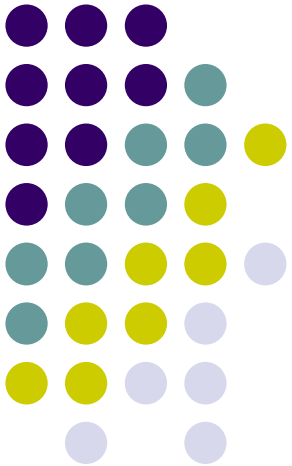
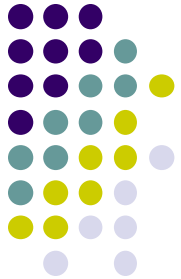
FOX path …



… navigation of physical, virtual and literal file systems

Hans-Jürgen Rennau, Traveltainment GmbH
Presented at xmlprague 2017, February 11, 2016

Trees! Trees! Trees!



Tree-structured information is ubiquitous.

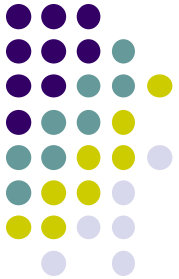
- Resource contents

  (XML, HTML, JSON, CSV, .properties, ...)

- Resource systems

  (file systems, .jar, .zip, SVN, ...)

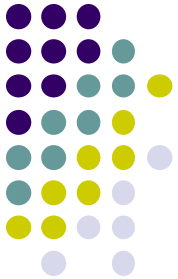
XPath!



- Expression language
- Powerful, readable, elegant ...
- *File content navigation* (XML, JSON, ...)

```
//route/arrival[airport = 'JFK']
/../departure[@t > '18:00']/airport
```

FOXpath!

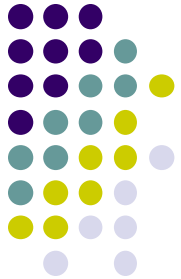


- Expression language
- Powerful, readable, elegant ...
- *File system navigation*
- *File content navigation* (XML, JSON, ...)

```
/projects/air
//route*.xml[not(ancestor~::blacklisted)]
\\route\arrival[airport = 'JFK']
\..\departure[@t > '18:00']\airport
```

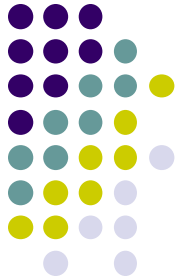
https://github.com/hrennau/foxpath

FOXpath =
superset(*XPath 3.0*)



- XPath 3.0 + URI navigation + Etc
- URI navigation ::=
  - URI axis step — UriAxis~::Name[Expr]
  - URI path operator — Expr1 / Expr2
- Etc
  - Semantic extensions (values instead of errors)
  - Global and external variables
  - Namespace declarations
  - Simple FLWOR (any mixture of `let` and `for`)

FOXpath examples



```
/github/oxygenxml/userguide//*.dita => count()
```

1984

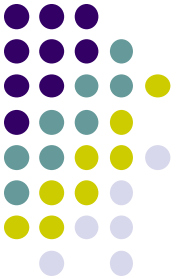
```
/github/oxygenxml/userguide
//*.dita[not(ancestor~::not_used)] => count()
```

1746

```
/github/oxygenxml/userguide
//*.dita[not(ancestor~::not_used)]
\*\local-name() => frequencies(15)
```

```
concept          (53)
glossentry       (9)
task             (160)
topic            (1524)
```

FOXpath examples



```
let $names :=
    /github/oxygenxml/userguide
    //*.dita[not(ancestor~::not_used)]
    /file-name(.)

let $refNames :=
    /github/oxygenxml/userguide
    //(*.dita, *.ditamap)
    \\@href\replace(., '.*/|#.*', '')
return
    count($names except $refNames)
```

43

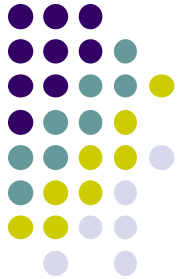
Generalization: challenge

Physical FS navigation (2016)

Logical FS navigation (2017)

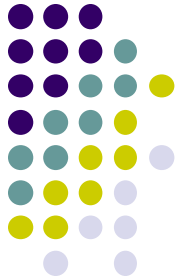
„To build a conceptual framework relating the abstract URI navigation defined by the FOXpath language to an implementation-defined set of logical file system types.“

Generalization: key concepts



- Logical file system
- URI relationships
- URI operations
- URI dispatchal

Logical file system



- Physical file systems (OS-defined)
- Virtual file systems
  - Archives (zip, jar, ...)
  - Document databases (BaseX, ...)
  - Version control repositories (SVN, ...)
  - Github project repository (https://github.com)
- Literal file systems
  - UTREE
  - UGRAPH

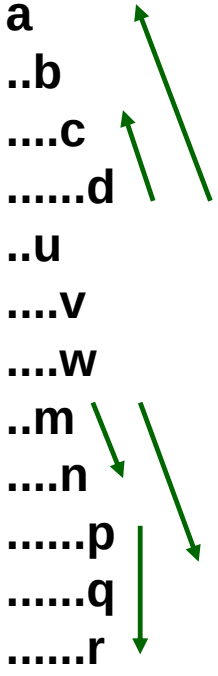
URI relationships

- URI: foo://a/b/c
- Child URI: foo://a/b/c/d
- Parent URI: foo://a/b

URI relationships

URI tree

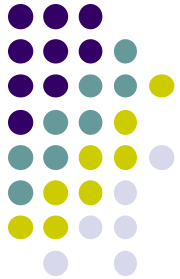
URI axes



(collection of path-structured URIs with common prefix)

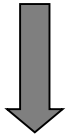
(child, descendant, parent, ancestor, self, ...)

URI operations



Impl(navigation) constructed from building blocks:

- Traversal URI => URI*
- Inspection URI => resource properties
- Retrieval URI => file content

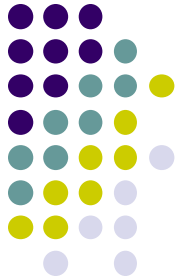


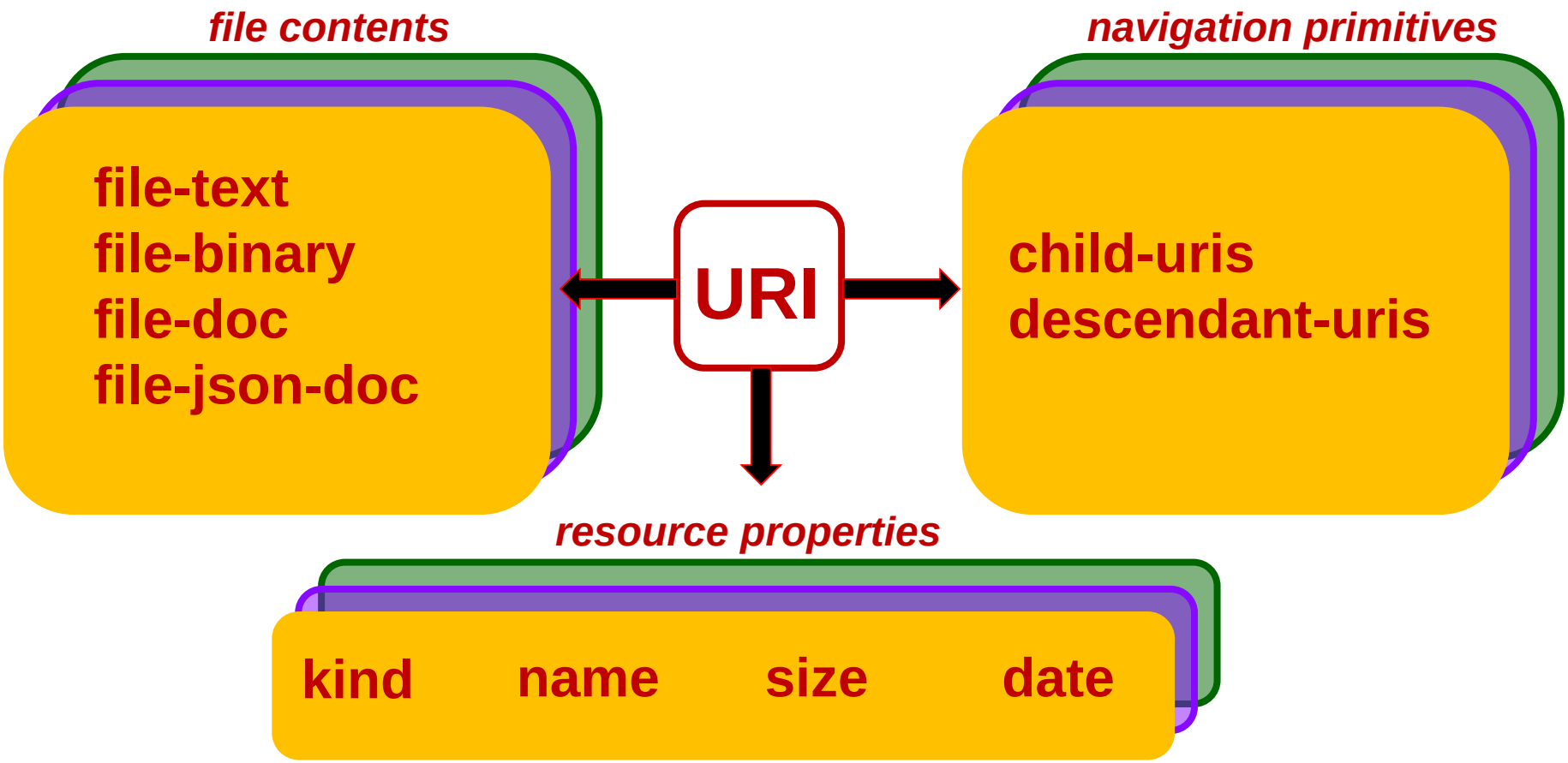
abstraction

URI operations

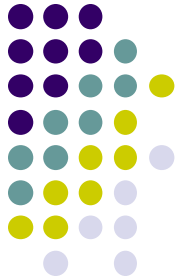
Navigation =
complex functionality
composed of URI operations

URI operations and URI processor



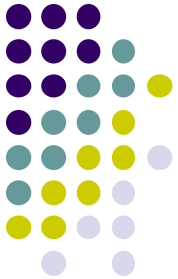


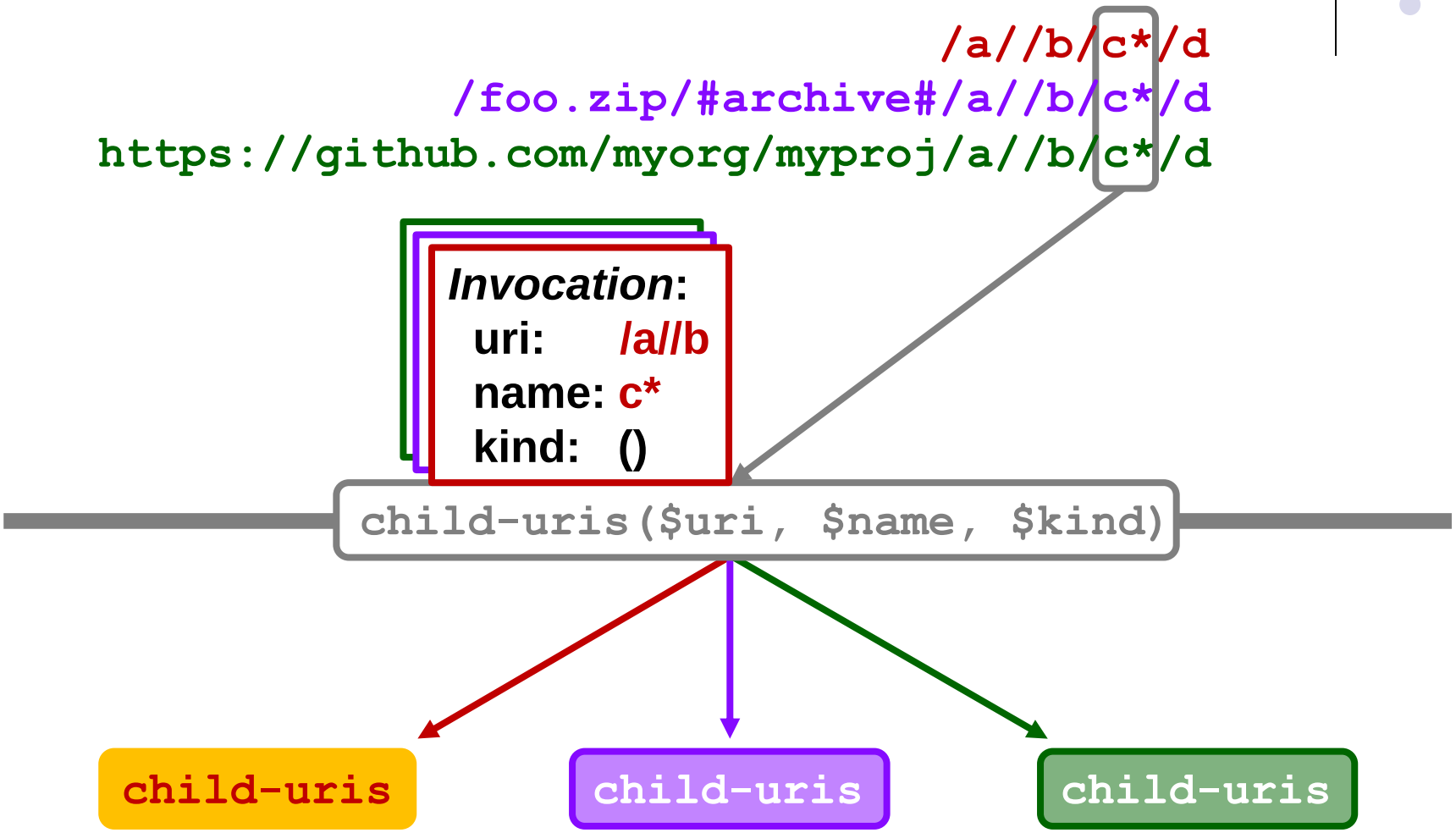
URI navigation – backing APIs

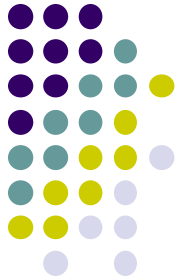


File system type	Backing APIs
Physical file system	XQuery module: file (EXPath)
Archives	XQuery module: file (EXPath) XQuery module: archive (BaseX)
SVN	*SVN Command Line Interface* XQuery module: proc (BaseX)
Github projects	*Github REST API* XQuery module: http (EXPath) XQuery module: convert (BaseX)
BaseX databases	XQuery module: db (BaseX)

URI operations = layer of abstraction



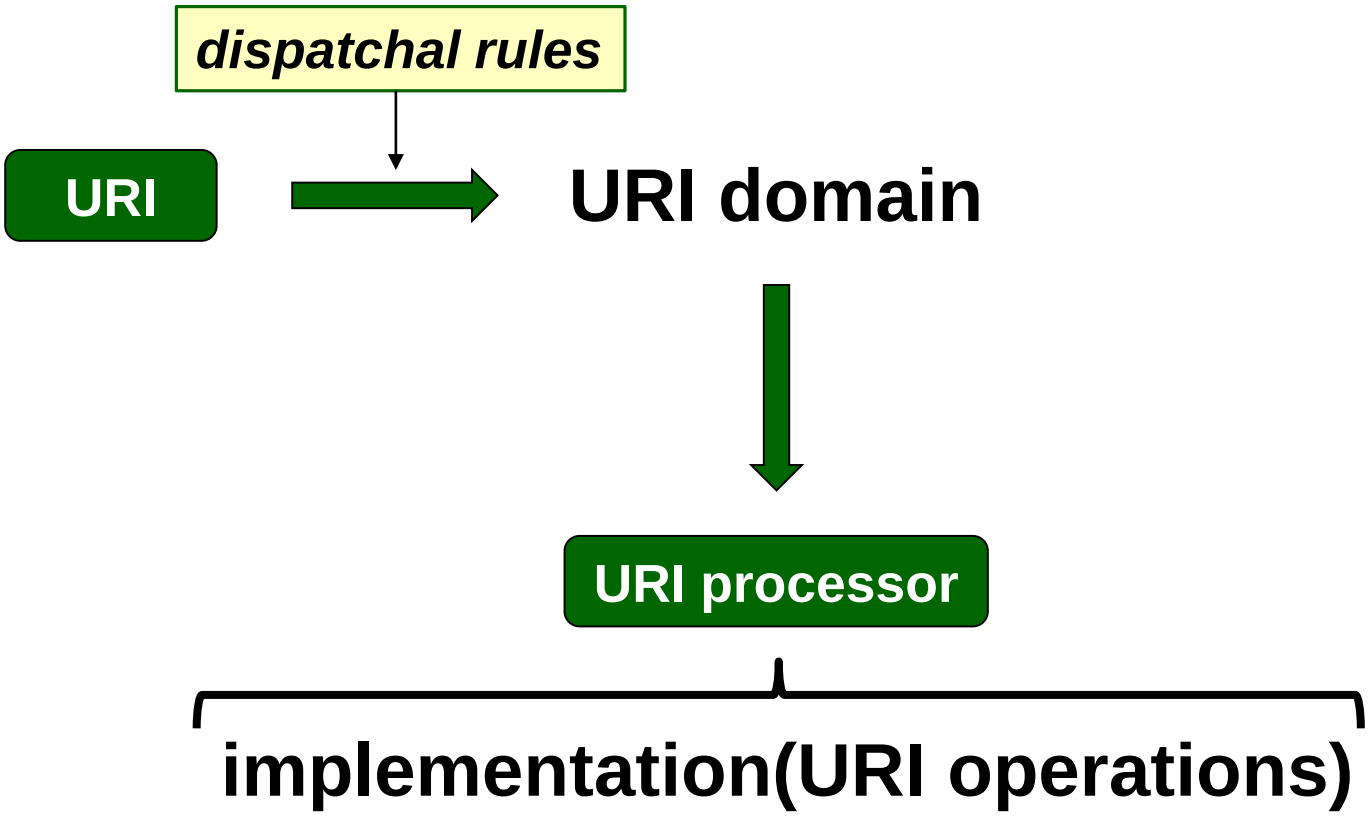




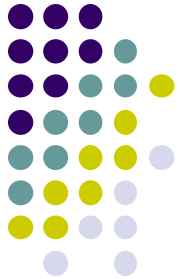
Think of *URI operations* as a generic interface exposed by a logical file system of any type.

***URI dispatchal* binds the generic interface to a specific implementation.**

URI dispatchal

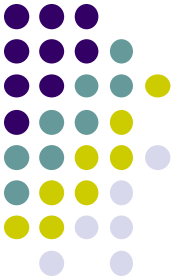


URI dispatchal rules



URI matches ...	URI domain
.../#archive#/...	ARCHIVE
(one of a set of configured URI prefixes)	UGRAPH
(one of a set of configured URI prefixes)	UTREE
basex://...	BASEX
svn-...	SVN
https://github.com/...	GITHUB
file://...	FILE
/...	FILE
*	NONE

Examples: navigation of virtual file systems



```
/downloads/userguide.zip/#archive#
//*.jar/#archive#
//*.xsl
\\@match => distinct-values() => sort()
```

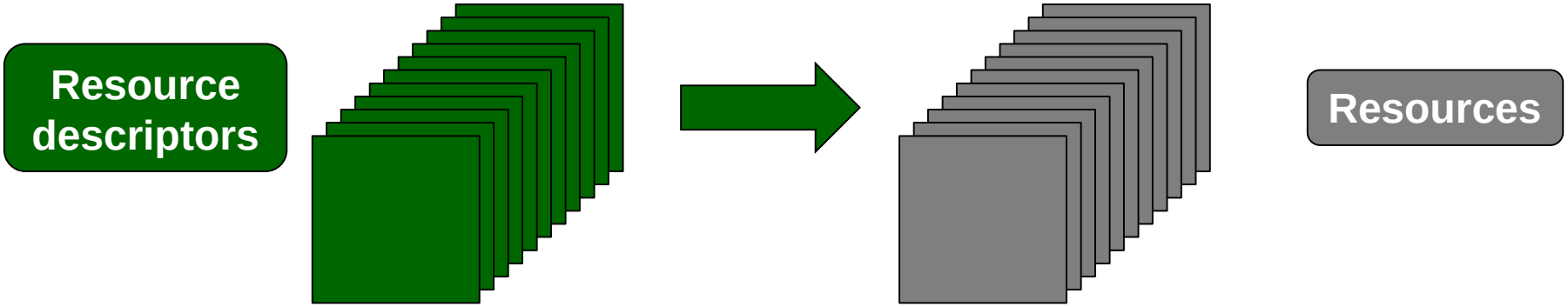
```
*:emph
*:title
*|text()|@*
sch:pattern[@abstract='true']
sch:schema
text()
```

```
svn-https://svn.apache.org/repos/asf/xalan
/java/trunk//*.xsl
\*\@version => frequencies(10)
```

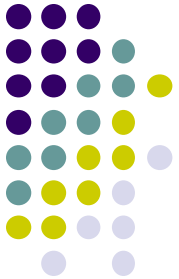
1.0 (53)

Literal file systems

- Collection of resources described by a collection of resource descriptors
  - UTREE description = XML document
  - UGRAPH description = RDF graph



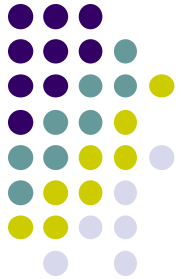
Resource descriptor



- Represents a single folder or file
- Information conveyed:
  - Navigation URI
  - Retrieval URI
  - Date of last modification *(optional)*
  - Size *(optional)*

Logical structure
!=
physical structure

Resource descriptor: UGRAPH



```
fs:file15549 a fs:file ;
   fs:navURI          "https://github.com/
                       marklogic/marklogic-jena/README.md" ;
   fs:accessURI       "https://api.github.com/repos/
                       marklogic/marklogic-jena/git/blobs/
                       21c55a33...3189aae5572a409ebae6ba"  ;
   fs:parentDir       fs:dir15495 ;
   fs:name            "README.md";
   fs:lastModified    "2016-03-25T23:07:55Z" ;
   fs:fileSize        "2401" .
```

/>

RDF triples

lifis tool

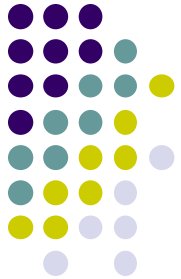
Create UTREE

```
lifis "github?org=oxygenxml,format=utree"
   > /utree/oxygenxml/utree-oxygenxml.xml
```

Create UGRAPH

```
lifis "github?org=oxygenxml,format=ugraph"
   > /ugraph/oxygenxml/ugraph-oxygenxml.ttl
```

Prepare UGRAPH endpoint



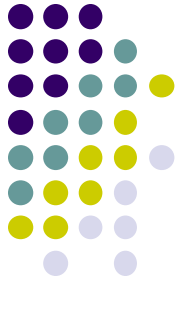
Load UGRAPH into RDF database

```
tdbloader --loc=/tdb/ugraph-oxygenxml
   /ugraph/oxygenxml/ugraph-oxygenxml.ttl
```

Launch UGRAPH endpoint

```
fuseki-server --loc=/tdb/ugraph-oxygenxml
                  /oxygenxml
```

Navigate github via UTREE / UGRAPH



Use UTREE

UTREE folder

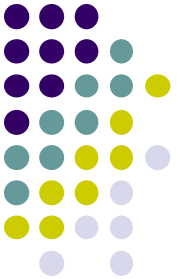
```
fox -t /utree/github/ml
"https://github.com/oxygenxml//*.xsd
 \*\@targetNamespace => frequencies(30)"
```

Use UGRAPH

UGRAPH endpoint

```
fox -g http://localhost:3030/marklogic
"https://github.com/oxygenxml//*.xsd
 \*\@targetNamespace => frequencies(30)"
```

Examples: github navigation (via UTREE | UGRAPH)



```
https://github.com/oxygenxml//*.xsl
/file-lines() => count()
```

83540

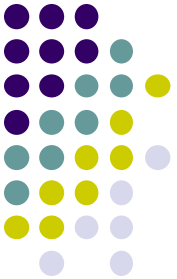
```
https://github.com/oxygenxml/*[.//*.xsl]
=> count()
```

15

```
https://github.com/oxygenxml/*
[.//*.xsl\*\@version\xs:decimal(.) > 2]
/file-name()
```

dita-css
dita-glass

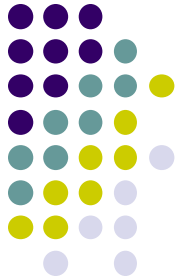
Goliath @ github



```
let $projs := https://github.com/google/*
return (
  '#projects: ' || $projs                  => count() => lpad(9),
  '#files:    ' || $projs//*[is-file()] => count() => lpad(9),
  '#xml:      ' || $projs//*.xml           => count() => lpad(9),
  '#xsd:      ' || $projs//*.xsd*          => count() => lpad(9),
  '#xsl:      ' || $projs//*.xsl*          => count() => lpad(9),
  'xsl versions:',
  $projs//*.xsl*\child::*\@version => frequencies(8)
)
```

```
#projects:       910
#files:       568358
#xml:           8484
#xsd:             80
#xsl:             44
xsl versions:
1.0      (42)
```

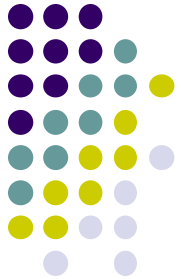
FOXpath – URI navigation scope



- Physical file system
- Virtual file systems
  - Archives (.zip, .jar, .docx, .epub, ...)
  - SVN repo
  - BaseX databases
  - github project repo (via UTREE or UGRAPH)
- Literal file systems
  - UTREE
  - UGRAPH

https://github.com/hrennau/foxpath

FOXpath 2017 - take away



A single navigation model covering
* URI navigation
* Node navigation

Seamless integration of U+N navigation

URI navigation is based on URI operations and URI dispatchal

Scope of supported file system types
implementation-defined and easily extensible

Thank you!