Protecting Identifiers in Cross-Domain Environments

Sam Chamberlain, Ph.D.

US Army Research Laboratory
In Support of The Joint Staff / J-8 / MASO
(410) 278-8948 // DSN 298
***chambesc@js.pentagon.mil*, or**
sam.chamberlain@us.army.mil

14th ICCRTS, Washington, DC 17 June 2009

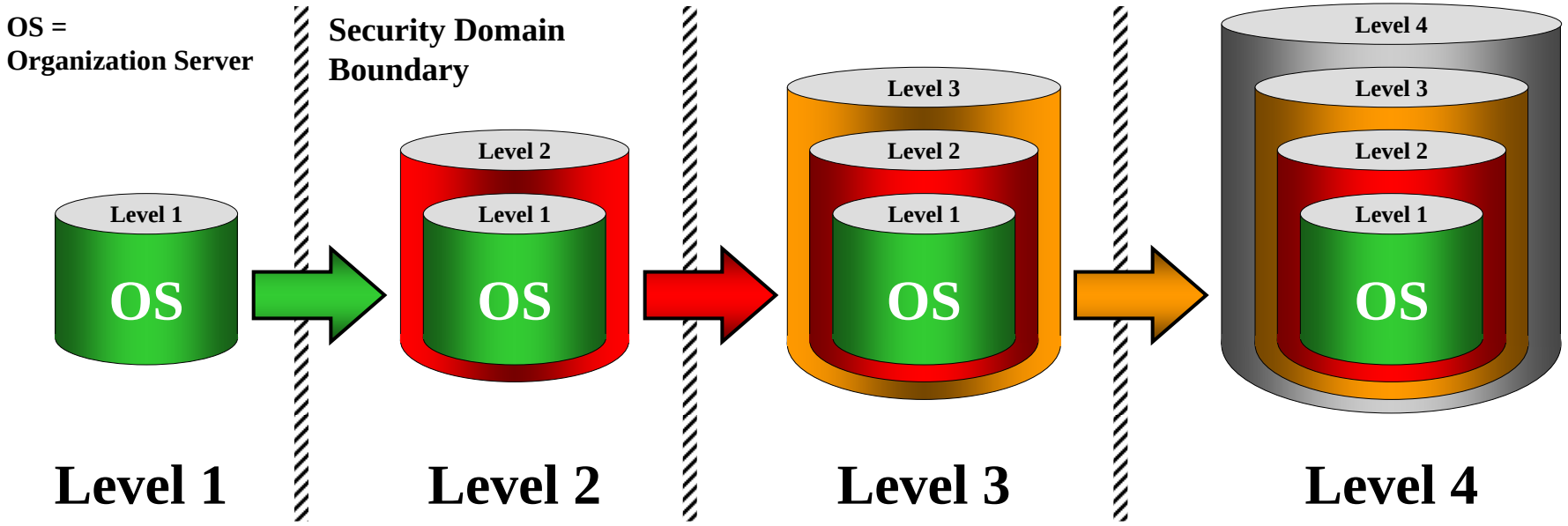
BLUF (Bottom Line Up Front)

Universally unique identifiers, even unintelligent ones, should be treated with the same classification level as the data they identify. This includes when they are a sole attribute.

History

- **Global Force Management Community of Interest (GFM COI) initiated the GFM Data Initiative (GFM DI) to unify force structure data and semantics across the DOD (Services, Joint, OSD, and Intel Communities).**
- **Unified front: the seven data sources, or Organization (Org) Servers will appear as one – common semantics and interface.**
- **All data is tagged with an enterprise-wide unique identifier.**
- **Each Org Server will replicate its data to the next higher security domain.**

Data Propagation Up Security Echelons



Level 1	Level 2	Level 3	Level 4
Create Level 1 Data Using Level 1 Identifiers	*Create Level 2 Data Using Level 2 Identifiers*	*Create Level 3 Data Using Level 3 Identifiers*	*Create Level 4 Data Using Level 4 Identifiers*

Data Entities and Attributes

- **Common semantics via the GFM XSD.**
- **A data entity (object) is composed of attributes; or, attributes are clustered into entities.**
- **An attribute is composed of a name and a value.**
- **An entity, not each attribute, is tagged with an enterprise-wide unique identifier that is also an attribute.**
- **An entity has a security classification, not each attribute.**
- **Data is created and tagged at the lowest security domain (e.g., Unclassified data created in the unclassified domain).**

Data is Structured as an Attribute Name and a Value

Obj

Attributes	
Name 0:	Value 0:
Name 1:	Value 1:
Name 2:	Value 2:
Name 3:	Value 3:
Name 4:	Value 4:
Name 5:	Value 5:

Template

Platform

Name	Value
ID:	0x0D74FF16
Type:	Aircraft
Model:	F-99
Length:	70 ft
Span:	45 ft
Speed:	1,000 MPH

Instance (Example)

An Attribute is a <u>name</u> with a <u>value</u> – NOT just a value.

Resolution of Classification

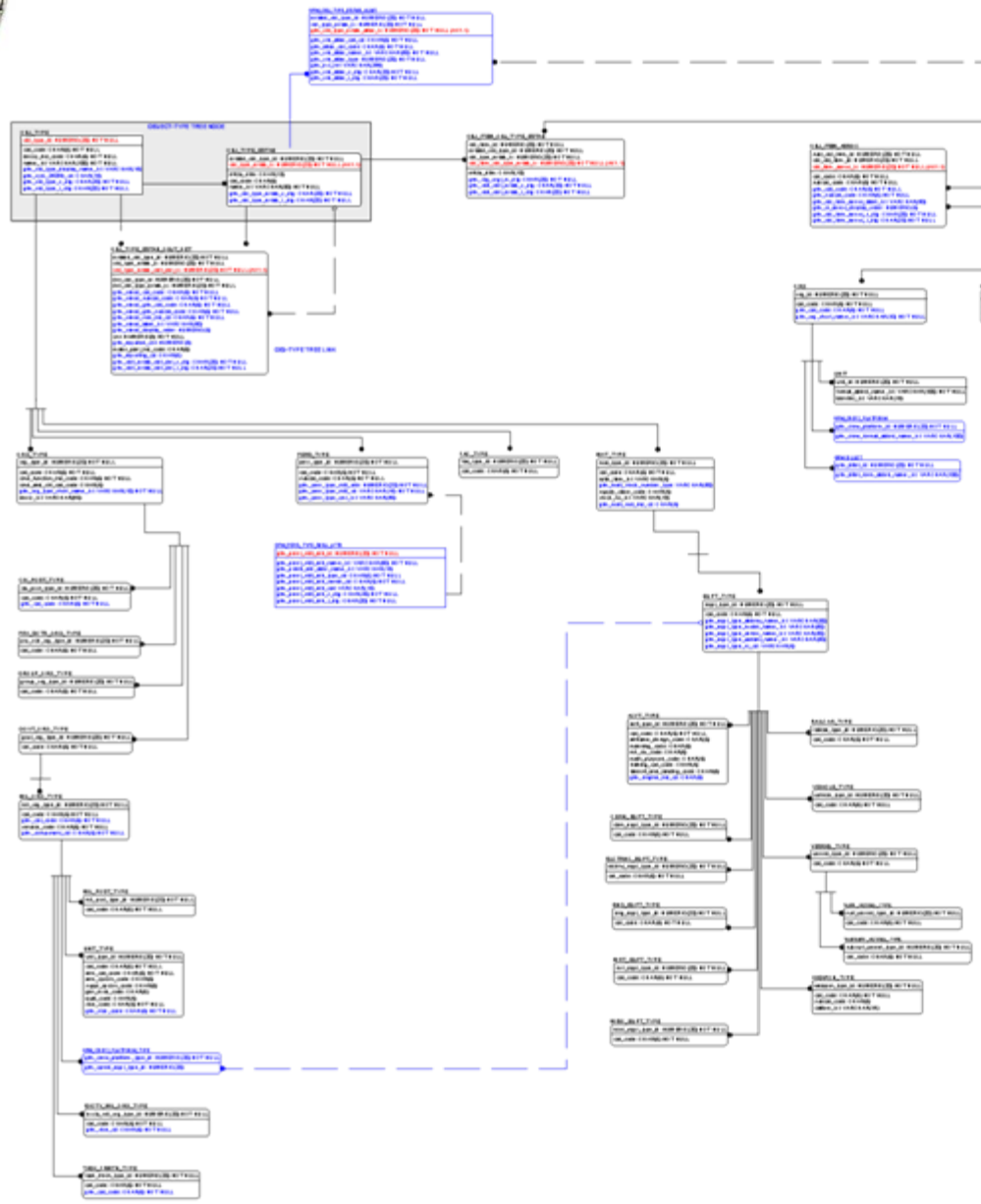
"Atomicity Boundary": the resolution to which the data can be identified as classified.

- **Class 1**: **Entity Resolution**: the existence of an entity is to be hidden; therefore, its classification can not be ascribed to any specific attribute, but only to the entity as a whole.
- **Class 2**: **Attribute Resolution**: one or more specific attributes of the entity can be identified as having values whose sensitivity is higher than the other attributes in the same entity, thus making the classification of the entity that of the most sensitive attribute.

Attributes Should be Portioned by Classification

- **Classification should be a part of the decision of how to cluster attributes into entities.**
- **That is, data should be arranged such that Class 2, or attribute resolution, does not happen. Attributes should be placed into entities with consistent classifications.**
- **One shouldn't mix classification levels within entities – but it is often done.**
- **Example: Data Masking.**

Partitioning Example – Normalized Data Model



Normal Forms **distribute or partition data to make management easier – classification level should be part of this process.**

Unique Identification Issues

Unique identifiers (with a wide scope) exacerbate this problem because a single, “world-wide” identifier immediately pin-points a single entity or unifies many attributes or entities that refer to it.

Unique Identifier Persistence in Partitions

Because of the Unique ID:
Entity X = Entities X_1 - X_5

ID: 0x0D74FF16
Type: Aircraft
Model: F-99
Length: 70 ft
Span: 45 ft
Speed: 1,000 MPH

X

X_1	ID: 0x0D74FF16 Type: Aircraft
X_2	ID: 0x0D74FF16 Model: F-99
X_3	ID: 0x0D74FF16 Length: 70 ft
X_4	ID: 0x0D74FF16 Span: 45 ft
X_5	ID: 0x0D74FF16 Speed: 1,000 MPG

Suppose X is type Class 1;
Then:
$CL(X_1 - X_5) = CL(X)$ and
CL(all attributes) = CL(X)
Therefore:
CL(ID=0x0D74FF16) = CL(X)

Unique Identifier Classification

Universally unique identifiers, even unintelligent ones, should be treated with the same classification level as the data they identify. This includes when they are a sole attribute.

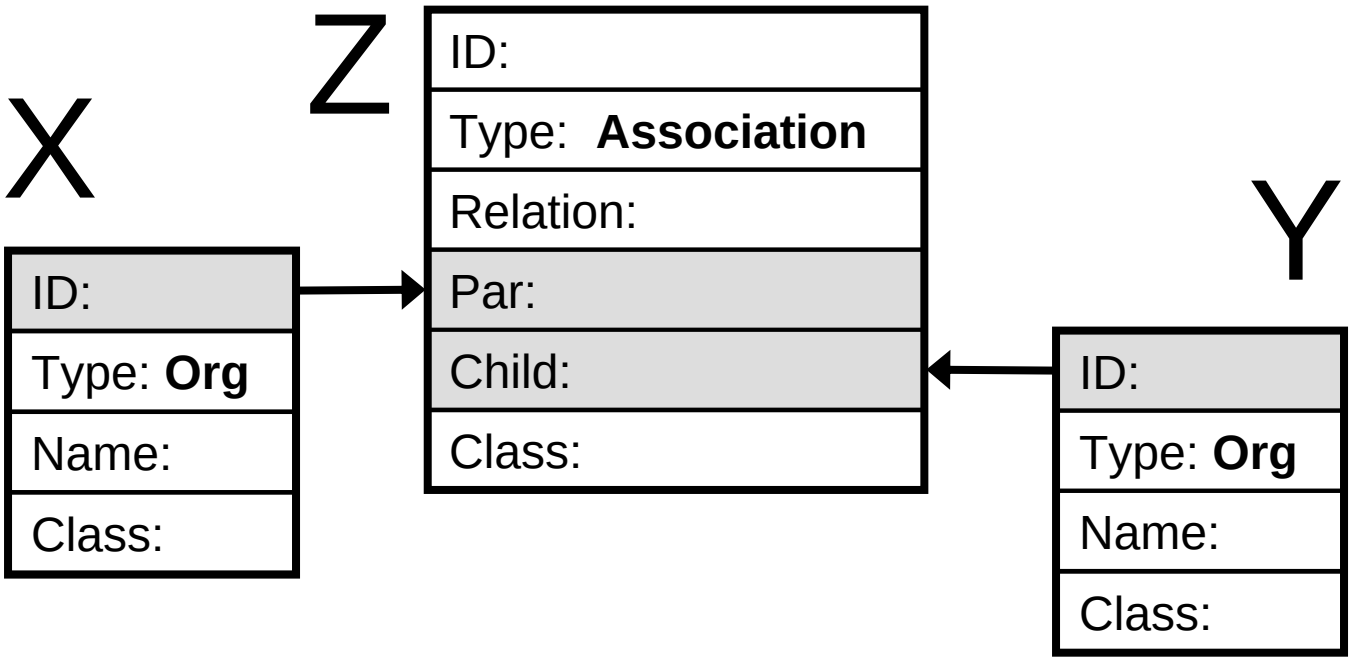
If Entity X has Classification Level CL= Z,
then attribute "ID = 0x0D74FF16" has CL=Z.

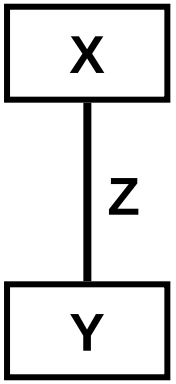
But, the value "0x0D74FF16" has no classification.

Intuitive reason: because attribute "ID=0x0D74FF16" ties together a set of data parts.

This policy simplifies many operations when applied to associating data (importing attributes).

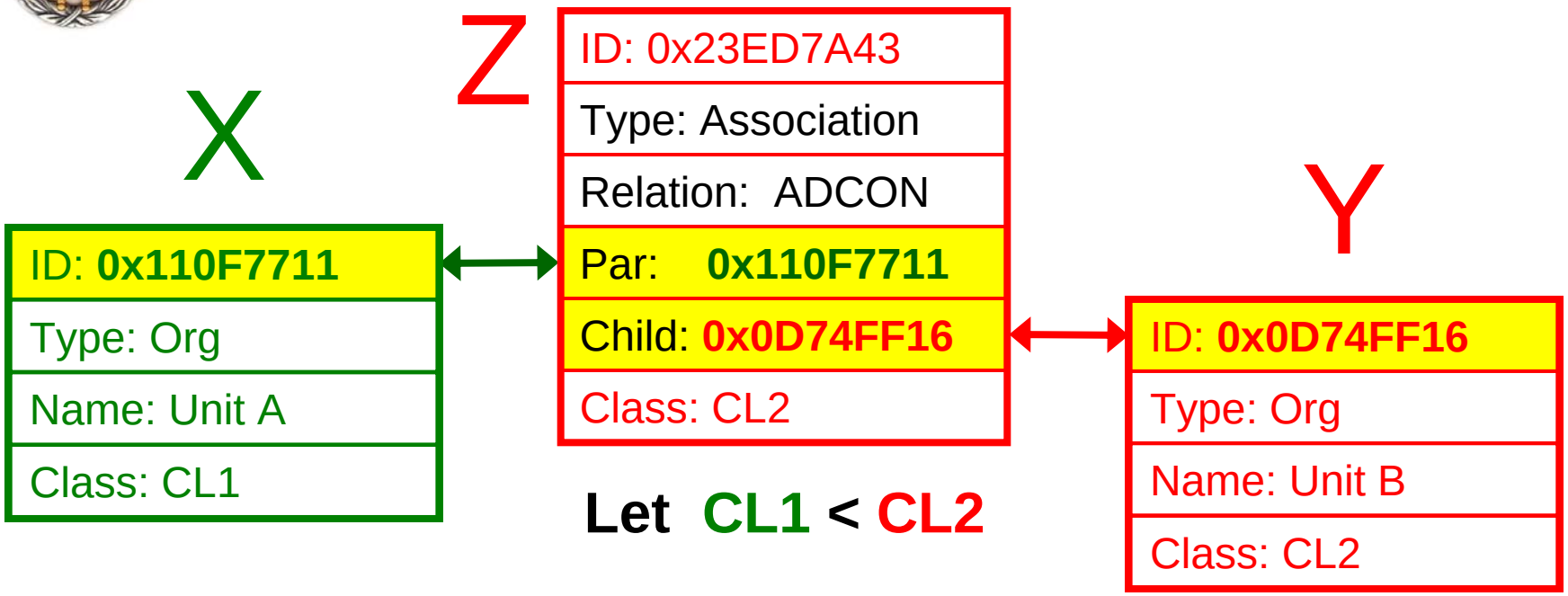
Associations Via Imported Attributes

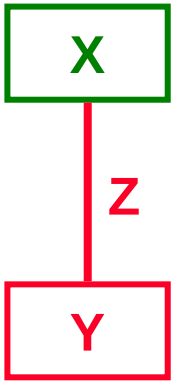




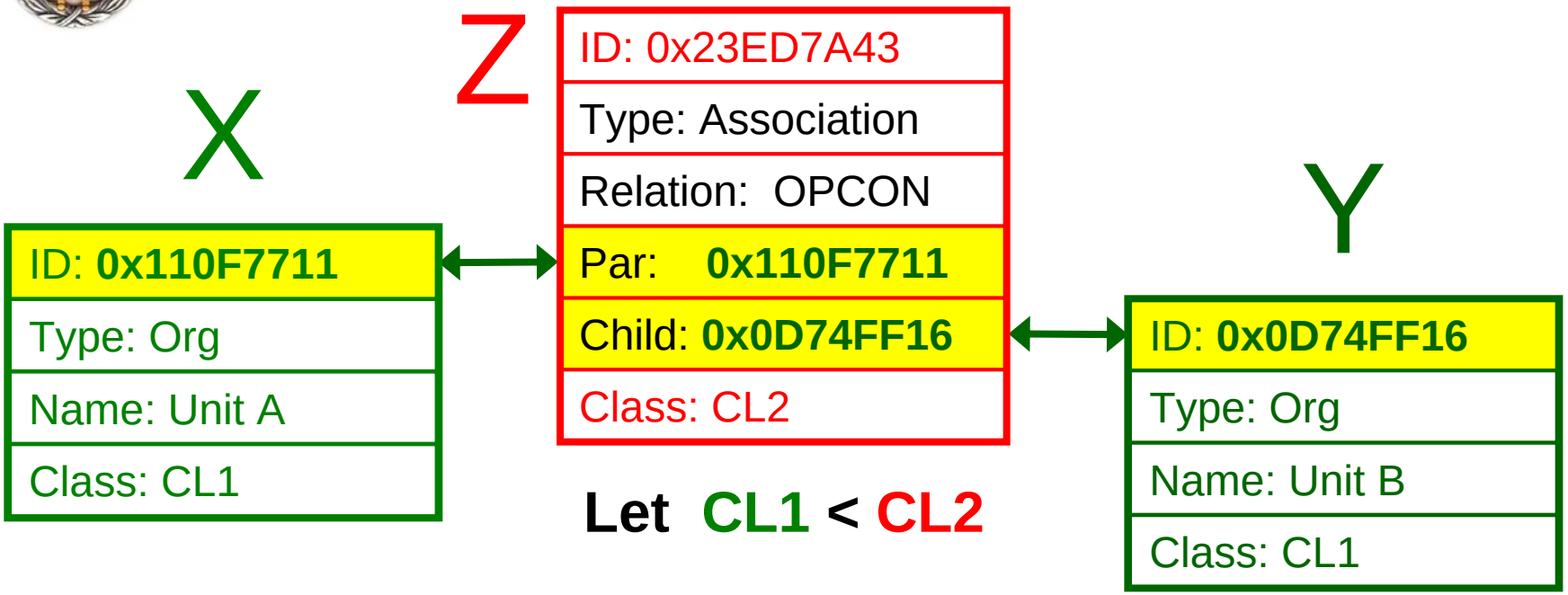
"Y is a child of X by association Z"

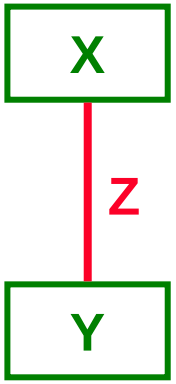
Classification Level Propagation Via Imported Attributes



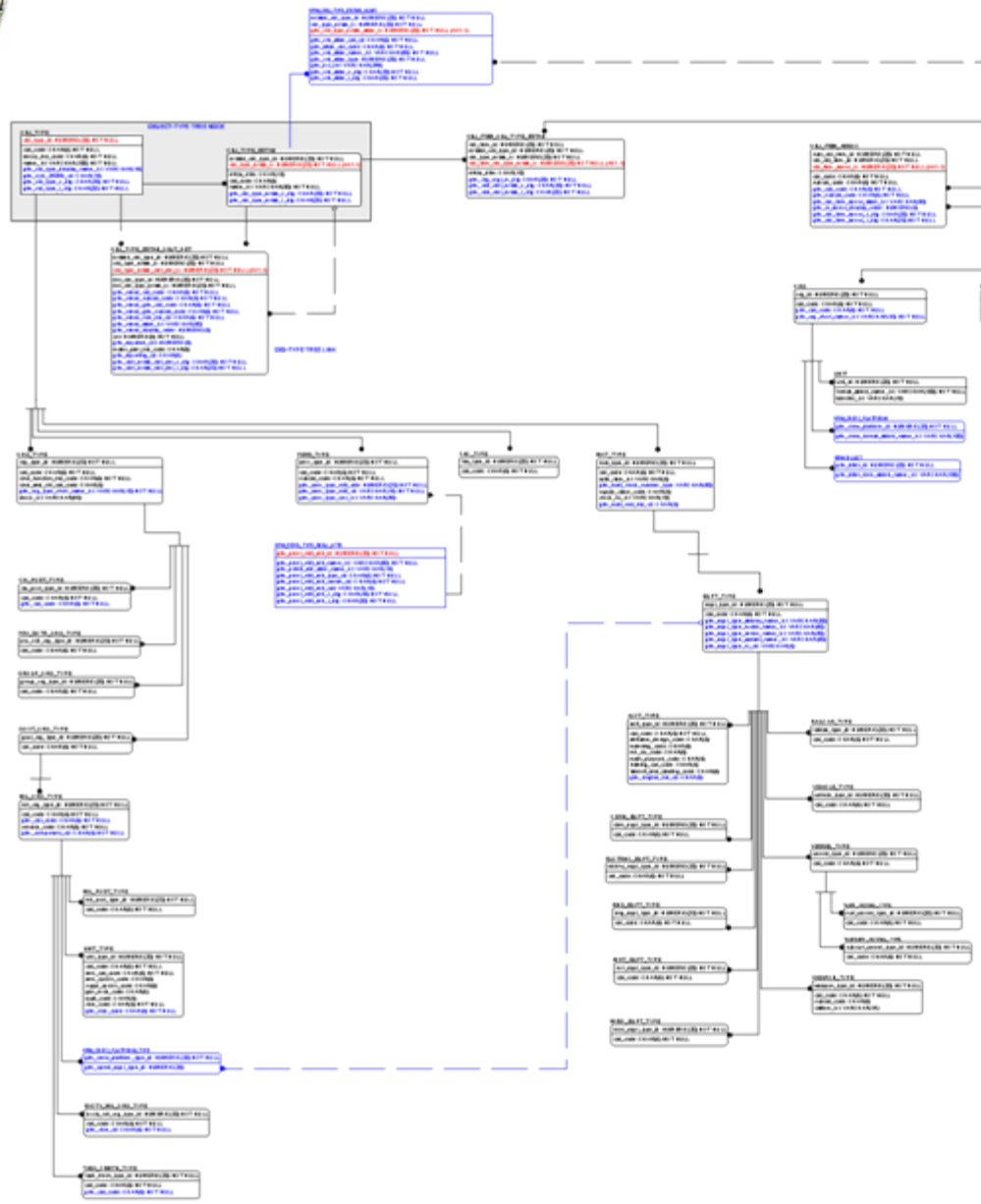


Classification Level Independent of Imported Attributes



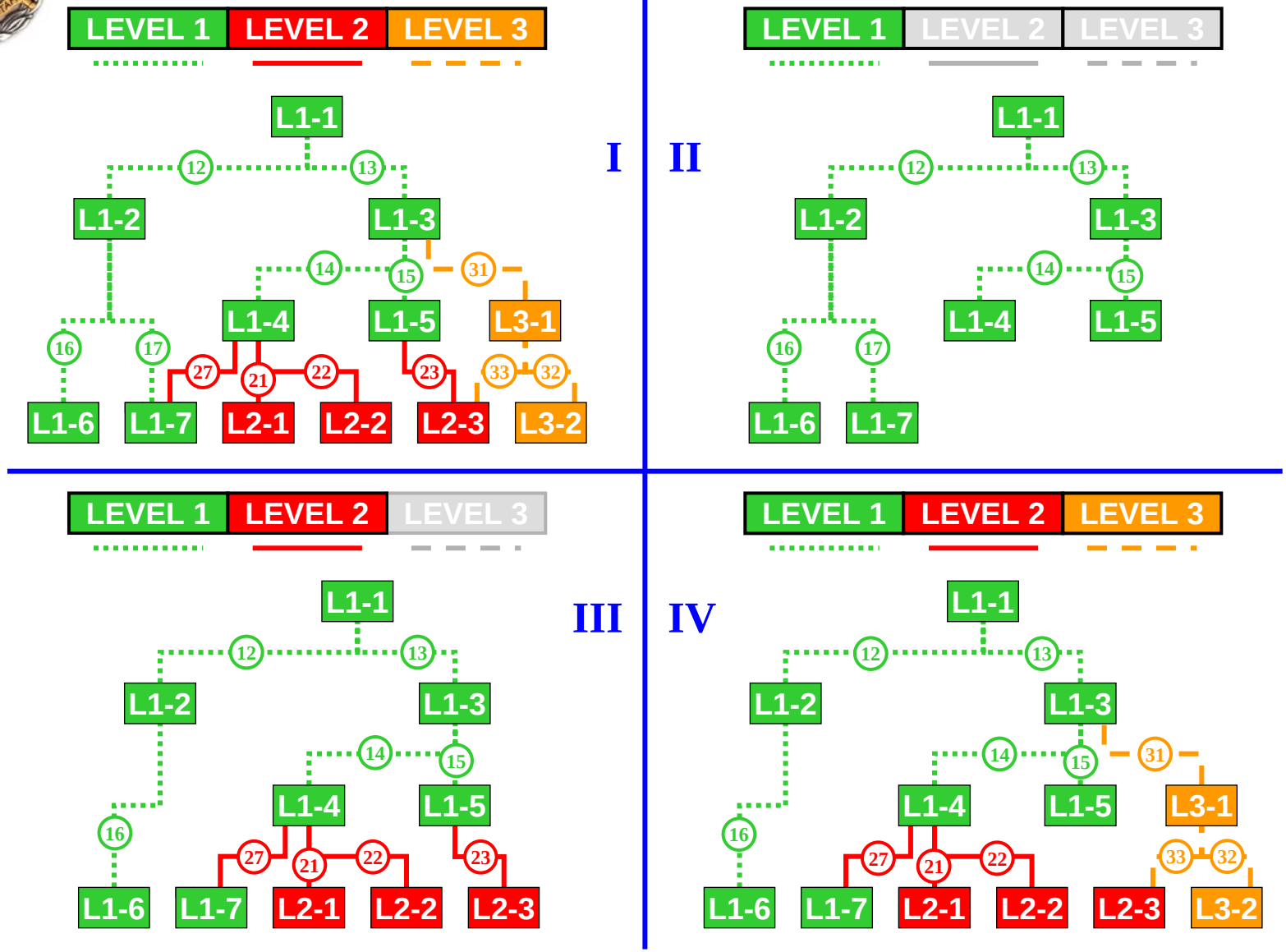


Classification Policy Applied to a Normalized Data Model

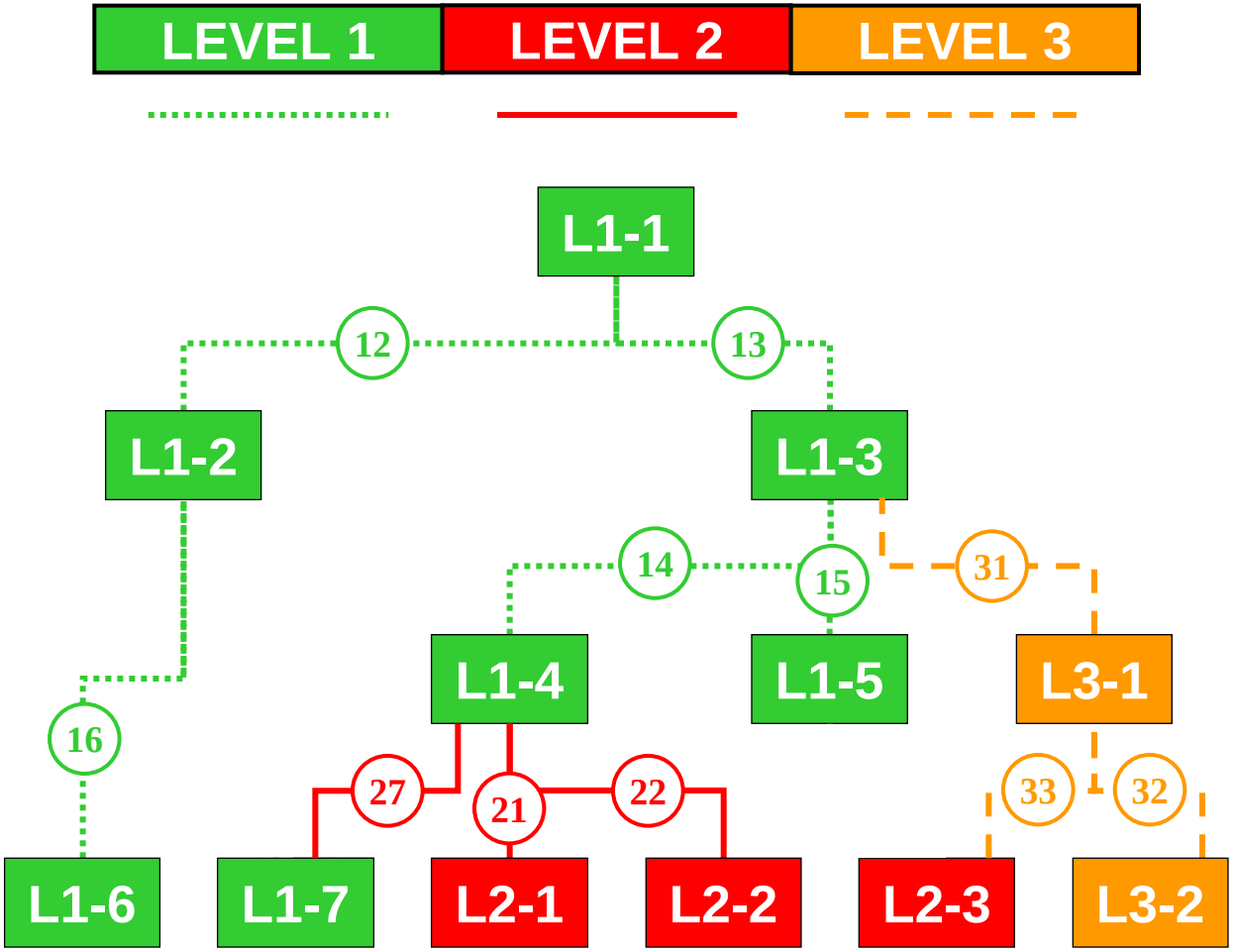


All Imported Attributes Follows These Rules.

Examples of Multi-Level Data



Examples of Multi-Level Data



Summary

Unique identifiers (with a wide scope) introduce new issues because a single, "world-wide" identifier immediately pin-points a single entity or unifies many attributes or entities that refer to it.

Universally unique identifiers, even unintelligent ones, should be treated with the same classification level as the data they identify. This includes when they are alone.