# HealthSOAF:
# BaseX and Its Usage in the Healthcare Domain

Author: Marco Lettere

# Me …

- Computer Scientist (MD 2001 University of Pisa)
- Currently employed as a tech lead and senior software engineer in the RnD department
- Background in
  - HPC and distributed computing
  - Ambient Intelligence
  - System integration, Web Services and XML stuff
- Even if I also like CG, UI development, Game development
- … and playing soccer

# The Company ...

- Dedalus is an Italian company involved in IT for public and private Healthcare
- Several sites in Italy and currently expanding world wide
  - China, South Africa, Eastern Europe, Latin America, …
- Strong focus on Healthcare related standards
  - IHE
  - HL7
  - Dicom
- Site:
  - http://www.dedalus.eu

# The Context (1) - the Pillars

- HL7
  - Standardization of procedures, formats, models and technologies for the software in the healthcare domain
  - http://www.hl7.org
- OMG
  - Standardization …. well overall
  - http://www.omg.org
- HSSP - Health Service Specification Programme
  - Joint initiative to embrace SoA for Healthcare with focus on simplification and interoperability
  - http://hssp.wikispaces.com/

# The Context (2) – the Program

- HSSP represents a technical specification (CIM, PIM, PSM) for a bouquet of services typically required in large-scale, cross-enterprise healthcare institutions
  - RLUS – Resource Repository
  - IXS - Identity Management
  - CTS2 – CodeSystems and Terminology management
  - ServD – Service Directory
  - CDSS – Clinical Decision Support

# The Context (3) – the Project

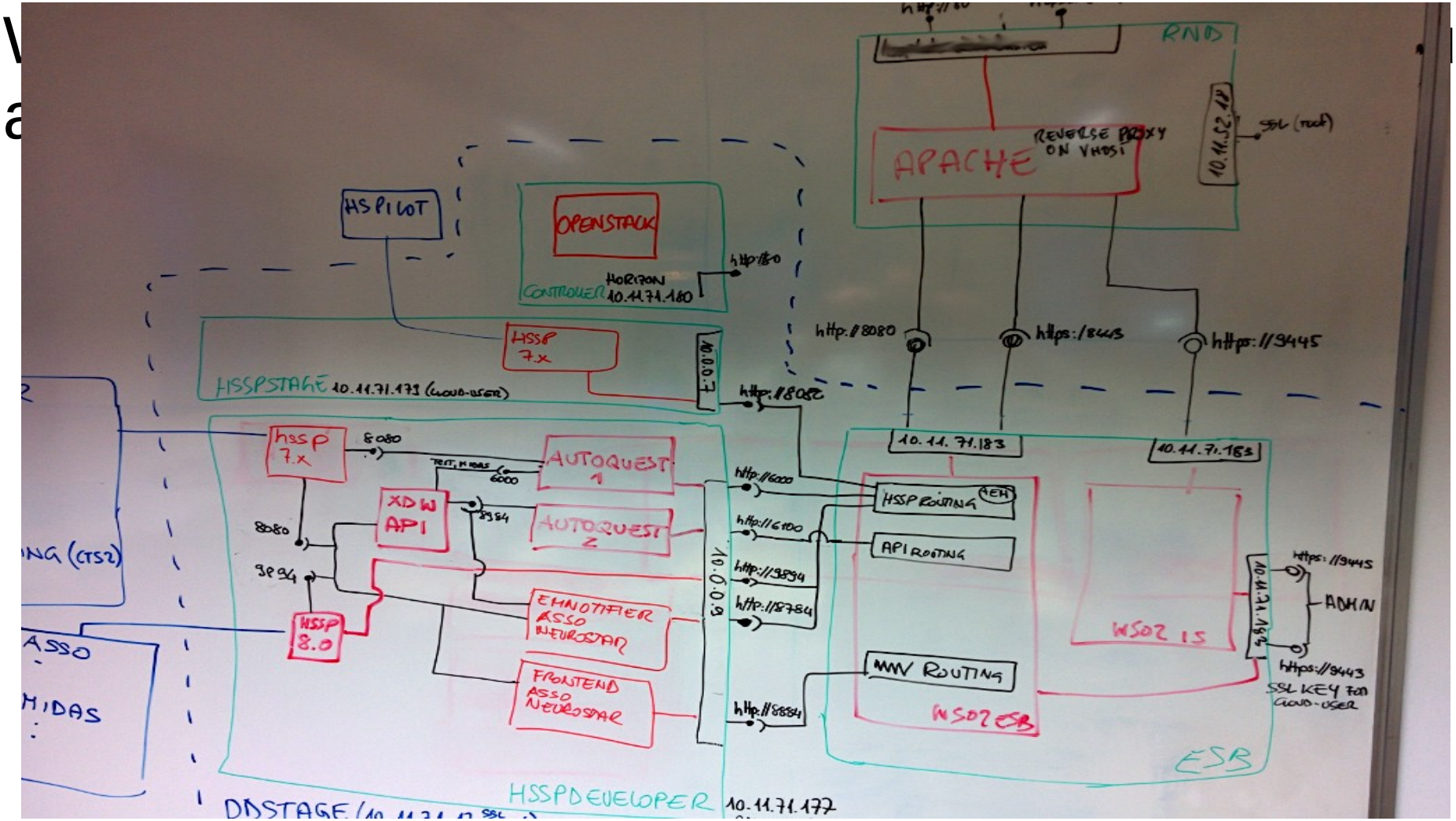
- HealthSOAF is a project funded by the Italian Ministry of Research and Education aiming at
  - delivering in depth study and a Model Driven analysis of the HSSP specifications
  - delivering the first complete Italian implementation
  - setting up a Pilot Site for medium term experimentation
  - producing implementation guidelines to facilitate other companies (providers or consumers) to adopt the HSSP specifications
- Site:
  - http://www.healthsoaf.it
- Partnership:
  - Dedalus SPA, Almaviva SPA, Simple Engineering, Università della Calabria, other smaller companies

# The Context (4) – our role in the movie

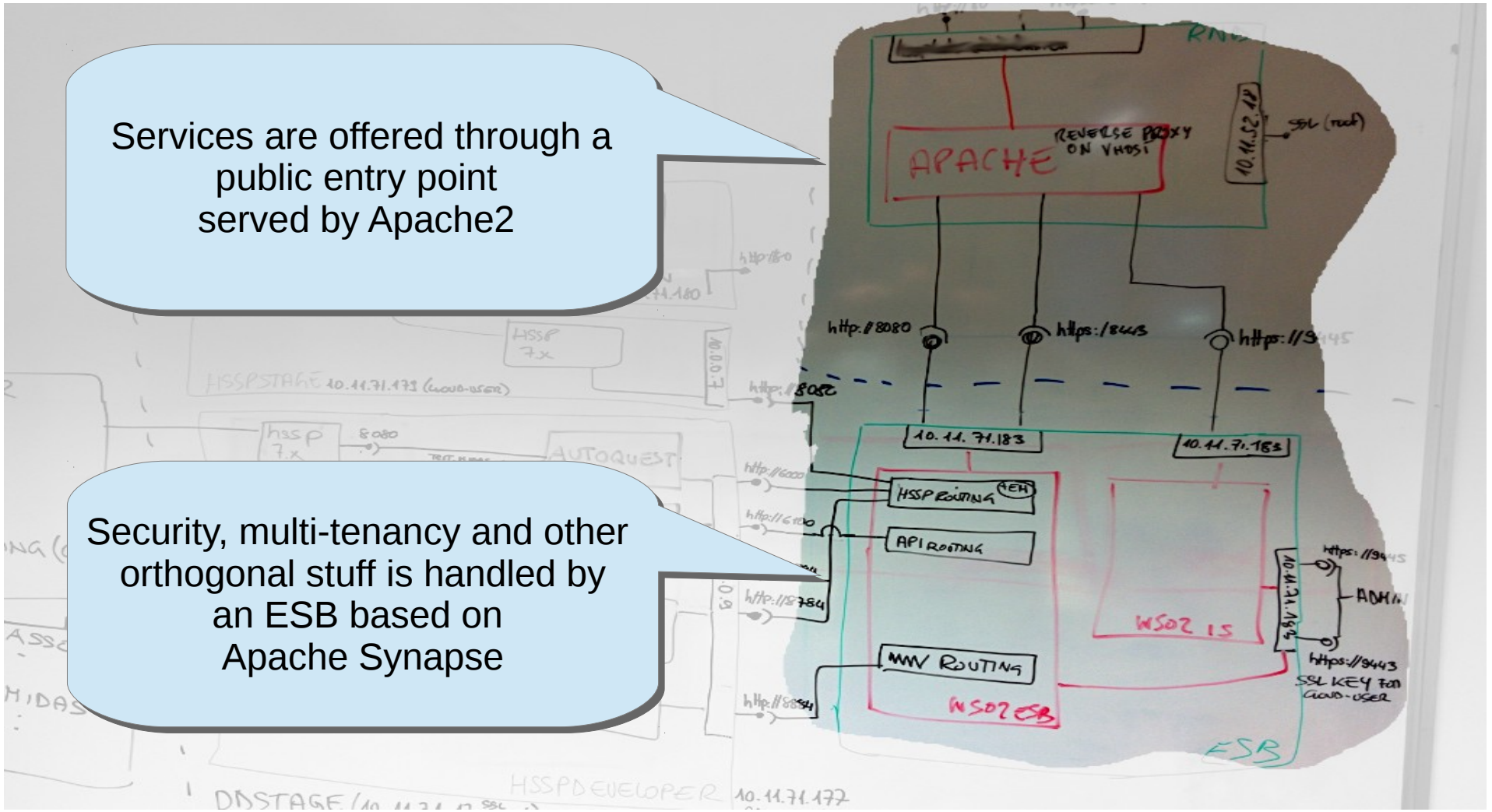
- Dedalus is in charge of implementing and maintaining the following three services:
  - RLUS
    - semantic aware repository of resources with operations for storing, retrieving and querying resources
    - Used for storing Clinical Reports and Observations, Medical Prescriptions, Workflow Documents, Accounts and Financial Transactions, ...
    - WS* and REST based interfaces
  - IXS
    - semantic aware identity index with operations for inserting identities, querying identities by ID or traits, linking and merging identities, reporting duplicates
    - Used for uniquely identifying patients, operators, devices, ...
    - WS* based interface
  - CTS2
    - Management of codesystems and terminologies with operations for importing and exporting codesystems, mapping codes, configuring valueset bound to conceptdomains, managing versioning and history
    - REST based interface

# And finally the interesting part ...

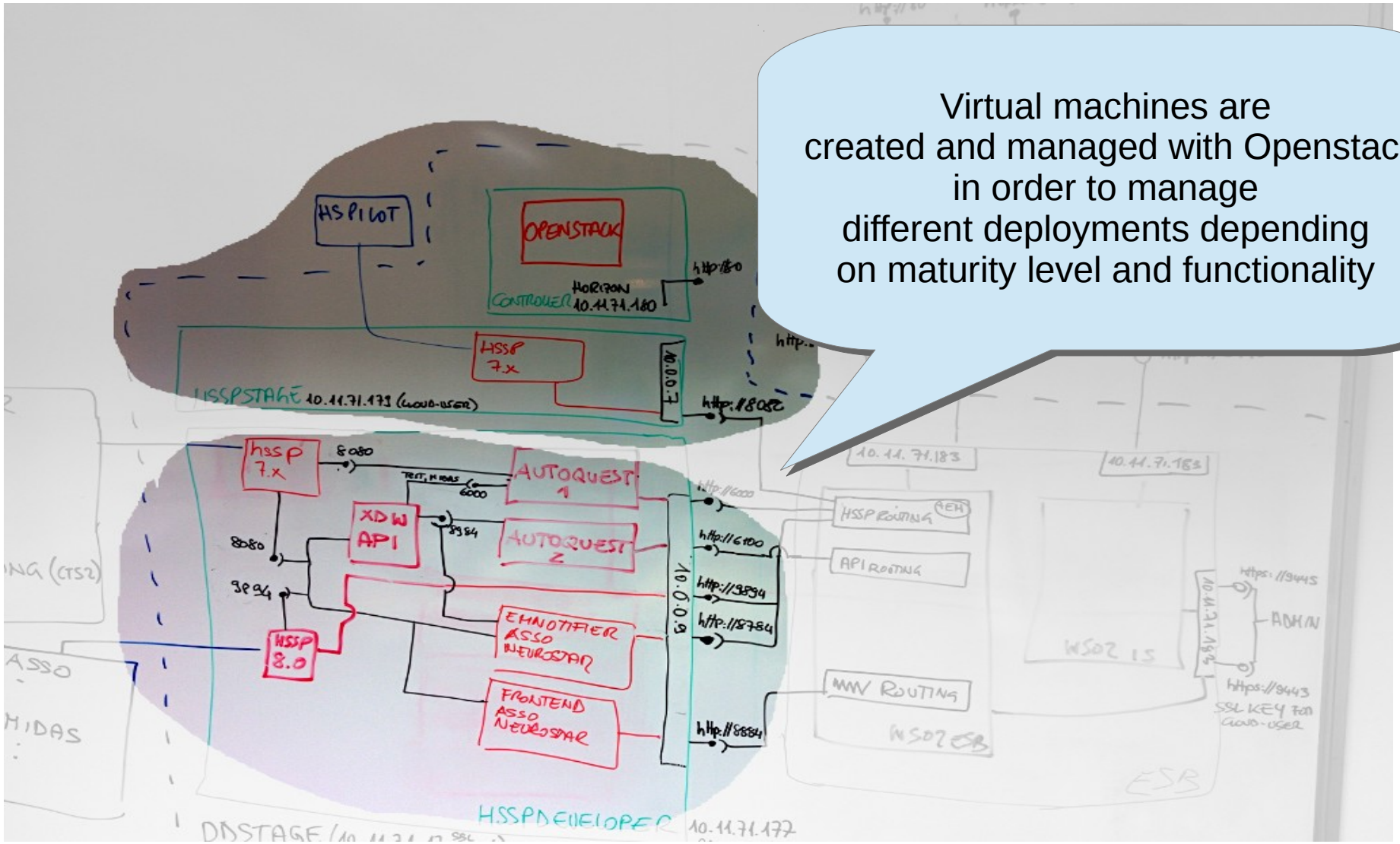
- V … s a



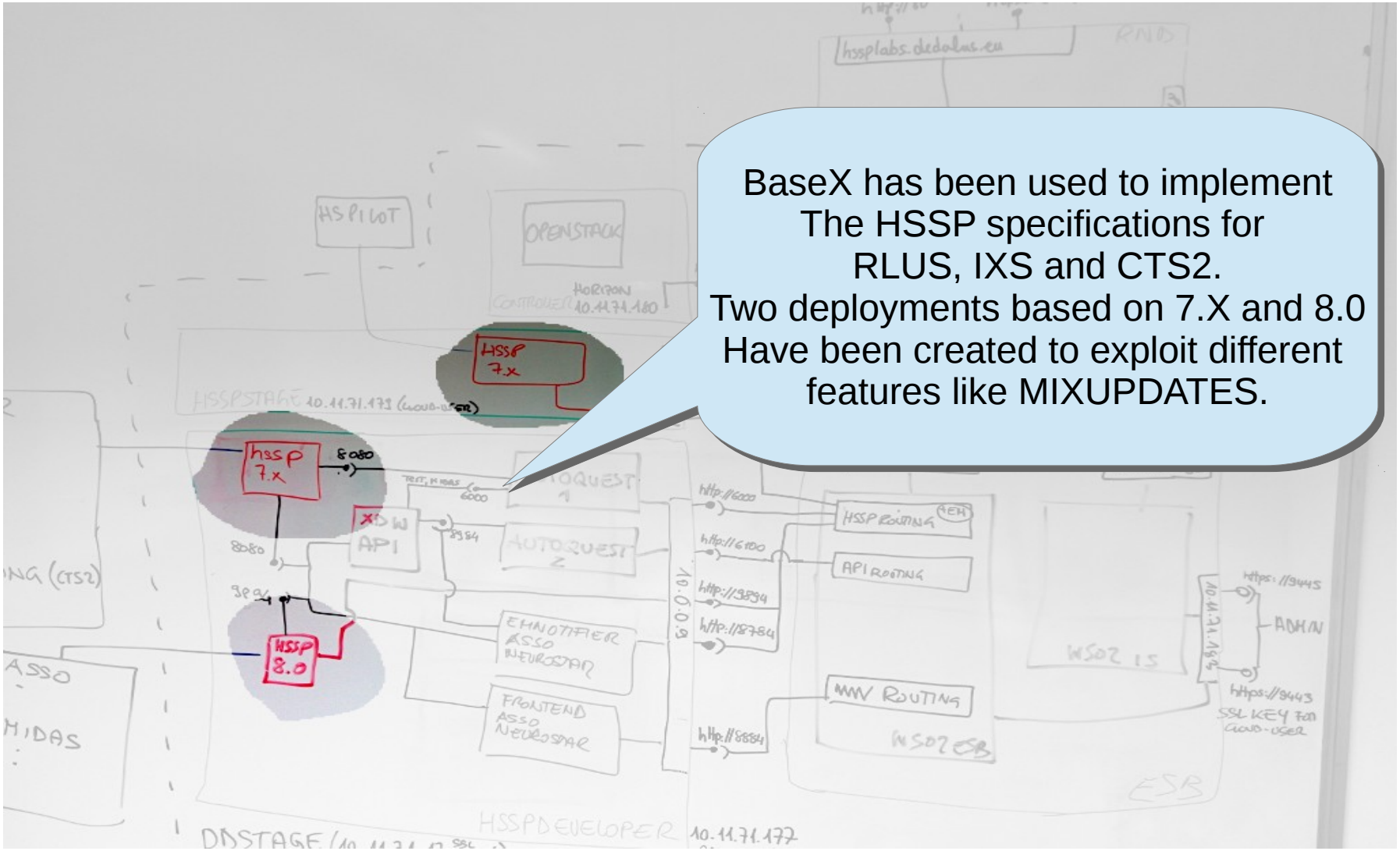
# Ok ... Clearing it up ...



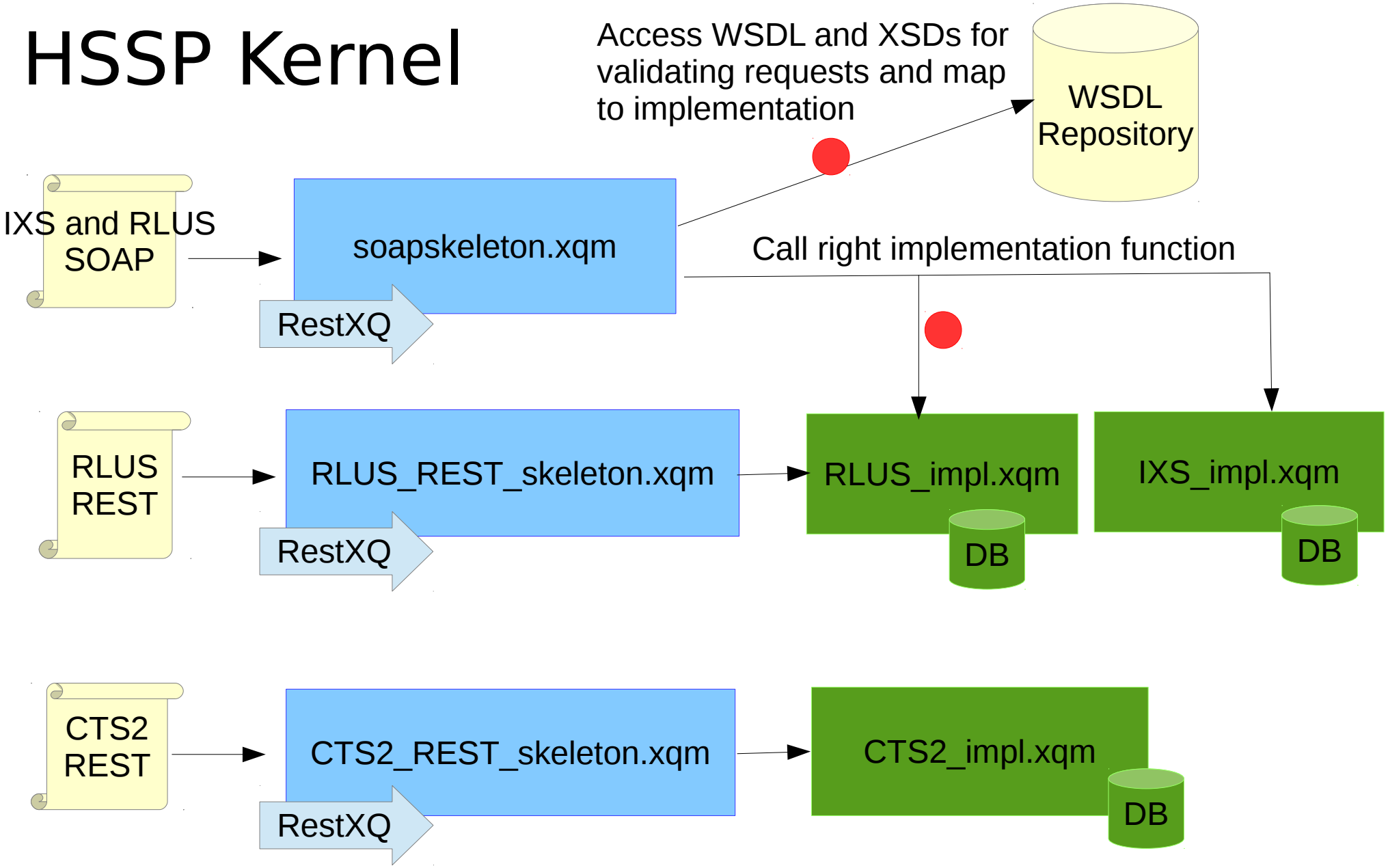
# Some more technology



# BaseX for the HSSP Kernels



# HSSP Kernel


Access WSDL and XSDs for validating requests and map to implementation
WSDL Repository
IXS and RLUS SOAP
soapskeleton.xqm
RestXQ
Call right implementation function
RLUS REST
RLUS_REST_skeleton.xqm
RestXQ
RLUS_impl.xqm
DB
IXS_impl.xqm
DB
CTS2 REST
CTS2_REST_skeleton.xqm
RestXQ
CTS2_impl.xqm
DB

# A small insight into soapskeleton

```
declare
        %rest:path("{$project}/{$service}/{$interface}/{$signifier}")
        %rest:POST("{$postbody}")
        %rest:consumes("application/soap+xml", "text/xml")
        %rest:header-param("SOAPAction", "{$soapaction}", "")
        %output:method("xml")
        %output:media-type("text/xml")

function soapskel:enter($project as xs:string, $service as xs:string,
                        $interface as xs:string, $signifier as xs:string,
                        $postbody as node(), $soapaction as xs:string){

    let $wsdlinfo := soapskel:validateandresolve($project, $service, $signifier, $postbody,
$soapaction)

    let $hsspresponse := try{
                            $wsdlinfo('function')($database, $service, $payload)
                        } catch * {  ... }
    ...
```

# A small insight into RLUS

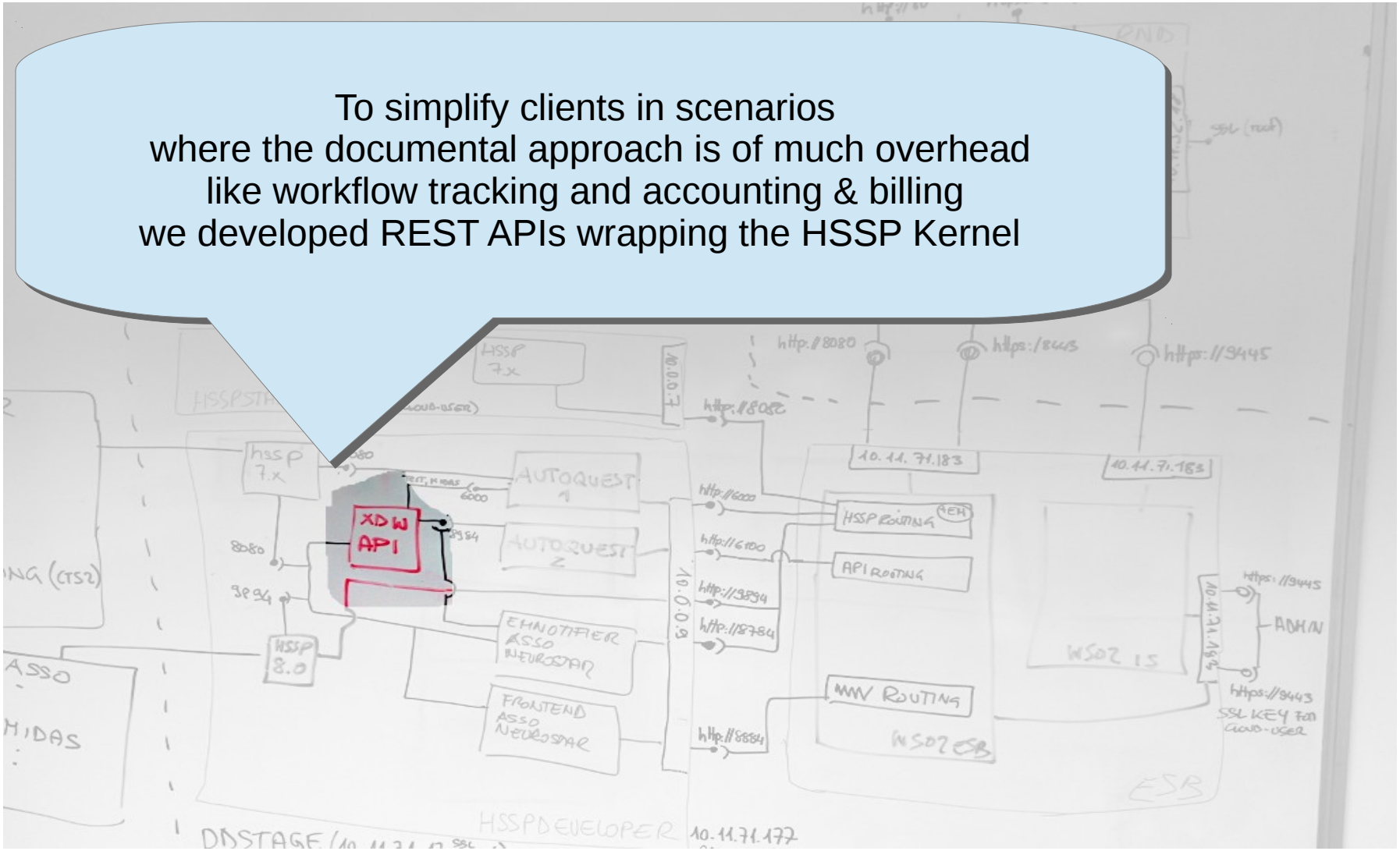
```
<rlusexp:BinaryExpression>
        <PrefixUnaryOperator type="Not"/>
        <PrefixTerm type="Other" text="true()"/>
</rlusexp:BinaryExpression>
<rlusexp:Operator type="Or"/>
<rlusexp:BinaryExpression>
    <rlusexp:BinaryExpression>
        <rlusexp:BetweenTerm text="#amount" type="Numeric"/>
         <rlusexp:Between/>
         <rlusexp:BetweenTerm text="5" type="Numeric"/>
         <rlusexp:And/>
         <rlusexp:BetweenTerm text="50" type="Numeric"/>
    </rlusexp:BinaryExpression>
    <rlusexp:Operator type="And"/>
    <rlusexp:BinaryExpression>
        <rlusexp:BinaryTerm text="#codicefiscale" type="Text"/>
        <rlusexp:Operator type="In"/>
        <rlusexp:BinaryTerm text="('AAA', 'BBB', 'CCC')" type="Other"/>
    </rlusexp:BinaryExpression>
</rlusexp:BinaryExpression>
```

# A small insight into RLUS

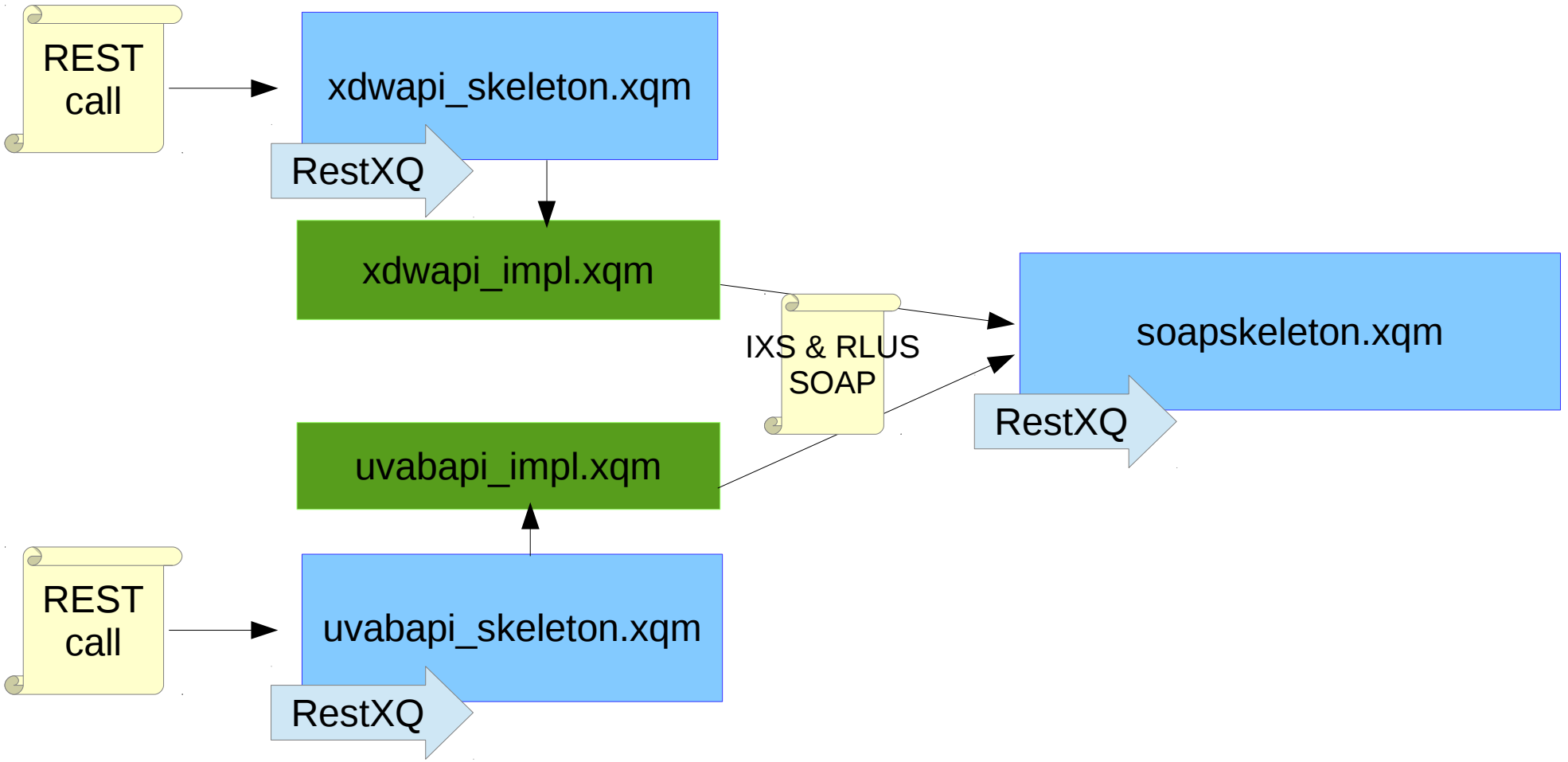
This is what gets to xquery:eval

```
declare namespace hl7v3='urn:hl7-org:v3';
declare variable $database external;

for $docs in $database

let $codicefiscale :=
$docs//hl7v3:subject/hl7v3:patientRole/hl7v3:patientPerson/hl7v3:id/hl7v3:item[@root='2.1
6.840.1.113883.2.9.4.3.2']/data(@extension)

let $amount := $docs//hl7v3:balanceAmt
let $time := $docs//hl7v3:effectiveTime/hl7v3:any/@value
where (not(true()) or (($amount >= 5 and $amount <= 50) and ($codicefiscale = ('AAA',
'BBB', 'CCC'))))
 return $docs
```

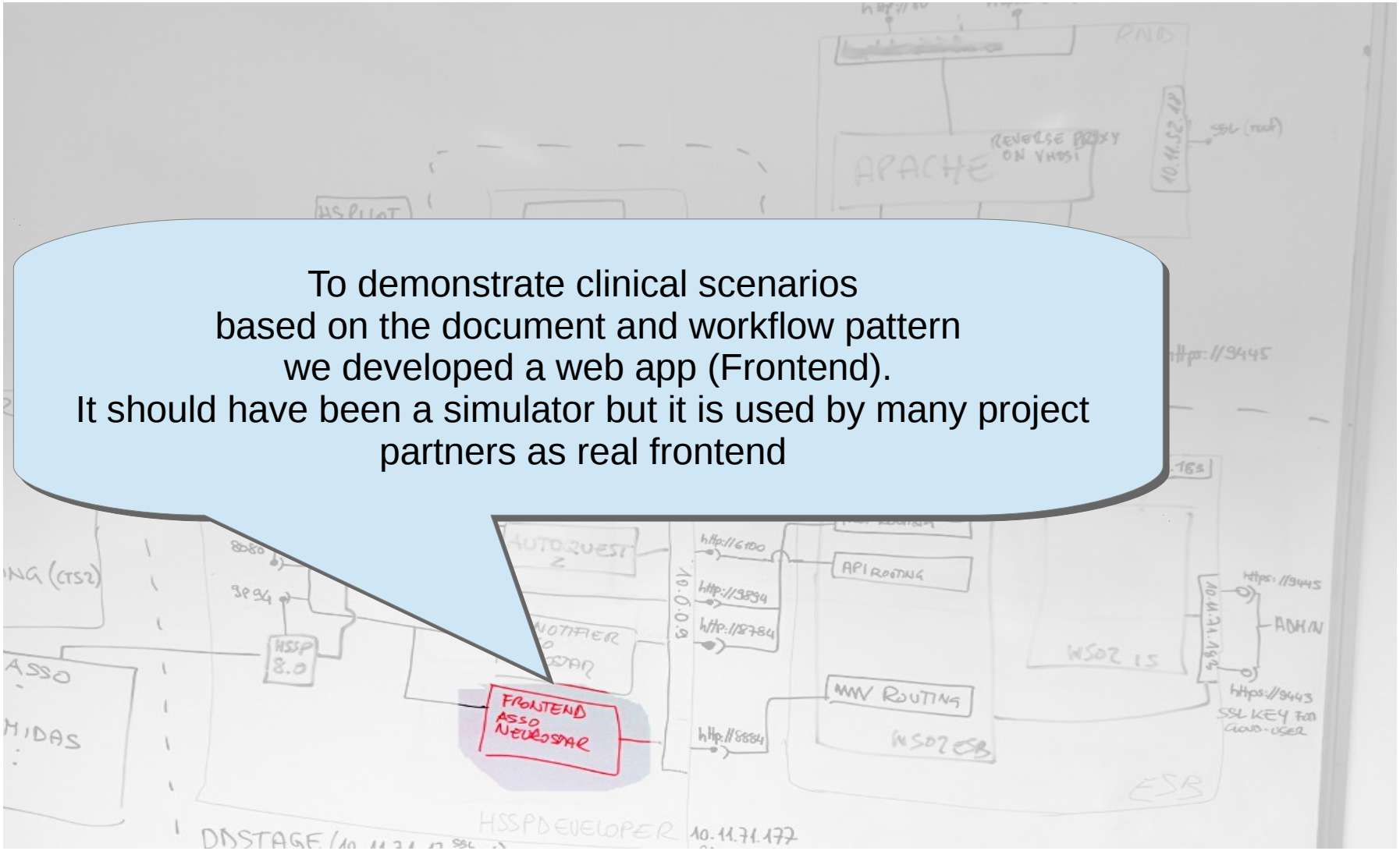
# BaseX for the REST APIs



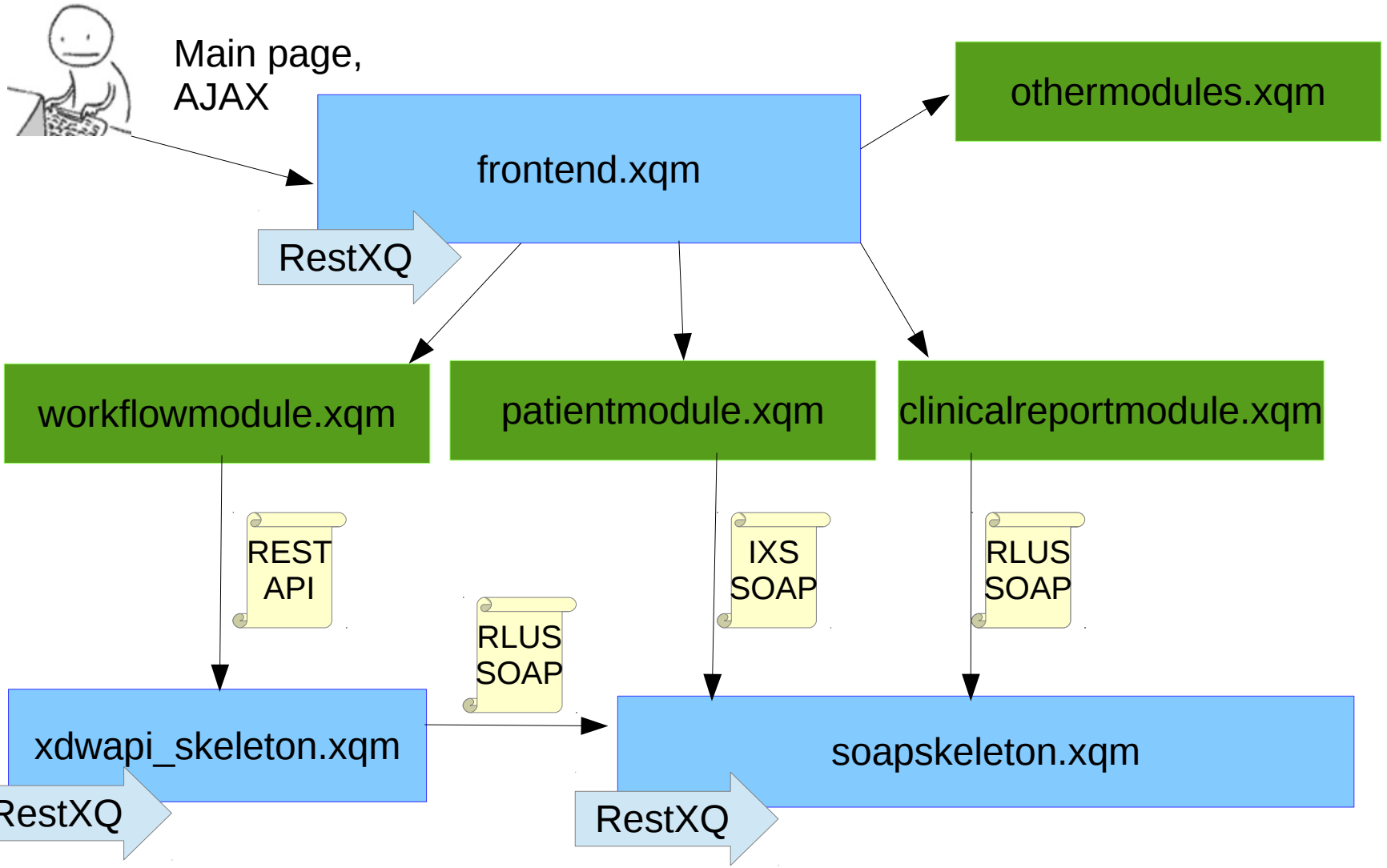
# REST APIs


REST call
xdwapi_skeleton.xqm
RestXQ
xdwapi_impl.xqm
IXS & RLUS SOAP
soapskeleton.xqm
RestXQ
uvabapi_impl.xqm
REST call
uvabapi_skeleton.xqm
RestXQ

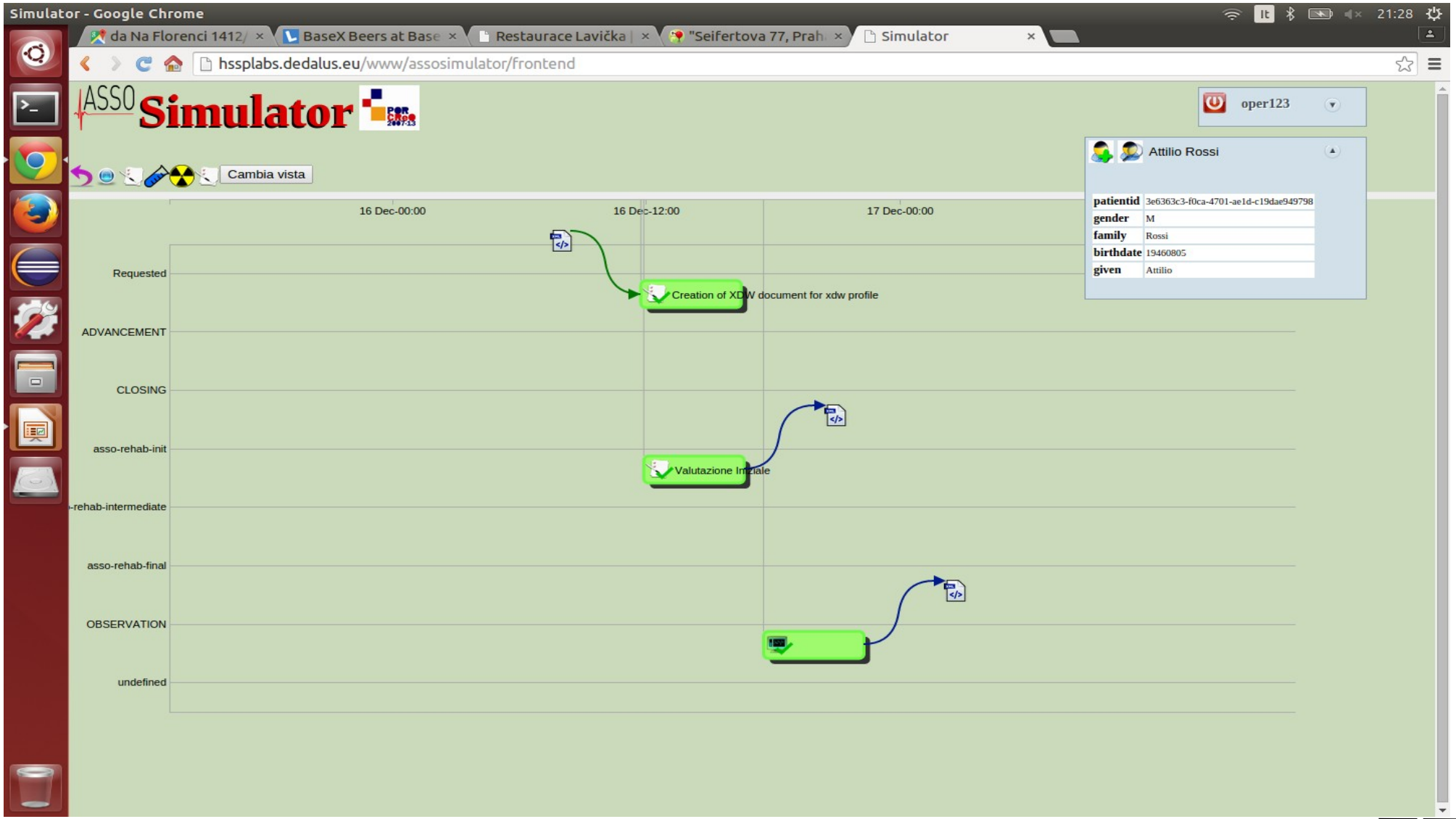
# BaseX as a Frontend Server



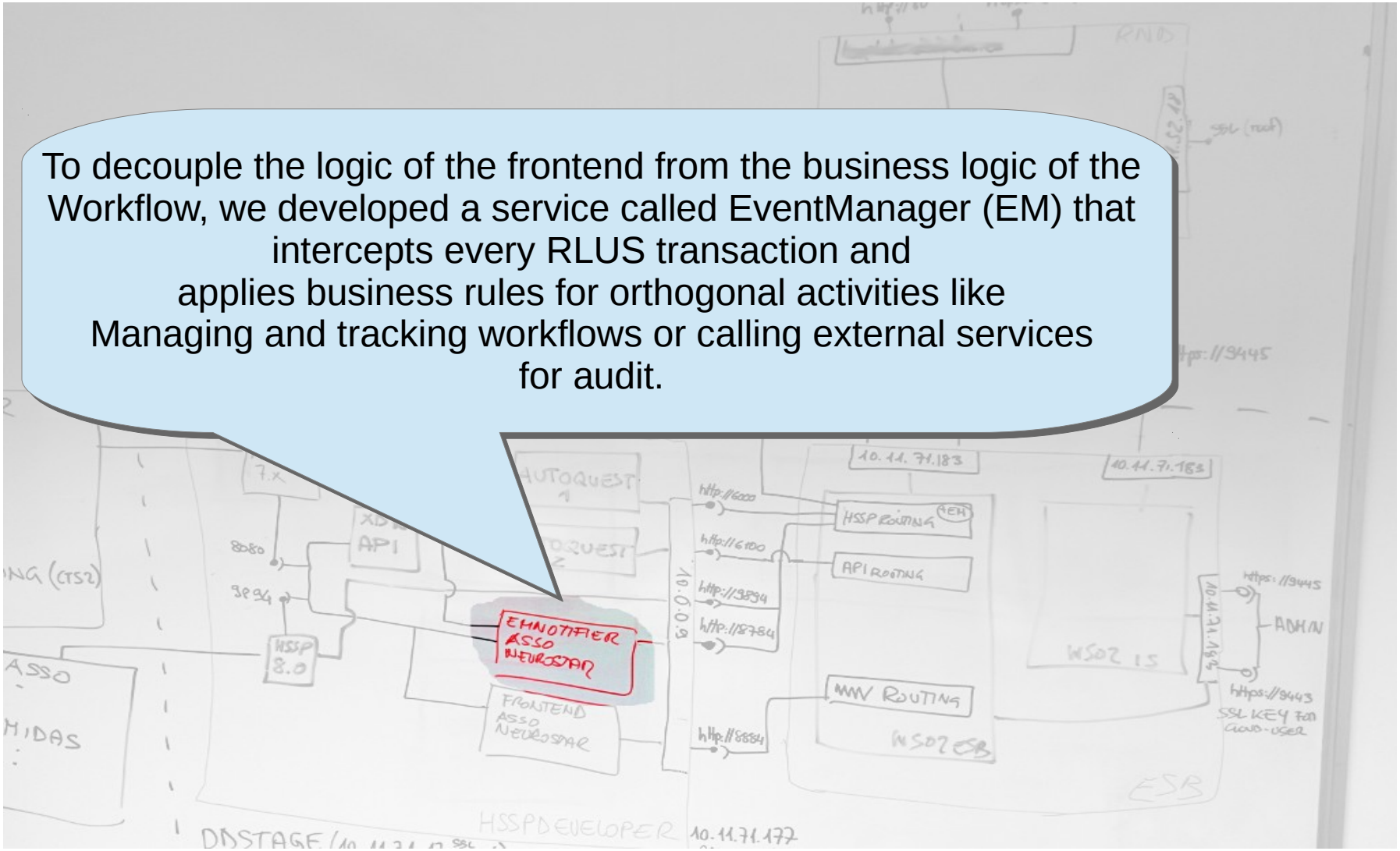
# The Frontend Server



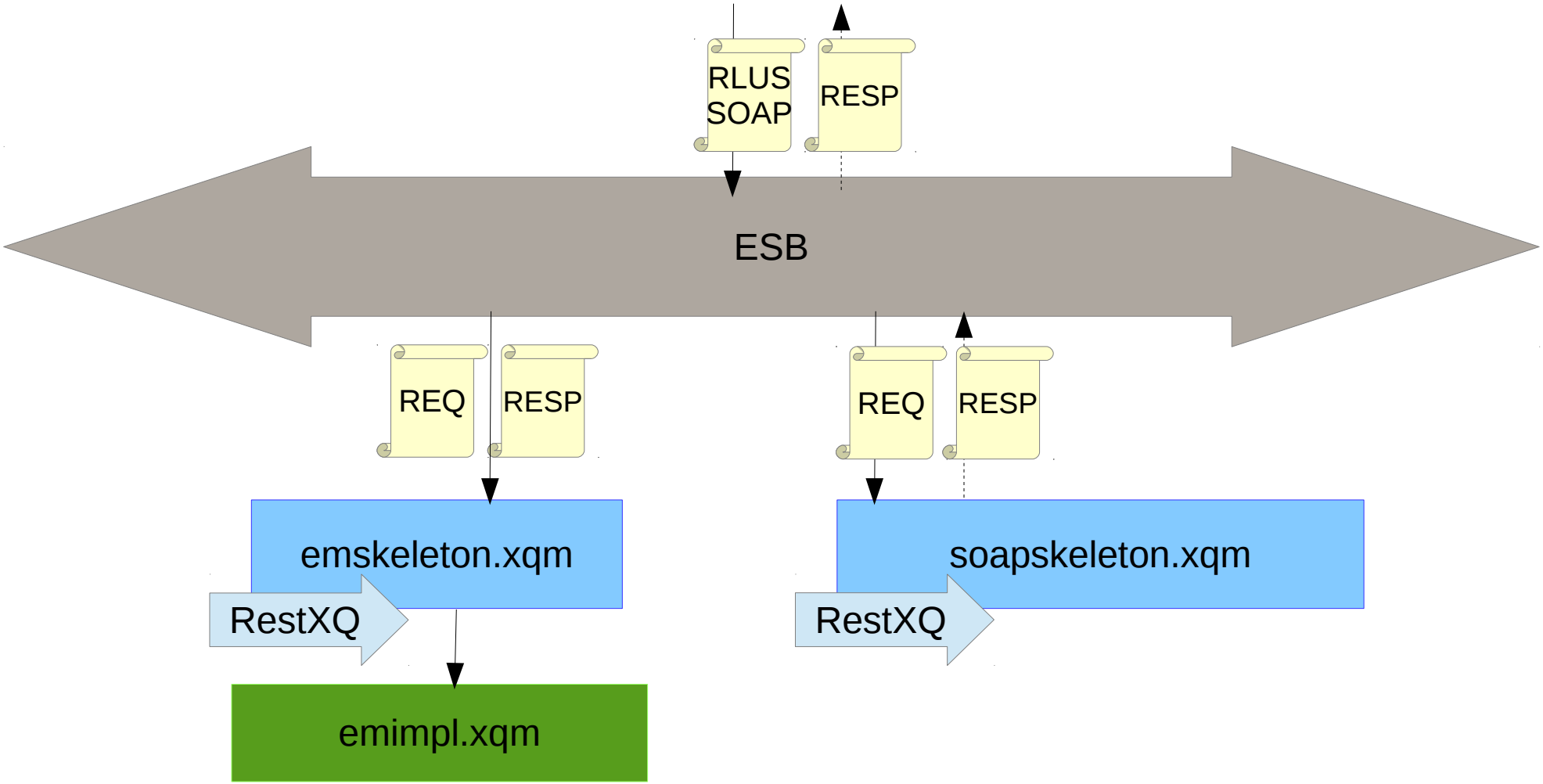
# Example of the Frontend



# BaseX as a workflow router



# The Event Manager



# Last but not least … Xtest

- We have developed a Unit Testing tool named Xtest to test our services
- It's based on a declarative XML model supporting:
  - Test campaigns and single Test Cases
  - A minimal set of Control Structures (env variables, grouping, linking, iteration)
- It's able to do the following phases of testing:
  - Test execution
  - Test arbitration
  - Test reporting
- It leverages:
  - xquery:eval (for env expansion and custom xquery based arbitration)
  - http:send-request (With expath formalism to access requests and responses)
  - validate:xsd-info, xslt:* (for xsd and schematron based arbitration)

# Example (I) a test campaign

```
<testcampaign id="xdwapi_campaign">
    <title>uvabapi functional test</title>
    <env name="basedir">/pathto/dir</env>
    <env name="dbname">accountingrlus</env>
    <env name="project">accounting</env>
    <testcase id="t002" xlink:href="{$basedir}/testcase_cleardb.xml"/>
    <env name=”subjects”>('test1','test2')</env>
    <env name="status">201</env>
    <foreach var="subjectid" expression="{$subjects}">
        <testcase id="t020" xlink:href="{$basedir}/tc1.xml"/>
    </foreach>
    <env name="expectedamount">10.0</env>
    <foreach var="accountid" expression="('account1')">
        <testcase xlink:href="{$basedir}/tc2.xml"/>
    </foreach>
[...]
```

# Example (II) a test case

```
<testcase id="testcase_retrieveAccount">
    <title>Retrieve account {$accountid} with expectation {$status}</title>
    <request name="request" xmlns:http="http://expath.org/ns/http-client">
        <data>
            <http:request method="get" href="{$baseuri}/{$project}/{$accountid}"/>
        </data>
    </request>
    <response>
        <customxq>
            if ($data/@status = $status) then () else  fn:error([...])
        </customxq>
    </response>
    <output name="out1">
        <xsd xlink:href=”{$pathtoxsd}”/>
        <customxq>[...]</customxq>
    </output>
</testcase>
```

# Example (III) Xtest reporting

**Test campaign**

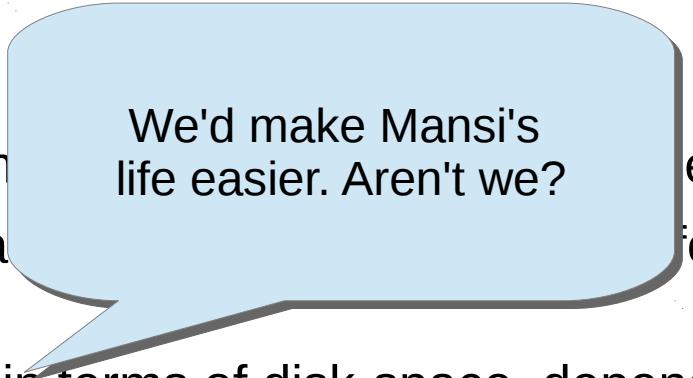
| HSPILOT_INSERT_TEST_RECORD | Test campaign of hspilot IXS project to Upload test data |
|---|---|
| **Testcases** | |
| Total | 5 |
| Success | 5 |
| Failures | 0 |
| Error | 0 |
| **Time Taken** | |
| Total | 3277.805ms |
| Top 3 | • testcase 1 took 808.36 ms<br>• testcase 4 took 599.007 ms<br>• testcase 3 took 582.098 ms |
| Bottom 3 | • testcase 2 took 437.885 ms<br>• testcase 5 took 472.871 ms<br>• testcase 3 took 582.098 ms |

**Test cases**

| | | | | |
|---|---|---|---|---|
| 1 | Test case of the IXS Register Entity with Identity - Database hspilotixs : Inserting pocdhealthcarefacility_101.xml | input<br>request<br>response<br>output | data<br>data<br>data, customxq<br>data | Total : 0 Failure,5 Success<br>Fired Rule(s) : 0<br>Time taken : 808.36ms |
| 2 | Test case of the IXS Register Entity with Identity - Database hspilotixs : Inserting pocdhealthcarefacility_102.xml | input<br>request<br>response<br>output | data<br>data<br>data, customxq<br>data | Total : 0 Failure,5 Success<br>Fired Rule(s) : 0<br>Time taken : 437.885ms |
| 3 | Test case of the IXS Register Entity with Identity - Database hspilotixs : Inserting pocdhealthcarefacility_103.xml | input<br>request<br>response<br>output | data<br>data<br>data, customxq<br>data | Total : 0 Failure,5 Success<br>Fired Rule(s) : 0<br>Time taken : 582.098ms |
| 4 | Test case of the IXS Register Entity with Identity - Database hspilotixs : Inserting pocdhealthcarefacility_104.xml | input<br>request<br>response<br>output | data<br>data<br>data, customxq<br>data | Total : 0 Failure,5 Success<br>Fired Rule(s) : 0<br>Time taken : 599.007ms |

# Conclusions

- We learned about BaseX four years ago (we were using eXist at that time) and now we are using it pervasively (sort of a swiss army knife)
- According to our long experience there is no better opensource alternative of working with the tons of heavily structured XML data which is typical in our domain
- The reasons are:
  - XQuery 3, RESTXQ and all th…es.
  - Small codebase to maintain a…fort. In other words improved workflow.
  - The incredibly small footprint in terms of disk-space, dependencies and simplicity of installation
- I also use BaseX for internal training and evangelism



# Cons and Future

- After decades of XML the healthcare domain is also starting to move towards Json … need to understand if BaseX still can play the main role there
- Exactly in these days we ar… performance issues
  - Very “insertion intensive” sce… …nents inserted per day. We aren't able… …keep the pace anymore
  - We are struggling to experiment with several Xquery and architectural optimizations and I'm looking forward to tell this (hopefully happy-ending) story in XMLPrague 2016 ...

Looking forward to test new UPDINDEX implementation.
I was one of the guys hit by it...

If there are no questions …

I'd like to thank BaseX Community and Christian in particular for this exciting opportunity...

… and all of you for the attention!