

LDP-DL: A Language to define the design of Linked Data Platforms

Noorani Bakerally, Antoine Zimmermann, Olivier Boissier
Dpt Informatique, Institut Henri Fayol, EMSE,
Connected intelligence, Laboratoire Hubert
Curien, UMR CNRS 5516



W3C Linked Data Platform 1.0

- Linked Data principles
- REST principles
- W3C Linked Data Platform (LDP) 1.0
- Standardize RESTful access to Linked Data
- HATEOAS in LDP:
 - Decouples LDP server and client
 - Enables development of generic LDP clients
- In short, LDPs can be used to:
 - provide a homogeneous view and access to data
 - facilitating both the task of data publishers & consumers

Overview of LDP standard

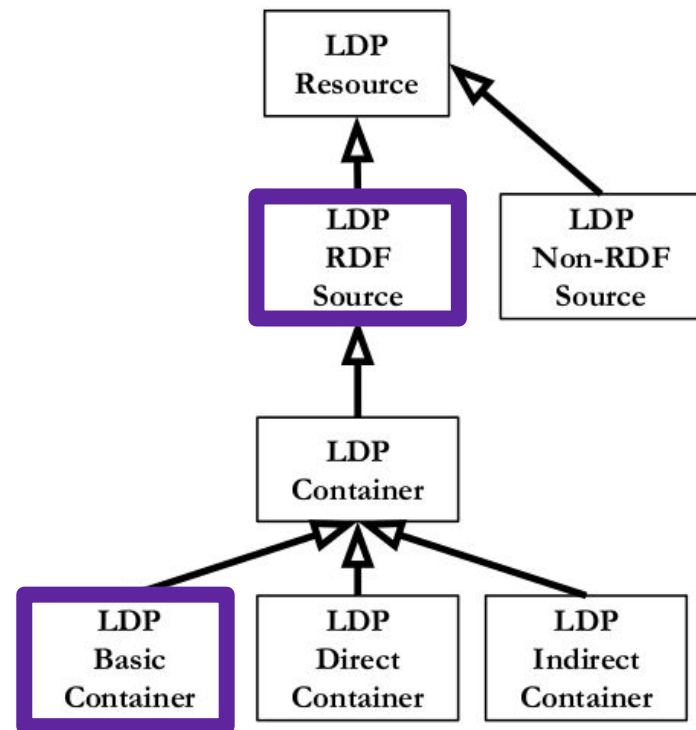
Parts of LDP standard:

• Domain Model

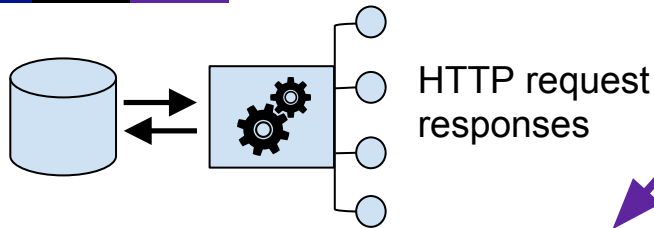
- Different types of LDP resources
- Containers to group resources
- Basic containers contains members that are either LDP RDF or Non-RDF sources

• Interaction Model

- Manipulate LDP resources through HTTP methods
- E.g. specifies forms of HTTP request and responses



Current LDP Implementation

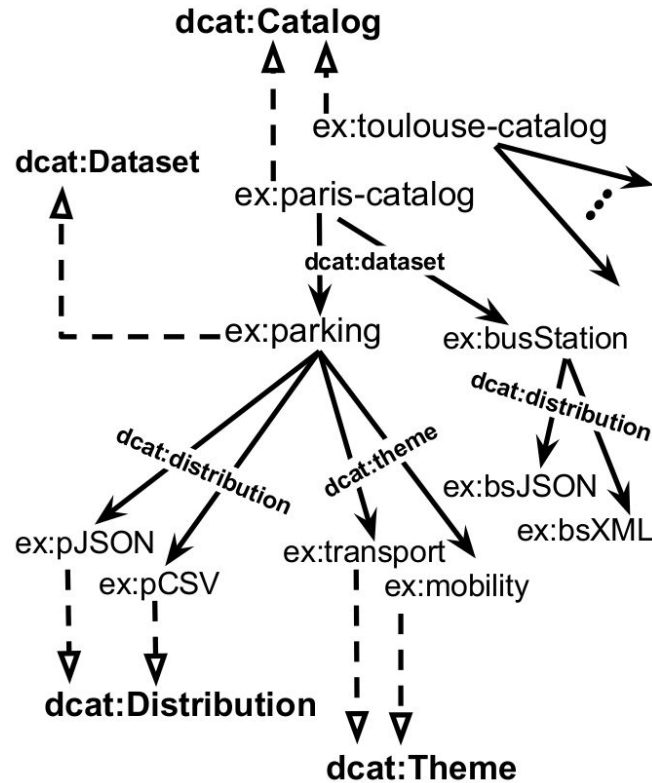


- LDP Resource Management System
 - repository for storing LDP resources
 - Allows read-write operations to manipulate resources through HTTP methods
- LDP Frameworks:
 - API to facilitate development of LDPs
 - E.g. helper methods, generic class templates
- Use of LDP standard requires data **design** and data **deployment**
- Current LDP impl provide no **automatic support** neither for data **design** nor **deployment** even if the data is already in **RDF** !!
- **Our solution**
 - LDP-DL(LDP design language) to describe the design of an LDP
 - Generate an LDP using an LDP-DL model

Outline

- Overview of LDP-DL
- Syntax of LDP-DL
- Semantics of LDP-DL
- Implementation
- Evaluation

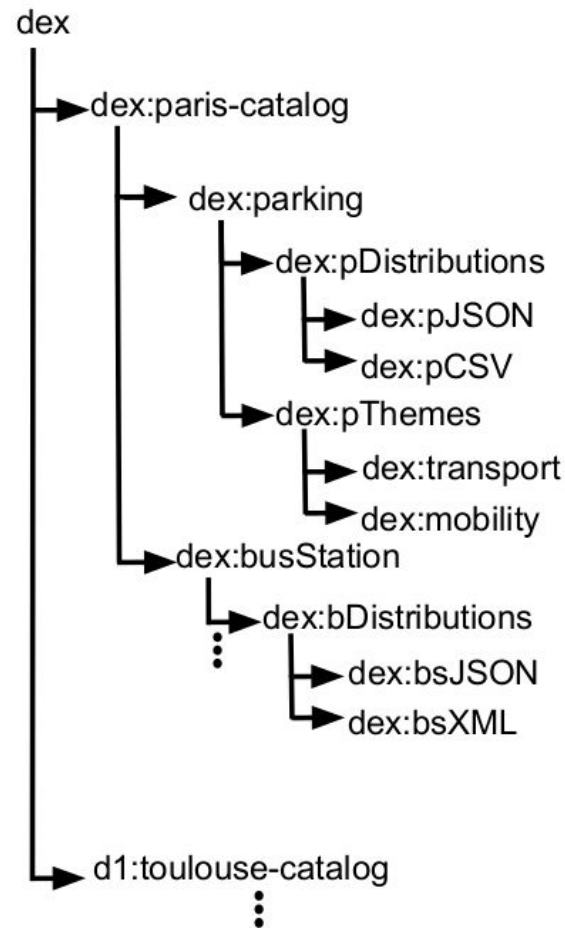
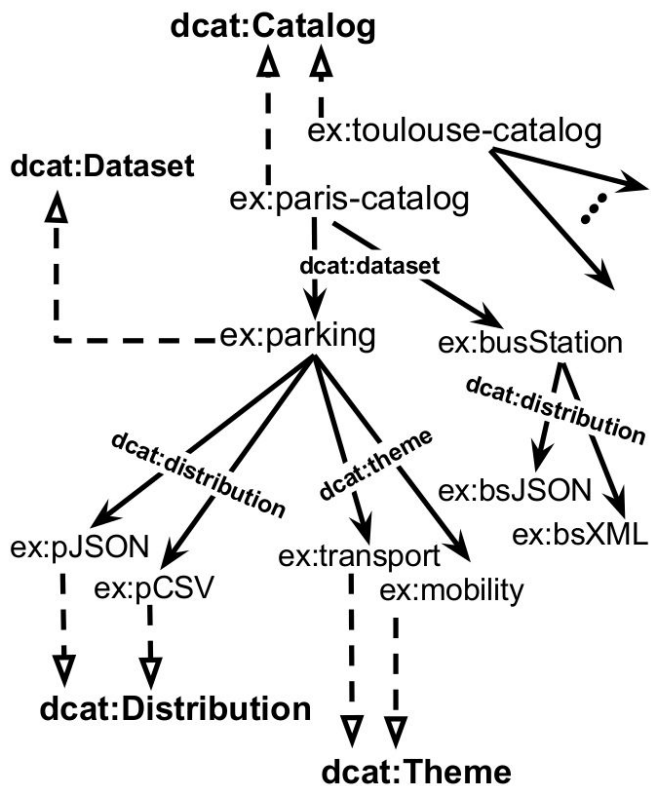
Overview of LDP-DL



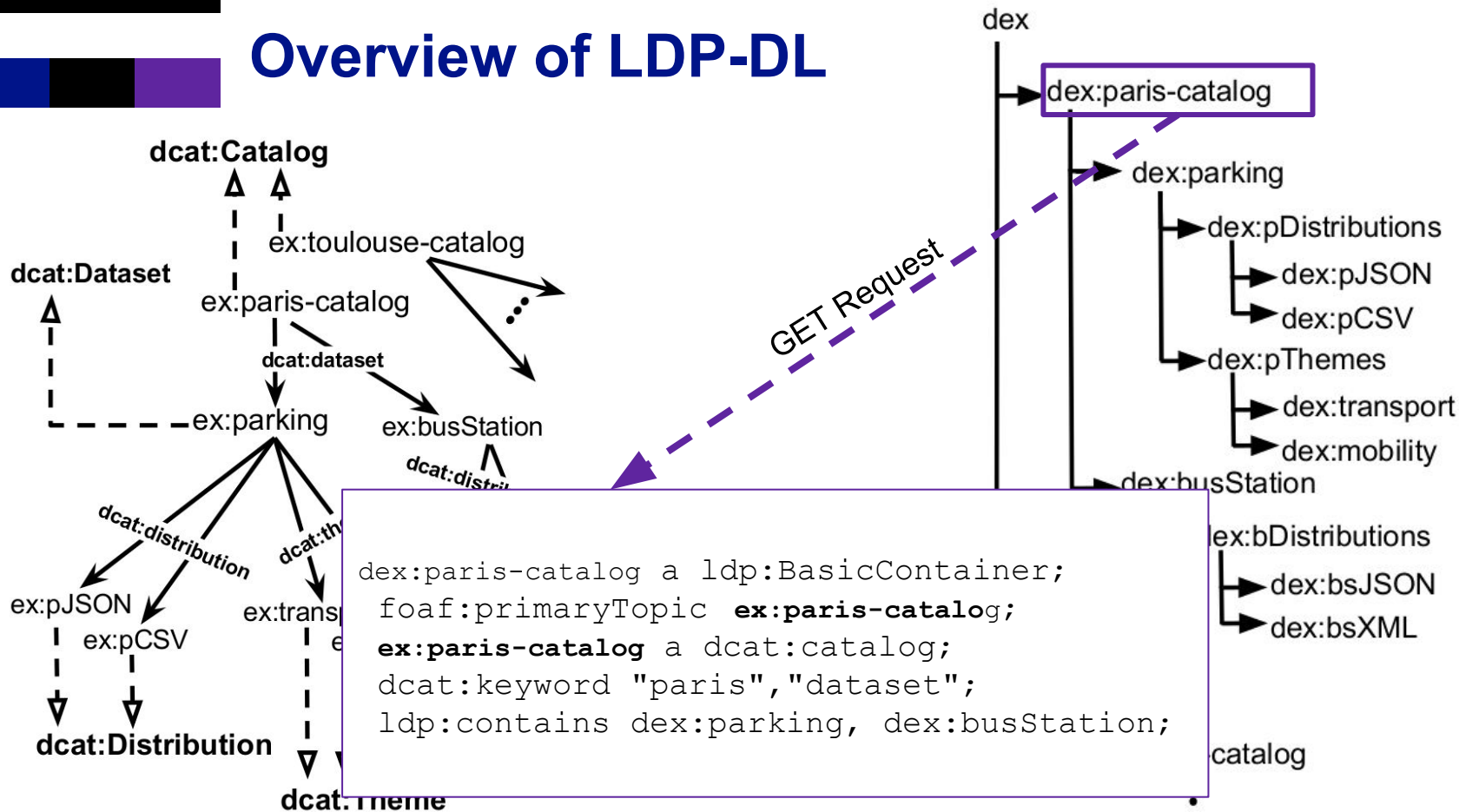
Overview of LDP-DL

- Minimum set of questions:
 - What are the LDP resources?
 - In which containers are they?
 - What are the content of LDP resources:
 - LDP RDF source (non-container)
 - LDP Basic Container (container)

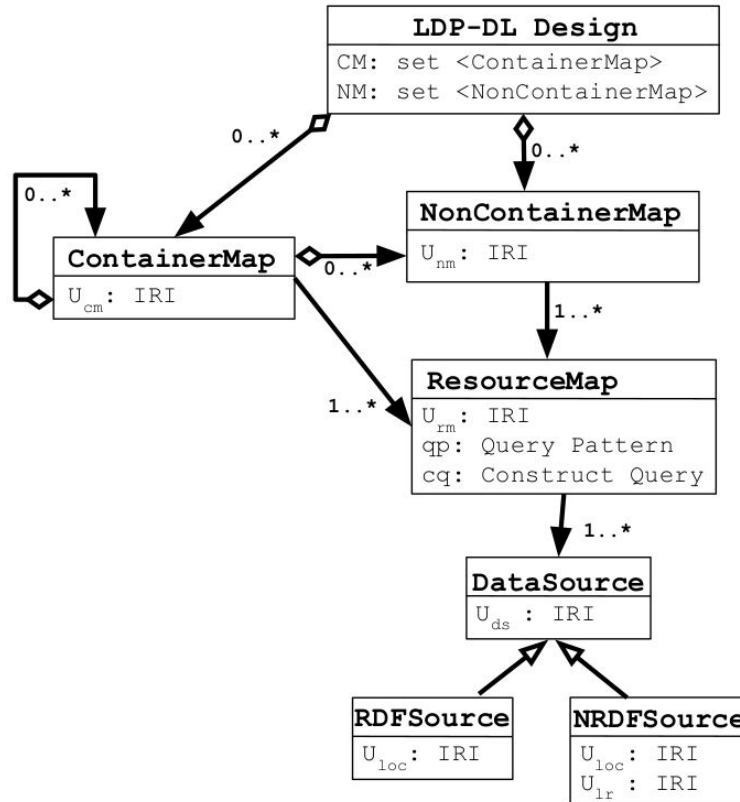
Overview of LDP-DL



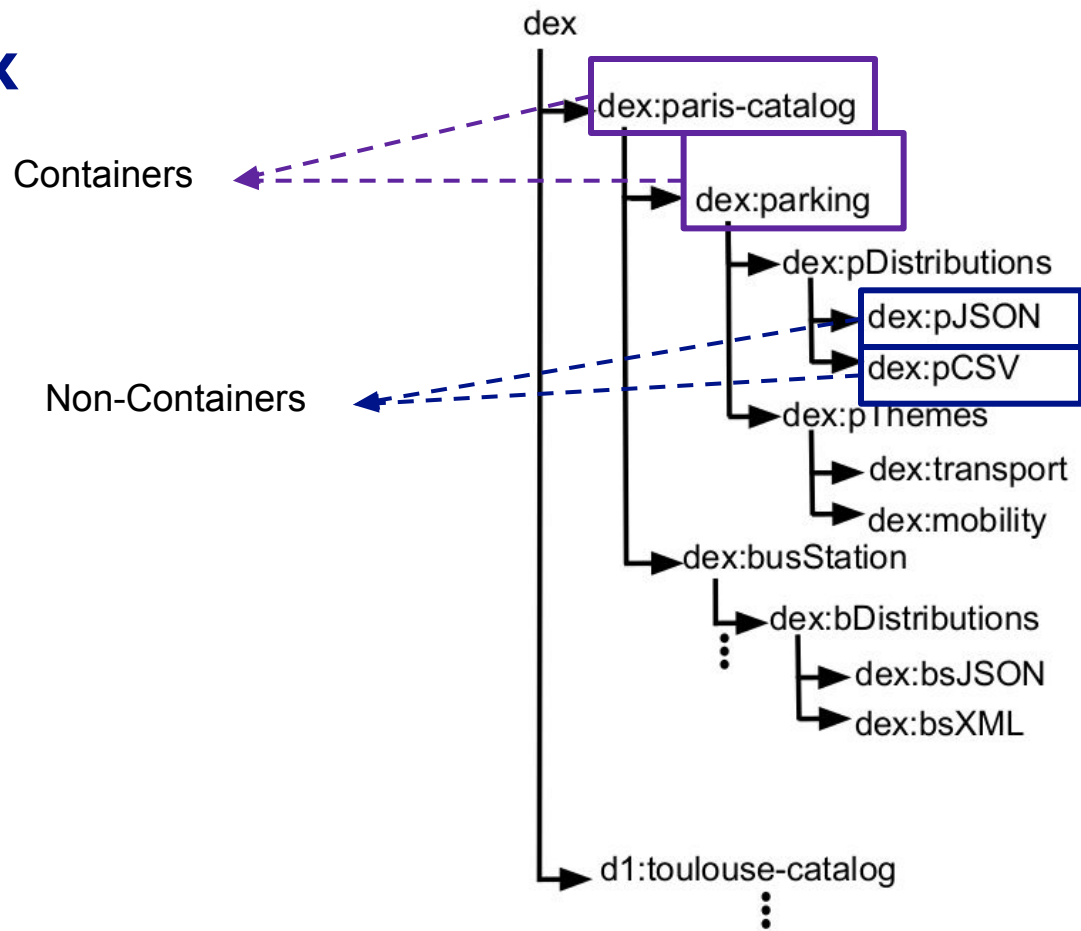
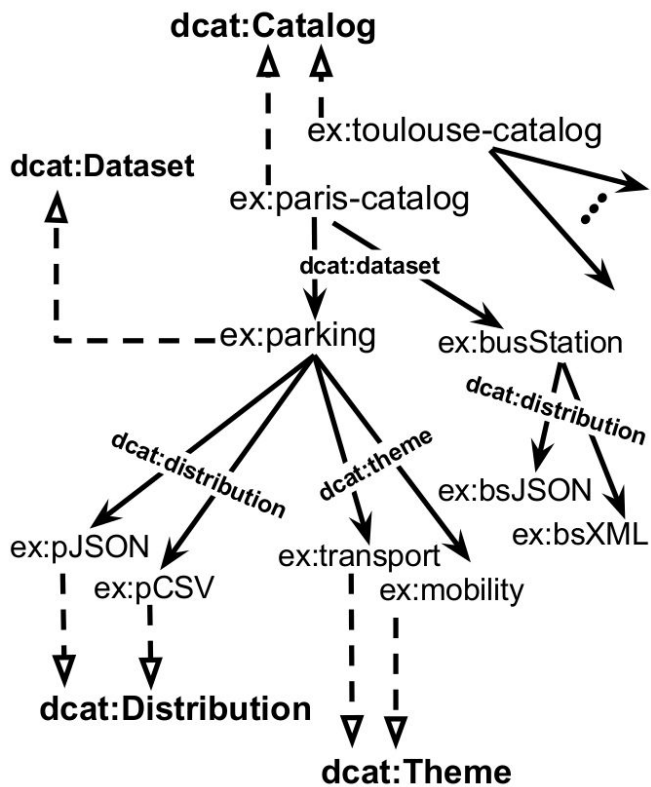
Overview of LDP-DL



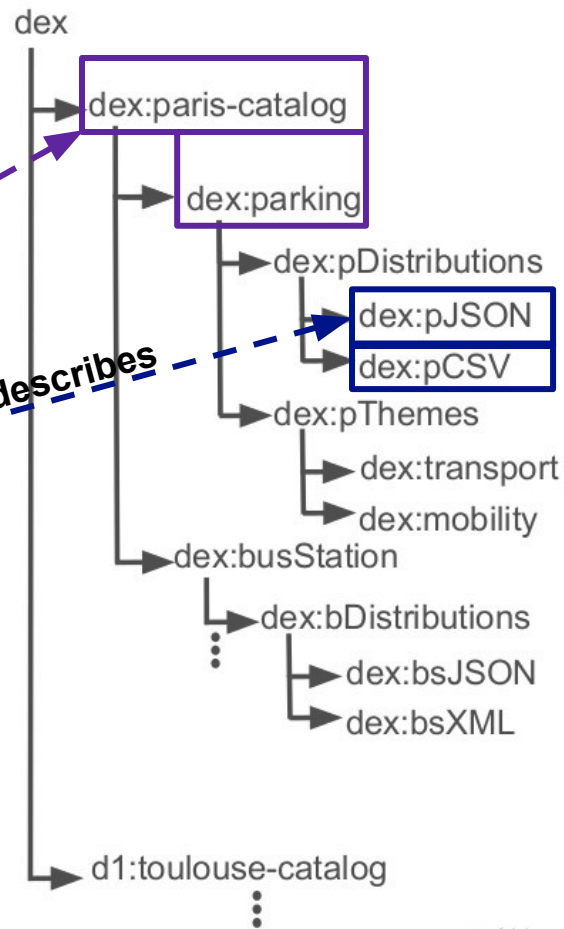
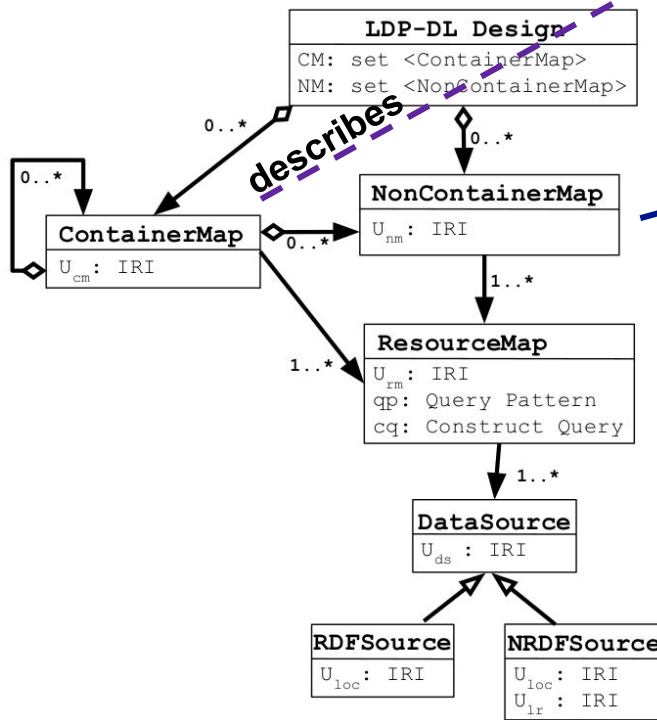
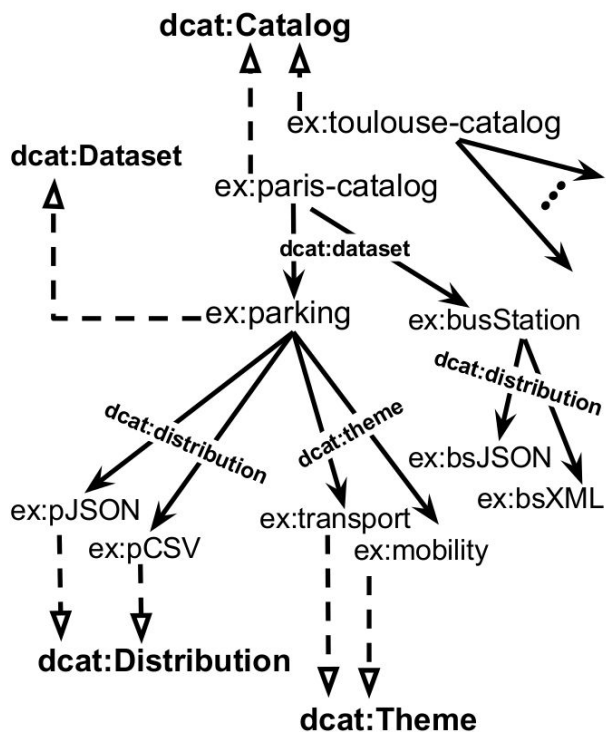
LDP-DL Syntax



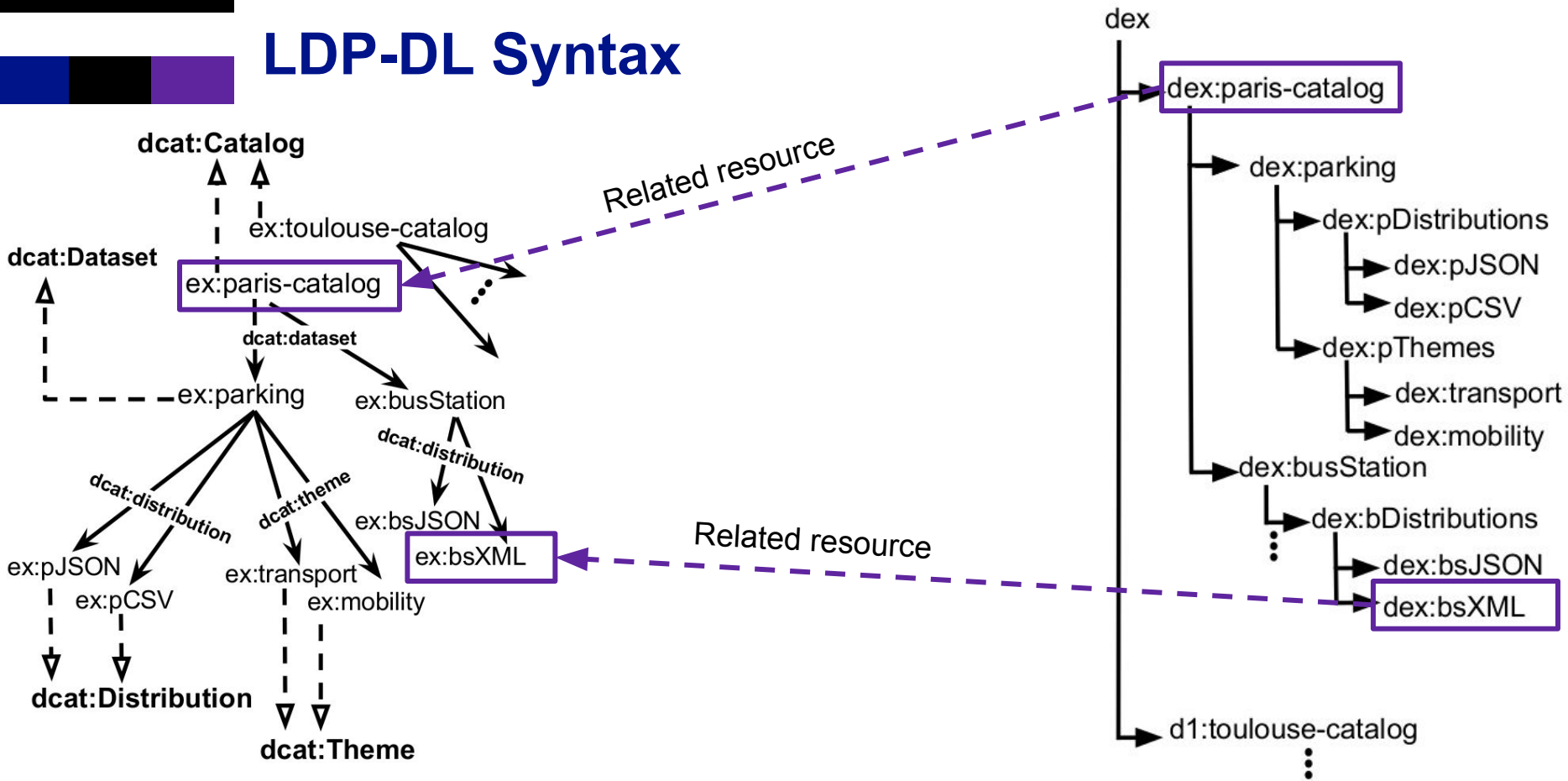
LDP-DL Syntax



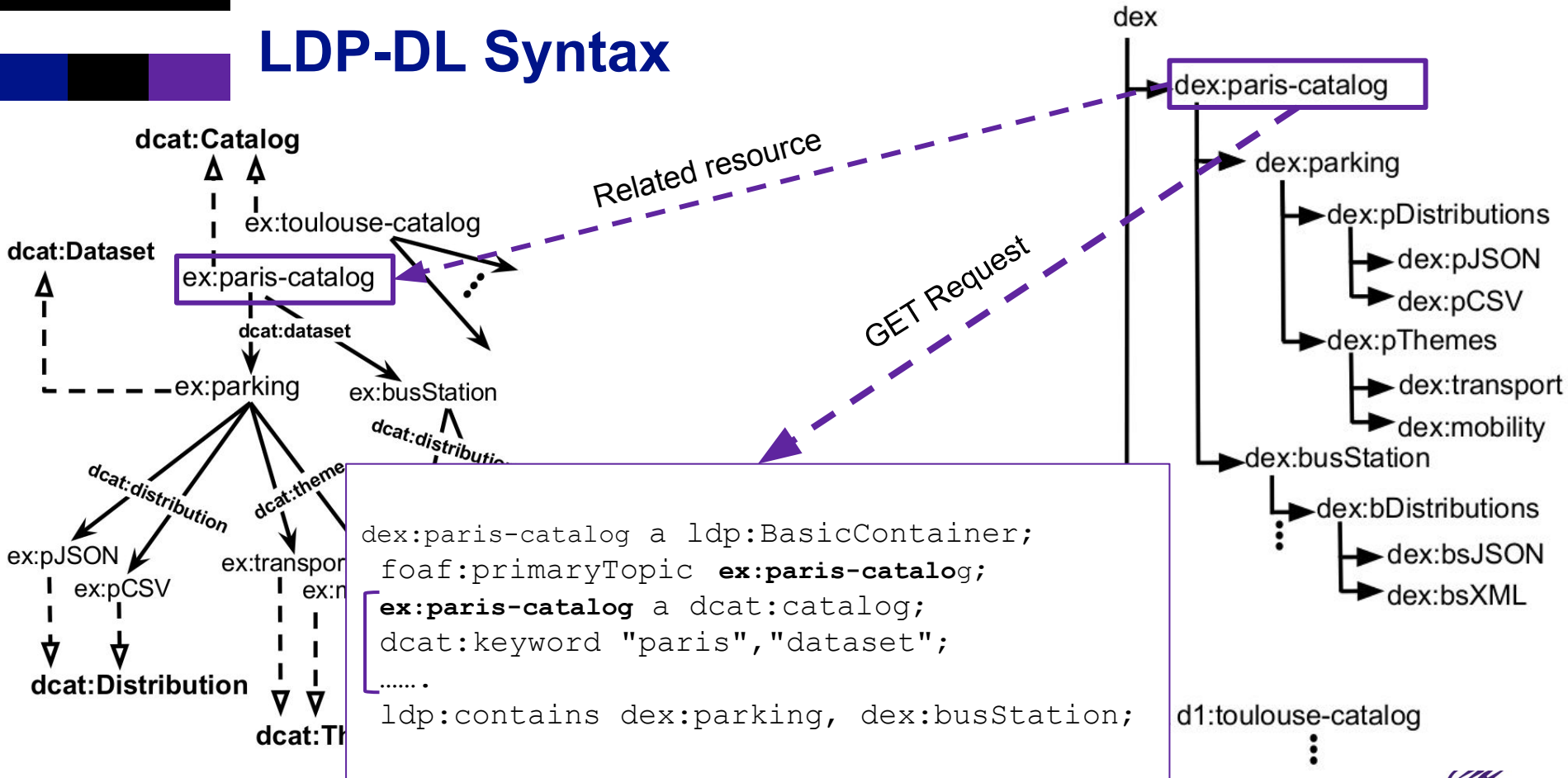
LDP-DL Syntax



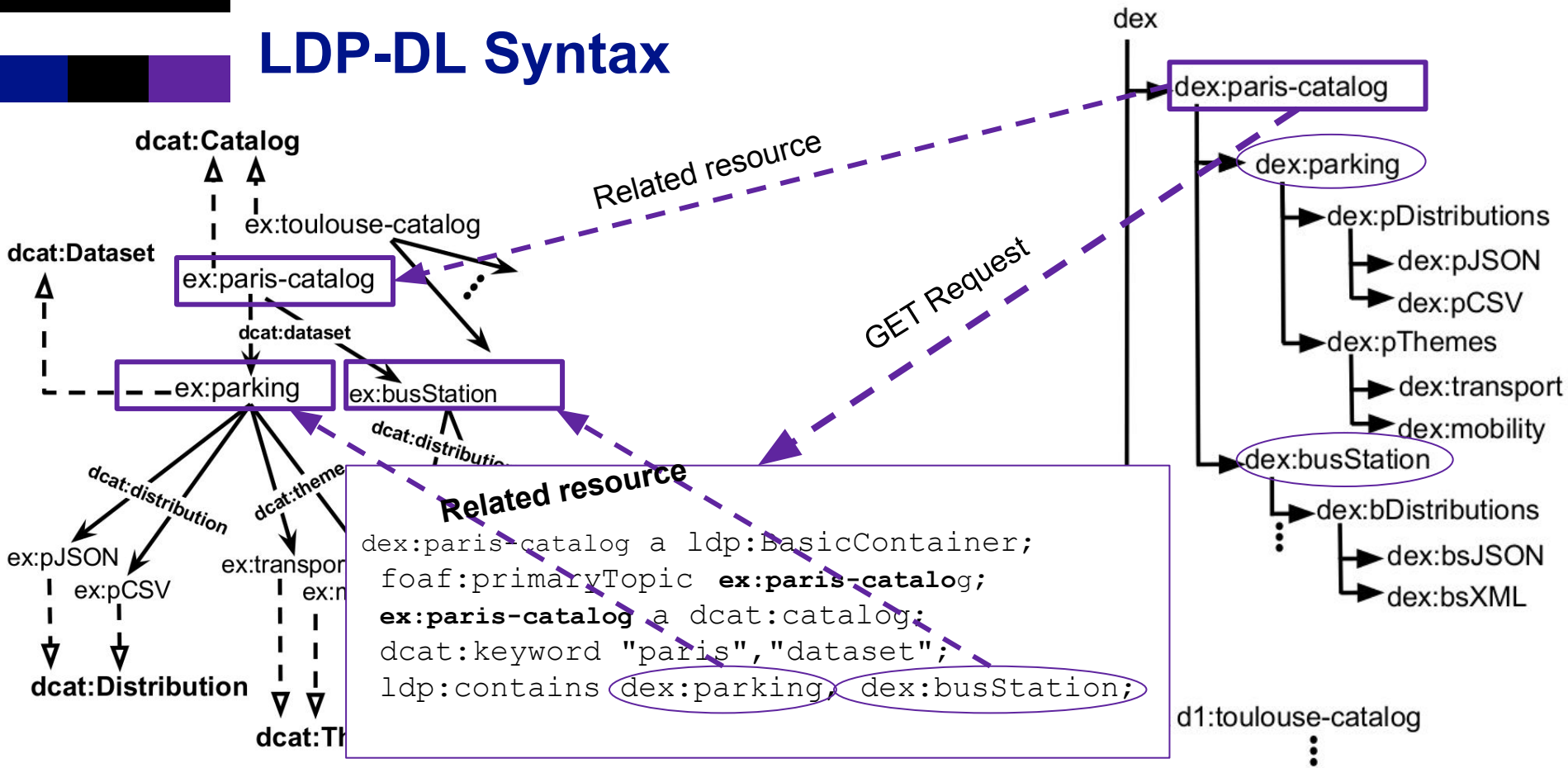
LDP-DL Syntax



LDP-DL Syntax

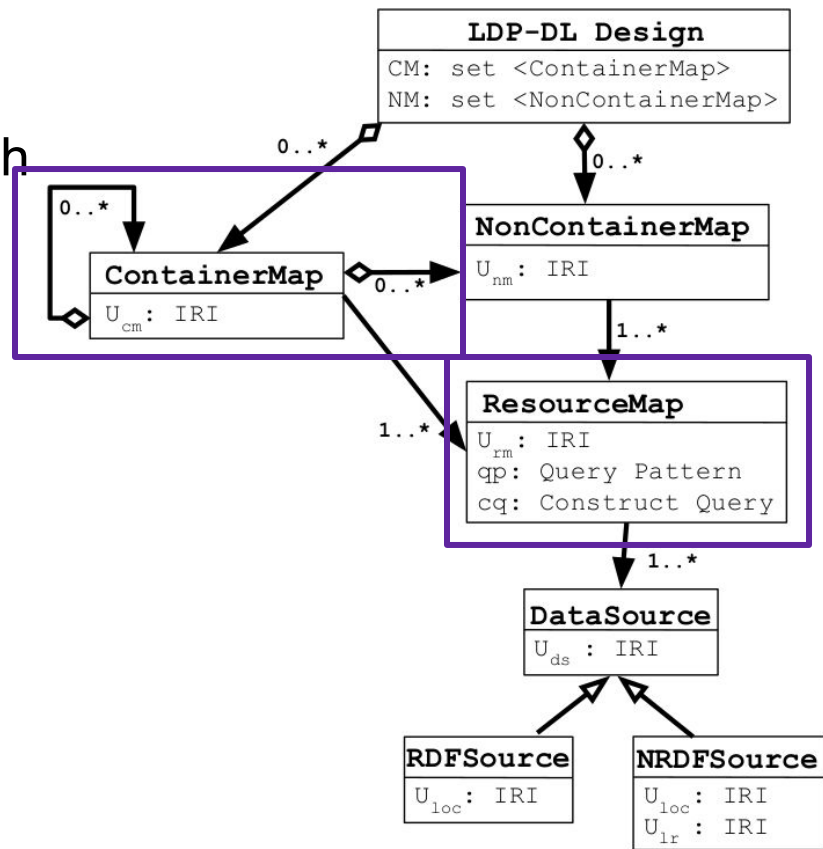


LDP-DL Syntax

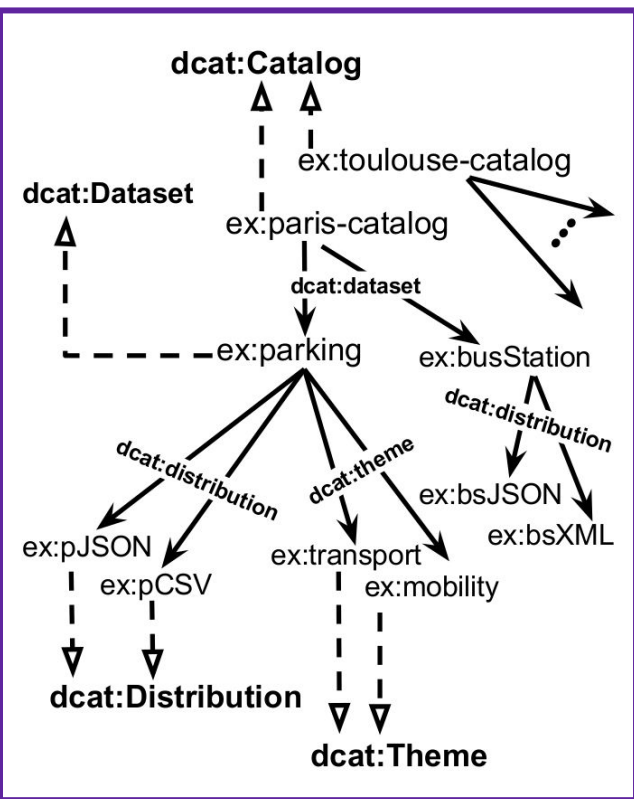


LDP-DL Syntax

- **ResourceMap** are used to identify related resources and describe the graph of LDP resources
- Then, **ContainerMap** and **NonContainerMap** uses **ResourceMap** to create containers or non-container(2)
- To describe members of containers, **ContainerMap** allows nested **ContainerMap** & **NonContainerMap** (3)



LDP-DL Semantics

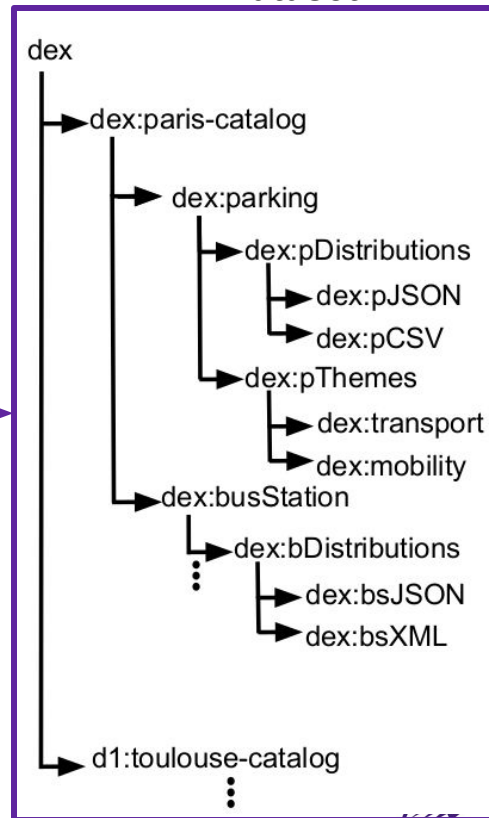


Design document



LDP-DL Interpreter
(LDPizer)

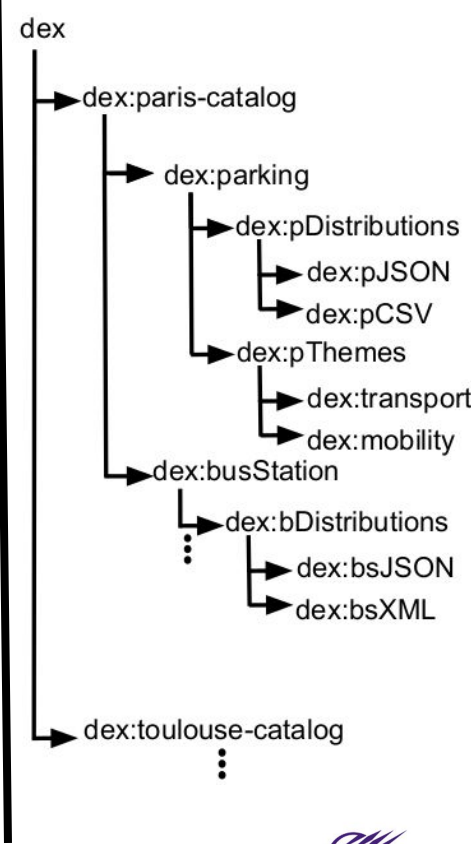
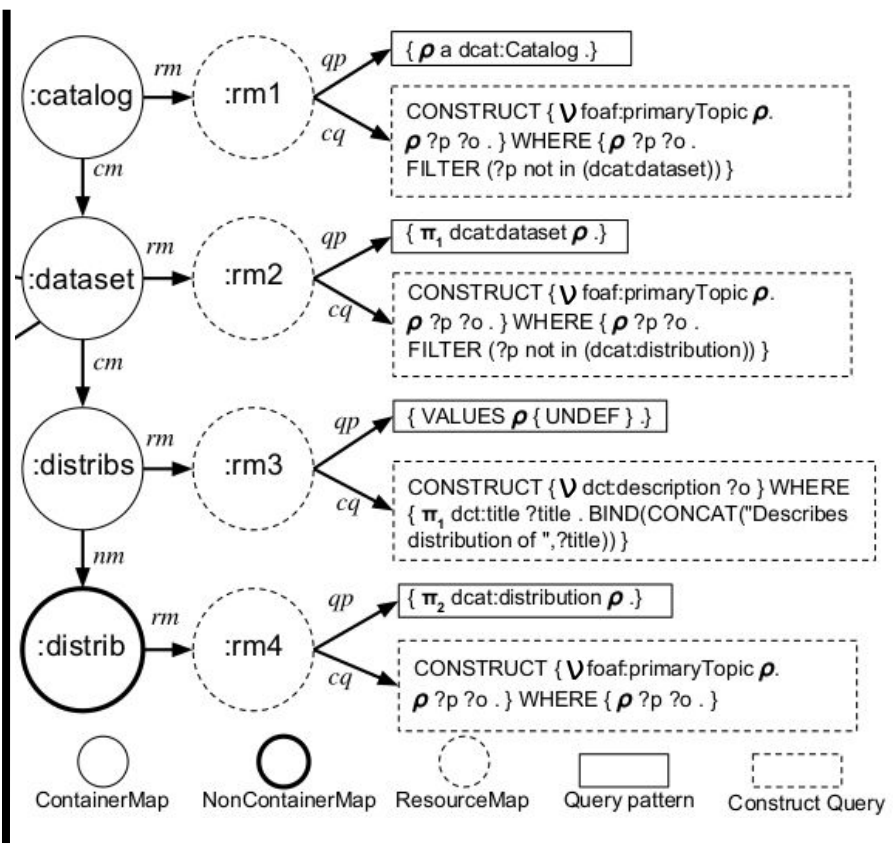
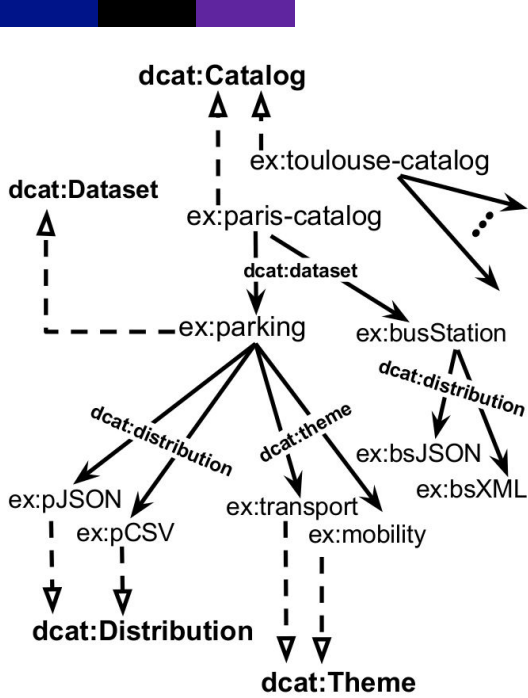
LDP Dataset



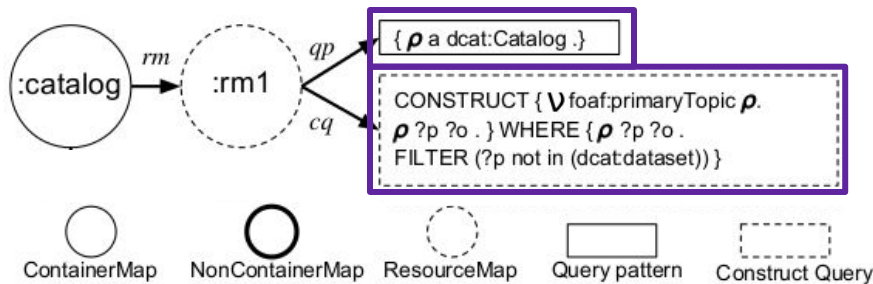
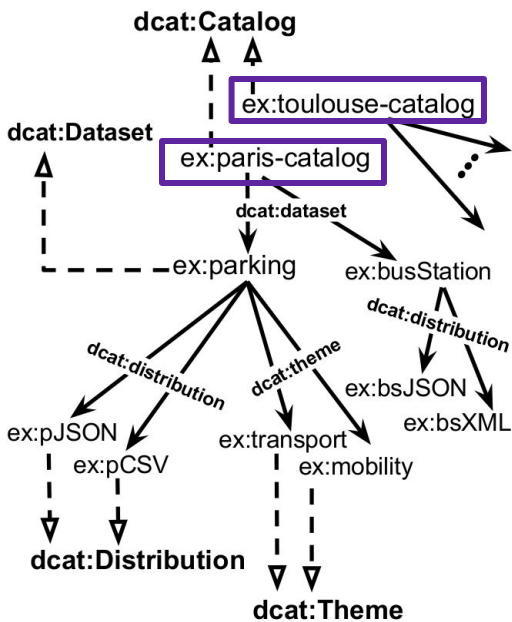
LDP-DL Semantics

- Aim of the formal semantics is to associate an **LDP dataset** to a **design document**
- **LDP Dataset** consists of:
 - **Set of named graphs (n,g)**
 - describes non-containers where n is the IRI of the non-container and g its RDF graph
 - **Set of container structures (n,g,M)**
 - describes containers n is the IRI of the container and g its RDF graph, and M is a set of IRIs representing the members of container n

LDP-DL Semantics

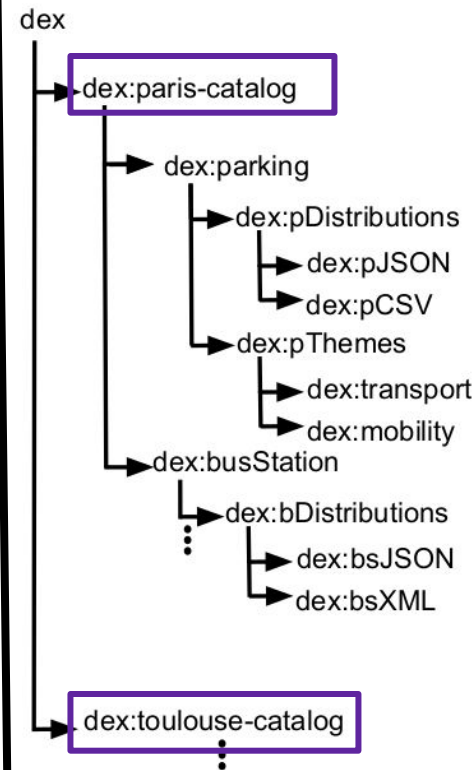


LDP-DL Semantics

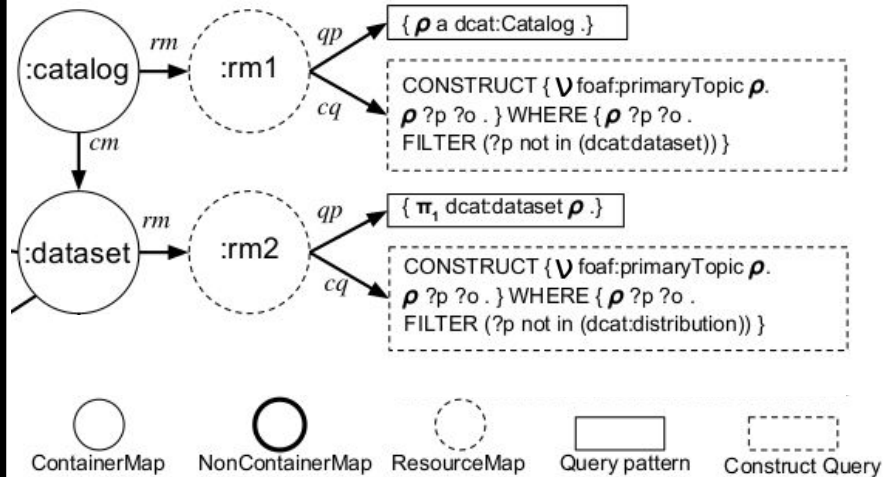
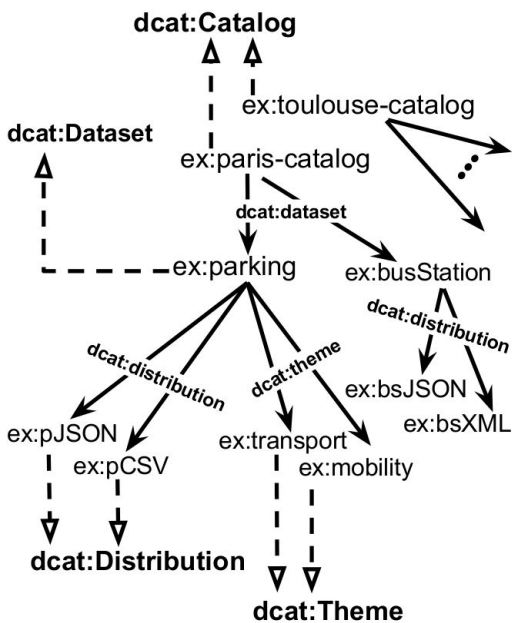


1. Eval of qp returns **{ ρ ← ex:paris-catalog }** and **{ ρ ← ex:toulouse-catalog }**
2. for each of them, a new resource is created
3. consider **{ ρ ← ex:paris-catalog }**
4. the new resource (ν) is **dex:paris-catalog**
5. To generate graph of **dex:paris-catalog**, cq is evaluated on the source with the bindings **{ ρ ← ex:paris-catalog }, { ν ← dex:paris-catalog }**

ρ: related resource, ν: new LDP resource

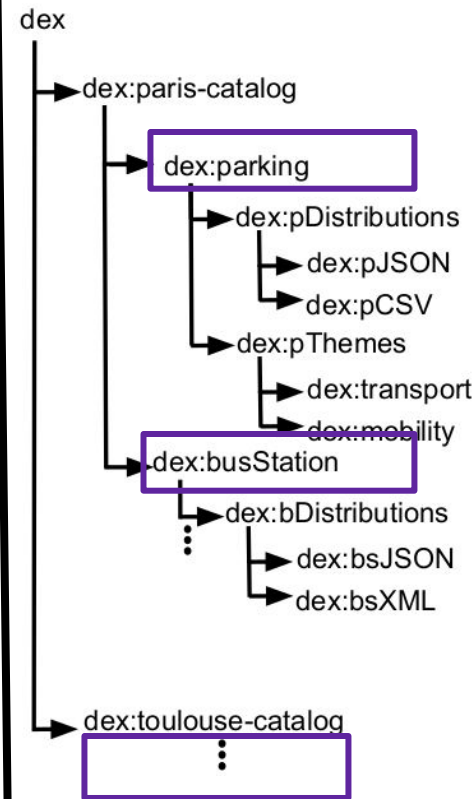


LDP-DL Semantics

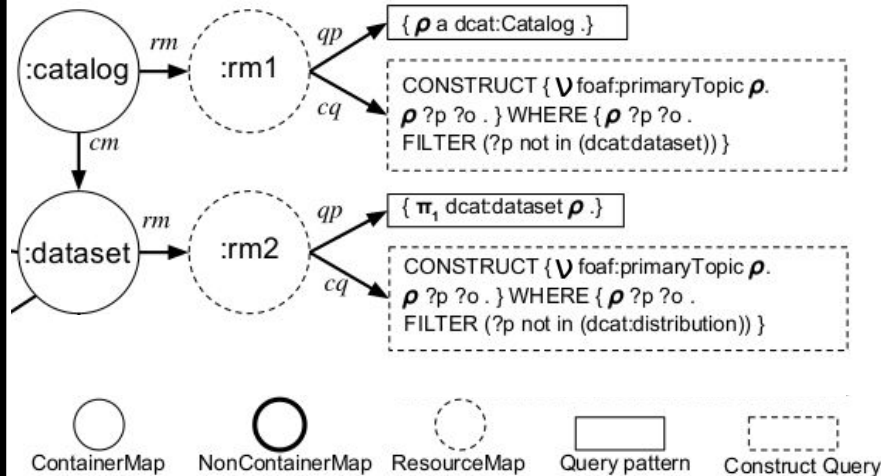
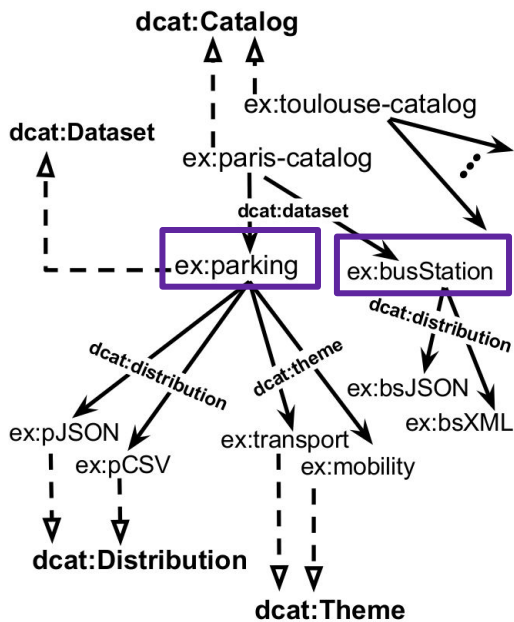


`:dataset` ContainerMap

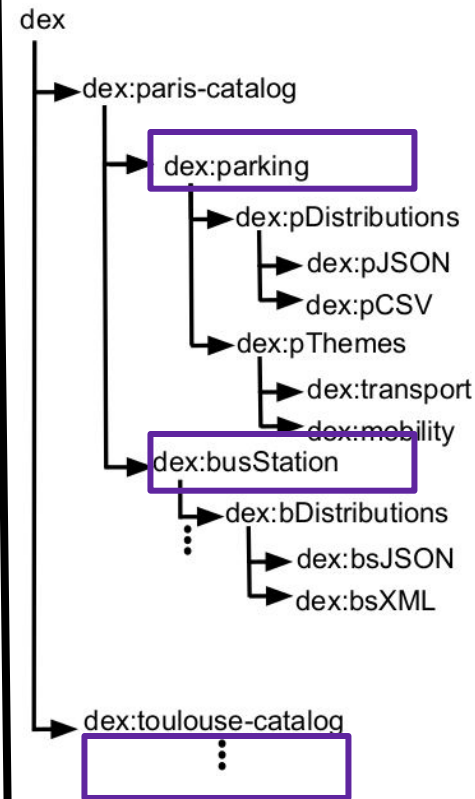
members of `dex:paris-catalog` and `dex:toulouse-catalogs`



LDP-DL Semantics



- Consider eval of **:dataset** to generate members of **dex:paris-catalog**
- members of **dex:paris-catalog** describes **dcat:datasets** of **ex:paris-catalog** (related resource)
- eval of qp is done with bindings $\{\pi_1 \leftarrow \text{ex:paris-catalog}\}$

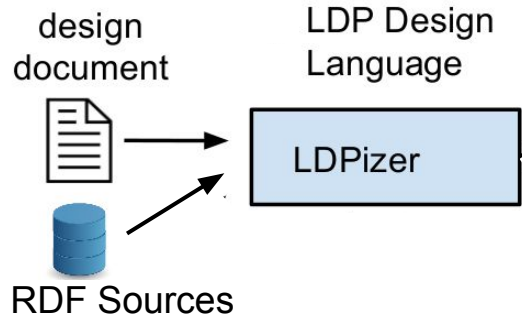


Implementation

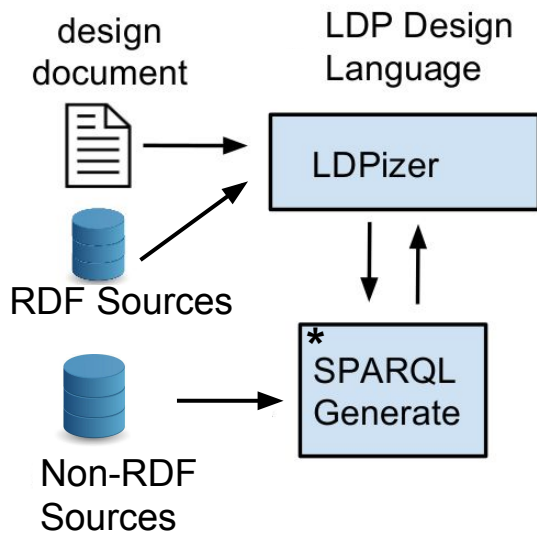
design
document



Implementation

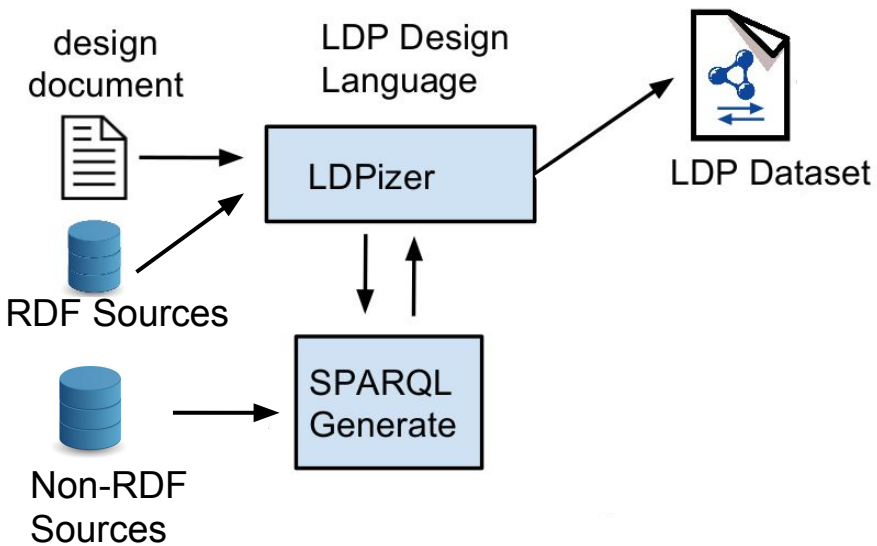


Implementation

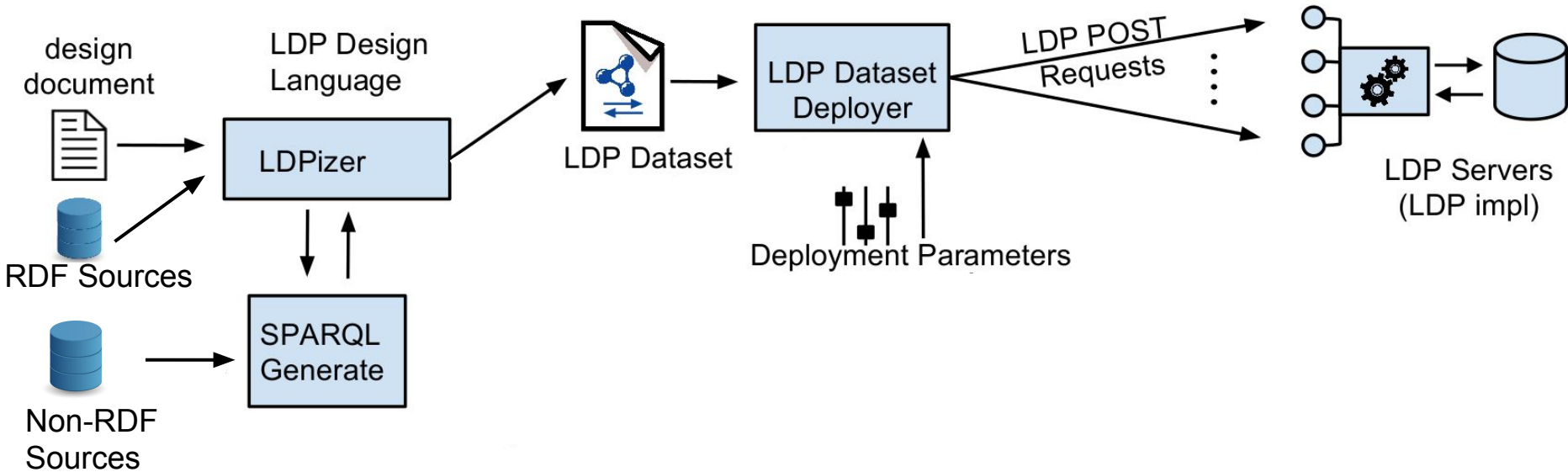


*Maxime Lefrançois, Antoine Zimmermann, and **Noorani Bakerally**. "A SPARQL extension for generating RDF from heterogeneous formats." *European Semantic Web Conference*. Springer, Cham, 2017.

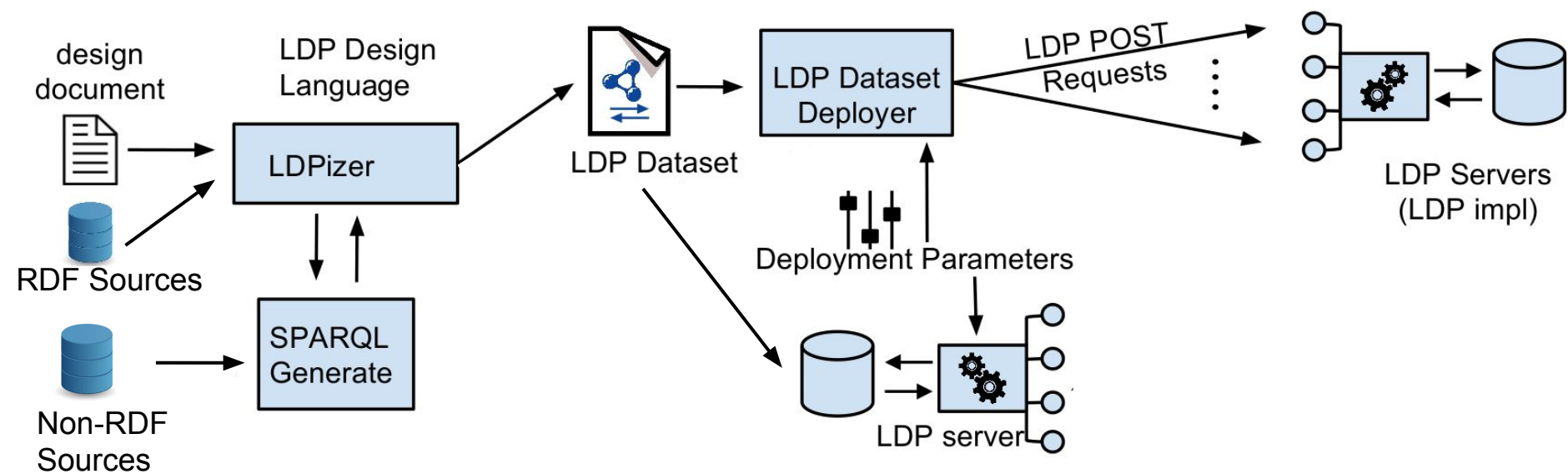
Implementation



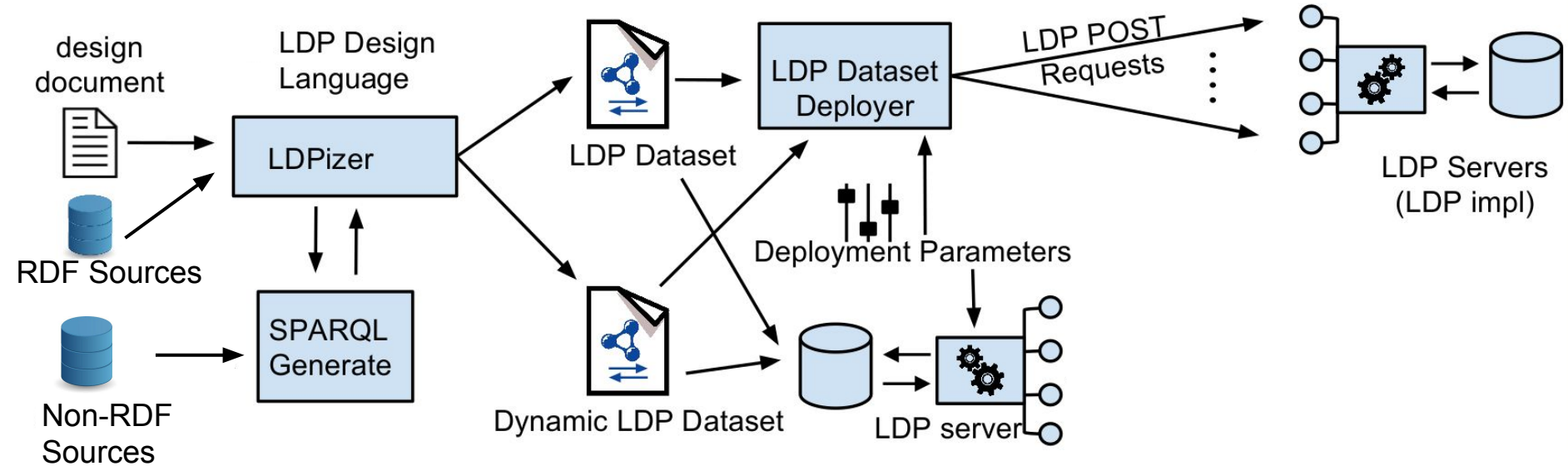
Implementation



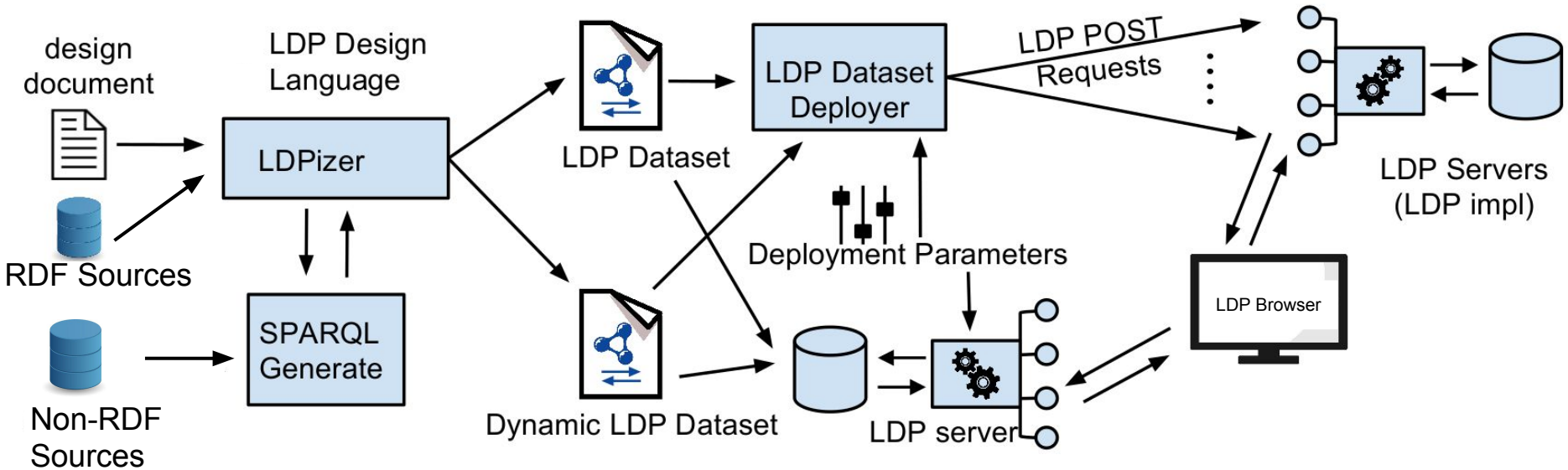
Implementation



Implementation



Implementation



Evaluation

- Experiments:
 - Different design documents
 - Heterogeneous & real-time data sources
 - Generated over than 100 LDPs
- **Reusability**: the same design on different data sources
- **Flexibility**: different designs on the same data source
- **Loose coupling**: different LDPs with the same design
- **Partial design automatization**: generic designs that can be used on RDFS/OWL vocabularies
- Compatibility with existing LDP implementations
- ability to deal with heterogeneous & real-time data sources

Conclusion

- LDP design language (LDP-DL)
- Implementation
- Evaluation
- Future works:
 - extend LDP-DL to other types of LDP resources such as Direct/Indirect container and non-RDF source
 - support other design aspects such as access rights
 - LDP Algebra
 - support paging, a desired feature for large datasets

**Thank you for you attention,
Questions ?**