Software System Architectures (NSWI130)
C4 model

- Martin Nečaský, Ph.D.
- Department of Software Engineering
- Faculty of Mathematics and Physics
- Charles University in Prague

C4 model introduction

Structurizr environment

Project in C4 model

Quality attributes and C4 model

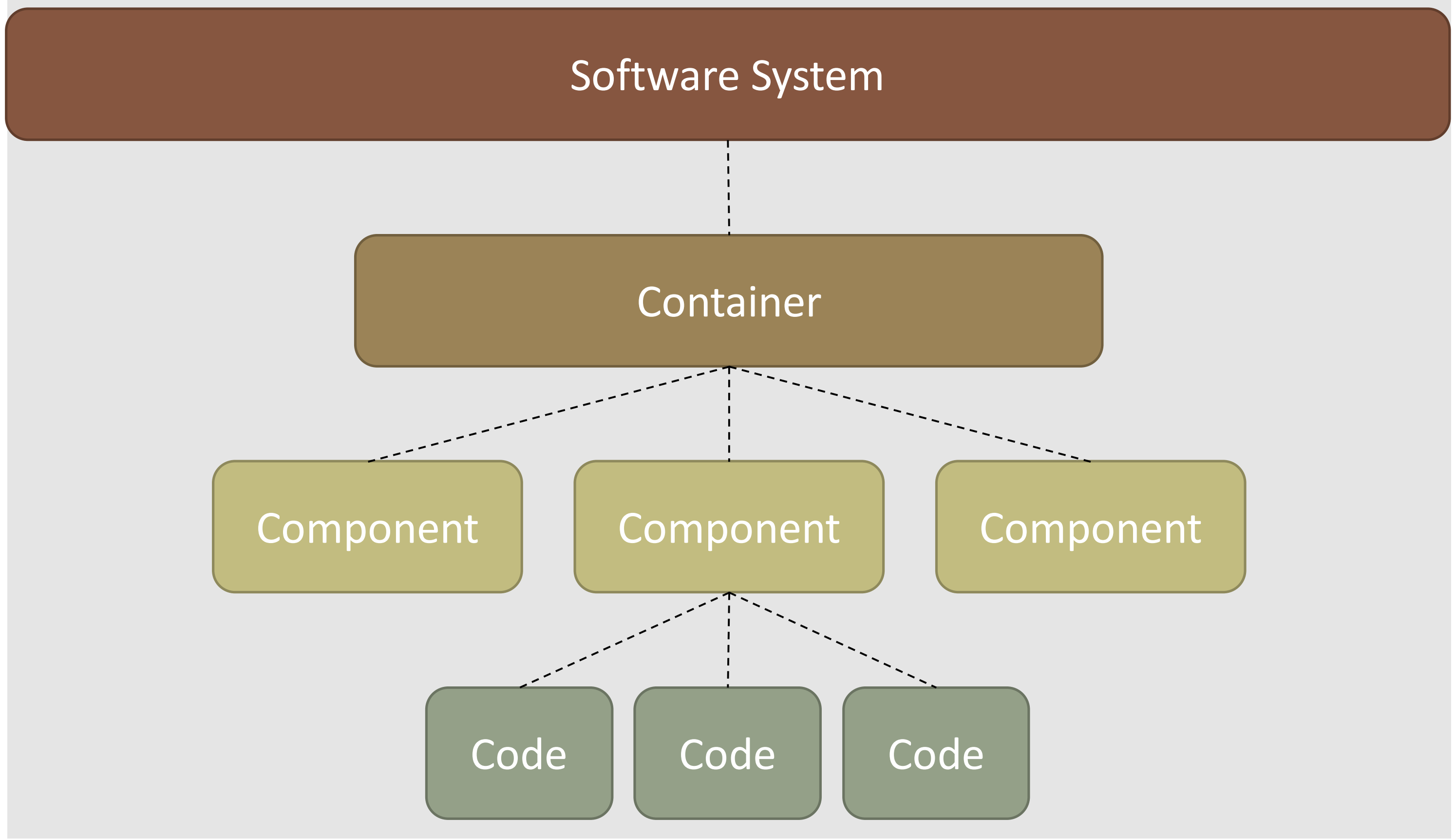
What is C4 model?

- A lightweight model for describing software architectures
- The primary interest is the static structure (module viewpoint from Bass, Clements, Kazman (BCK))
- https://c4model.com/

C4 model terminology

- A **software system** is made up of one or more **containers** (web, mobile and desktop applications, databases, file systems, …).
- A container contains one or more **components**, which in turn are implemented by one or more **code** elements (e.g., classes, interfaces, objects, functions, …).
- And **people** use the software systems that we build.

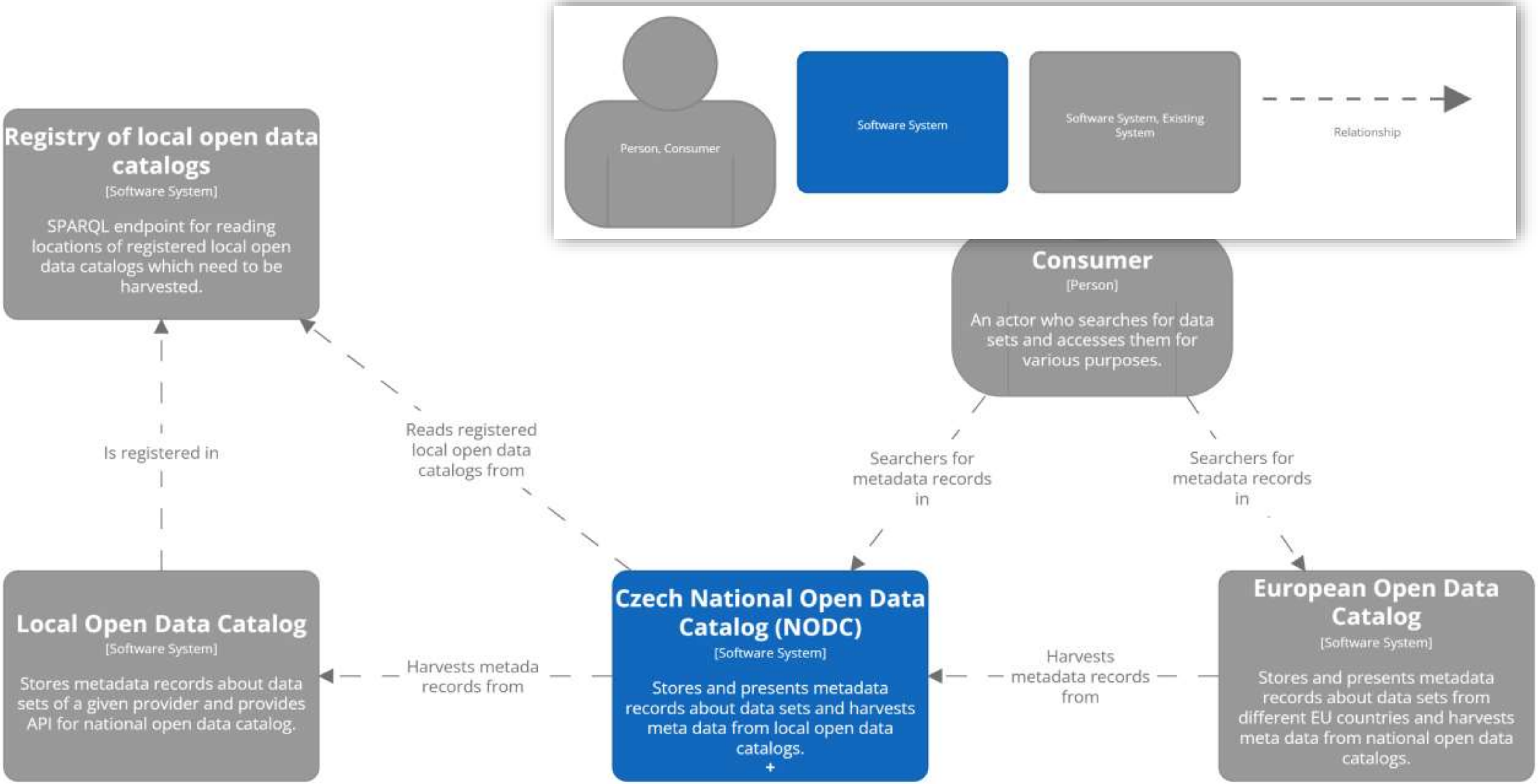
Abstraction levels in C4 model



- Level 1 : Software System
- Level 2 : Container
- Level 3 : Component
- Level 4 : Code

Level 1 : Software System

- highest level of abstraction
- delivers value to its users (human or not)
- presented with **system context diagrams**



Person, Consumer

Software System

Software System, Existing System

Relationship

Registry of local open data catalogs

[Software System]

SPARQL endpoint for reading locations of registered local open data catalogs which need to be harvested.

Consumer

[Person]

An actor who searches for data sets and accesses them for various purposes.

Is registered in

Reads registered local open data catalogs from

Searchers for metadata records in

Searchers for metadata records in

Local Open Data Catalog

[Software System]

Stores metadata records about data sets of a given provider and provides API for national open data catalog.

Harvests metada records from

Czech National Open Data Catalog (NODC)

[Software System]

Stores and presents metadata records about data sets and harvests meta data from local open data catalogs.

+

Harvests metadata records from

European Open Data Catalog

[Software System]

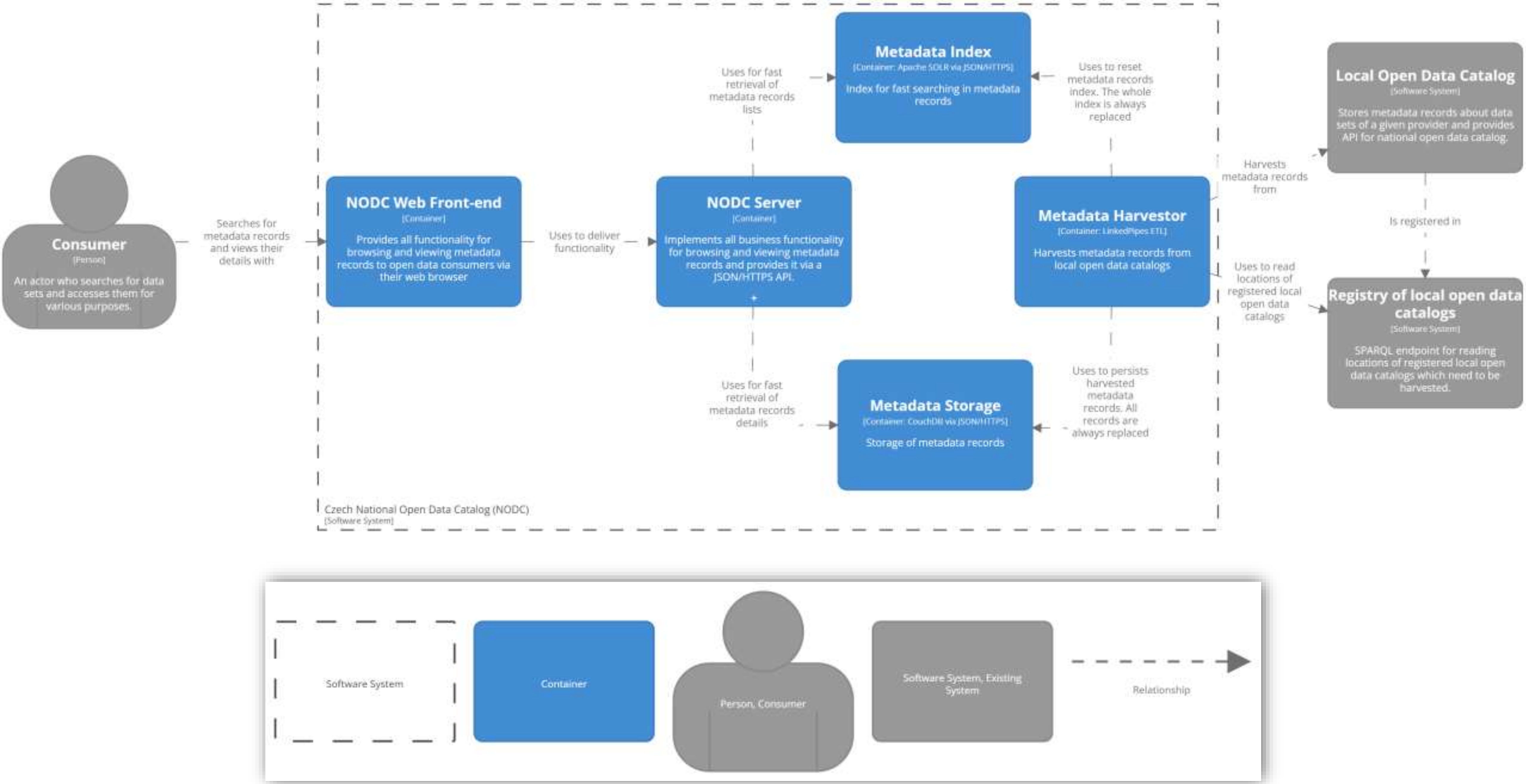
Stores and presents metadata records about data sets from different EU countries and harvests meta data from national open data catalogs.

Level 2 : Container

- represents a context inside which some code is executed (application, service, script, …)
- or some data is stored (database, file system, …)

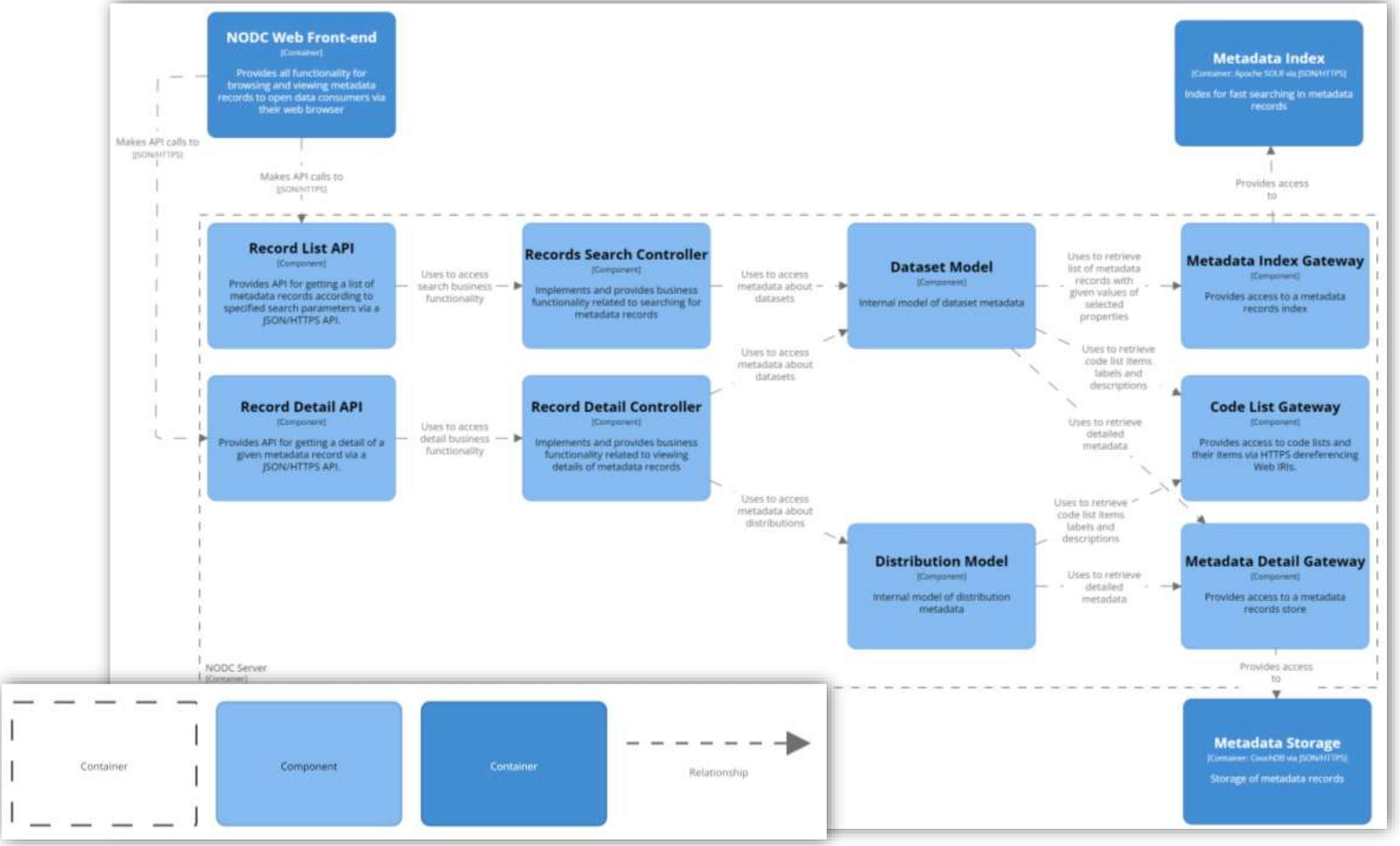
Level 2 : Container

- needs to run but we consider it from the static point of view
  - we are not interested in its runtime behavior
- separately deployable
- presented with **container diagrams**


Consumer
[Person]
An actor who searches for data sets and accesses them for various purposes.
Searches for metadata records and views their details with
NODC Web Front-end
[Container]
Provides all functionality for browsing and viewing metadata records to open data consumers via their web browser
Uses to deliver functionality
NODC Server
[Container]
Implements all business functionality for browsing and viewing metadata records and provides it via a JSON/HTTPS API.
Uses for fast retrieval of metadata records lists
Metadata Index
[Container: Apache SOLR via JSON/HTTPS]
Index for fast searching in metadata records
Uses to reset metadata records index. The whole index is always replaced
Metadata Harvestor
[Container: LinkedPipes ETL]
Harvests metadata records from local open data catalogs
Uses for fast retrieval of metadata records details
Metadata Storage
[Container: CouchDB via JSON/HTTPS]
Storage of metadata records
Uses to persists harvested metadata records. All records are always replaced
Czech National Open Data Catalog (NODC)
[Software System]
Harvests metadata records from
Local Open Data Catalog
[Software System]
Stores metadata records about data sets of a given provider and provides API for national open data catalog.
Is registered in
Uses to read locations of registered local open data catalogs
Registry of local open data catalogs
[Software System]
SPARQL endpoint for reading locations of registered local open data catalogs which need to be harvested.
Software System
Container
Person, Consumer
Software System, Existing System
Relationship

Level 3 : Component

- represents a grouping of related functionality encapsulated behind a well-defined interface
- corresponds to module from BCK model but it is not further decomposed to submodules
- presented with **component diagrams**



NODC Web Front-end
[Container]
Provides all functionality for browsing and viewing metadata records to open data consumers via their web browser

Metadata Index
[Container: Apache SOLR via JSON/HTTPS]
Index for fast searching in metadata records

Makes API calls to
[JSON/HTTPS]

Makes API calls to
[JSON/HTTPS]

Provides access to

Record List API
[Component]
Provides API for getting a list of metadata records according to specified search parameters via a JSON/HTTPS API.

Uses to access search business functionality

Records Search Controller
[Component]
Implements and provides business functionality related to searching for metadata records

Uses to access metadata about datasets

Dataset Model
[Component]
Internal model of dataset metadata

Uses to retrieve list of metadata records with given values of selected properties

Metadata Index Gateway
[Component]
Provides access to a metadata records index

Uses to access metadata about datasets

Uses to retrieve code list items labels and descriptions

Record Detail API
[Component]
Provides API for getting a detail of a given metadata record via a JSON/HTTPS API.

Uses to access detail business functionality

Record Detail Controller
[Component]
Implements and provides business functionality related to viewing details of metadata records

Uses to retrieve detailed metadata

Code List Gateway
[Component]
Provides access to code lists and their items via HTTPS dereferencing Web IRIs.

Uses to access metadata about distributions

Uses to retrieve code list items labels and descriptions

Distribution Model
[Component]
Internal model of distribution metadata

Uses to retrieve detailed metadata

Metadata Detail Gateway
[Component]
Provides access to a metadata records store

NODC Server
[Container]

Provides access to

Metadata Storage
[Container: CouchDB via JSON/HTTPS]
Storage of metadata records

Container

Component

Container

Relationship

Level 4 : Code

- code elements forming components
- basic building blocks of programming languages, e.g., classes, interfaces, functions, …
- presented with UML class diagrams

Structurizr

Web-based rendering tool for software development teams to design and document software architecture with C4 model

Structurizr DSL for creating architectural diagrams as code from a single model (C4 model system context, container and component diagrams)

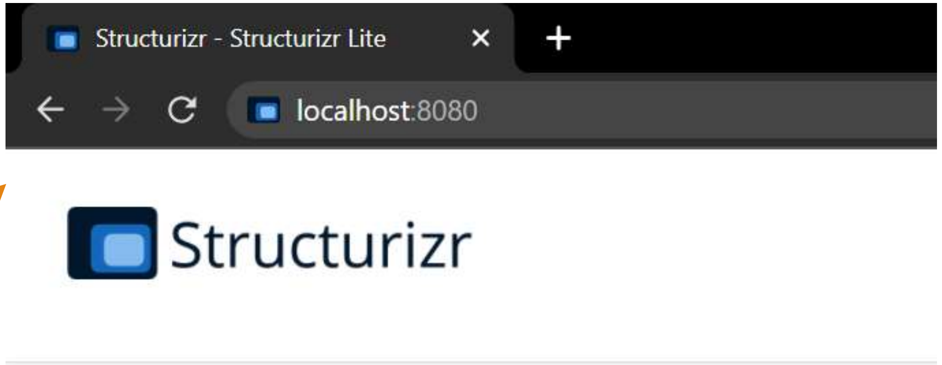
Markdown/AsciiDoc for documentation

Structurizr Lite

- Packaged as a Docker container
- Displays diagrams in a workspace stored in workspace.dsl or workspace.json in a directory on WORKSPACE_PATH (replace WORKSPACE_PATH with your local directory path)

```
docker pull structurizr/lite

docker run -it --rm -p 8080:8080 -v
WORKSPACE_PATH:/usr/local/structurizr structurizr/lite
```



workspace.dsl

- Main file of your Structurizr project
- Sample project from the lecture