28msec

XQuery Design Patterns

Reusable Solutions for Large-Scale XQuery Applications

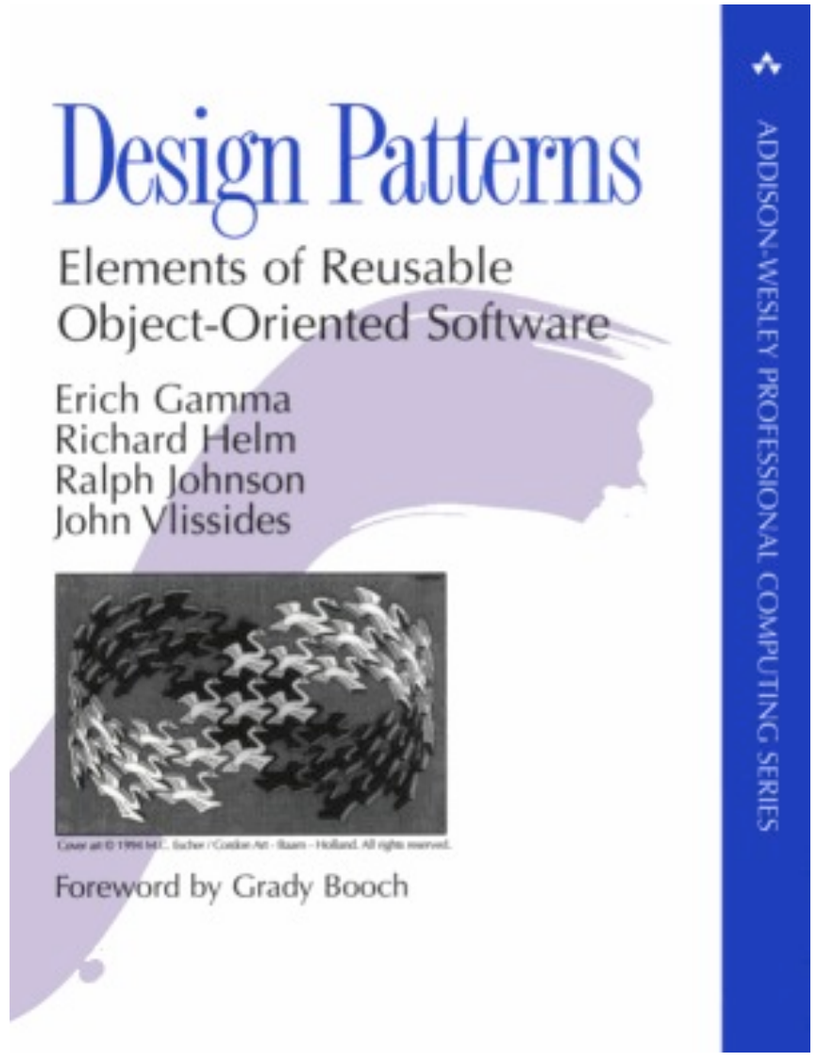
William Candillon {william.candillon@28msec.com}

Balisage 2010

Design Patterns

28msec

- Very mature in the object community
- Reusable Software and Design
- Documentation
- Communication and Teaching



Because Design Patterns bills itself as being concerned with OOP software alone, I fear that software developers outside the object community may ignore it.

Tom DeMarco, IEEE Software, 1996

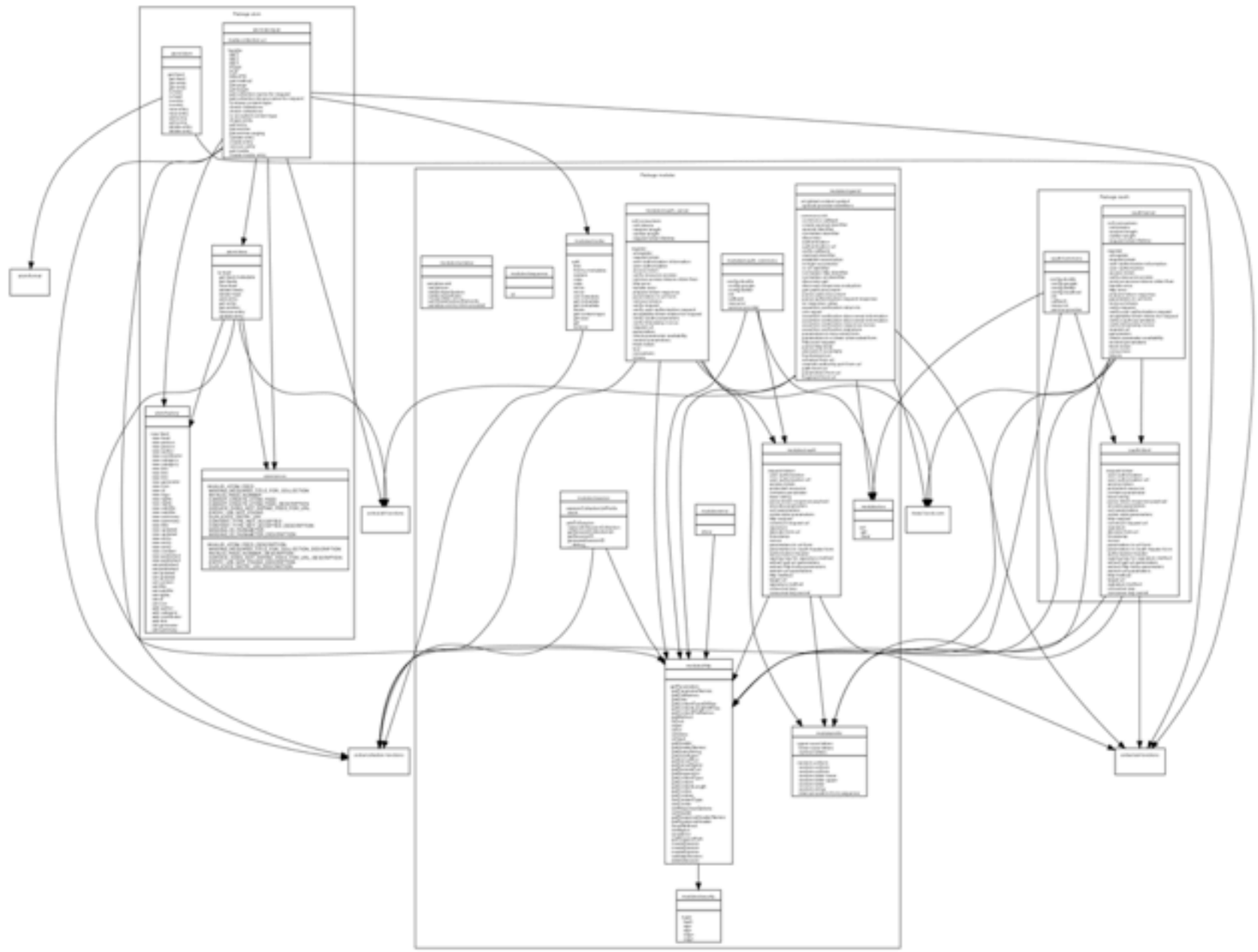
XQuery has grown...

- Update Facility
- Scripting Extension
- Full-Text Support
- Data Definition Facility (Zorba)
- Rich variety of module Libraries (EXPath / EXQuery)

XQuery has grown...

28msec

...so have its applications



An XQuery Application (~15k LOC)

Existing XQuery Application

28msec

- Enterprise Resource Planning application entirely written in XQuery
- Featuring
  - Web Front-end and APIs
  - Workflow Engine
  - Wiki Engine
- 28 000 lines of code
- 135 XQuery modules

Symptoms

28msec

- Strong coupling between modules
- Low extensibility
- Heterogeneous vocabulary

Strong Coupling Between Modules

28msec

```
module namespace oauth = "http://wwww.example.com/modules/oauth/client";

import module namespace io      = "http://www.zorba-xquery.com/modules/readline";
import module namespace fs      = "http://www.zorba-xquery.com/modules/file";
import module namespace xqddf   = "http://www.zorba-xquery.com/modules/xqddf";

import module namespace utils   = "http://www.28msec.com/modules/utils";
import module namespace random  = "http://www.28msec.com/modules/random";
import module namespace cookies = "http://www.28msec.com/modules/http/cookies";
import module namespace http    = "http://www.28msec.com/modules/http";

import module namespace rest    = "http://expath.org/ns/http-client";
```

Low Extensibility

28msec

```
module namespace oauth = "http://wwww.example.com/modules/oauth/client";

module namespace atom = "http://www.example.com/modules/atom/client";

declare function atom:get($feed-uri) { (: ... :) };
declare function atom:post($feed-uri, $entry) { (: ... :) };
declare function atom:put($feed-uri, $entry-uri, $entry) { (: ... :) };
declare function atom:delete($entry-uri) { (: ... :) };
```

Heterogeneous Vocabulary

28msec

Function Summary

	check-params($request as element(https:request)?, $href as xs:string?, $bodies as item()*) as xs:boolean
	create-body($body as element(https:body)) as element(https:body)
	create-multipart($multipart as element(https:multipart)) as element(https:multipart)
	read($href as xs:string) as item()+
	read($request as element(https:request)?, $href as xs:string?) as item()+
	read($request as element(https:request)?, $href as xs:string?, $bodies as item()*) as item()+
S	send-request($request as element(https:request)) as item()+
S	send-request($request as element(https:request)?, $href as xs:string?) as item()+
S	send-request($request as element(https:request)?, $href as xs:string?, $bodies as item()*) as item()+
	set-content-type($request as element(https:request)?) as element(https:request)?

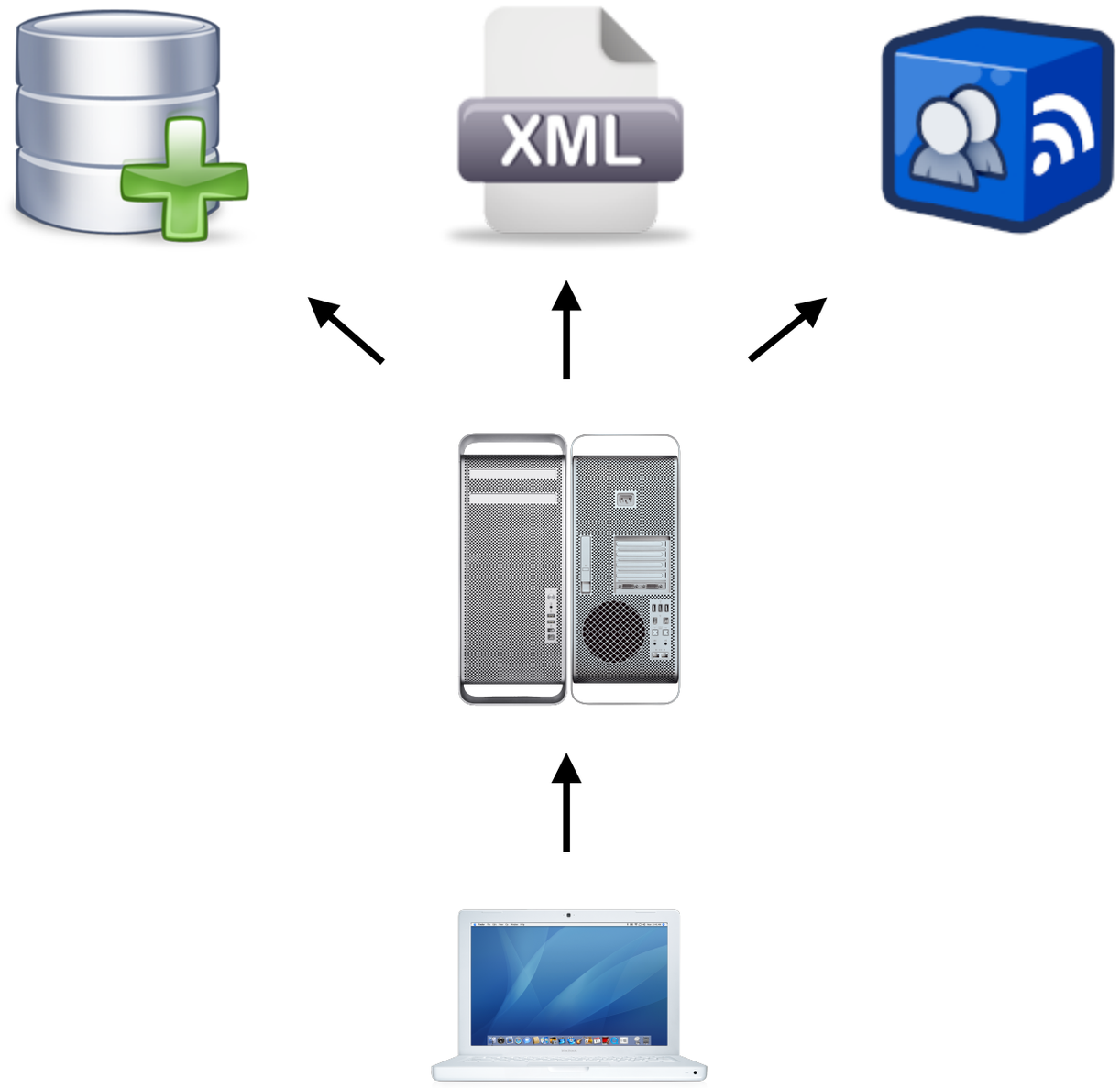
28msec

AtomPub Client / Server

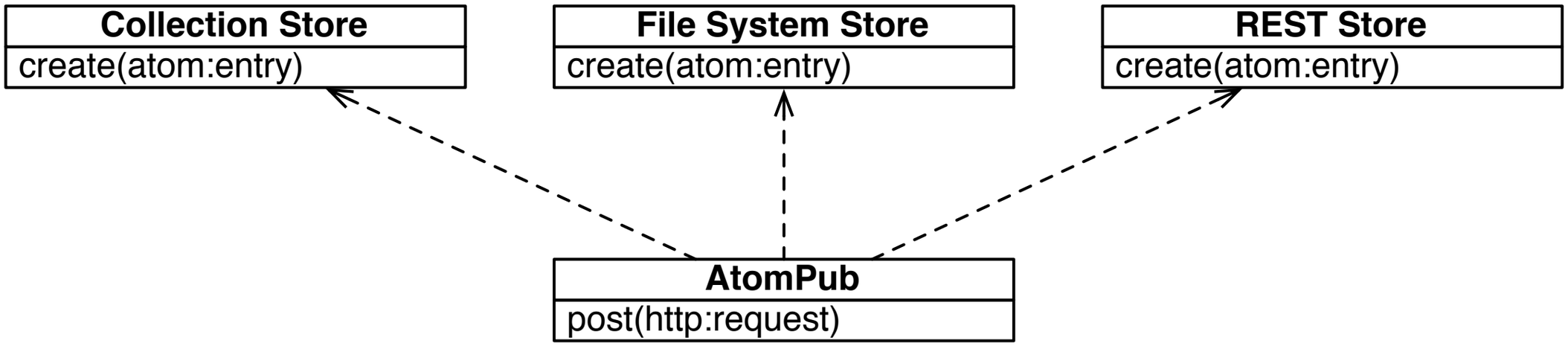
	Description	Pattern	Language
1	Store an Atom entry on the server		
2	Transform an Atom entry into XHTML		
3	Notify Twitter for each new Atom entry		

28msec

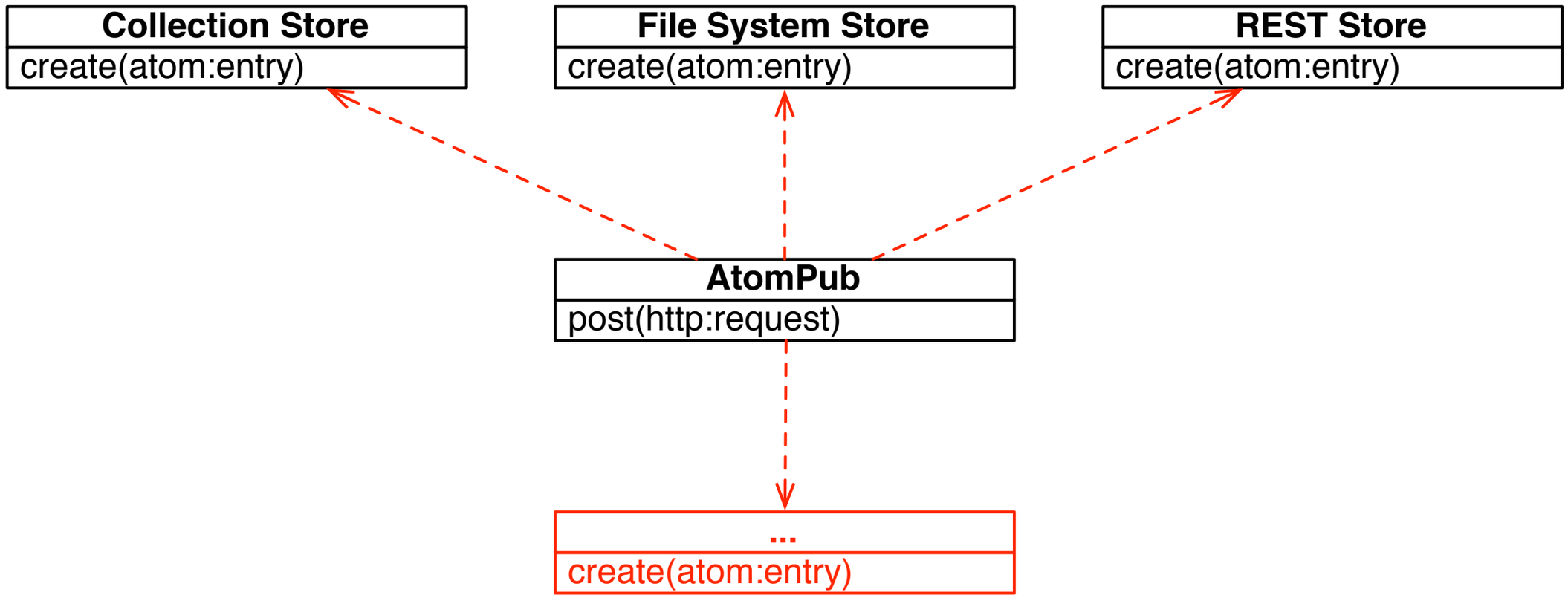
Store an Atom entry on the server



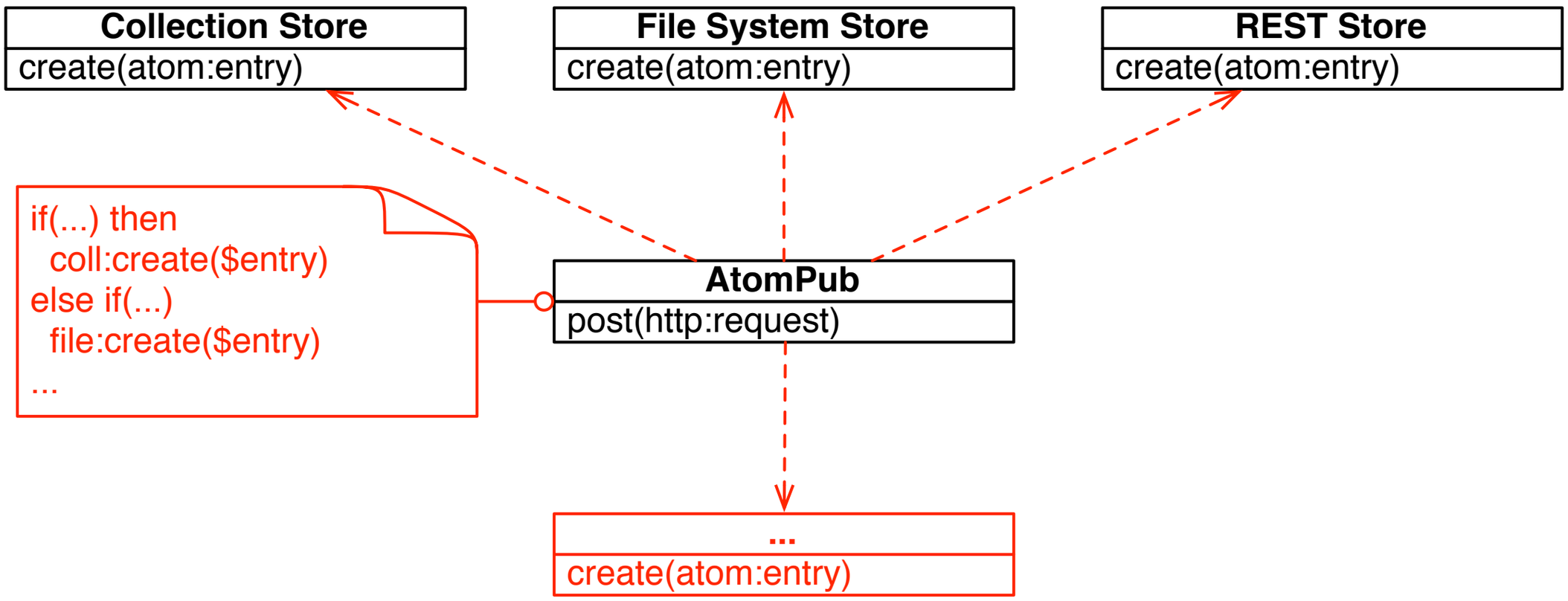
Store an Atom entry on the server



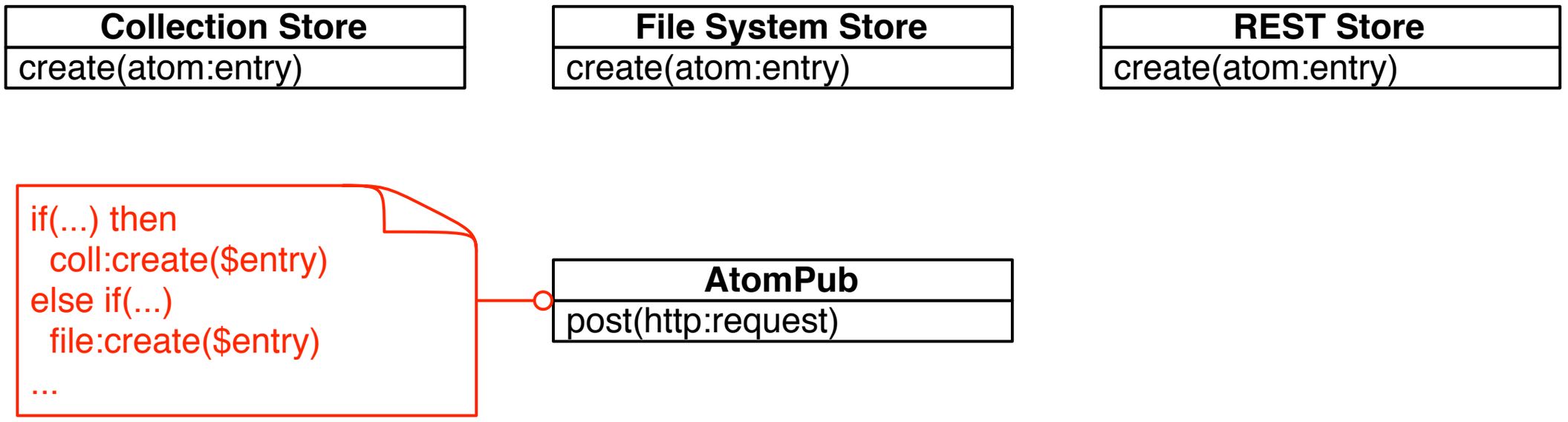
Use-Case 1


Collection Store
create(atom:entry)
File System Store
create(atom:entry)
REST Store
create(atom:entry)
AtomPub
post(http:request)
...
create(atom:entry)

Use-Case I


Collection Store
create(atom:entry)
File System Store
create(atom:entry)
REST Store
create(atom:entry)
if(...) then
coll:create($entry)
else if(...)
file:create($entry)
...
AtomPub
post(http:request)
...
create(atom:entry)

Use-Case 1


Collection Store
create(atom:entry)
File System Store
create(atom:entry)
REST Store
create(atom:entry)
if(...) then
coll:create($entry)
else if(...)
file:create($entry)
...
AtomPub
post(http:request)

Use-Case 1

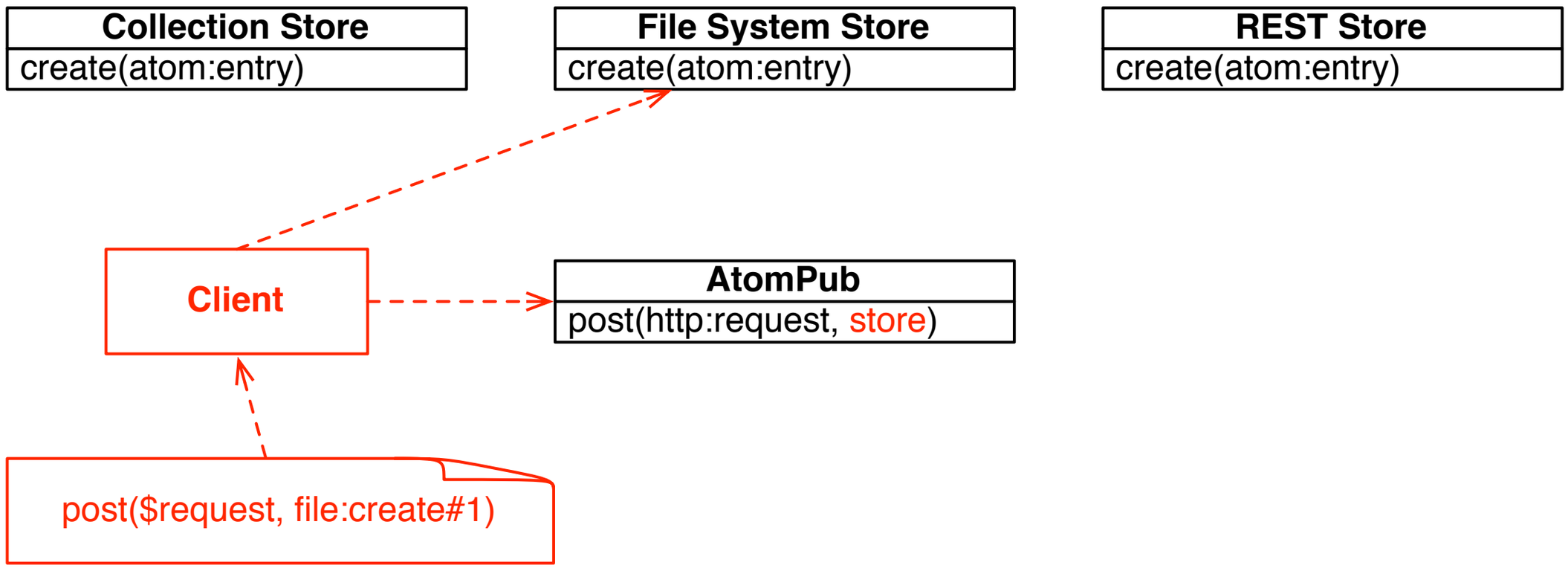
Collection Store
create(atom:entry)

File System Store
create(atom:entry)

REST Store
create(atom:entry)

AtomPub
post(http:request, store)

Strategy


Collection Store
create(atom:entry)
File System Store
create(atom:entry)
REST Store
create(atom:entry)
Client
AtomPub
post(http:request, store)
post($request, file:create#1)

Strategy

28msec

```
declare sequential function atompub:post(
  $feed-uri as xs:string,
  $entry as schema-element(atom:entry),
  $store as (function(xs:string, atom:entry) as item()*))
) {
  (: Processing ...:)
  (: Storing the entry :)
  $store($feed-uri, $entry-uri)
};
```

Benefits

- Reduced Coupling
- Reusability
- Flexibility

Use it when

- Multiple implementation share the same interface
- Hide specific implementation details from a module
- Large amount of conditional statements

Use-Cases

28msec

	Description	Pattern	Language
1	**Store an Atom entry on the server**	**Strategy**	**XQuery 1.1**
2	Transform an Atom entry into XHTML		
3	Notify Twitter for each new Atom entry		

Use-Case 2

28msec

- Transform an Atom entry into XHTML

```
let $title := $feed/atom:title/text()
return
  <html xmlns="http://www.w3.org/1999/xhtml">
    <head>
      <title>{$title}</title>
    </head>
    <body>
      <h1>{$title}</h1>
      {
        for $entry in $feed/atom:entry
        return <div id="{$entry/atom:id/text()}">
                 <h2>{$entry/atom:title/text()}</h2>
                 {$entry/atom:content/*}
               </div>
      }
      </body>
    </html>
```

Use-Case 2

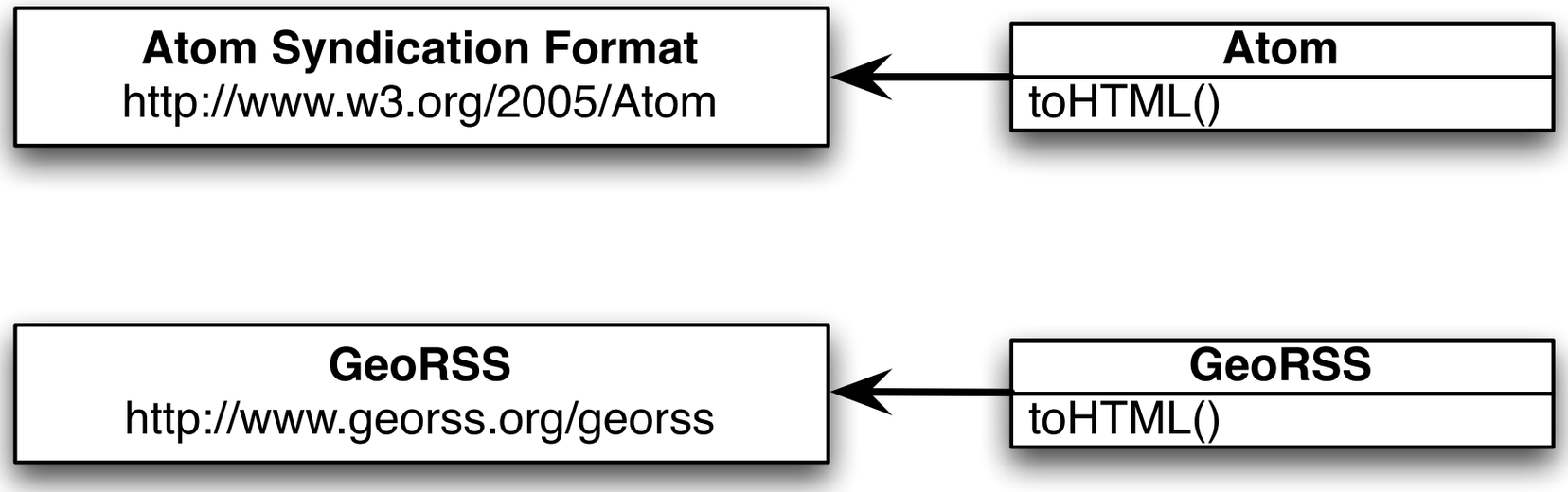
28msec

- Providing transformations for a particular data type

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:georss="http://www.georss.org/georss">
  <title>Where is Waldo?</title>
  <link href="http://example.org/"/>
  <entry>
    <title>Cacilienstrasse 5, 8032 Zurich, Switzerland</title>
    <link href="http://example.org/2009/09/09/atom01"/>
    <updated>2009-08-17T07:02:32Z</updated>
    <georss:point>45.256 -71.92</georss:point>
  </entry>
</feed>
```

Use-Case 2

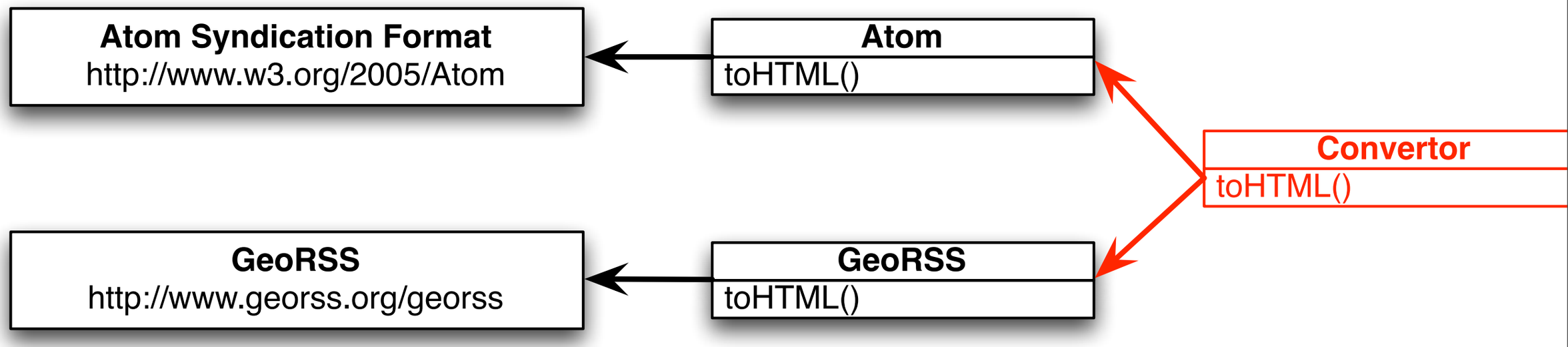
- Providing transformations for a particular data type



Use-Case 2

28msec

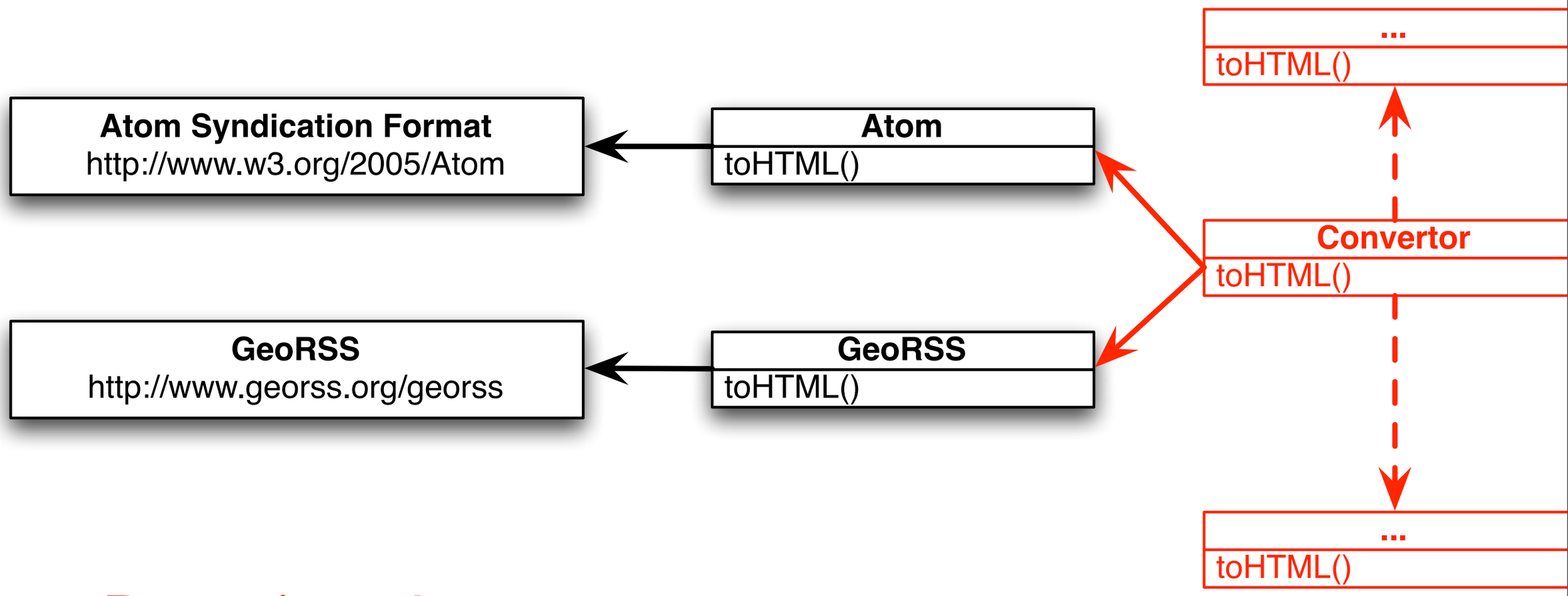
- Providing transformations for a particular data type
- How to combine them?



Use-Case 2

28msec

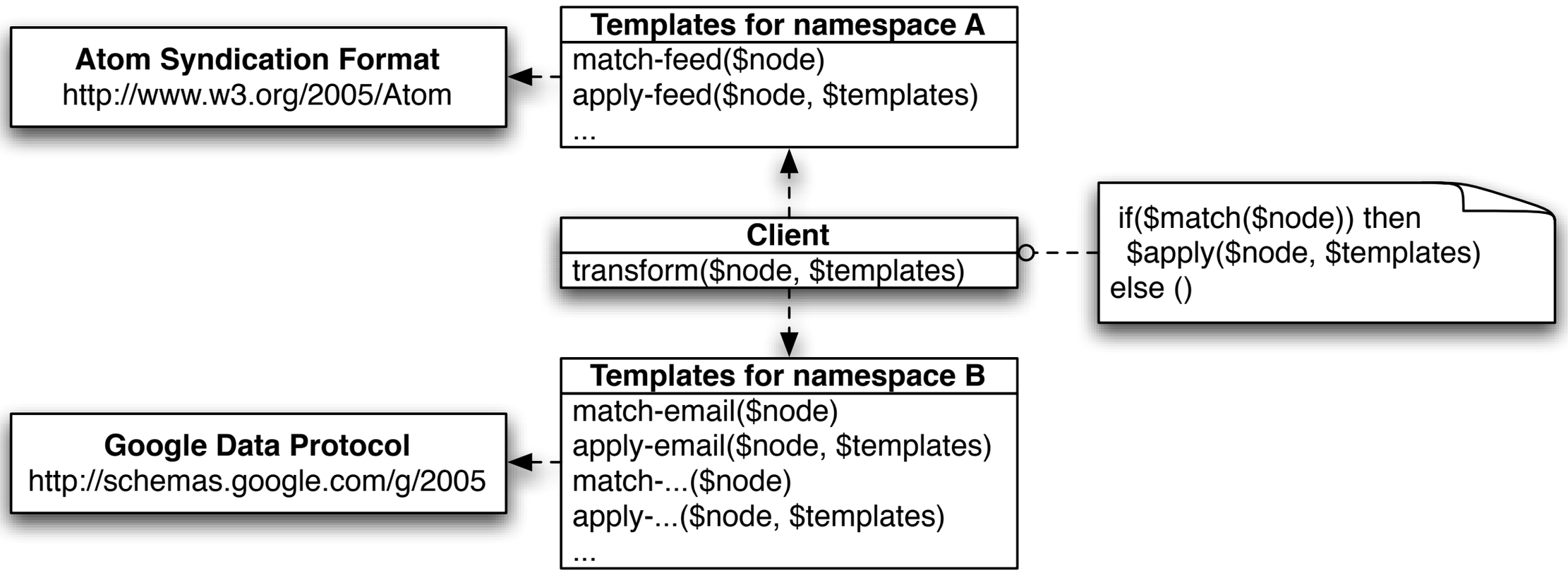
- Providing transformations for a particular data type
- How to combine them?



- Doesn't scale

Translator

- Separate interpretations of heterogeneous elements.
- Bringing the XSLT programming paradigm into XQuery.



Template - Match Function

28msec

```
declare function atom:feed-template(
  $transform, $node, $templates) as function(){
((: match function :)
  function($node) as xs:boolean
  {
    typeswitch ($node)
      case $n as schema-element(atom:feed) return true()
      default return false()
  },
  (: apply :)
  function($transform, $feed, $templates) as element(html:html)
  {
    <html xmlns="http://www.w3.org/1999/xhtml">
      <h1>Feed</h1>
      <div id="entries">
      {$transform($feed/atom:entry, $templates)}
      </div>
    </html>
  })
};
```

Template - Apply Function

28msec

```
declare function atom:feed-template(
  $transform, $node, $templates) as function(){
(: match :)
  function($node) as xs:boolean
  {
    typeswitch ($node)
      case $n as schema-element(atom:feed) return true()
      default return false()
  },
  (: apply function :)
  function($transform, $feed, $templates) as element(html:html)
  {
    <html xmlns="http://www.w3.org/1999/xhtml">
      <h1>Feed</h1>
      <div id="entries">
      {$transform($feed/atom:entry, $templates)}
      </div>
    </html>
  })
};
```

Translator - Transform Function

28msec

```
declare function local:transform($node, $templates) as item()*
{
  for $tpl in $templates
  let $template := $tpl()
  let $match := $template[1]
  let $apply := $template[2]
  return
    if($match($node)) then
      $apply(local:transform#2, $node, $templates)
    else ()
};
```

Translator - Conclusion

28msec

Benefits

- Independent modules can easily collaborate on the same XDM instance
- Extending the XDM translation is easy

Use it when:

- Transform/Interpret an heterogeneous XML document
- Use the power of XSLT paradigm in XQuery

Use-Cases

28msec

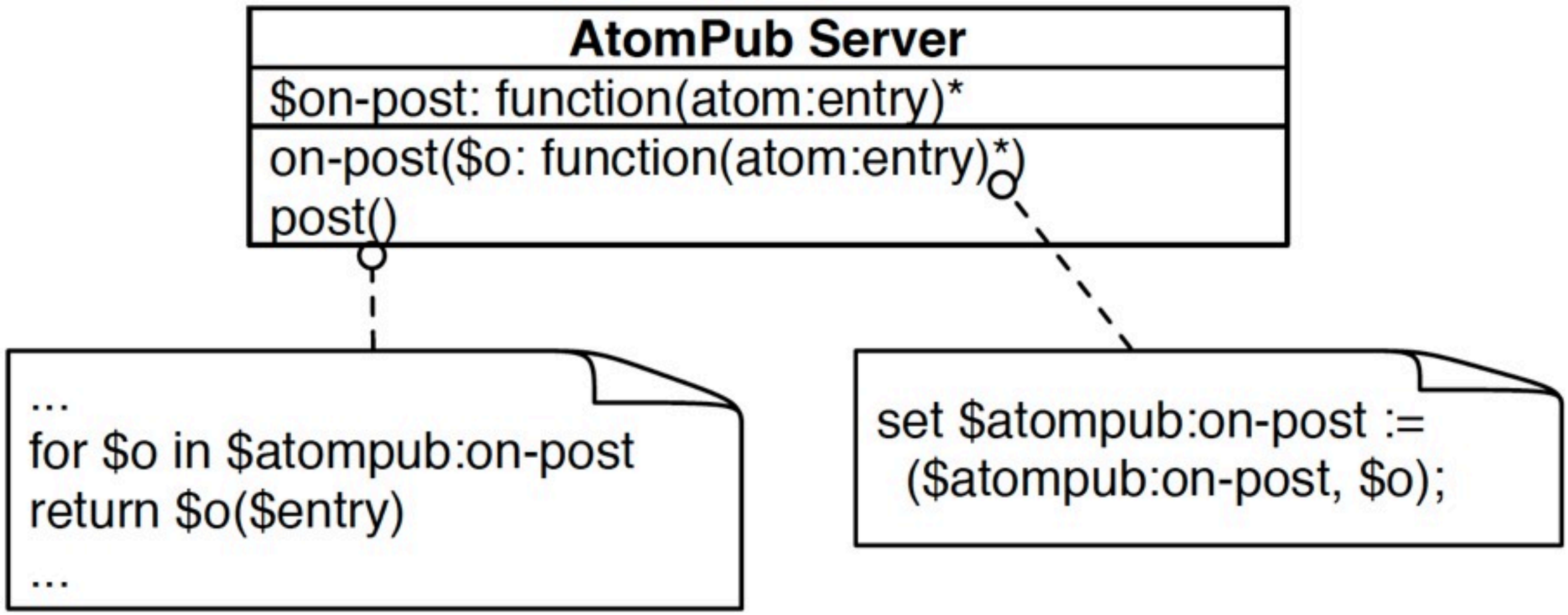
	Description	Pattern	Language
1	Store an Atom entry on the server	Strategy	XQuery 1.1
2	**Transform an Atom entry into XHTML**	**Translator**	**XQuery 1.1**
3	Notify Twitter for each new Atom entry		

Use-Case 3

28msec

- Publish/Subscribe mechanism for the Atom server
- Enable new kind of collaboration with an arbitrary number of modules
- Example: Sending a message on Twitter for each new Atom entry

Observer


AtomPub Server
$on-post: function(atom:entry)*
on-post($o: function(atom:entry)*)
post()
...
for $o in $atompub:on-post
return $o($entry)
...
set $atompub:on-post :=
($atompub:on-post, $o);

Observer

28msec

```
(: Hold the observer references :)
declare variable $atompub:on-post := ();

(: Add an observer to the post request :)
declare sequential function atompub:on-post($o) as item()*)){
  set $atompub:on-post := ($atompub:on-post, $o)
};

declare sequential function atompub:post() {
  (: Processing :)
  (: Notify observers :)
  for $o in $atompub:on-post
  return $o($entry);
};
```

Observer - Conclusion

28msec

Benefits

- Extensible behavior
- Collaboration with an arbitrary number of modules
- Low coupling

Applicability

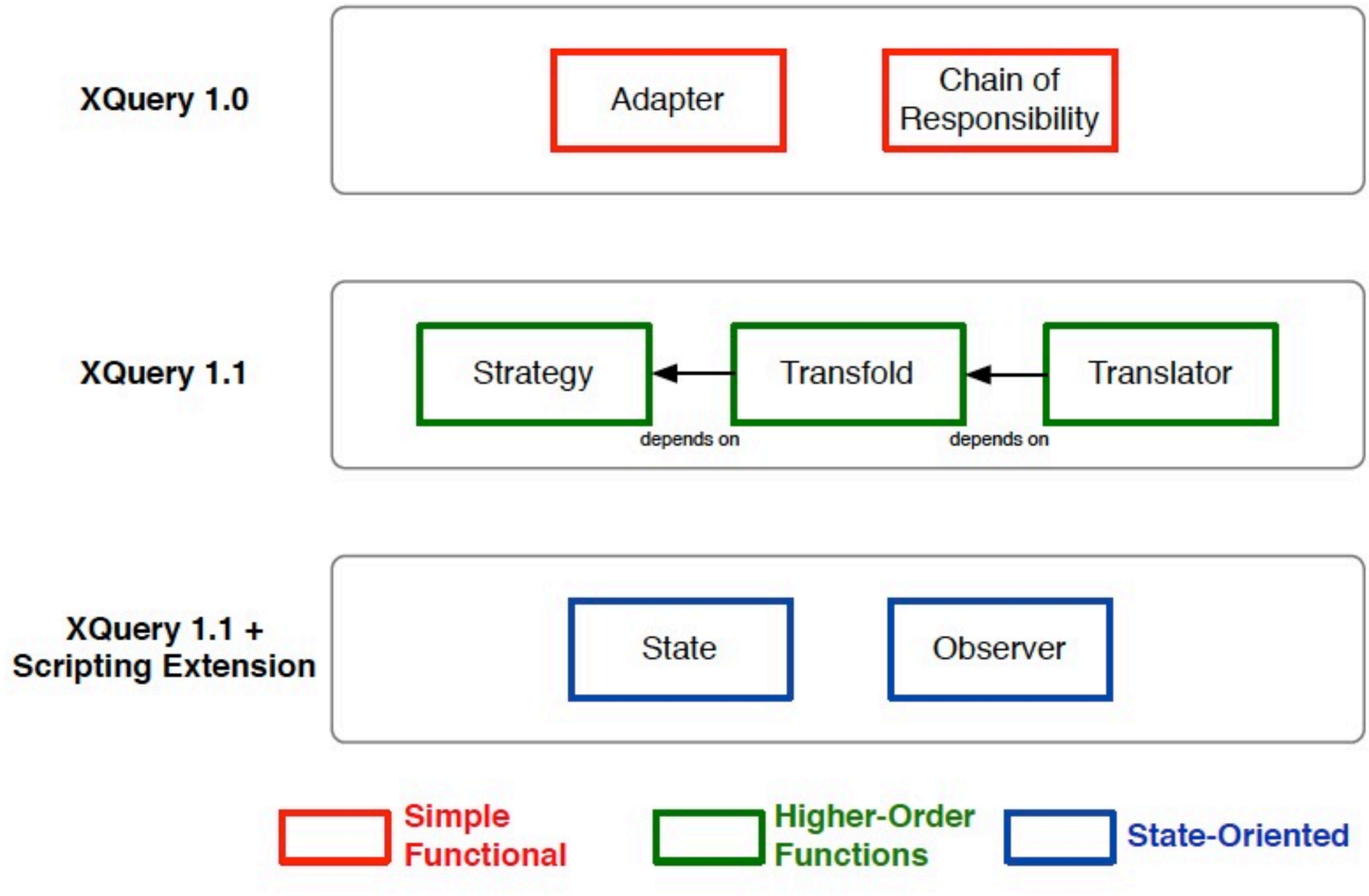
- Publish/Subscribe paradigm within XQuery
- Keep consistency between module states

Use-Cases

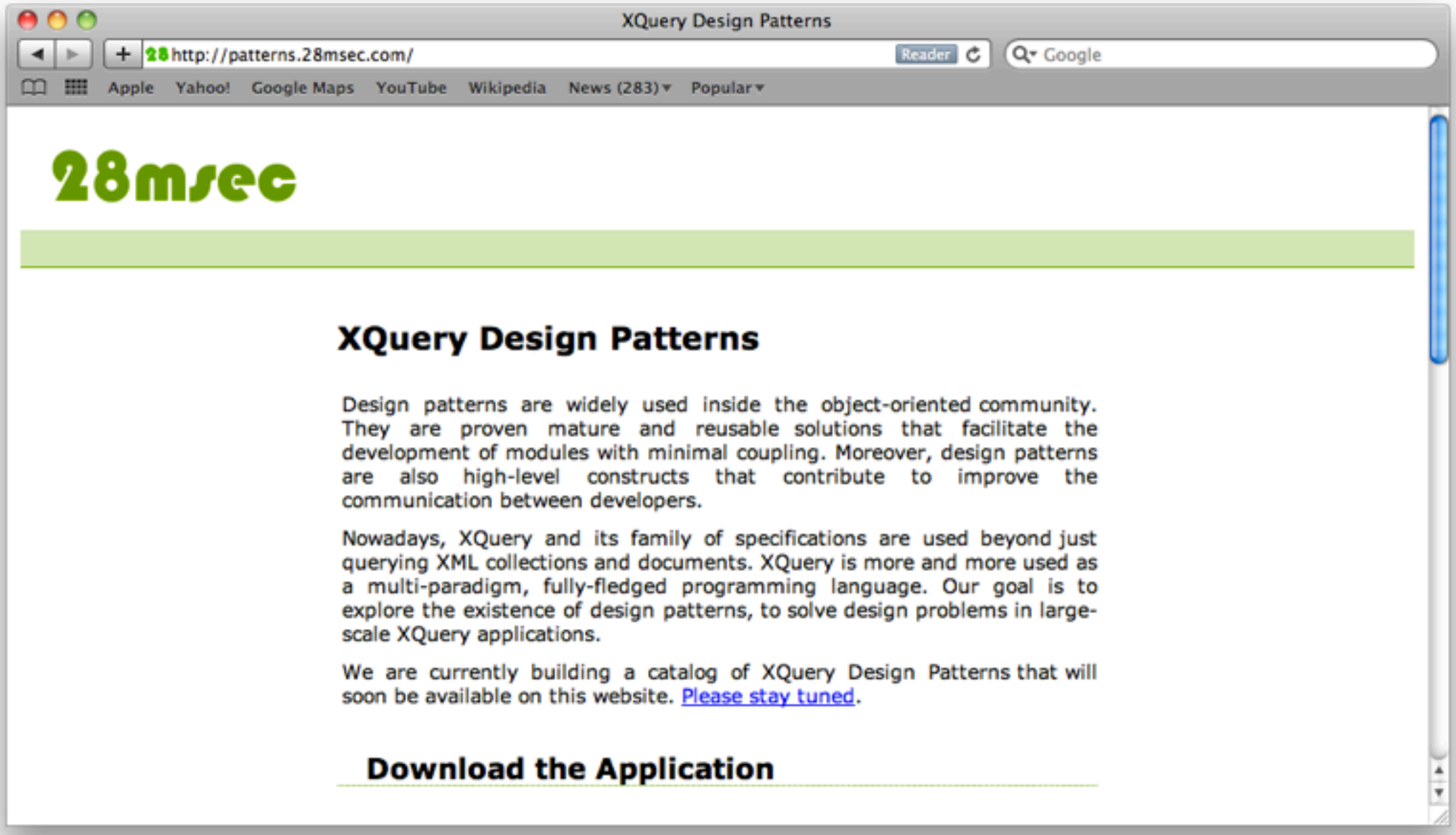
28msec

	Description	Pattern	Language
1	Store an Atom entry on the server	Strategy	XQuery 1.1
2	Transform an Atom entry into XHTML	Translator	XQuery 1.1
3	**Notify Twitter for each new Atom entry**	**Observer**	**Scripting**

XQuery Design Pattern Catalog



XQuery Design Pattern Catalog



Join the community: http://patterns.28msec.com

Conclusion

28msec

- Design Patterns are going beyond the object community
- Companies are building large-scale applications in XQuery
- Promoting better designs and low coupling
- XQuery Design Patterns...
  - ...are pragmatic solutions
  - ...they belong to the community

Questions?

More info

http://patterns.28msec.com