Software System Architectures (NSWI130)

Architectural Views

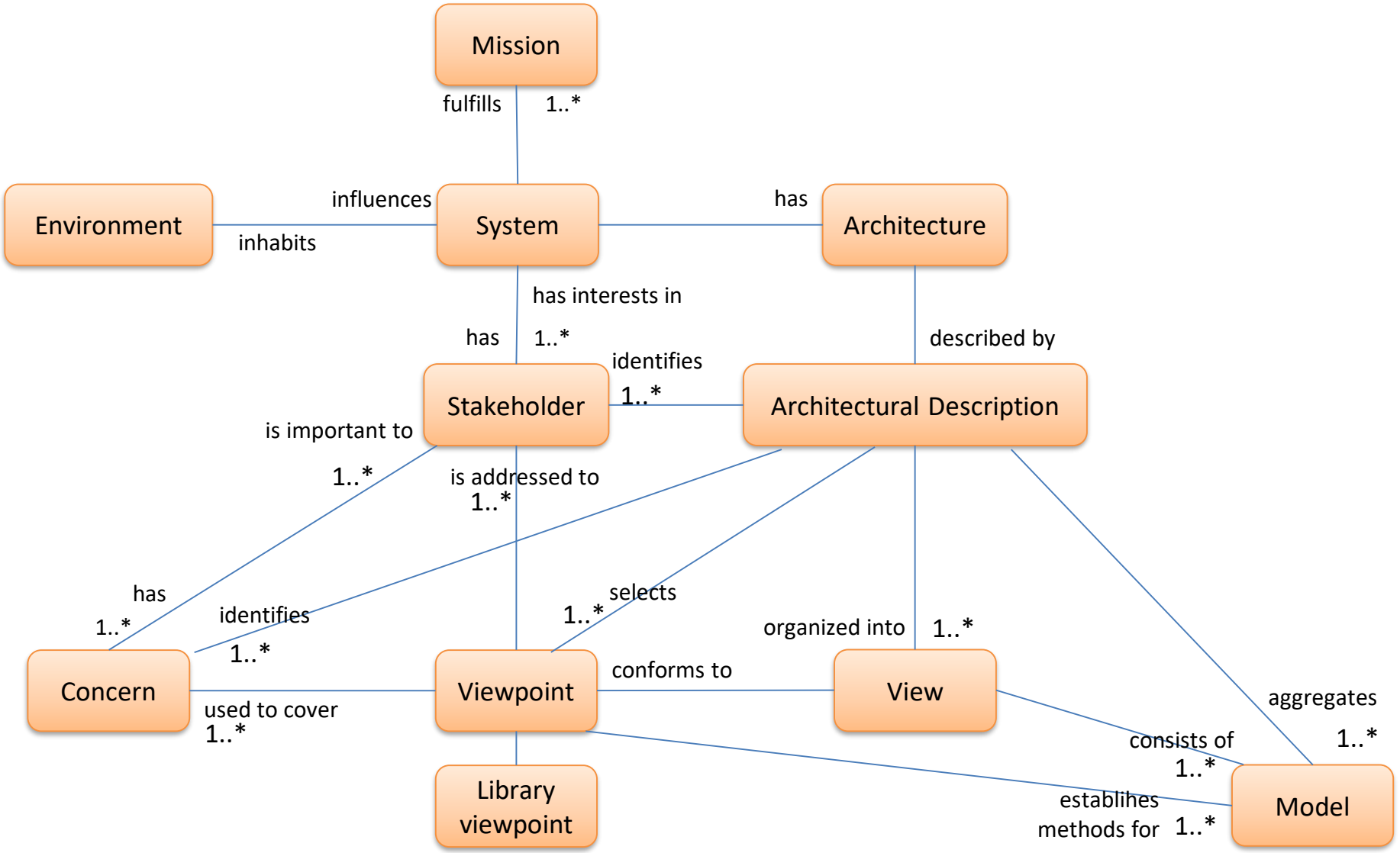
Martin Nečaský, Ph.D.

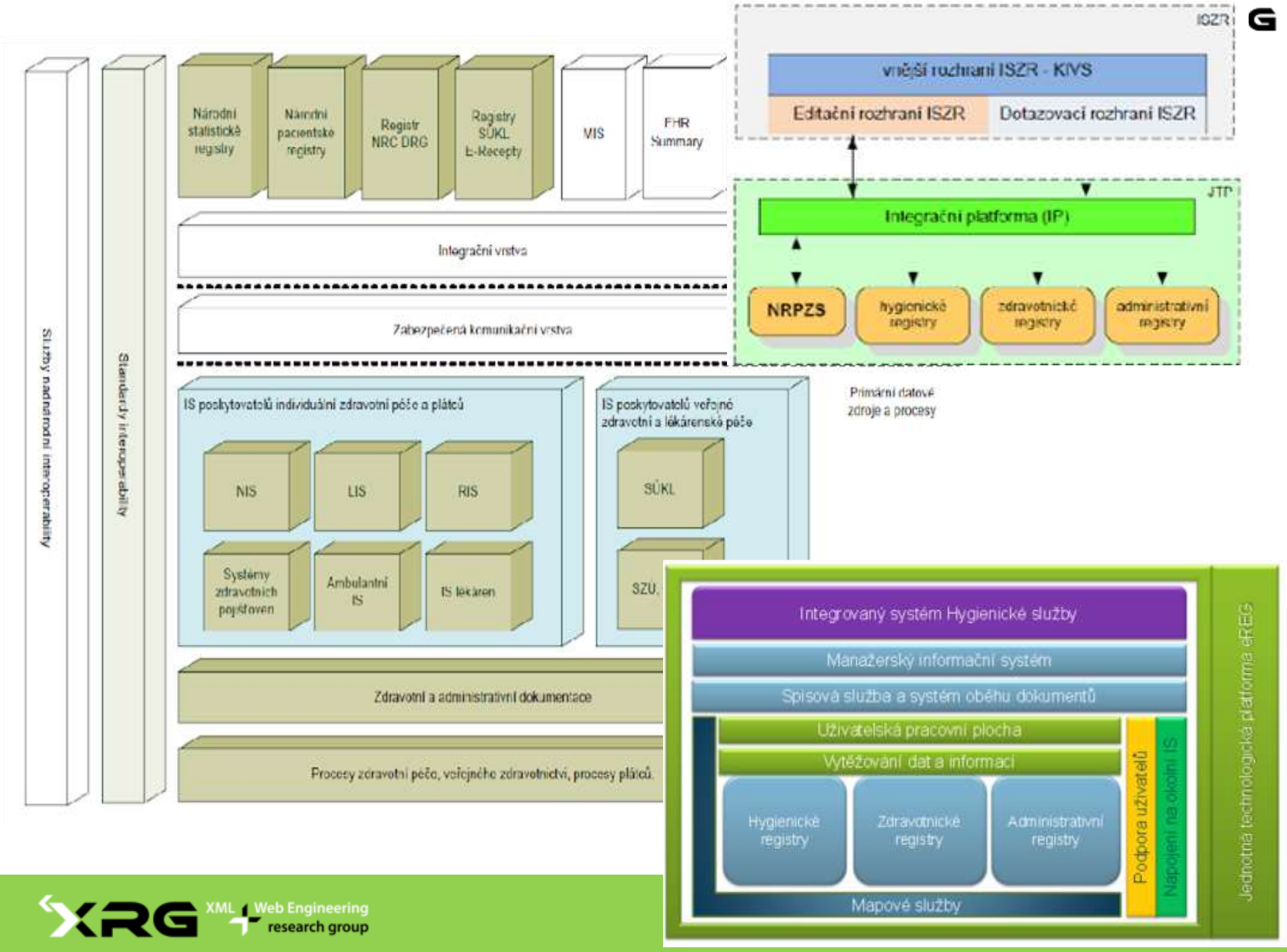
Department of Software Engineering

Faculty of Mathematics and Physics

Charles University in Prague

IEEE 1471 Conceptual Framework




ISZR
vnější rozhraní ISZR - KIVS
Editační rozhraní ISZR
Dotazovací rozhraní ISZR
JTP
Integrační platforma (IP)
NRPZS
hygienické registry
zdravotnické registry
administrativní registry
Národní statistické registry
Národní pacientské registry
Registr NRC DRG
Registry SÚKL E-Recepty
VIS
EHR Summary
Integrační vrstva
Zabezpečená komunikační vrstva
Služby nadnárodní interoperability
Standardy interoperability
IS poskytovatelů individuální zdravotní péče a plátců
NIS
LIS
RIS
Systémy zdravotních pojišťoven
Ambulantní IS
IS lékáren
IS poskytovatelů veřejné zdravotní a lékárenské péče
SÚKL
SZÚ
Primární datové zdroje a procesy
Zdravotní a administrativní dokumentace
Procesy zdravotní péče, veřejného zdravotnictví, procesy plátců
Integrovaný systém Hygienické služby
Manažerský informační systém
Spisová služba a systém oběhu dokumentů
Uživatelská pracovní plocha
Vytěžování dat a informací
Hygienické registry
Zdravotnické registry
Administrativní registry
Mapové služby
Podpora uživatelů
Napojení na okolní IS
Jednotná technologická platforma eREG

Architectural Views

- different approaches (models) in the literature
  - **Bass & Clemens & Kazman**
  - **Kruchten's 4+1 view**
  - Rozanski & Woods
  - C4 model
- this lecture is about principles of architectural views
  - not about their graphical notations and documentation
  - may have standardized graphical notation

<xs:complexType name="CategoryType">
<xs:sequence>
<xs:element name="description" type="xs:string" />
<xs:element name="category" type="CategoryType"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="books">
</xs:sequence>
</xs:complexType>

(Bass, Clemens, Kazman)'s Architectural Views

(Bass, Clemens, Kazman)'s Architectural Views

- module viewpoint
  - system structured into code units
- component-and-connector viewpoint
  - system structured into elements with runtime behavior
- allocation viewpoint
  - relationship of the system to non-software structures

Module Viewpoint Definition

- static implementation units of the system and their
- code-based way of considering a system
- **elements** = **modules**
  - implementation unit that provides a coherent set of responsibilities
  - has properties
    - name, ...
- **relationships** = static relationships between modules
  - part of, depends on (uses) and is a

Why Module Viewpoint

- What is the functional responsibility of a module?
- What other modules a module uses?
- What modules are related to a module by part-of or generalization relationships?

Why Module Viewpoint

- Shows decomposition of code into units.
- Determines assumptions each unit can make about other units.
- Helps with project planning and work assignments.

Why Module Viewpoint

- Helps to plan incremental development.
- Improves software reuse.

Why Module Viewpoint

- Allows for change-impact analysis.
- Allows for requirements traceability analysis.

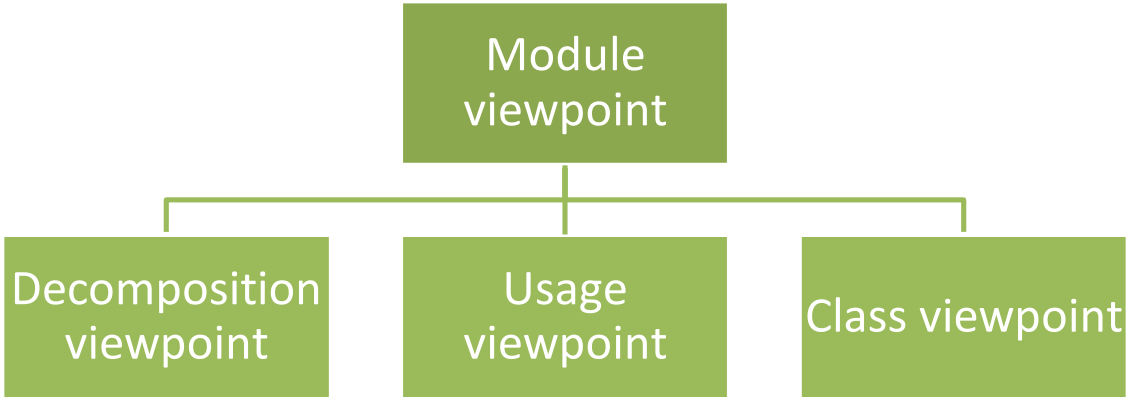
Why Module Viewpoint

- Helps to communicate the functionality of a system and the structure of the code.

Module Properties

- Name
- Visibility
- Functional responsibilities
- Implementation details
  - mapping to source code, testing information, management information, restrictions
- Revision history

Specific Kinds of Module Viewpoint



Decomposition Viewpoint

- modules are logically decomposed to smaller ones until they are small enough
- modules related by “is a sub-module of” relationship

Module in Decomposition Viewpoint

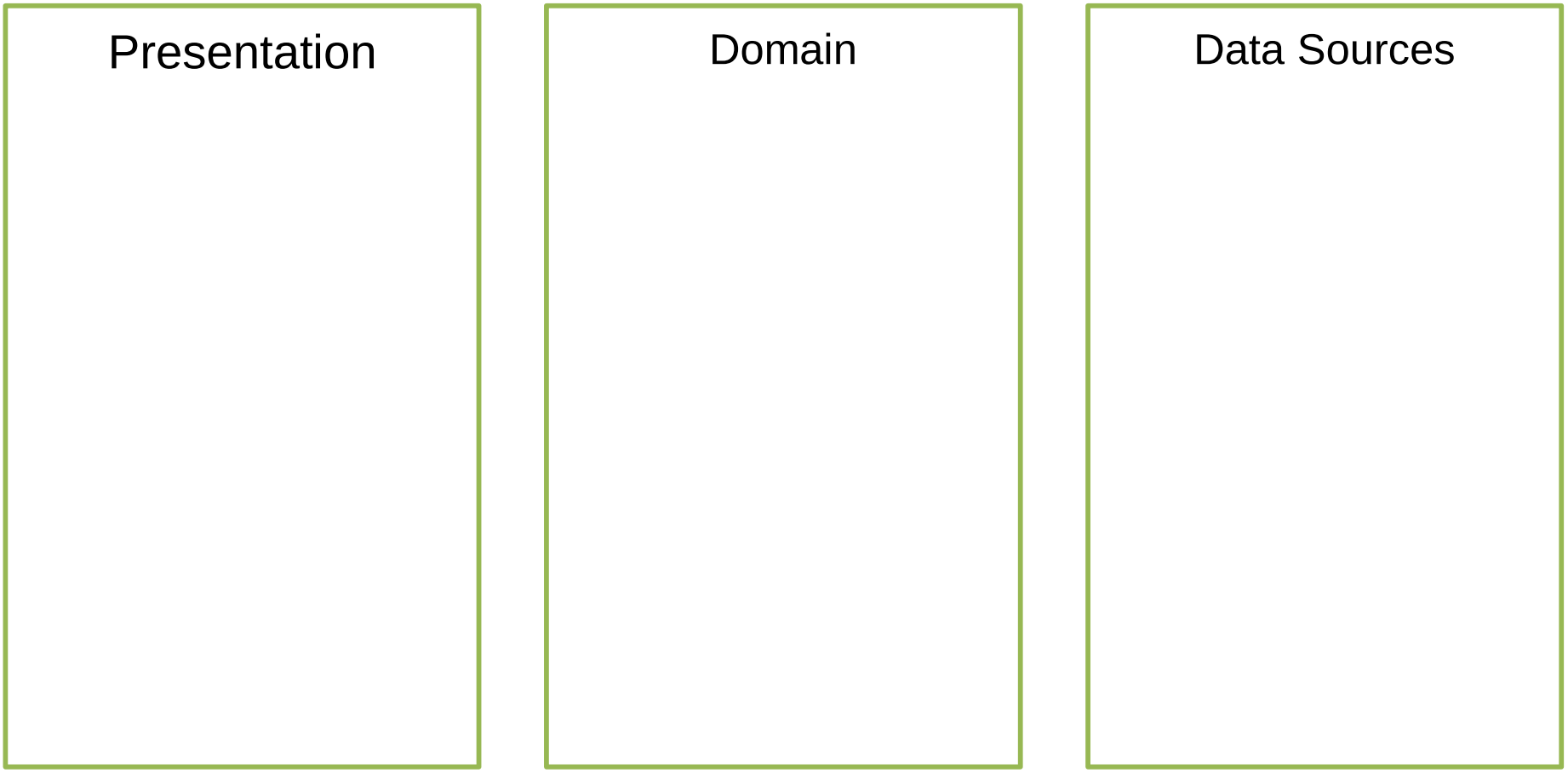
- represents a common starting point for the design
- has associated various products
- has its interface and secrets (information hiding)

Why Decomposition Viewpoint

- to divide and conquer
- to support system's modifiability
- to support project managers

National Open Data Catalog

National Open Data Catalog



National Open Data Catalog

Presentation

Domain

Services

Model

Data Sources

National Open Data Catalog

Presentation

Domain

Services

Model

Dataset

Distribution

Data Sources

National Open Data Catalog

Presentation

Domain

Services

- Search Datasets
- Get Dataset Detail

Model

- Dataset
- Distribution

Data Sources

National Open Data Catalog

Presentation

Domain

Services

- Search Datasets
- Get Dataset Detail

Model

- Dataset
- Distribution

Data Sources

- Dataset Source

National Open Data Catalog

Presentation

Domain

Services

- Search Datasets
- Get Dataset Detail

Model

- Dataset
- Distribution

Data Sources

Dataset Source

- Dataset Index
- Dataset Gateway

National Open Data Catalog

Presentation

Web application

Public API

Domain

Services

Search Datasets

Get Dataset Detail

Model

Dataset

Distribution

Data Sources

Dataset Source

Dataset Index

Dataset Gateway

National Open Data Catalog

Presentation

Web application

Public API

- Dataset List
- Dataset Detail

Domain

Services

- Search Datasets
- Get Dataset Detail

Model

- Dataset
- Distribution

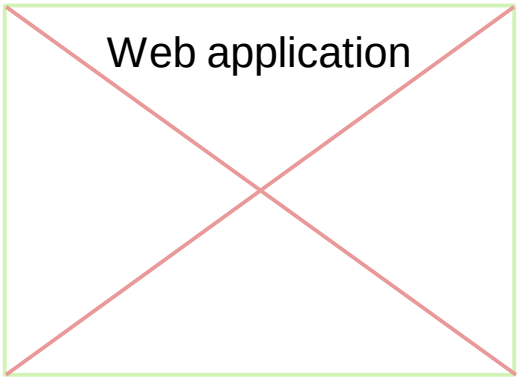
Data Sources

Dataset Source

- Dataset Index
- Dataset Gateway

National Open Data Catalog

Presentation



Public API

- Dataset List
- Dataset Detail

Domain

Services

- Search Datasets
- Get Dataset Detail

Model

- Dataset
- Distribution

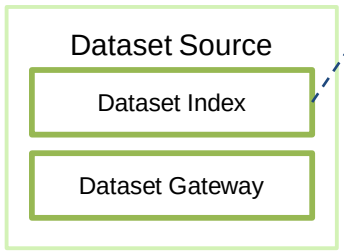
Data Sources

Dataset Source

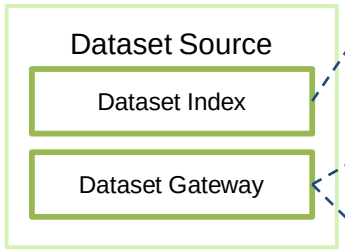
- Dataset Index
- Dataset Gateway

Modules and Source Code

```
class DatasetIndex {
  public List<DatasetValueObject> find(String namePattern, ...) {
    …
  }
  …
}
```



Modules and Source Code

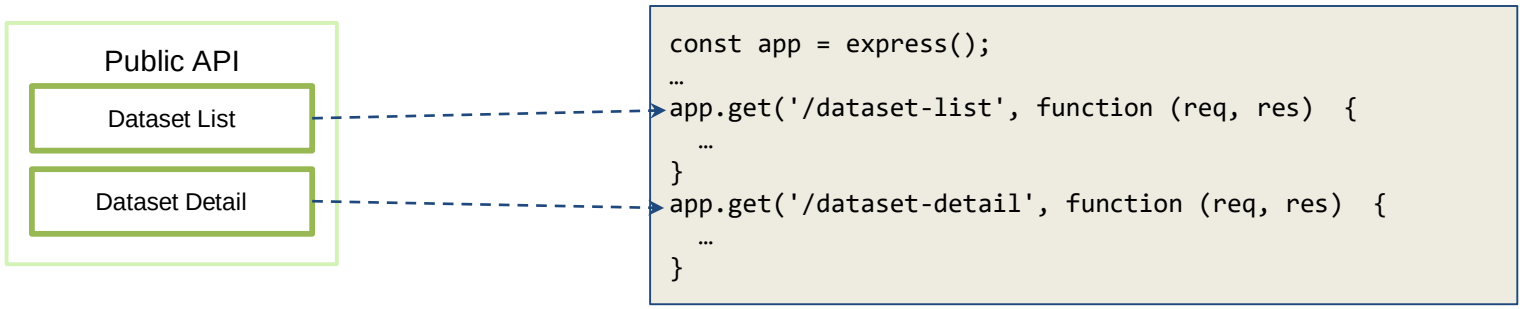



```
class DatasetIndex {
  public List<DatasetValueObject> find(String pattern) {
    …
  }
  …
}
```

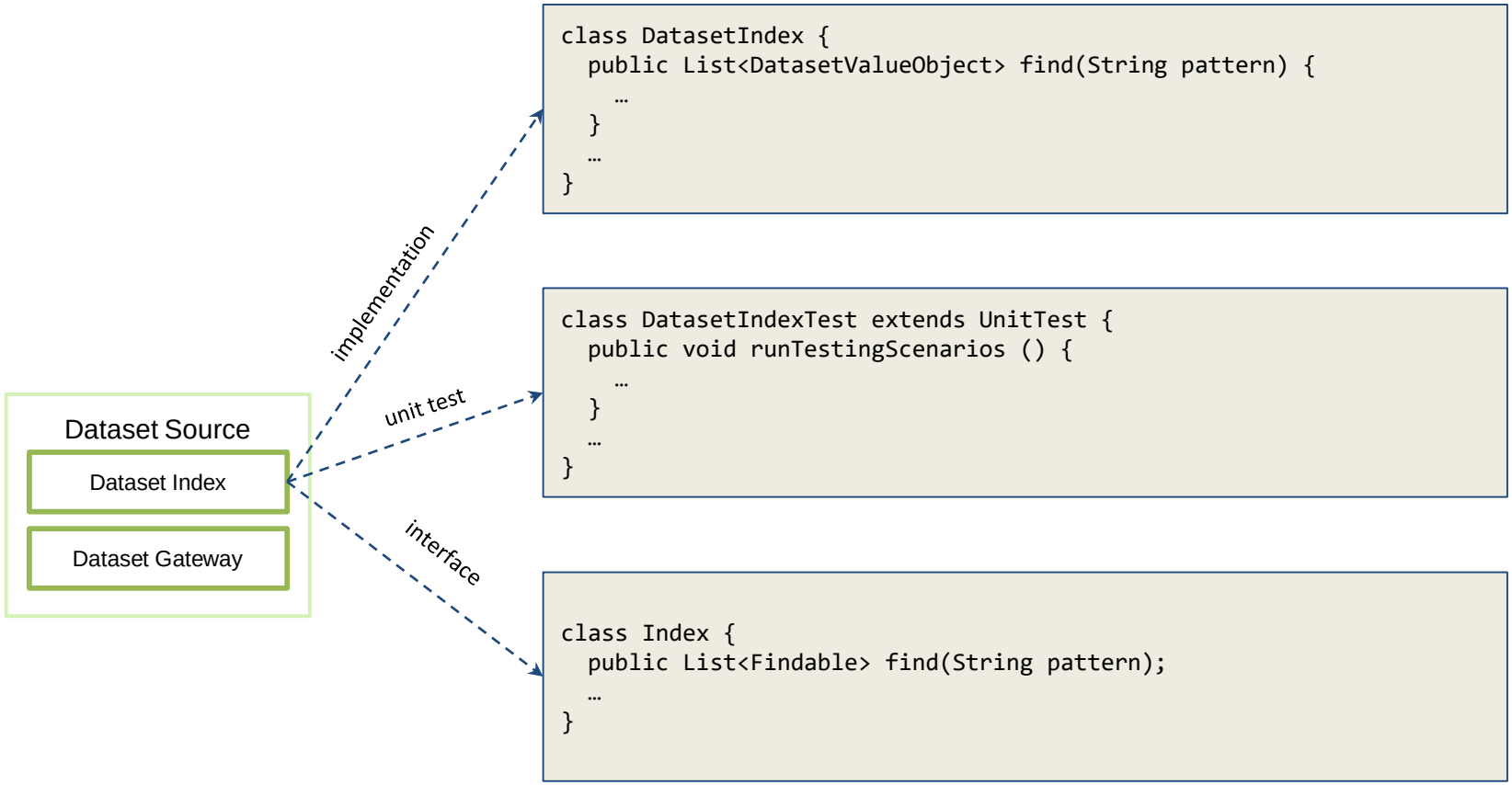
```
class DatasetGateway {
  public static DatasetValueObject load (IRI dsIRI) {
    …
  }
  public static void insert (DatasetValueObject ds) {
    …
  }
  …
}
```

```
class DatasetValueObject {
  private String name ;
  private List<DistributionValueObject> distributions ;
  public String getName() { … }
  …
}
class DistributionValueObject {
  private String name ;
  private IRI downloadURL ;
  public String getName() { … }
  …
}
```

Modules and Source Code



Modules and Source Code



Usage Viewpoint

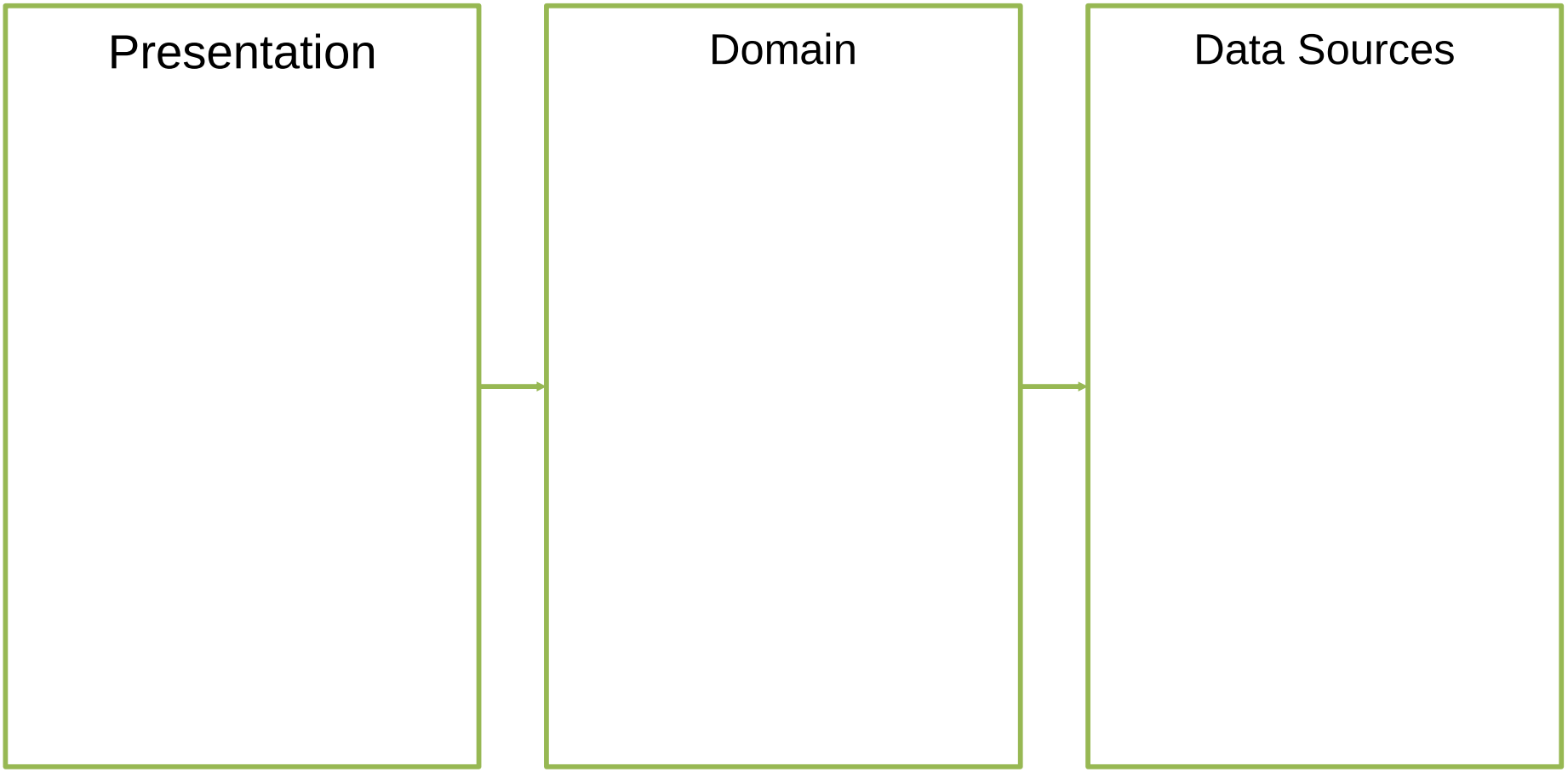
- how modules use other modules

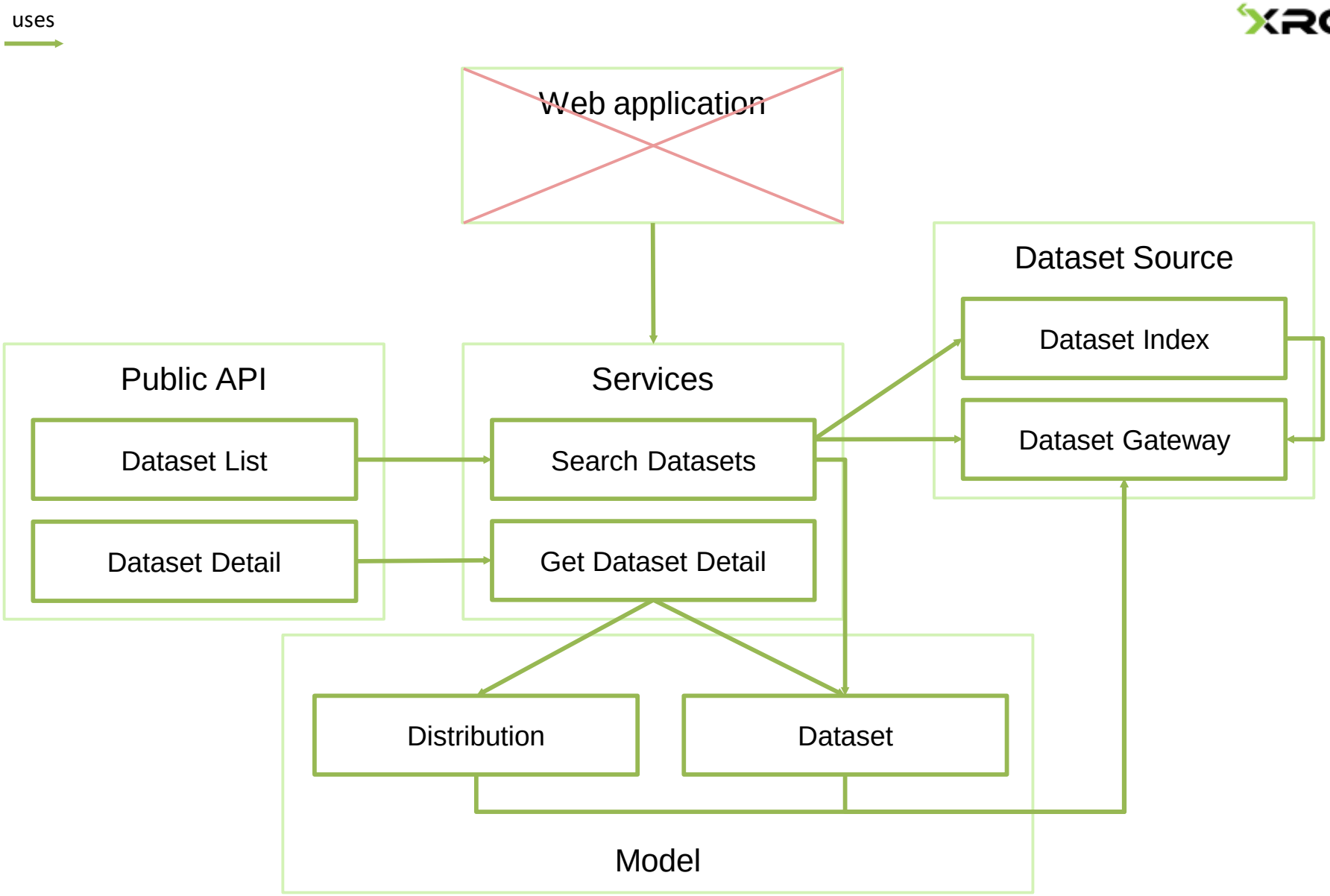
"A module uses another if the correctness of the first requires the presence of a correct version of the other."

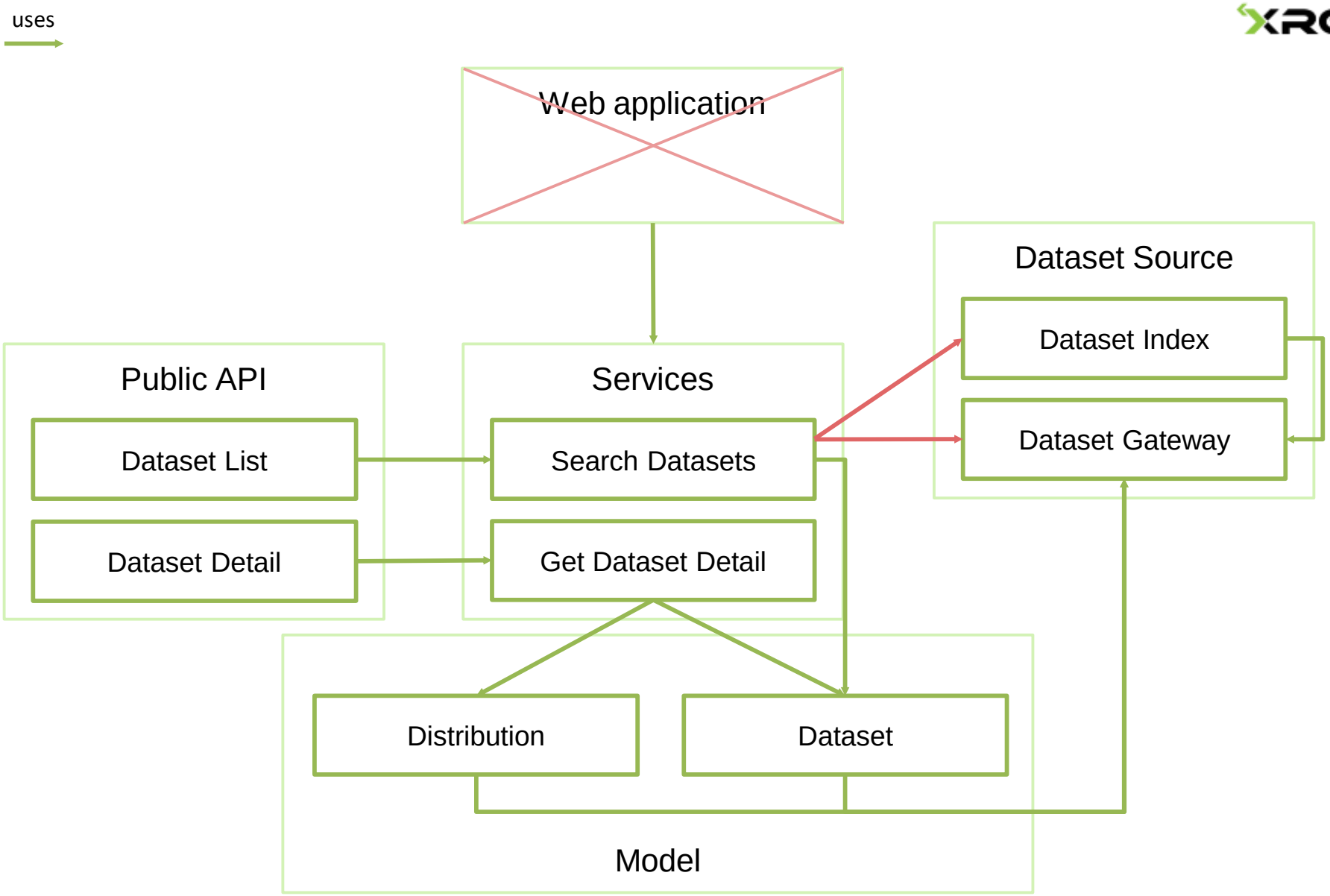
Why Usage Viewpoint

- development planning
- team allocation
- system extensibility and modifiability
- software reuse

National Open Data Catalog




uses
Web application
Dataset Source
Dataset Index
Dataset Gateway
Public API
Dataset List
Dataset Detail
Services
Search Datasets
Get Dataset Detail
Distribution
Dataset
Model

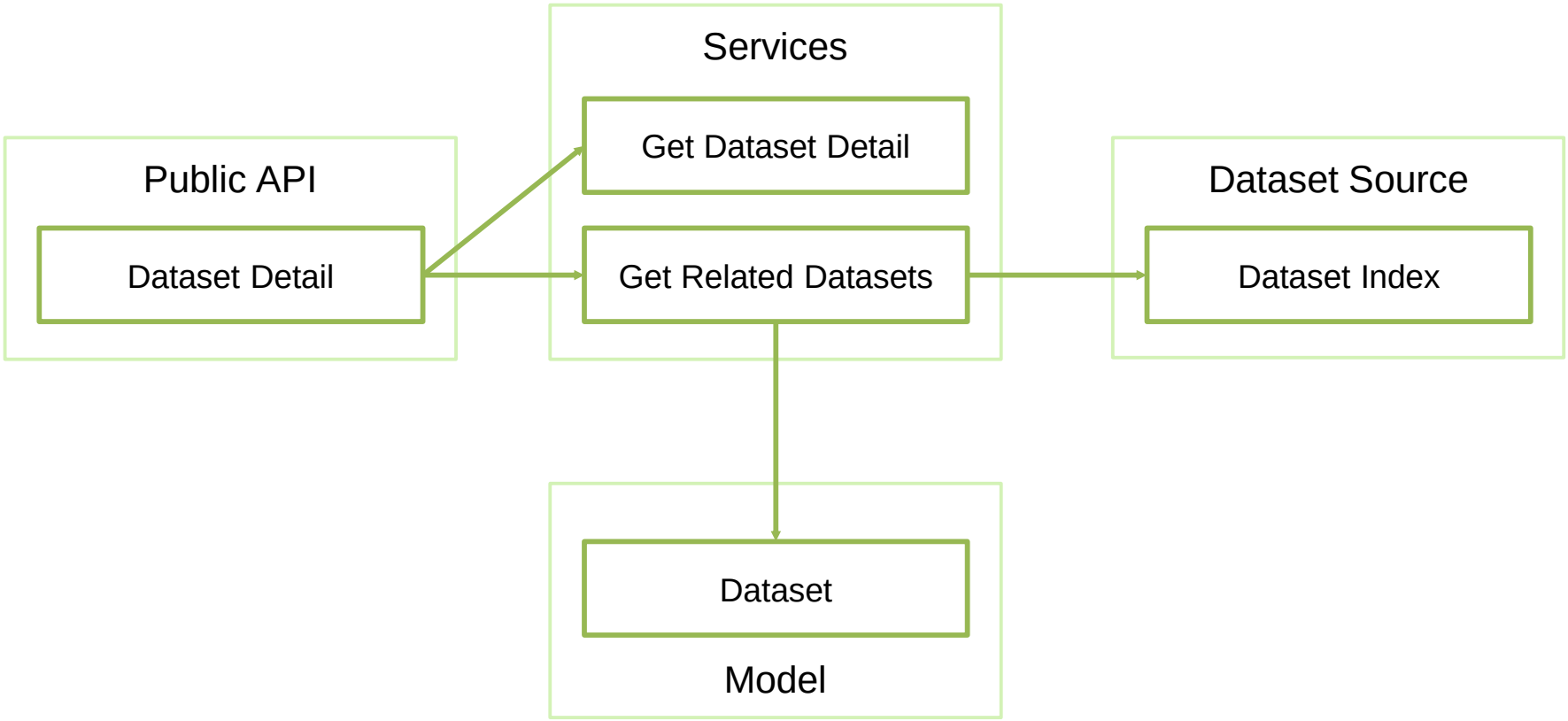

uses
Web application
Dataset Source
Dataset Index
Dataset Gateway
Public API
Dataset List
Dataset Detail
Services
Search Datasets
Get Dataset Detail
Distribution
Dataset
Model

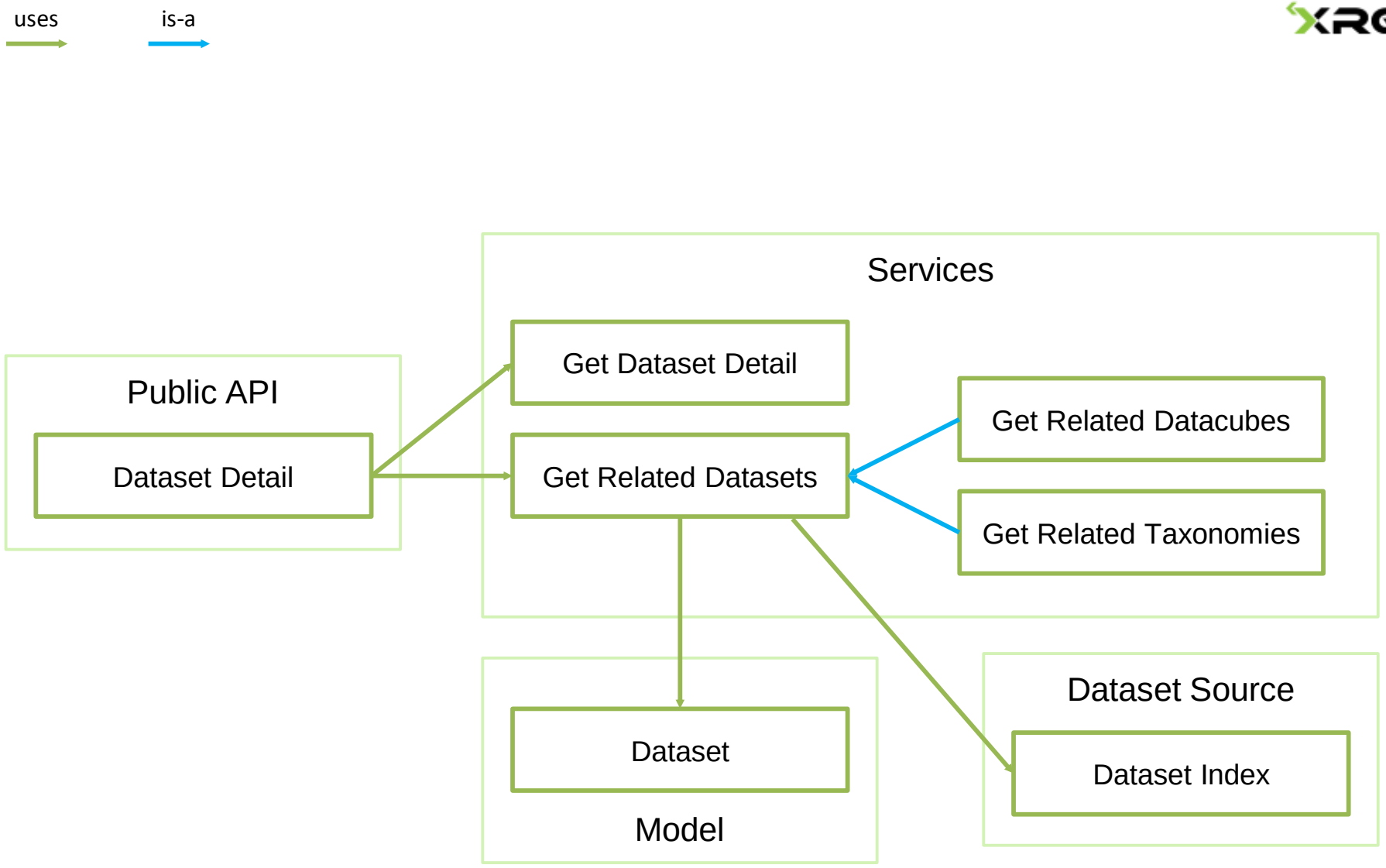
Usage Viewpoint

- in addition to “uses” we can also consider “is-allowed-to-use”, “depends” and “calls”
  - *calls* : A is required to call B but A does not necessarily further depend on what B does
  - *uses* : B produces its result (either when called by A or independently of being called by A) and A uses the result
  - *is-allowed-to-use* : B produces its result (either when called by A or independently of being called by A) and A may use the result
  - *depends on* : A may run only when B performs its functionality independently of being called by A but it does not necessarily use the result of B

Class Viewpoint

- Are there any similarities in behavior or capabilities of modules which can be captured by sub-classing?
- "is-a" relationship
- allows us to reason about re-use and incremental addition of functionality


Services
Get Dataset Detail
Public API
Dataset Detail
Get Related Datasets
Dataset Source
Dataset Index
Dataset
Model


uses
is-a
Services
Public API
Get Dataset Detail
Dataset Detail
Get Related Datasets
Get Related Datacubes
Get Related Taxonomies
Dataset
Model
Dataset Source
Dataset Index

The End

Check out https://www.ksi.mff.cuni.cz/ for amazing student project topics and bachelor's, master's and doctoral theses.