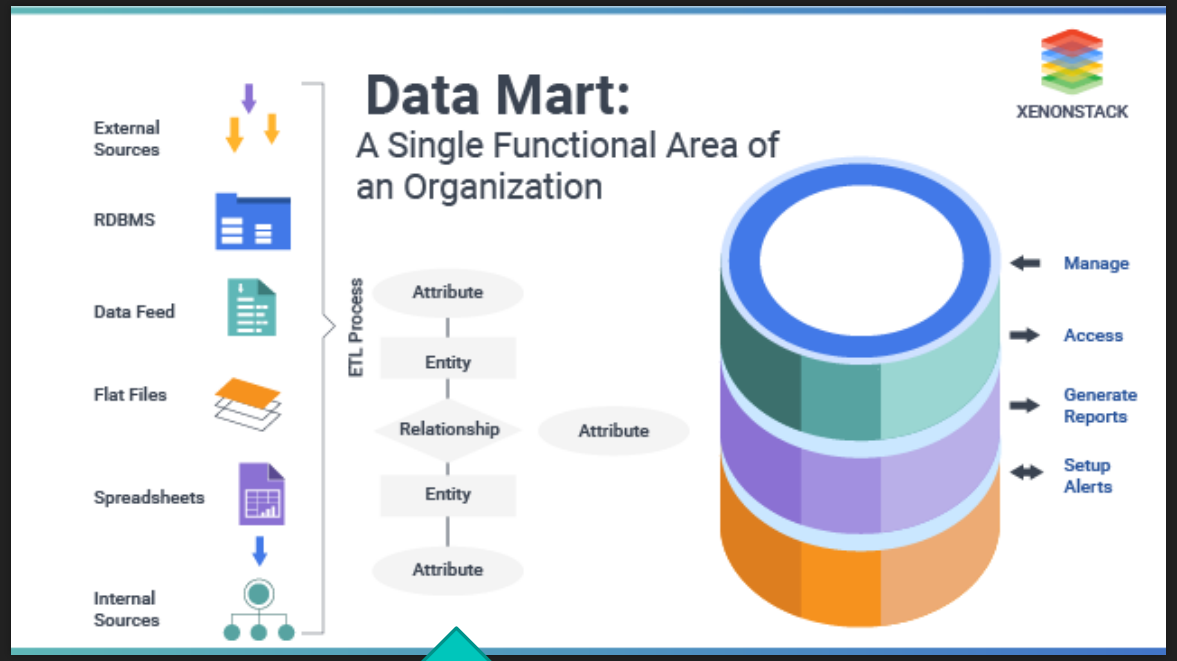
- Dimensions
- Fact
- Measure



Data Lake → ETL → TypeDB

Datamarts and ER





TypeDB