

# CERIF COURSE

## Session3: DataModel 1

Keith G Jeffery,

Director, IT CLRC

[k.g.jeffery@rl.ac.uk](mailto:k.g.jeffery@rl.ac.uk)

Anne Asserson,

University of Bergen

[anne.asserson@ub.uib.no](mailto:anne.asserson@ub.uib.no)

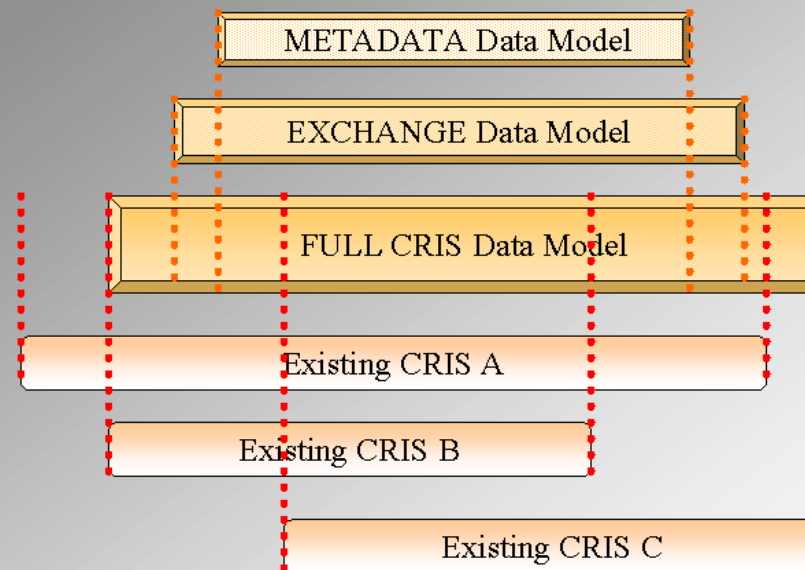
- Full, exchange and metadata models
- Full model – overview (nutshell)
- The concept of binary relations, linking relations and recursion
- The concept of character / language variants
- The concept of enumerated lists – dictionaries, thesauri, ontologies

- Full, exchange and metadata models
- Full model – overview (nutshell)
- The concept of binary relations, linking relations and recursion
- The concept of character / language variants
- The concept of enumerated lists – dictionaries, thesauri, ontologies

# Full, exchange and metadata models

Metadata Model is a subset of Exchange Model is a subset of Full Model

Full Model is intersection of existing CRISs excluding uncommon variants



# Structure of Session

- Full, exchange and metadata models
- Full model – overview (nutshell)
- The concept of binary relations, linking relations and recursion
- The concept of character / language variants
- The concept of enumerated lists – dictionaries, thesauri, ontologies

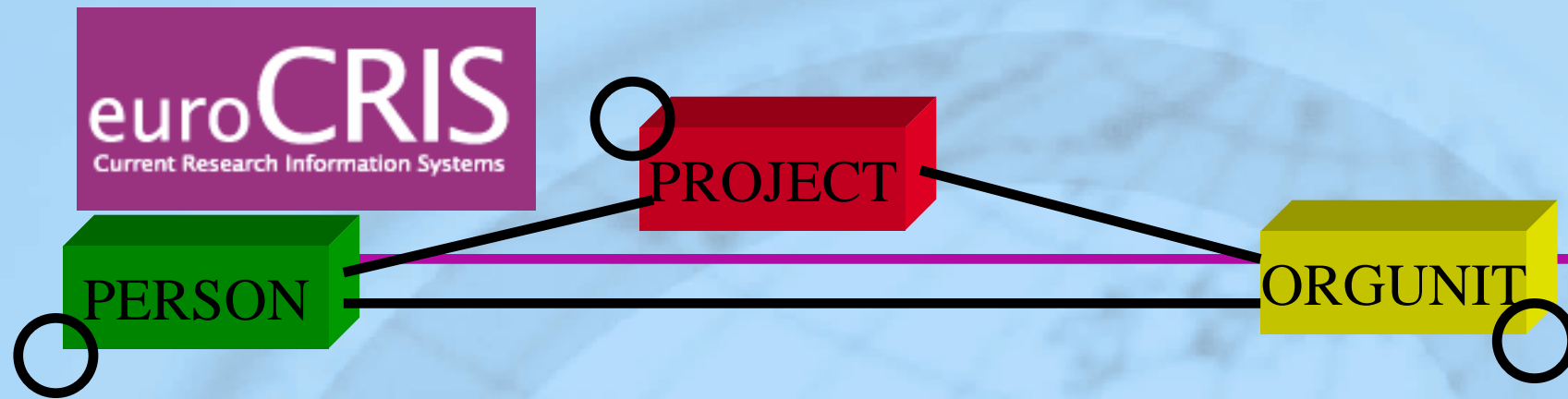
# CERIF2000 Data model

- Extended relational model
- Linking relations with attributes (roles and time stamp)
- 3 base entities Person, Organisation, Project
- 12 secondary base entities (linked to base entities)
- 36 Look up tables (to ensure data quality)
- 39 Link tables (flexibility)
- all text fields have multiple language fields
- maximum representativity with minimum complexity

# CERIF2000 in a Nutshell

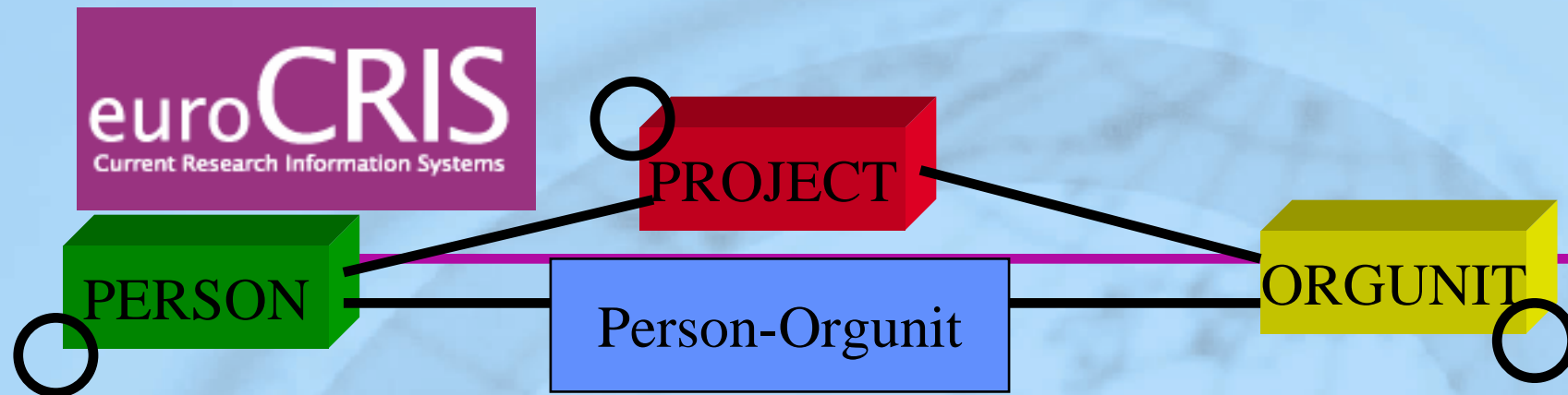


## Three Primary Entities



### Concepts:

- (1) entities that reflect main 'views of entry' into CRISs
- (2) entities with no direct functional dependency on each other
- (3) entities that can refer to themselves (recursion)
- (4) entities linked in pairs by 'linking relations'
- (5) 'linking relations' represent temporally-bound roles
- (6) 'linking relations' have primary key of each entity, role, date/time start, date/time end and any other constraints



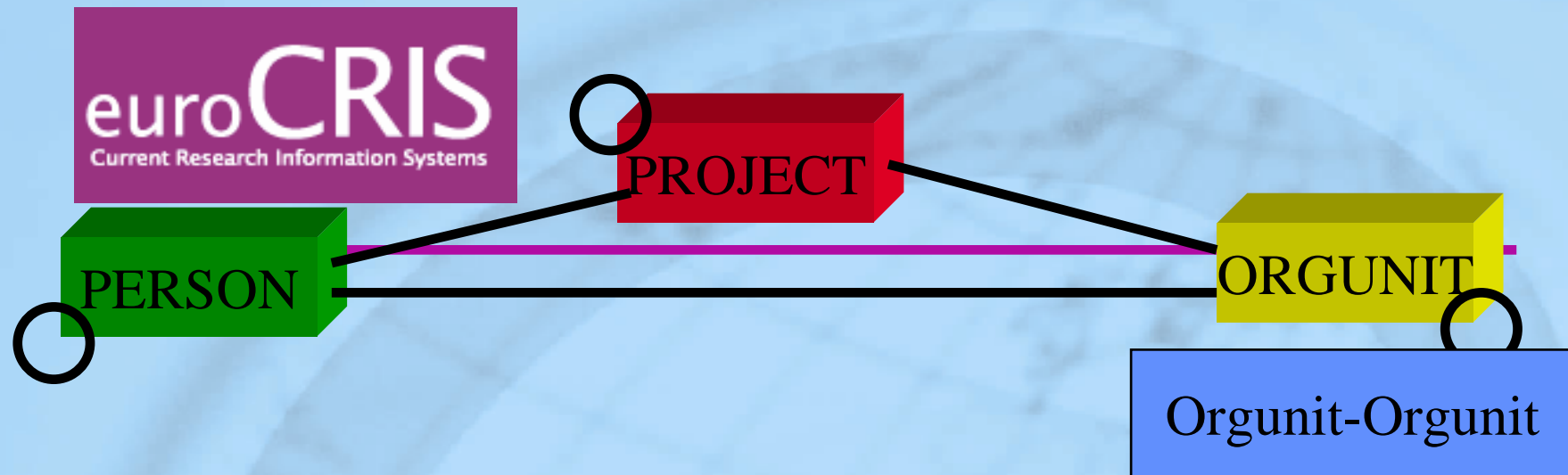
As an Example: PERSON-ORGUNIT

Concepts:

- (1) May have many instances of the relationship for each instance of PERSON and ORGUNIT due to role and temporal bounding
- (2) Role: the purpose of the relationship e.g. employee | head | ....
- (3) Temporal: the use of <Start Date/Time> and <End Date/Time> defines the duration of this relationship

Analagous for PROJECT\_ORGUNIT and PERSON\_PROJECT

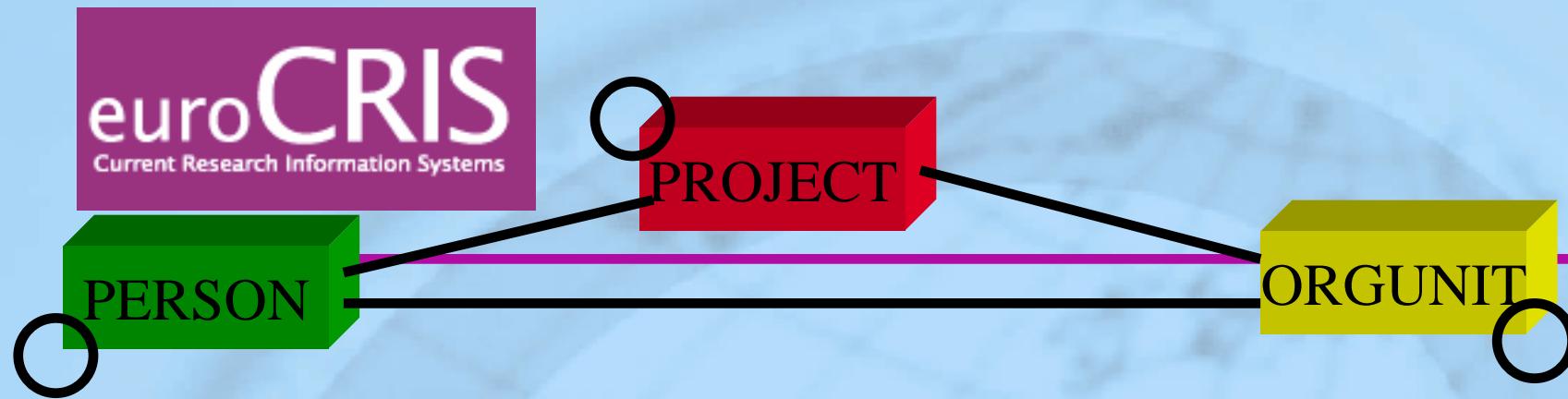
## Primary Base Entity: **ORGUNIT**



### Concepts:

- (1) ORGUNIT may have an organisationally subordinate relationship to another ORGUNIT e.g. a Group within a Department
- (2) ORGUNIT may have a symbiotic relationship to another ORGUNIT e.g. two Groups that have a cooperation agreement
- (3) ORGUNIT may have a financial relationship to another ORGUNIT e.g. customer - contractor

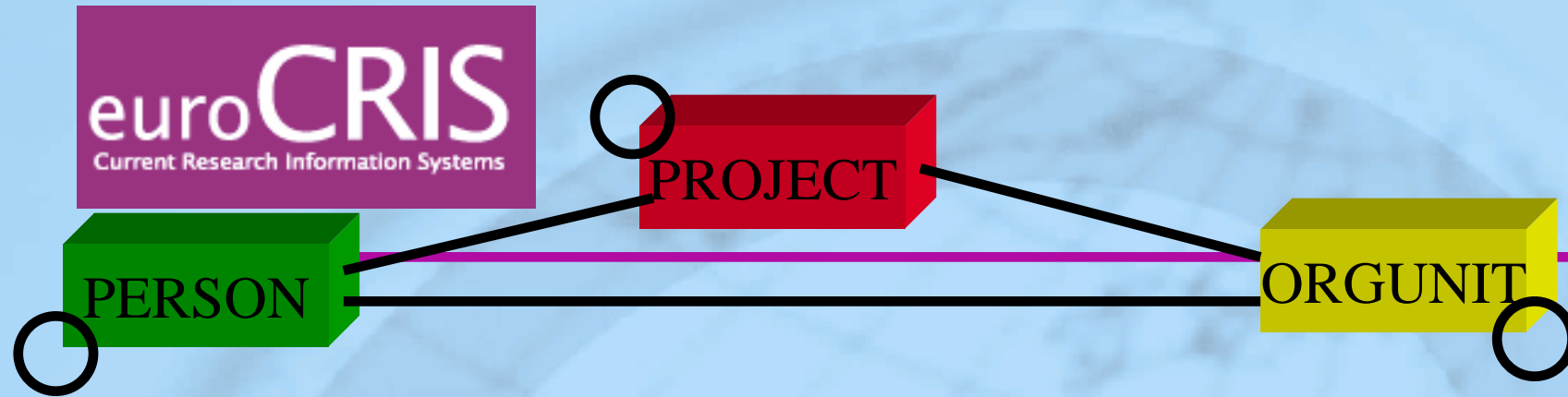
## Primary Base Entity: **PROJECT**



### Concepts:

- (1) PROJECT may have an organisationally subordinate relationship to another PROJECT e.g. a sub-Project
- (2) PROJECT may have a symbiotic relationship to another PROJECT e.g. two Projects that cooperate by agreement
- (3) PROJECT may have a temporal relationship to another PROJECT e.g. one project follows on from another

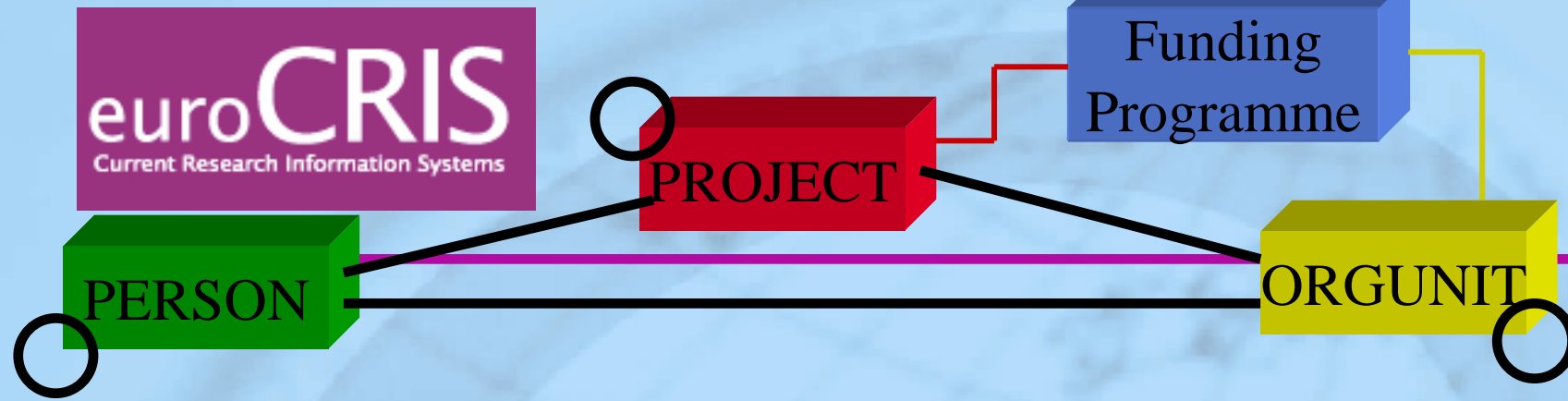
## Primary Base Entity: **PERSON**



### Concepts:

- (1) PERSON may have a socially subordinate relationship to another PERSON e.g. a child of a parent
- (2) PERSON may have a symbiotic relationship to another PERSON e.g. two researchers that cooperate by agreement
- (3) PERSON may have a temporal relationship to PERSON e.g. a lecturer (dates) becomes a reader (dates)

## Secondary Base Entities: **FUNDING PROGRAMME**

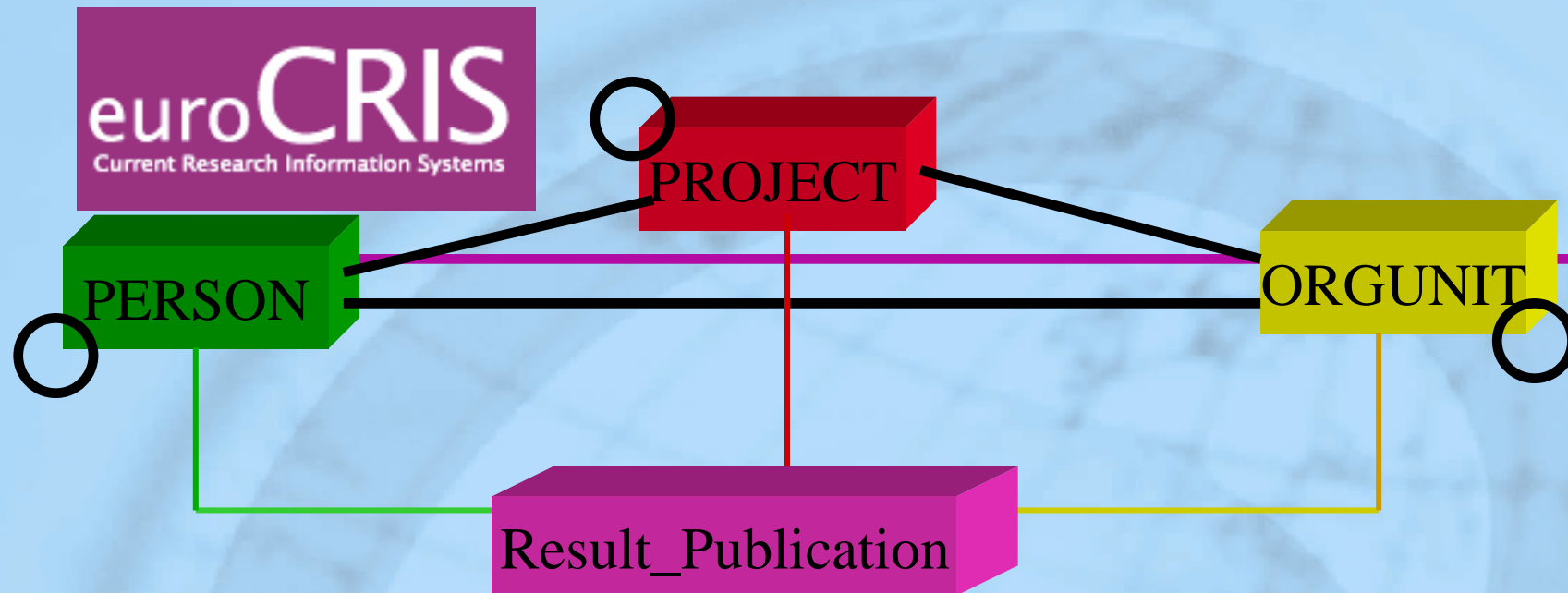


### Concepts:

- (1) Funding Programme is related to
  - (a) ORGUNIT and / or (b) PROJECT
- (2) A Person is only funded via
  - (a) ORGUNIT and / or (b) PROJECT
- (3) any other entities are only funded via
  - (a) ORGUNIT and / or (b) PROJECT



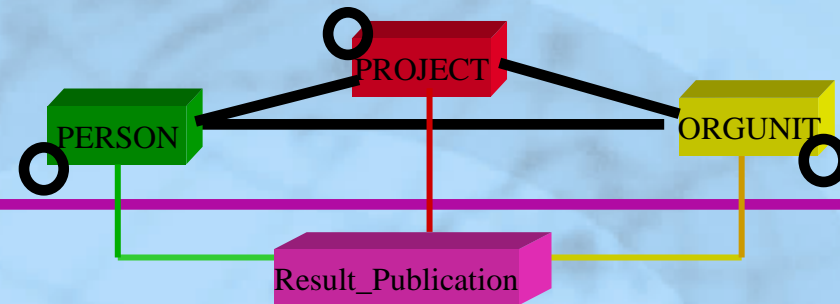
## Secondary Base Entities: example: **RESULT\_PUBLICATION**



### Concepts:

- (1) temporally-bound role linking relations
- (2) >1 linking relation : Result\_Publication and other entities
- (3) PERSON role may be author, co-author, editor, reviewer....
- (4) ORGUNIT role may be publisher, IPR or copyright owner..
- (5) PROJECT role may be the source of the idea

## Secondary Base Entities: example: RESULT\_PUBLICATION



Can Express: (where DT=date/time)

Person A (DT1 - DT2) (is author of) Publication X

Orgunit O (DT1 - DT2) (is owner of IPR in) Publication X

Person A (DT1 - DT2) (is employee of ) Orgunit O

Person A (DT1 - DT2) (is project leader of) Project P

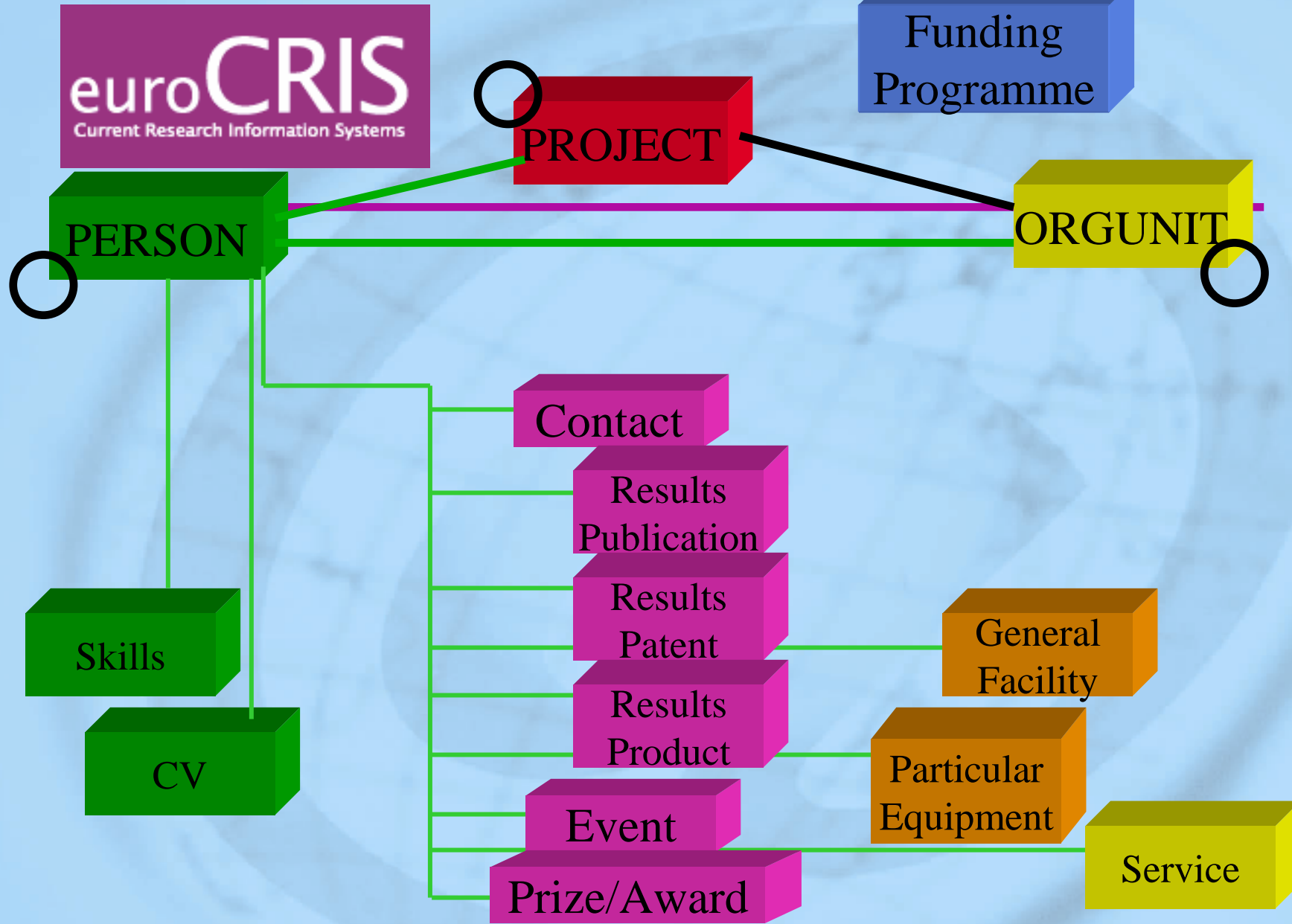
Person A (DT1-DT2) (is member of) Orgunit M

Person A (DT1-DT2) (is member of) Orgunit N

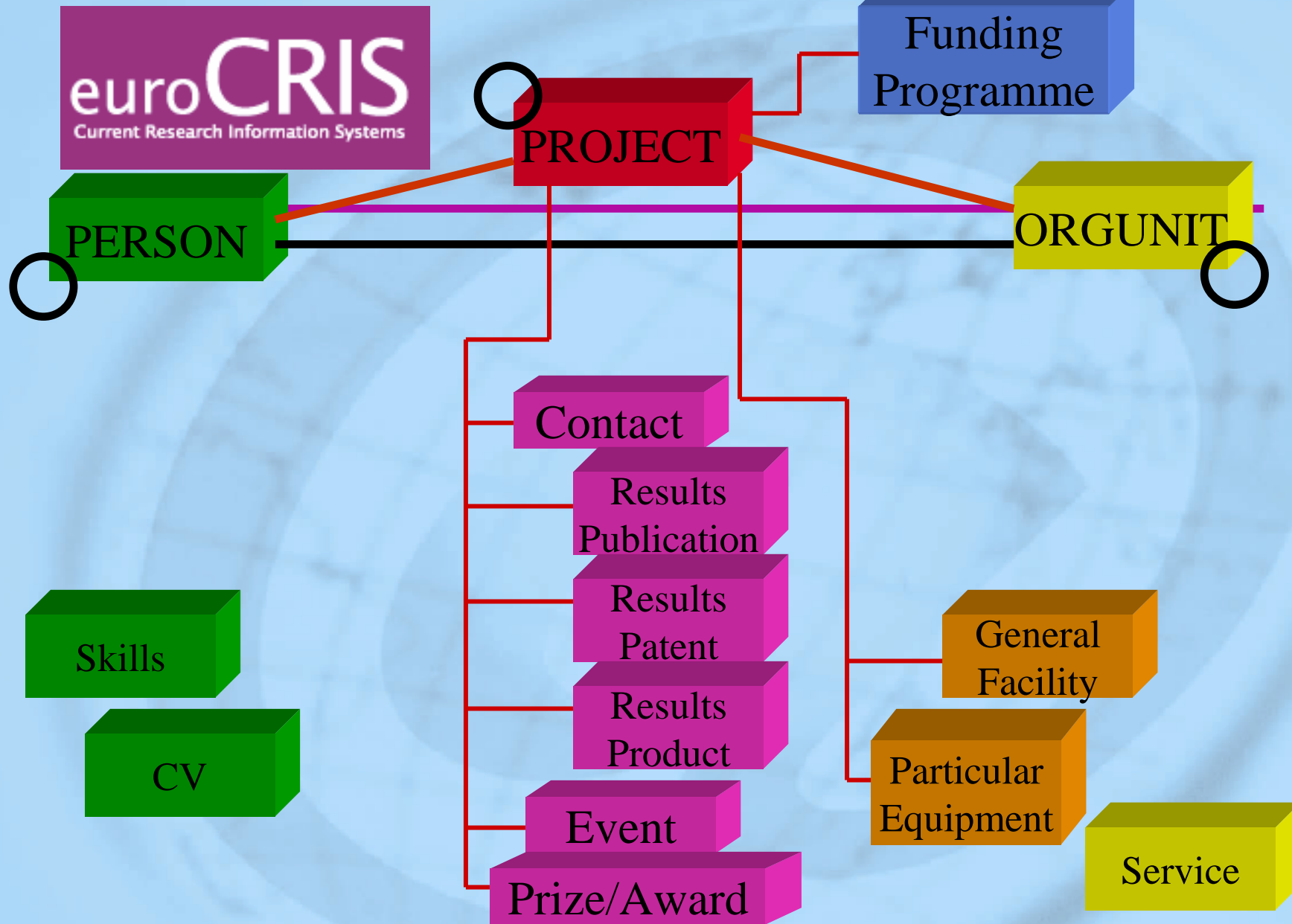
Orgunit M (DT1-DT2) (is part of) Orgunit O

Orgunit N (DT1-DT2) (is part of) Orgunit O

# PERSON Links

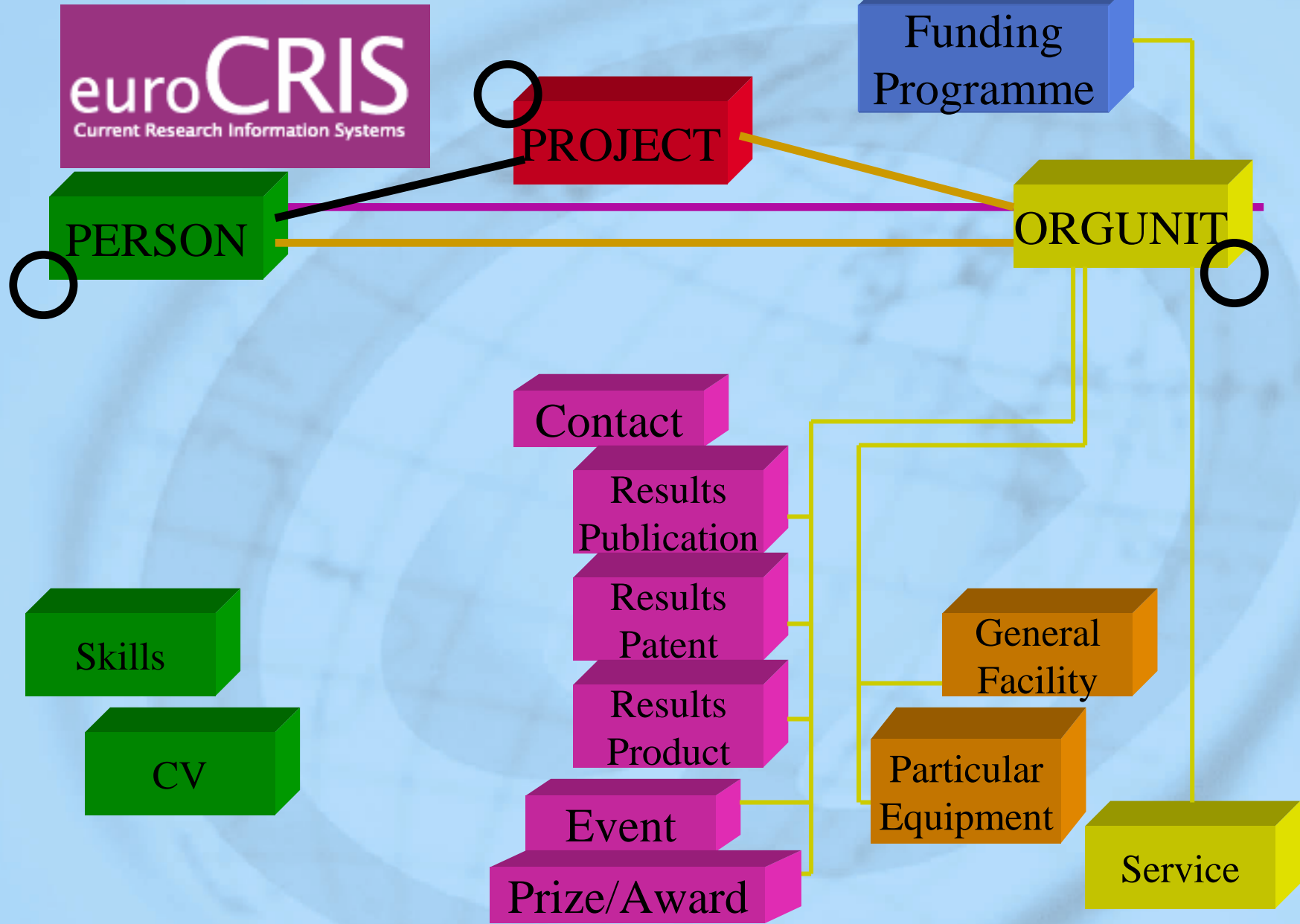


# PROJECT Links

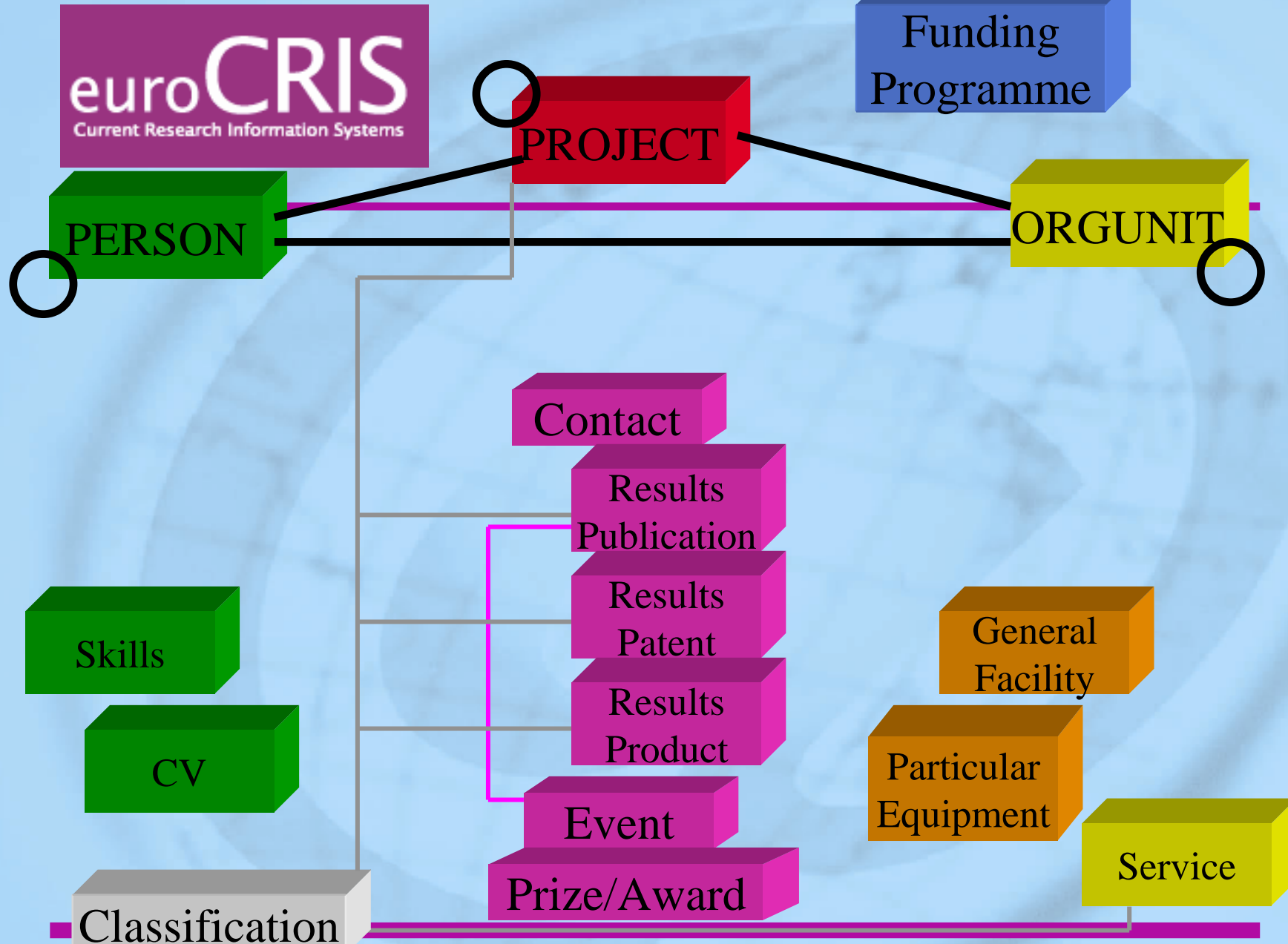


euroCRIS  
Current Research Information Systems

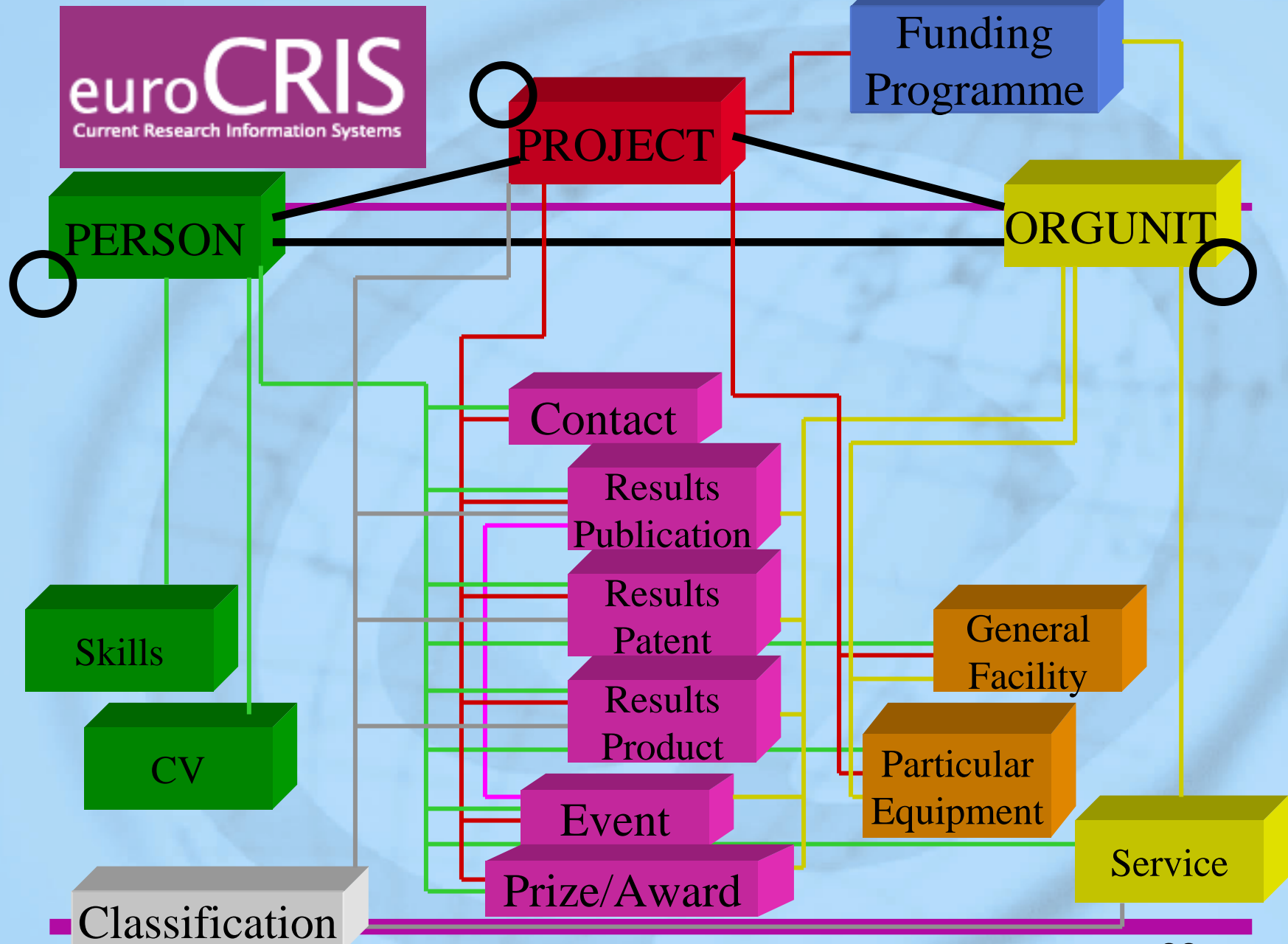
# ORGUNIT Links



# Classification



# The Whole Thing



# End of CERIF2000 in a Nutshell

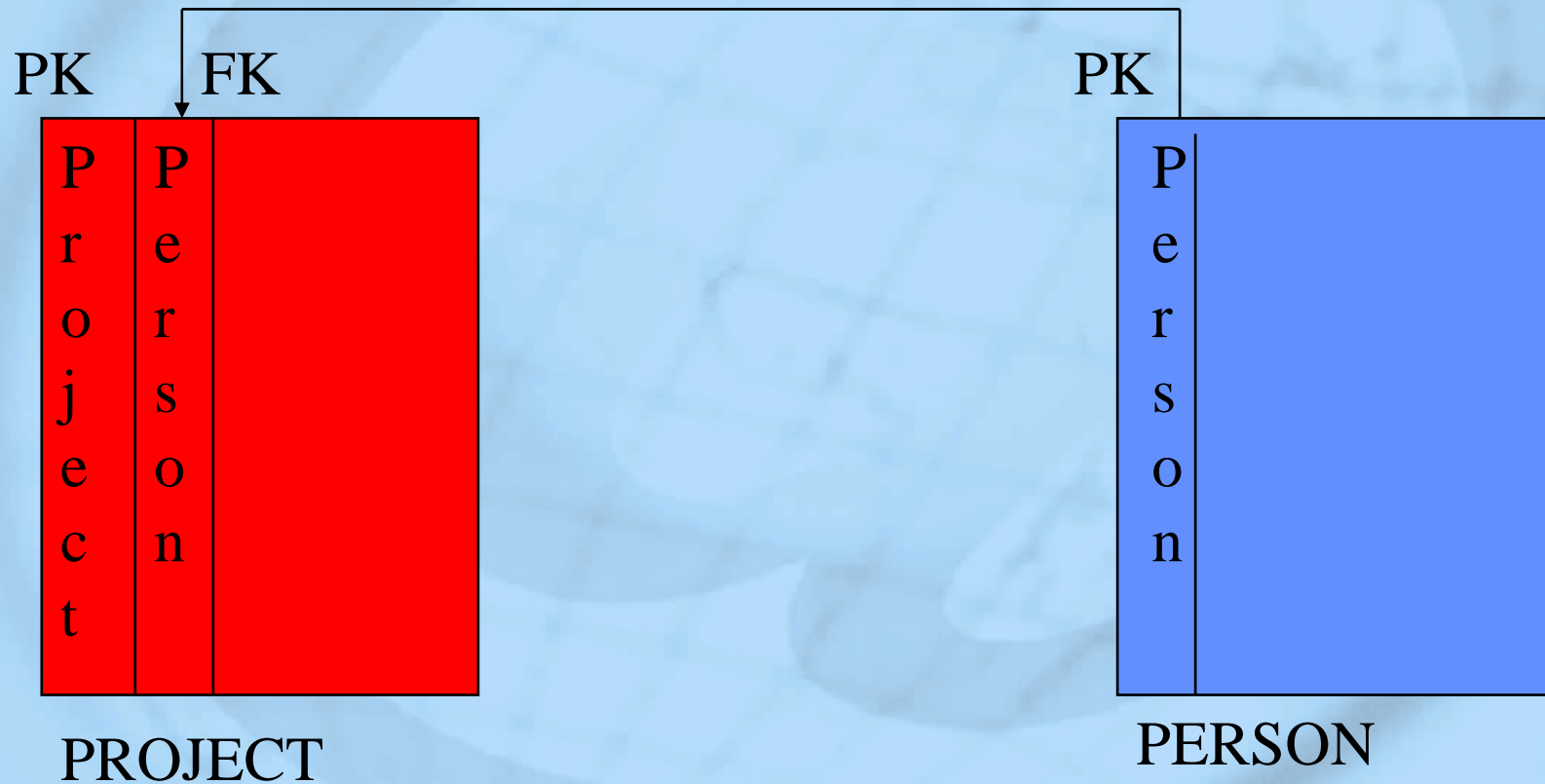
## Structure of Session

---

- Full, exchange and metadata models
- Full model – overview (nutshell)
- The concept of binary relations, linking relations and recursion
- The concept of character / language variants
- The concept of enumerated lists – dictionaries, thesauri, ontologies

- Wish to link flexibly
  - An instance in an entity to a related instance in another entity (relationship)
  - An instance in an entity to another instance in the same entity (recursion)
- Examples
  - Person  $\leftrightarrow$  Project e.g. x is leader of z
  - Person  $\leftrightarrow$  Person e.g. x is boss of y

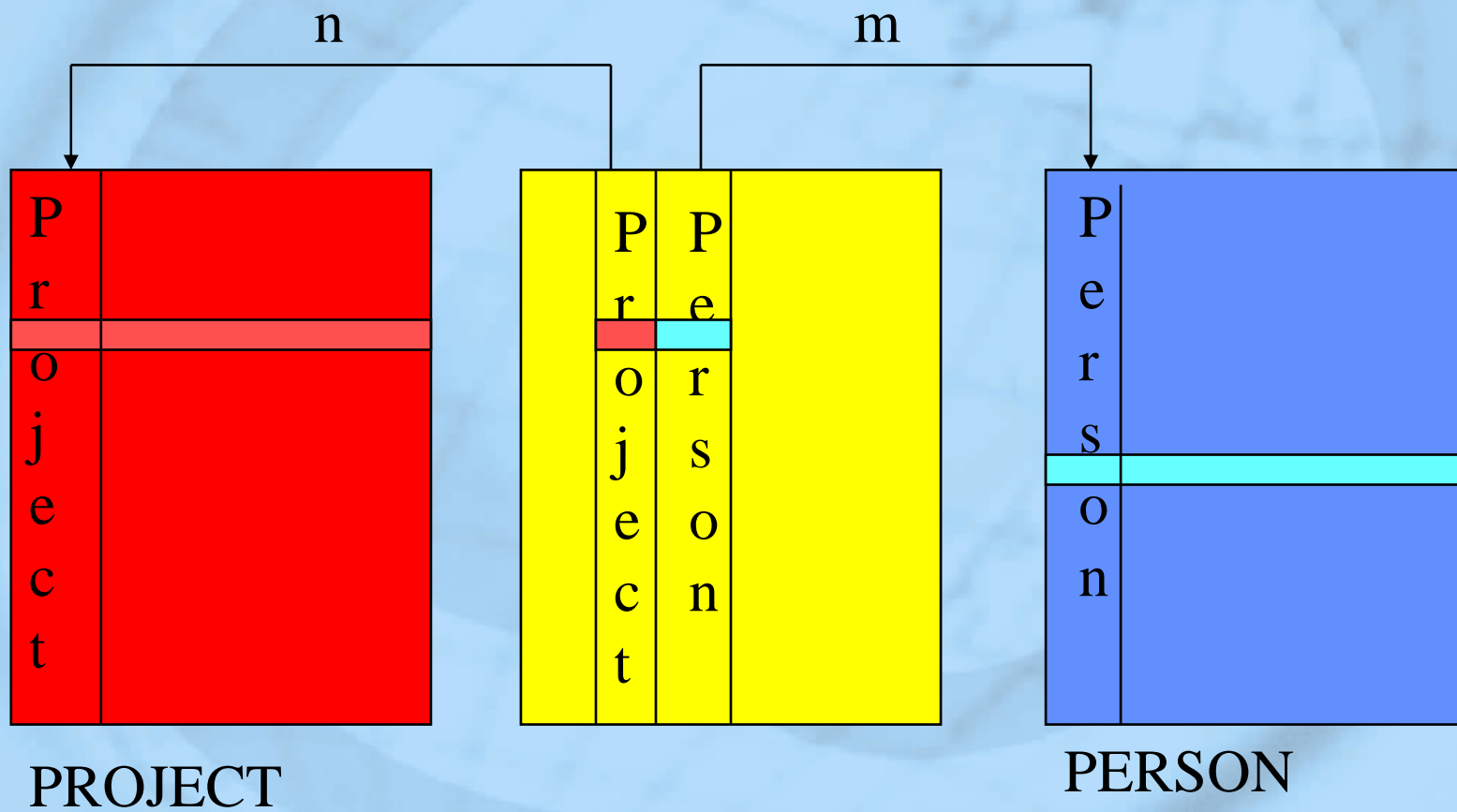
# Binary Relations Relationship Usual Relation



## Binary Relations Relationship Problem

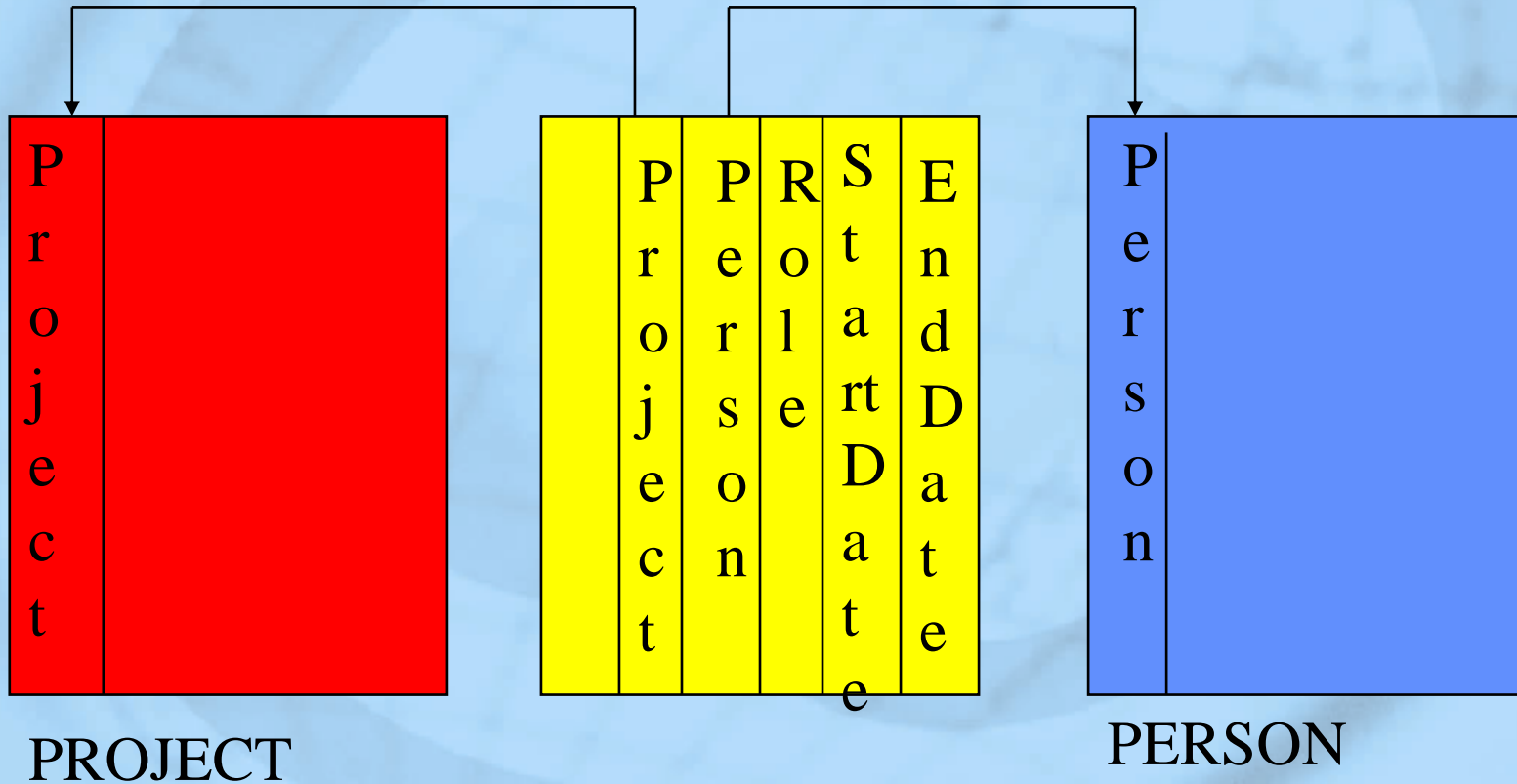
- Supports only 1 (**Project**) to n (**Persons**)
- i.e. the persons on any 1 project, with all their attributes (dependencies)
- In many cases need to indicate that
  - The same person works on several projects
  - In different roles (e.g. leader, programmer)
  - At different (or the same) time periods
- i.e. 1 (**Person**) to n (**Projects**)

# Binary Relations Relationship Binary Relation



# Binary Relations Relationship ~~Binary Relation~~

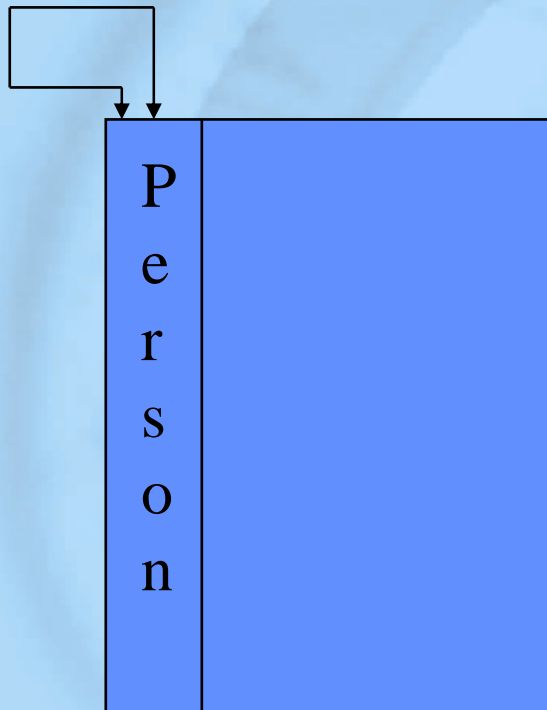
In practice usually have more attributes than Project / Person



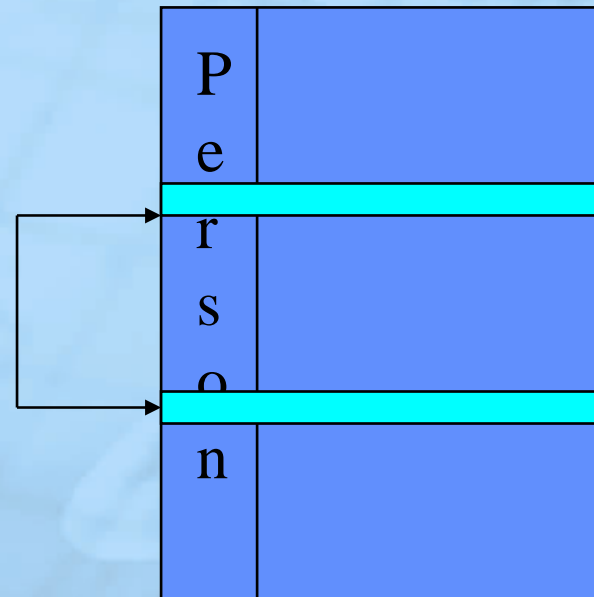
# Binary Relations Recursion ~~Usual Relation~~

PK & FK

Actually works like this

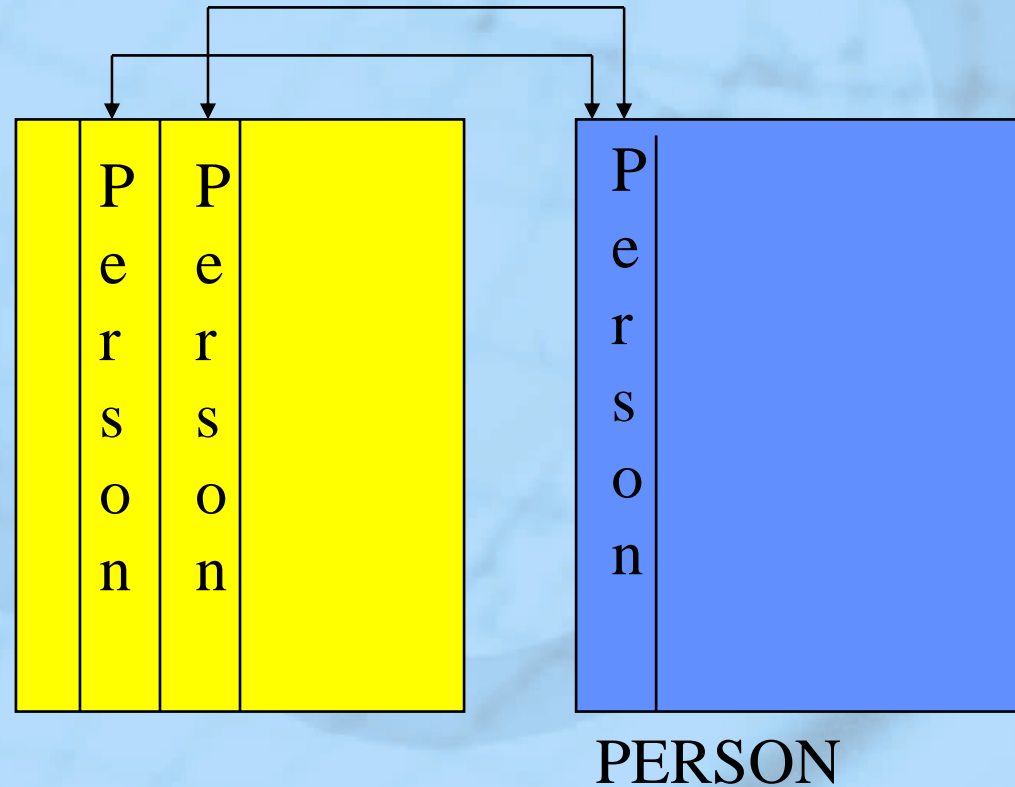


PERSON



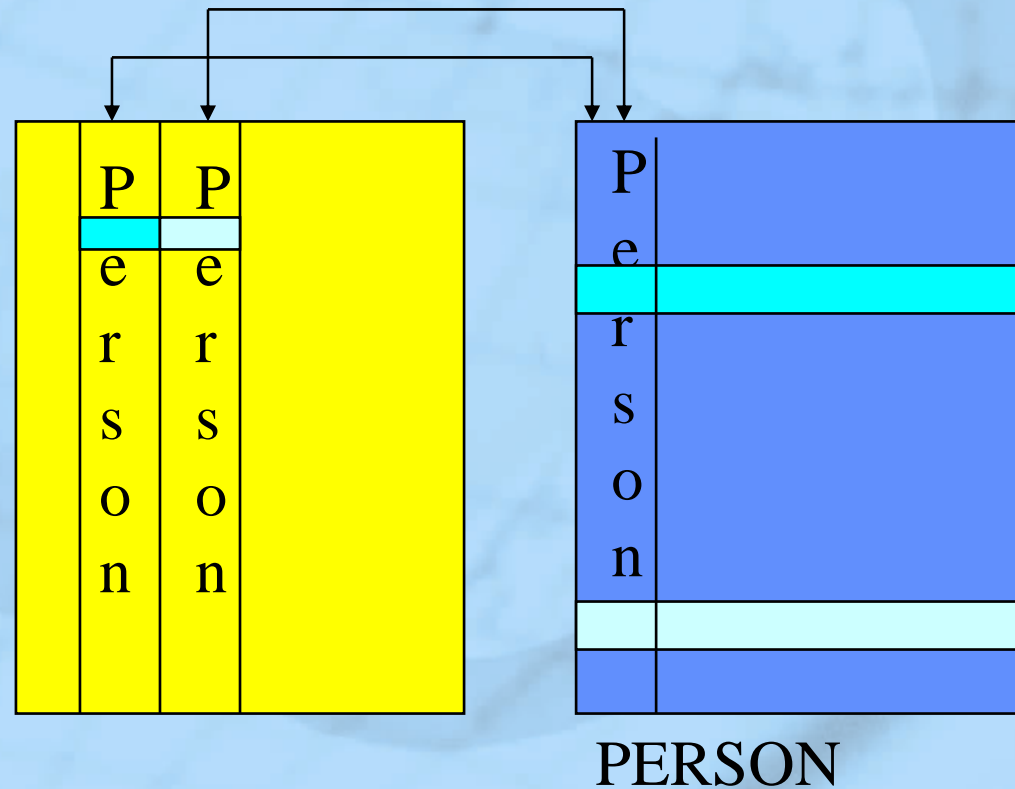
PERSON

# Binary Relations Recursion Binary Relation



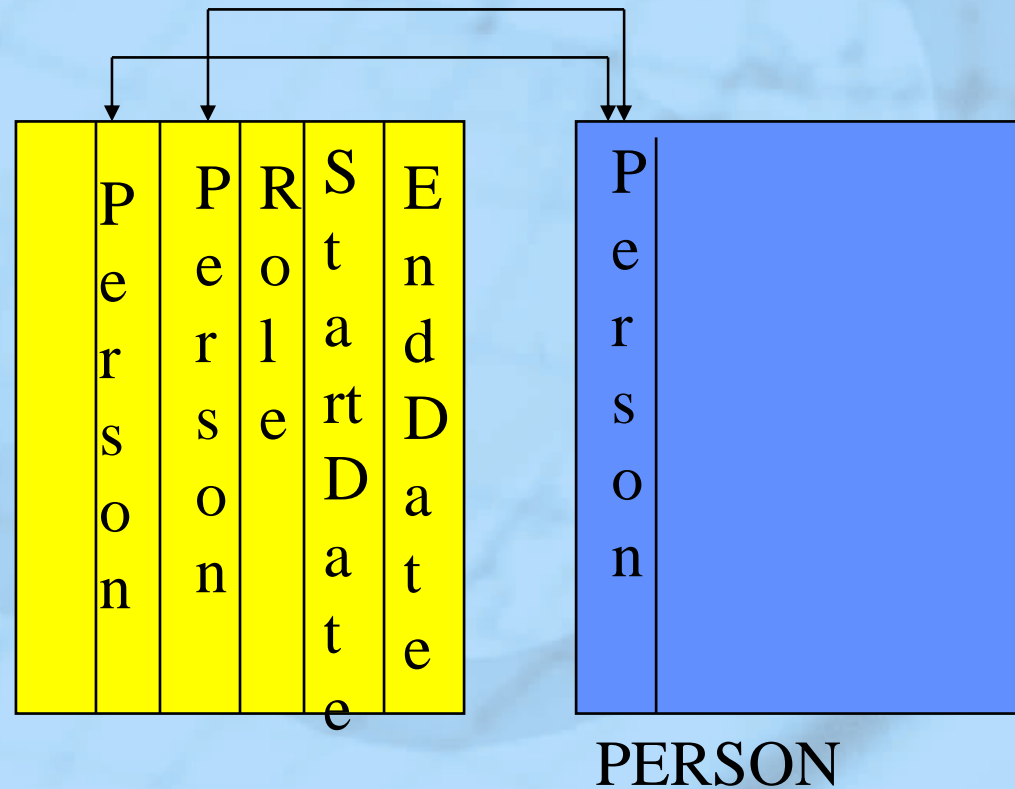
# Binary Relations Recursion Binary Relation

How the tuples from  
Person are represented  
in the binary relation



# Binary Relations Recursion Binary Relation

In practice usually  
have more  
attributes than  
Person / Person



## Binary Relations Binary Relation

---

- Flexible
- Allows  $n : m$
- With added attributes e.g. role, date/time
- Thus permitting
  - Conditional relationships
  - Temporal relationships
  - i.e. rich semantics

# Structure of Session

- Full, exchange and metadata models
- Full model – overview (nutshell)
- The concept of binary relations, linking relations and recursion
- **The concept of character / language variants**
- The concept of enumerated lists – dictionaries, thesauri, ontologies

- Character sets
  - Not only 'Latin-1' (need also to handle Greek, Arabic, Chinese...)
  - Can use escape codes technique but only works in linear data streams
  - Better to use a rich code that can handle any character from any language (including mathematics, financial currencies) as an atomic item - Unicode
  - But it requires more storage

- Language
- CERIF has many text fields
- Each field may exist in multiple languages
- For retrieval or update need to know the language (for text-matching)
- So have within the logical record multiple sub-records differentiated by language for each text field
- Example: Project.Abstract will usually exist in (US) English and original language and maybe language of country/region where stored

# Structure of Session

- Full, exchange and metadata models
- Full model – overview (nutshell)
- The concept of binary relations, linking relations and recursion
- The concept of character / language variants
- The concept of enumerated lists – dictionaries, thesauri, ontologies

- Purpose
  - Higher quality data: data validation
  - More accurate retrieval: query keywords limited and stored words (for any attribute) limited
  - Classification – allowing grouping and ranking by value of attribute

# Enumerated Lists, Dictionaries, Thesauri, Ontologies

## Enumerated List

- Example: Country Code
- There is an ISO standard list of valid 2-character and 3-character country codes
- On input can validate country code is from this list (commonly with a pull-down)
- If changes in countries, update the list in one place and whole system reconfigured

# Enumerated Lists, Dictionaries, Thesauri, Ontologies Dictionaries

- Example: meaning of a word (term)
  - Used in ensuring correct use of a value in an attribute
  - For explanation of result output
- Example: multilingual
  - Used in multilingual query (query in language 1 and retrieve from records stored in languages 2....n)
  - Used in result output – translate (crudely) to single language as required

# Enumerated Lists, Dictionaries, Thesauri, Ontologies

## Thesauri

- Provide the structural relationships of words (terms)
  - Synonym (different word same meaning)
  - Homonym (same word different meaning)
  - Antonym (word with opposite meaning)
  - Super-term (a word whose meaning includes the word being used e.g. person includes {student | worker | ....})
  - Sub-term (a word whose meaning is included in a Super-term)

# Enumerated Lists, Dictionaries, Thesauri, Ontologies Ontologies

- Ontology: philosophical study of existence and nature of reality
- In practice a resource of terms, their definitions and their logical inter-relationships
- E.g. For a publication to exist it is necessary to have a title, at least 1 author
- Publication  $\leftarrow [\exists \text{ title AND } \exists \geq 1 \text{ author}]$

# Enumerated Lists, Dictionaries, Thesauri, Ontologies Ontologies

- Domain Ontology: Ontology covering a domain (subject area of interest)
- Example Publication
- Publication  $\leftarrow [\exists \text{ title}] \text{ AND } \exists \text{ author}]$
- Collection  $\leftarrow [\exists \text{ title} + \exists > 1 \text{ author} + \exists \text{ editor}]$
- *If Publication has title, > 1 author and editor it is a collection*
- Publication is\_part\_of Collection
- Collection is\_a\_kind\_of Publication

- Domain Ontologies in IT
- A representation in first order logic allowing
  - Facts to be expressed
  - Relationships to be expressed
  - Constraints to be expressed
  - New facts and relationships to be deduced or induced

# Enumerated Lists, Dictionaries, Thesauri, Ontologies Ontologies

- Used
  - Data validation on input
  - Clarification and improvement of a query
  - Resolving heterogeneity of terms to homogeneity
  - Expanding super-terms to subterms and vice-versa conditionally
  - Deducing or inducing new facts and relationships from stored facts and relationships

- CERIF is a data model with 'levels'
  - Primary base entities
    - e.g. Person
  - Secondary base entities
    - e.g. Result\_Publication
  - Language-base entities
    - e.g. Abstract
  - Lookup Tables
    - e.g. Role of Person
  - Linking Relations
    - e.g. Project <-> Person