THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

HPCC SYSTEMS®

The Download: Community Tech Talks Episode 10

January 18, 2018

LexisNexis® RISK SOLUTIONS

Welcome!

- Please share: Let others know you are here with #HPCCTechTalks
- Ask questions! We will answer as many questions as we can following each speaker.
- Look for polls at the bottom of your screen. Exit full-screen mode or refresh your screen if you don’t see them.
- We welcome your feedback - please rate us before you leave today and visit our blog for information after the event.
- Want to be one of our featured speakers? Let us know! techtalks@hpccsystems.com

Community announcements

- HPCC Systems Platform updates
  - 6.4.6-1 is the latest gold version
  - 6.4.8 RC2 available now
- Reminder: 2018 Summer Internship Proposal Period Open
  - Interested candidates can submit proposals from the Ideas List
  - Visit the Student Wiki for more details
  - Deadline to submit is April 6, 2018
  - Don't delay as some proposals may get accepted earlier
  - Program runs late May through mid August
- 2018 HPCC Systems Summit Community Day
  - October in Atlanta
  - Pre-event workshop, Poster Competition, Public Admission & Sponsorship packages - All returning this year!

Dr. Flavio Villanustre
VP Technology
RELX Distinguished Technologist
LexisNexis® Risk Solutions
Flavio.Villanustre@lexisnexisrisk.com

Today's speakers

Featured Community Speaker

Chris Gropp

PhD Candidate

Clemson University

cgropp@g.clemson.edu

Chris Gropp is a PhD candidate at Clemson University. His research interests include machine learning, high performance computing, and data analysis. Chris is currently working on refining topic modeling approaches to text analysis, both by improving the algorithms themselves and by developing new methods to analyze output. He is also working with a number of other researchers to apply existing tools to new domains.

CLEMSON School of COMPUTING

Today's speakers

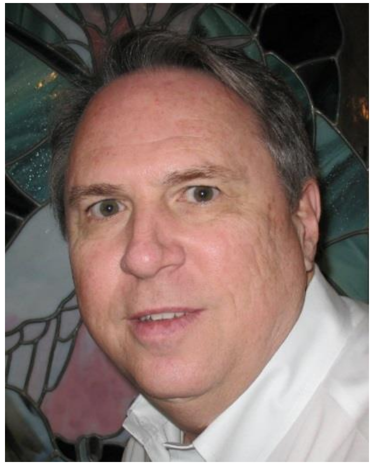
Rodrigo Pastrana

Software Architect

LexisNexis Risk Solutions

Rodrigo.Pastrana@lexisnexisrisk.com

Rodrigo is an Architect with the HPCC systems supercomputer focusing in platform integration and plug-in development. He has been a member of the HPCC core technology team for over five years and a member of the LexisNexis team for seven. Rodrigo is the principle developer of WsSQL, the HPCC JDBC connector, the HPCC Java APIs library and tools, and the Dynamic ESDL component. He has more than fifteen years of experience in design, research and development of state of the art technology including IBM's embedded text-to-speech and voice recognition products, Eclipse's device development environment. Rodrigo holds an MS and BS in Computer Engineering from the University of Florida and during his professional career has filed more than ten patent disclosures through the USPTO.



Richard Taylor

Chief Trainer, HPCC Systems

LexisNexis® Risk Solutions

richard.taylor@lexisnexisrisk.com

Richard Taylor has worked with the HPCC Systems technology platform and the ECL programming language for over 15 years. He is the original author of the ECL documentation, developer and designer of the HPCC Systems Training Courses, and is the Chief Instructor for all classroom and remote based training.

THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

HPCC SYSTEMS®

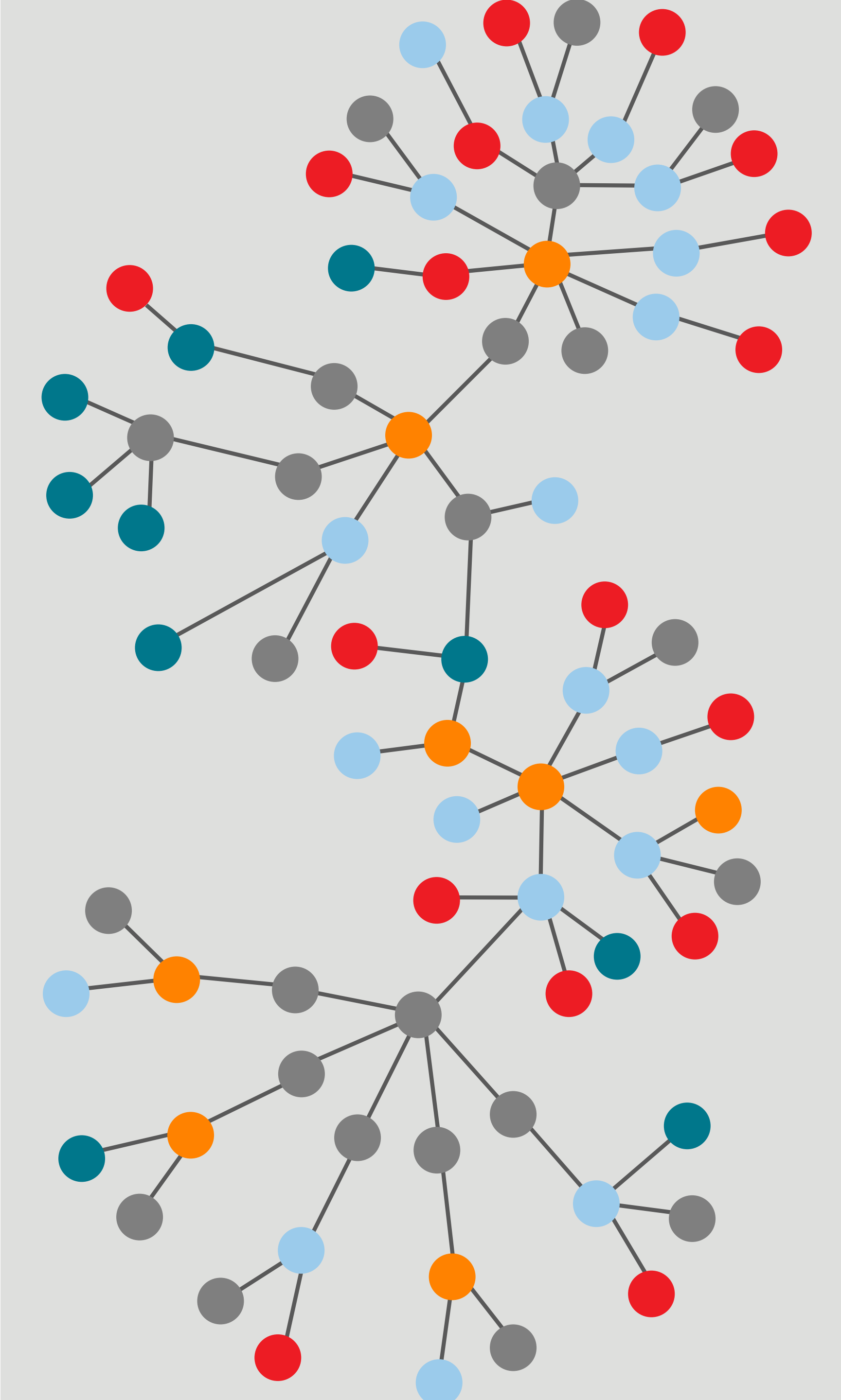
Asking the Right Questions with Machine Learning

Chris Gropp
PhD Candidate
Clemson University

CLEMSON
School of COMPUTING

Quick poll:
How long do you spend choosing a method to solve a problem?

See poll on bottom of presentation screen



HPCC SYSTEMS®

Overview

- No Silver Bullets – Be Careful What You Ask For
- Brief Introduction to Topic Models
  - Latent Dirichlet Allocation (LDA)
  - Dynamic Topic Models (DTM)
- The Quest for a Parallel Dynamic Topic Model
  - Parallelize DTM?
  - Step Back – What do we actually need?
  - Clustered Latent Dirichlet Allocation (CLDA)
- General Advice
  - Identify Requirements
  - How can I evaluate success?
  - Do my methods satisfy those requirements?

No Silver Bullets – Example Problem

- Suppose you have a text reading application.
  - The client wants to create something to read text to the vision impaired.
  - You have been contracted to convert images taken by a camera into text.

- Certainly, accuracy is important to this application, right?
  - Your client agrees!
  - Create the most accurate system you can.

No Silver Bullets – The “Solution”

- Most accurate text reading method I can think of:
  - Take the image and email it to 100 grad students.
  - Offer them $5 starbucks gift cards if they transcribe the text in the image for you.
  - Once you get enough responses that agree with each other, send that back to the device.
  - If you don’t get timely consensus, keep sending it to more people.

- Perfect accuracy, and therefore a perfect solution, right?

No Silver Bullets

- “Solution” scores well on the metric, but does not actually solve the problem
  - Nowhere near real time
  - Prohibitively expensive

- While this example is exaggerated, this danger is omnipresent

- Make sure you know what you actually need!

Brief Introduction to Topic Models – Latent Dirichlet Allocation

- Documents are assumed to be created via a generative process
    - For each word:
        - Sample a document's topic mixture to choose a topic
        - Sample the chosen topic to choose a word from the vocabulary
    - Repeat until document is complete
- Infer latent topics and topic mixtures from observed documents
    - Use variational inference or Gibbs sampling
    - Iterate over documents and modify prior estimates until satisfactorily converged

Brief Introduction to Topic Models – Dynamic Topic Models

- Modify Key Assumption of Latent Dirichlet Allocation
  - Suppose that documents are not all generated simultaneously
  - Separate documents into discrete timesteps
  - Each timestep has a distinct version of each topic
- What do we get:
  - Evolution of topics over time
  - Allow for language to change with new related concepts
  - Determine which topics are most important at each timestep
- Inferring this is hard
  - Each topic is linked to the version of it from the previous timestep
  - Complicates parallelization due to data dependencies
  - Time distribution and word distributions don’t play nicely with each other

Towards a Parallel Dynamic Topic Model – Obvious Approach

- Parallelize traditional DTM algorithm?
  - Lots of data dependency
  - Original code not designed for performance
  - Much more difficult than initially thought

Towards a Parallel Dynamic Topic Model – Requirements

- Hang on, do we actually need to parallelize traditional DTM?

- What we need:
  - Fast, presumably parallel code
  - Extract topics from distinct timesteps
  - Express information about topic evolution
- Parallel DTM would do that, but there's another way

Towards a Parallel Dynamic Topic Model – CLDA

- Rather than keep topics linked during inference, link them afterwards
- Infer all the topics with only local information, allowing for easy parallelism

- **Clustered Latent Dirichlet Allocation**
    - Run LDA independently on each timestep
    - Cluster resulting topics
    - Evaluate clusters the way you would dynamic topics

Towards a Parallel Dynamic Topic Model – CLDA (continued)

- How'd we do?
  - Two orders of magnitude faster than DTM
  - Provides more detailed topic evolution information than DTM
  - Allows for topics to arise and die off, unlike DTM
  - For more details, check out the paper: https://arxiv.org/abs/1610.07703

- CLDA implementations:
  - Original in python and C, available at https://github.com/groppcw/CLDA
  - Active project to construct CLDA using ECL!

General Application – Identify Requirements

- Don’t start with a tool you want to use; it might not be the right one!

- What is the problem you want to solve?
  - Start with the big picture
  - What is the final application?

- What does your solution have to look like?
  - What kinds of input do you have?
  - What does your output need to contain?
  - What other constraints are there? (Speed, memory, security, etc)

General Application – How do you evaluate success?

- What you can measure easily and what you need to measure may be different!

- Remember the problem:
  - What is the application evaluated on?
  - How do you distinguish a good solution from a bad one?
  - What can a good solution do that a bad solution can't?
  - How can you measure the difference?

General Application – Choose a Method

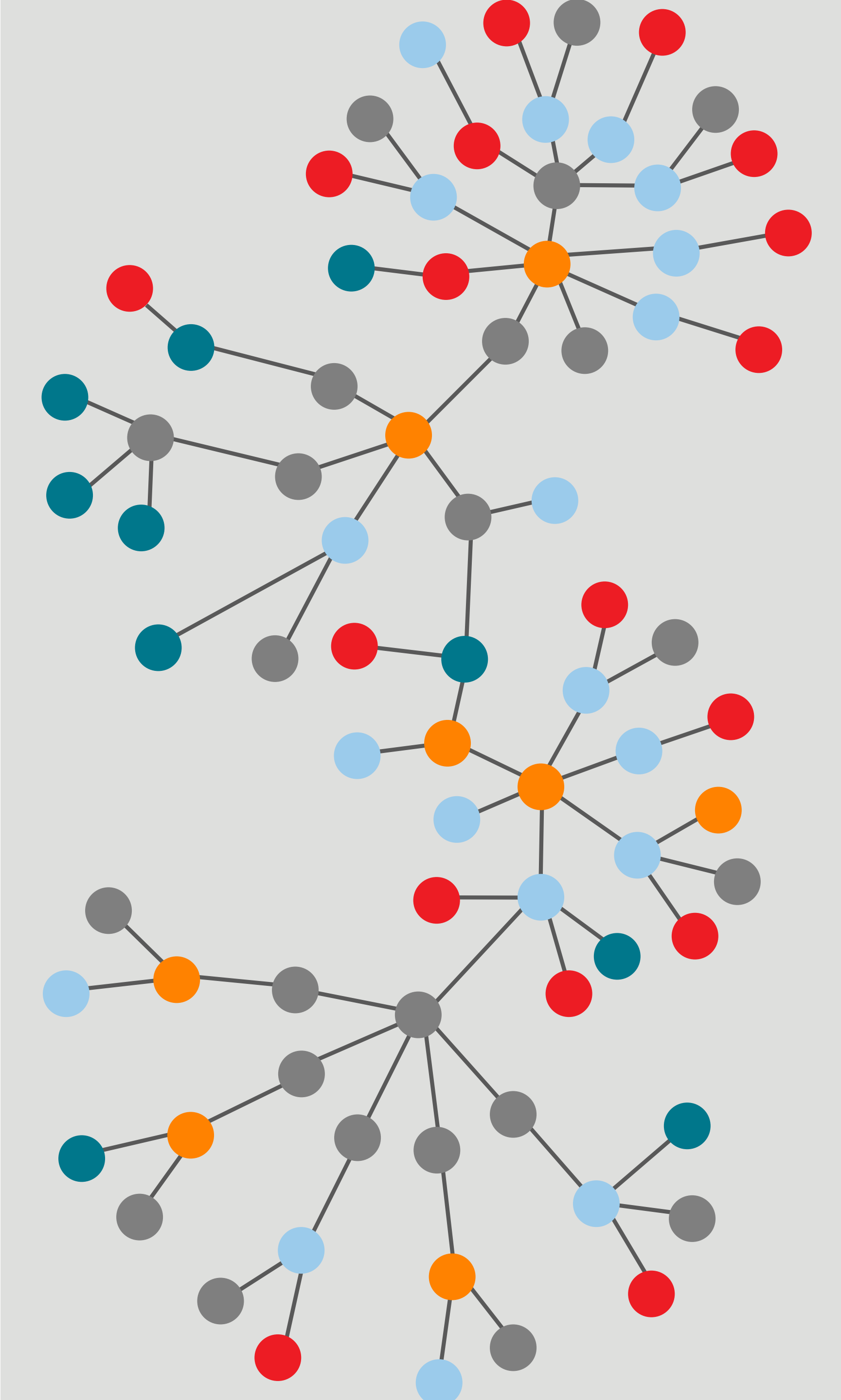
- Now that you know what you need, you can finally pick (or create) a method

- Look at your requirements:
  - Which methods process the type of input you have, and produce the type of output you need?
  - Which methods are within the constraints of your application?

- Look at your metrics:
  - Which candidate methods perform best where you need them to excel?
  - What trade-offs do the candidate methods have? Which best satisfy your priorities?

Quick poll:
Did you previously think a lot about what methods you used? Will you in the future?

See poll on bottom of presentation screen



HPCC SYSTEMS®

Questions?

Chris Gropp
PhD Candidate
Clemson University
cgropp@g.clemson.edu

THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

HPCC SYSTEMS®

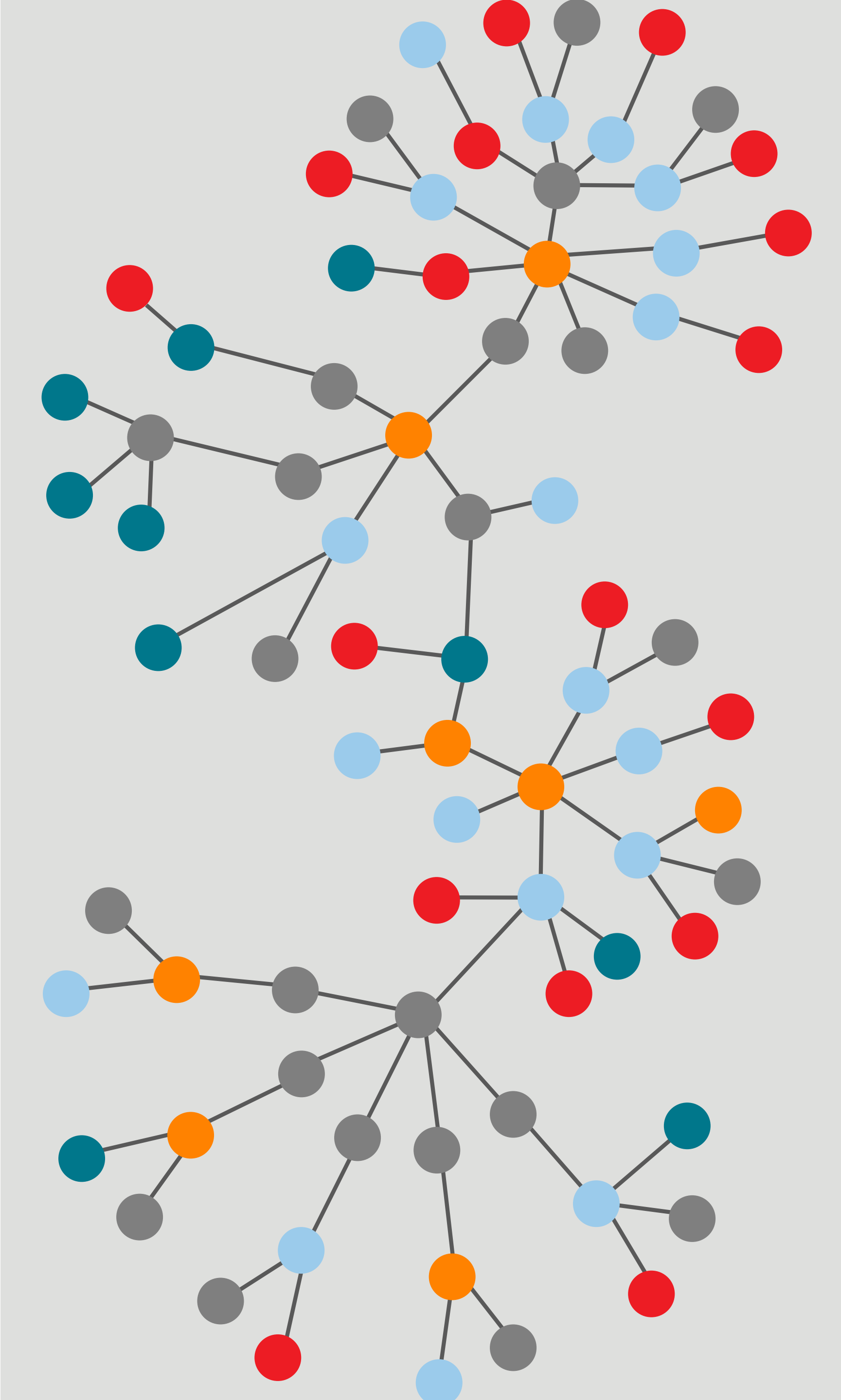
Discover HPCC Systems Web Services Framework for Delivering Query Data

Rodrigo Pastrana
Software Architect
LexisNexis® Risk Solutions

LexisNexis® RISK SOLUTIONS

Quick poll:
Are you involved with projects that deliver data/information to customers?

See poll on bottom of presentation screen



HPCC SYSTEMS®

CustomerFacing
Synchronous
HPCC
JSON
DynamicESDL
Distributed
Cluster
TestPages
SOAP
Queries
Forms
Fast
Accessible
AccessControl
HTTP
Security
WebServices
Transaction
Discovery
HTMLPages
IDE
Binding
Framework
Methods
ESDL
Messages
Automatic
ESP
LDAP
Clients
Interface
XML
REST
JavaScript
Protocol
Logging
Client
JS
C++
ECL
Data
Roxie
WSDL
Contract
OpenSource
BusinessLogic
BigData
Compliant
JAVA

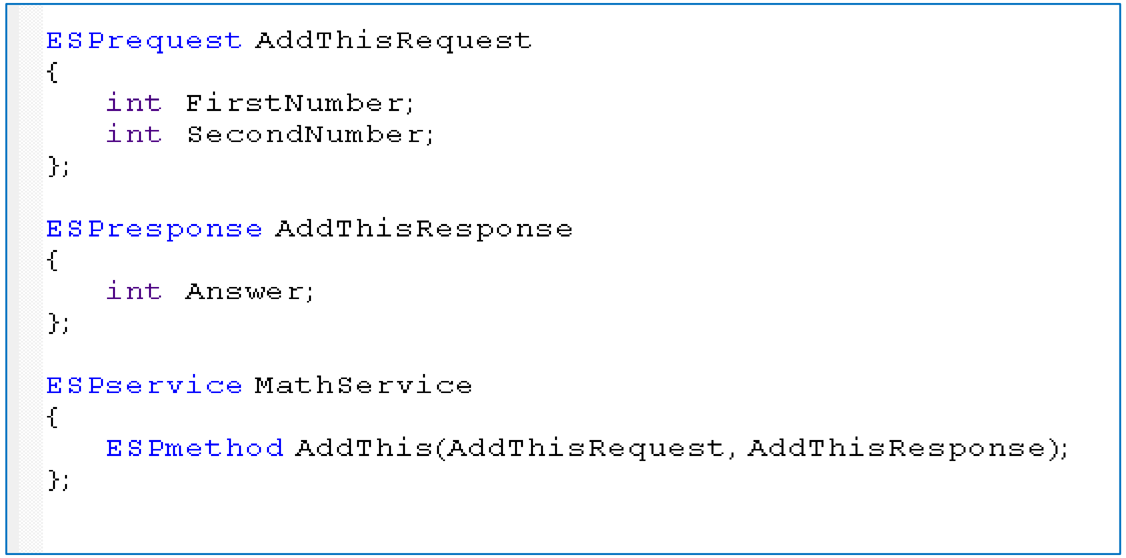
HPCC Systems Web Services Framework – Major Components

ESP Config Tools

HPCC Systems Web Services Framework - ESDL

Enterprise Service Description Language

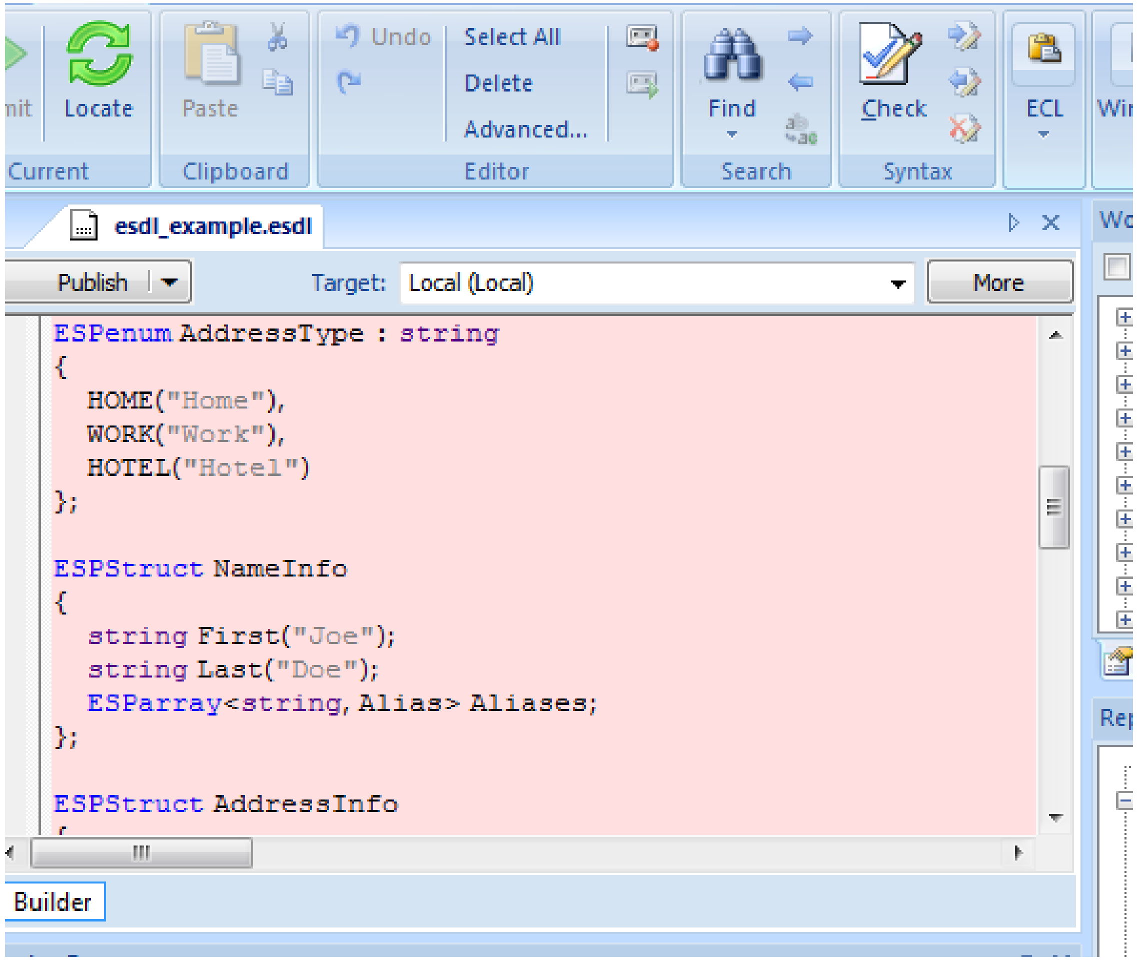
- Define Web Server interface using straight forward constructs
- Creates contract between WS and ECL Published Query
- Powerful version control via intuitive constructs



```
ESPrequest AddThisRequest
{
    int FirstNumber;
    int SecondNumber;
};

ESPresponse AddThisResponse
{
    int Answer;
};

ESPservice MathService
{
    ESPmethod AddThis(AddThisRequest, AddThisResponse);
};
```

HPCC Systems Web Services Framework - IDE

- Language structure support
- Context-aware help documentation
- One touch WS Interface publishing
- Generates ECL structures

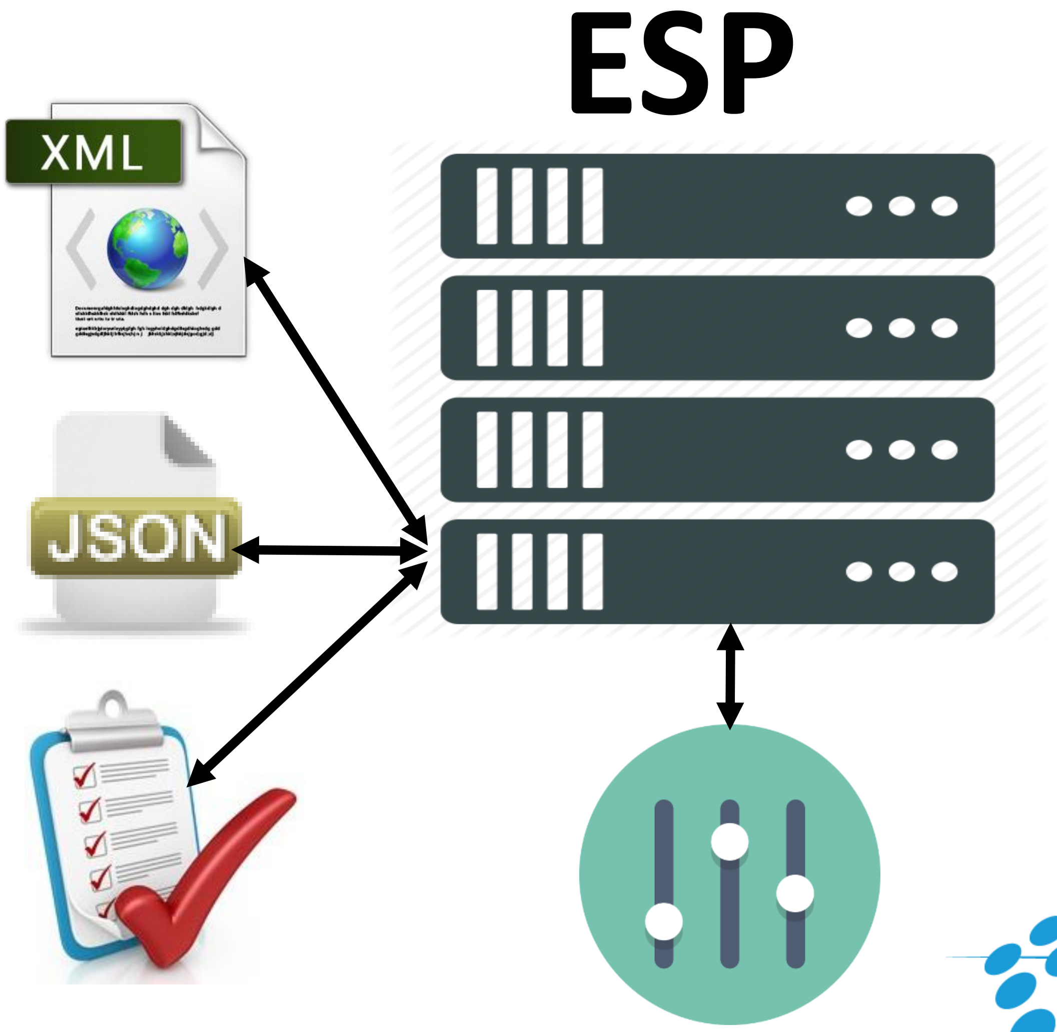


```
ESPenum AddressType : string
{
  HOME("Home"),
  WORK("Work"),
  HOTEL("Hotel")
};

ESPStruct NameInfo
{
  string First("Joe");
  string Last("Doe");
  ESParray<string, Alias> Aliases;
};

ESPStruct AddressInfo
```

HPCC Systems Web Services Framework - ESP

Enterprise Services Platform

- Core of the HPCC WS framework
- Distributed Web Server
- Automatic Service Forms
- Transaction tracing
- Dynamic Configuration
  - DESDL



HPCC Systems Web Services Framework – ESP Plug-in Framework

- Security
  - Abstracts how authorization and authentication are mapped to backend systems.
  - Support various backend systems
- Java
  - Web Service logic can be implemented in Java
- Protocols
  - Non-http based standards, or highly proprietary protocols can be created
  - Easy to migrate fix-len, binary, apps to ESP
- Transaction Logging and archival
  - Adaptive logging server maps transactions to alternate backend architectures
  - Data mapping billing, monitoring, accountability
  - Fault tolerant data persistence.

Security

JAVA

Protocols

Logging

HPCC Systems Web Services Framework – Config and Tools

- Most are utilized by IDE and ESP - but can be accessed directly
- Dynamic interface and configuration updates
- Publish ESDL based interface for public consumption
- Generate ECL, WSDL, Schema, Form HTTP pages, Sample req/resp
- Test SOAP-based requests
- Test ROXIE targeting requests
- Much more…

HPCC Systems Web Services Framework – Create a simple WS

- Let’s create a simple Web Service from scratch
- Math based service providing basic arithmetic functions
- Start with a single operation, add more dynamically
- Outline advanced tasks

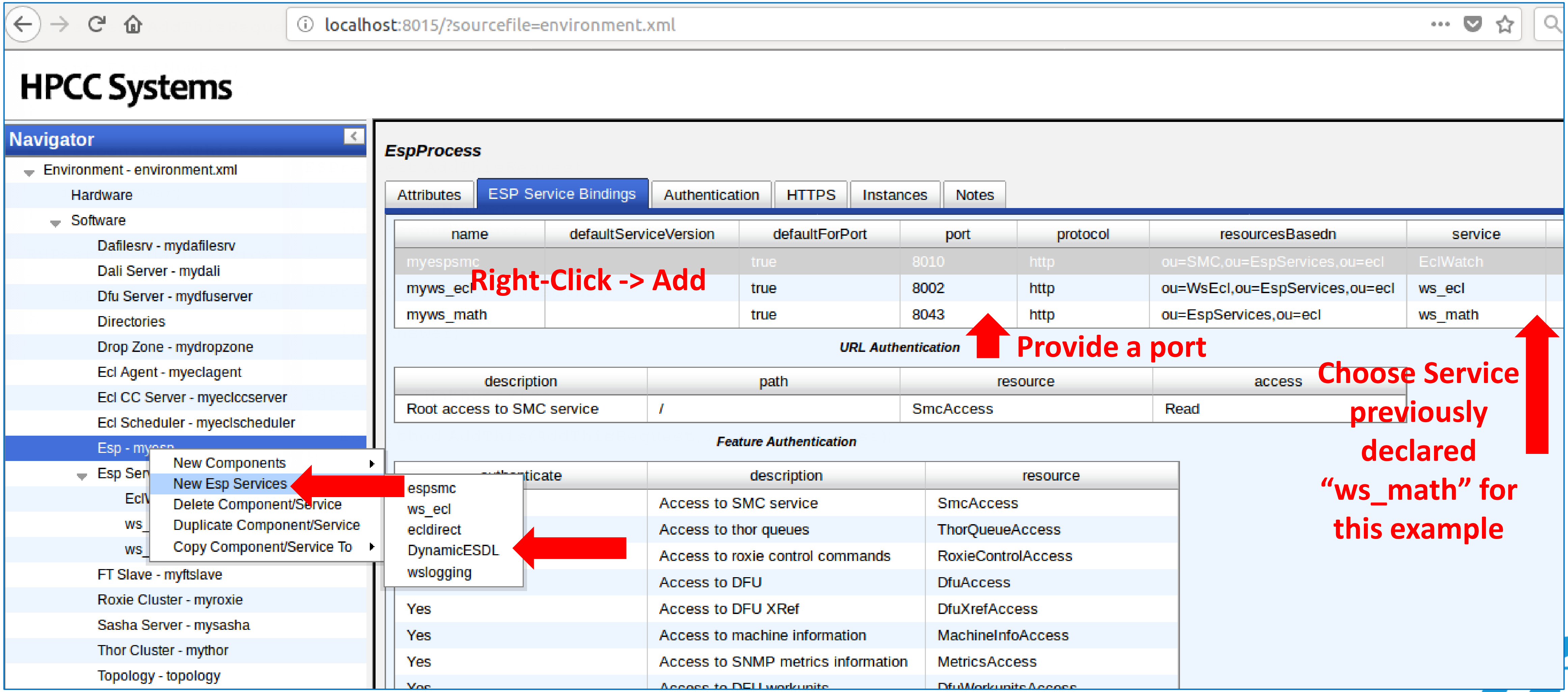
Create a Web Service – Basic steps

1. Declare your WS and reserve a listening port on your ESP
2. Define the WS interface – Fields making up the req/resp
3. Publish the interface – Make the interface available for use
4. Bind that interface to the WS declared earlier
5. Configure your WS – Link it to your back-end query
   - Back-end queries are usually not created by the WS developer
   - We'll go ahead and create a ROXIE query anyway using the generated ECL

Demo - Declare your Web Service on ESP

1. Use ConfigManager to declare a new DynamicESDL service
2. Provide a meaningful name
3. Bind the service to the ESP Process of your choice
4. Bind that interface to the WS declared earlier

Demo - Declare your Web Service on ESP



name	defaultServiceVersion	defaultForPort	port	protocol	resourcesBasedn	service
myespsmc		true	8010	http	ou=SMC,ou=EspServices,ou=ecl	EclWatch
myws_ecl		true	8002	http	ou=WsEcl,ou=EspServices,ou=ecl	ws_ecl
myws_math		true	8043	http	ou=EspServices,ou=ecl	ws_math

URL Authentication

description	path	resource	access
Root access to SMC service	/	SmcAccess	Read

Feature Authentication

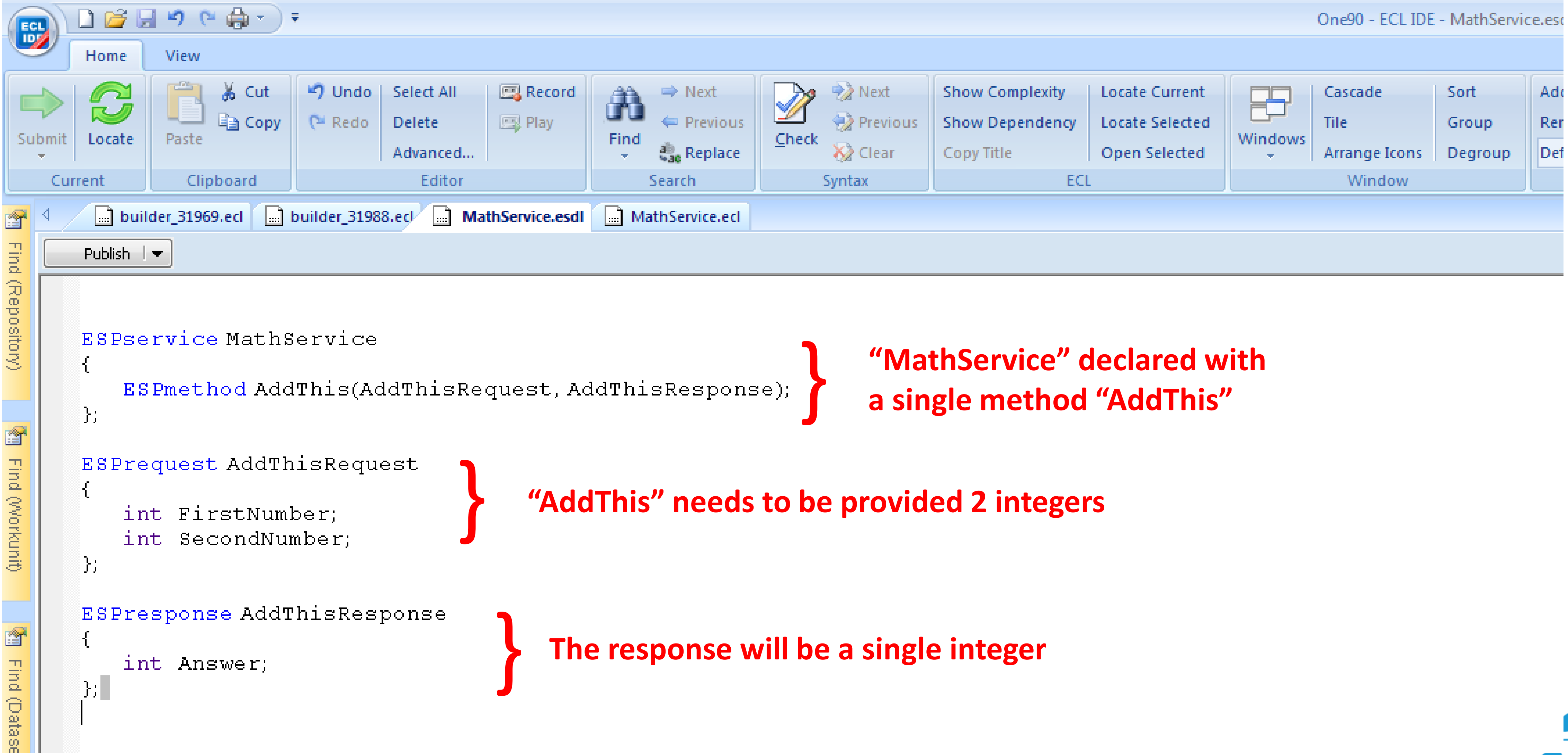
authenticate	description	resource
	Access to SMC service	SmcAccess
	Access to thor queues	ThorQueueAccess
	Access to roxie control commands	RoxieControlAccess
	Access to DFU	DfuAccess
Yes	Access to DFU XRef	DfuXrefAccess
Yes	Access to machine information	MachineInfoAccess
Yes	Access to SNMP metrics information	MetricsAccess
Yes	Access to DFU workunits	DfuWorkunitsAccess

Right-Click -> Add

Provide a port

Choose Service previously declared “ws_math” for this example

Demo - Define your Web Service's interface



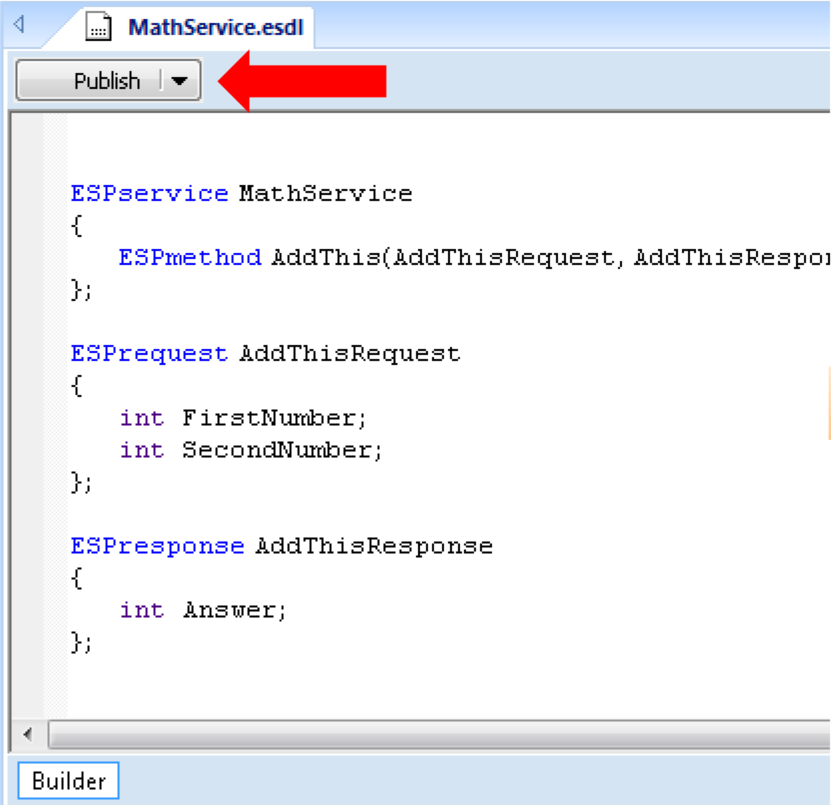
```
ESPservice MathService
{
    ESPmethod AddThis(AddThisRequest, AddThisResponse);
};

ESPrequest AddThisRequest
{
    int  FirstNumber;
    int  SecondNumber;
};

ESPresponse AddThisResponse
{
    int  Answer;
};
```

"MathService" declared with a single method "AddThis"

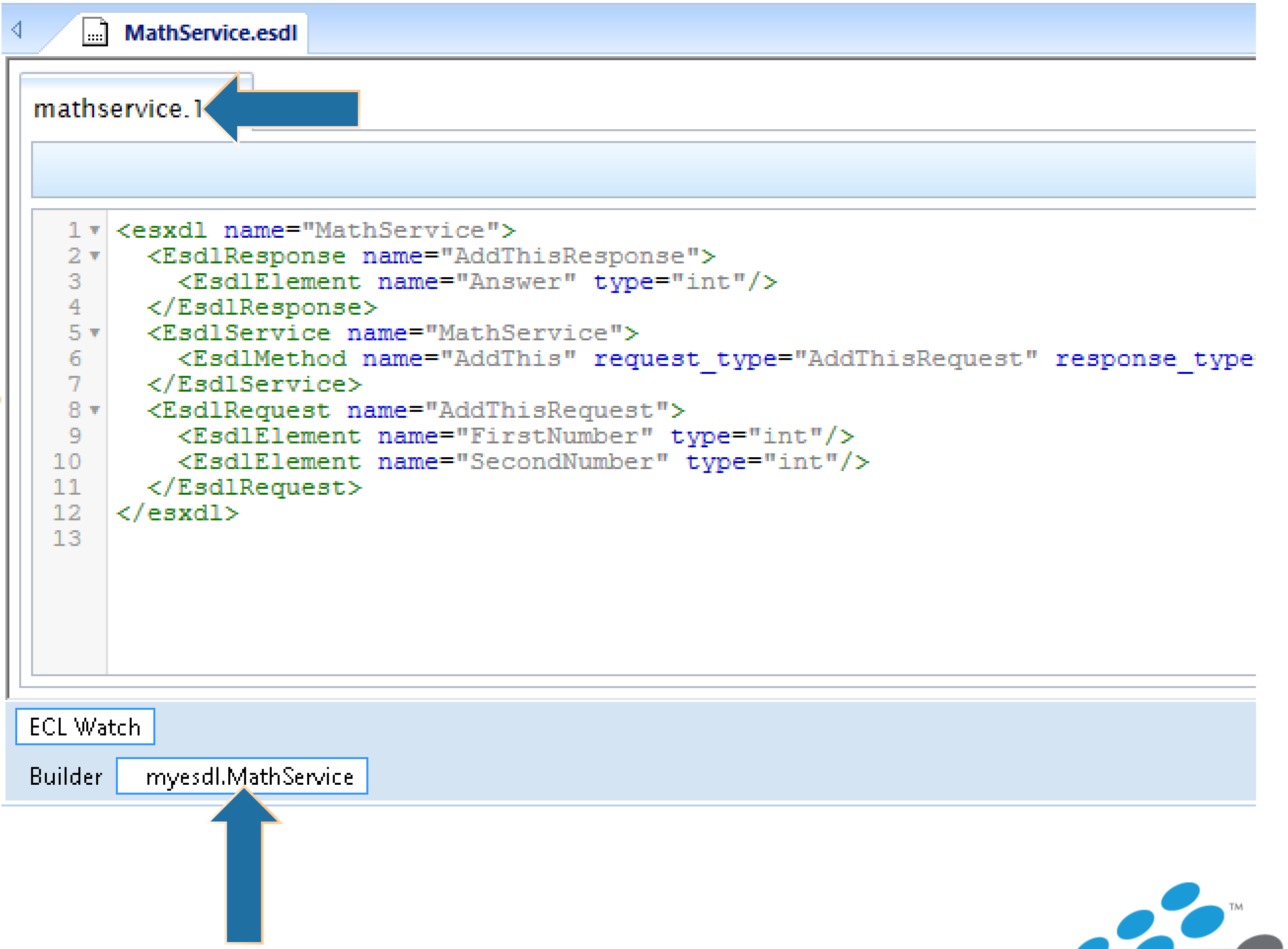
"AddThis" needs to be provided 2 integers

The response will be a single integer

Demo - Publish the interface

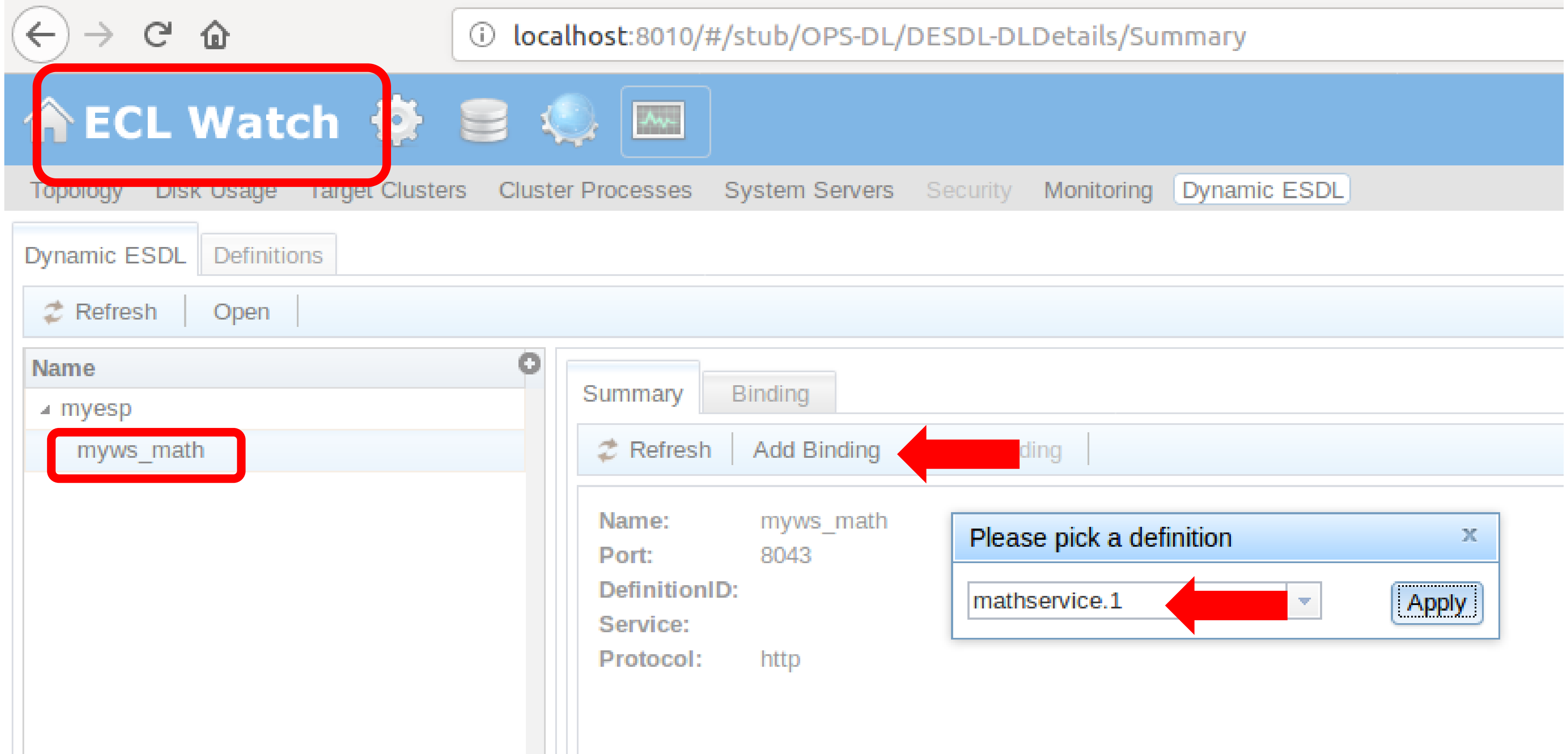


```
ESPservice MathService
{
    ESPmethod AddThis(AddThisRequest, AddThisRespo
};

ESPrequest AddThisRequest
{
    int FirstNumber;
    int SecondNumber;
};

ESPresponse AddThisResponse
{
    int Answer;
};
```



```
<esxdl name="MathService">
  <EsdlResponse name="AddThisResponse">
    <EsdlElement name="Answer" type="int"/>
  </EsdlResponse>
  <EsdlService name="MathService">
    <EsdlMethod name="AddThis" request_type="AddThisRequest" response_type
  </EsdlService>
  <EsdlRequest name="AddThisRequest">
    <EsdlElement name="FirstNumber" type="int"/>
    <EsdlElement name="SecondNumber" type="int"/>
  </EsdlRequest>
</esxdl>
```

Demo - Bind that interface to the WS declared earlier

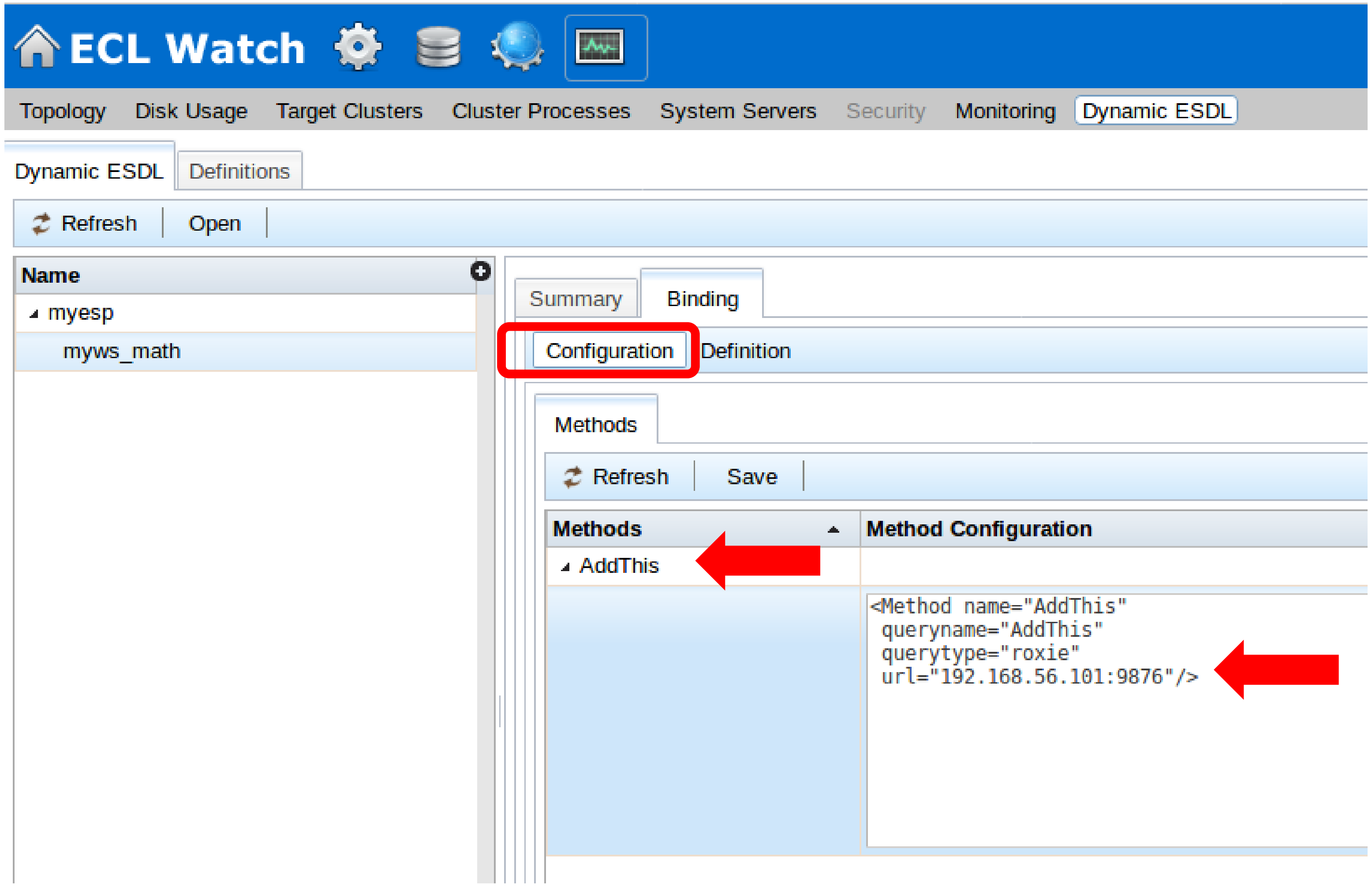


localhost:8010/#/stub/OPS-DL/DESDL-DLDetails/Summary

Name: myws_math
Port: 8043
DefinitionID:
Service:
Protocol: http

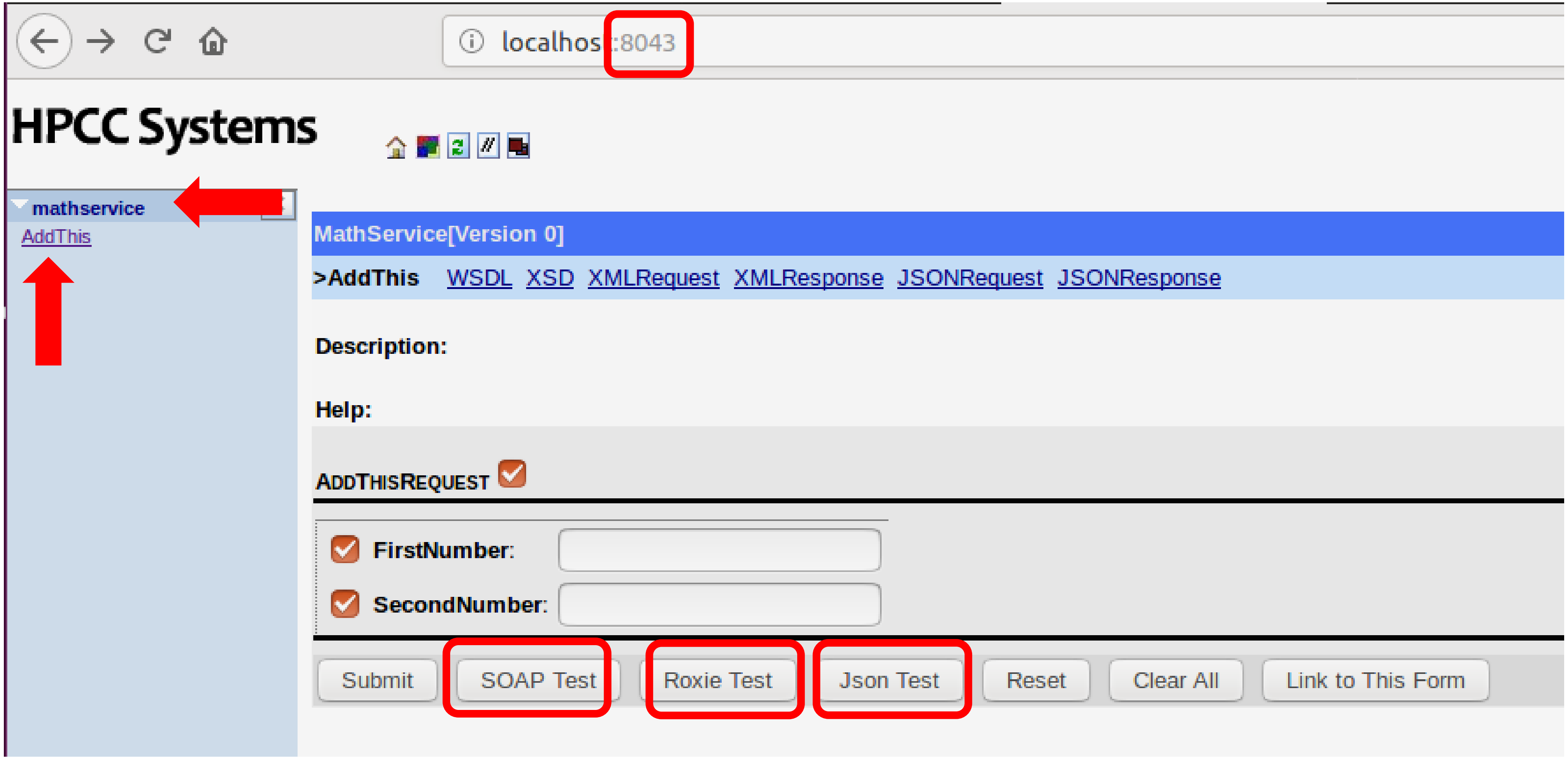
Please pick a definition
mathservice.1

Demo - Bind that interface to the WS declared earlier



```
<Method name="AddThis"
 queryname="AddThis"
 querytype="roxie"
 url="192.168.56.101:9876"/>
```

Demo - We’re done! Let’s look at our math web service



localhos :8043

HPCC Systems

mathservice

AddThis

MathService[Version 0]

>AddThis WSDL XSD XMLRequest XMLResponse JSONRequest JSONResponse

Description:

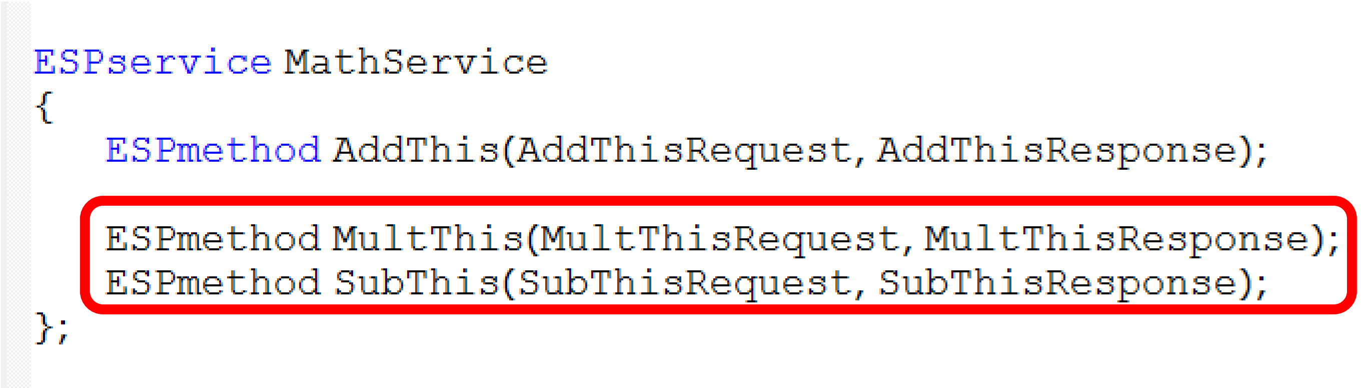
Help:

ADDTHISREQUEST ☑

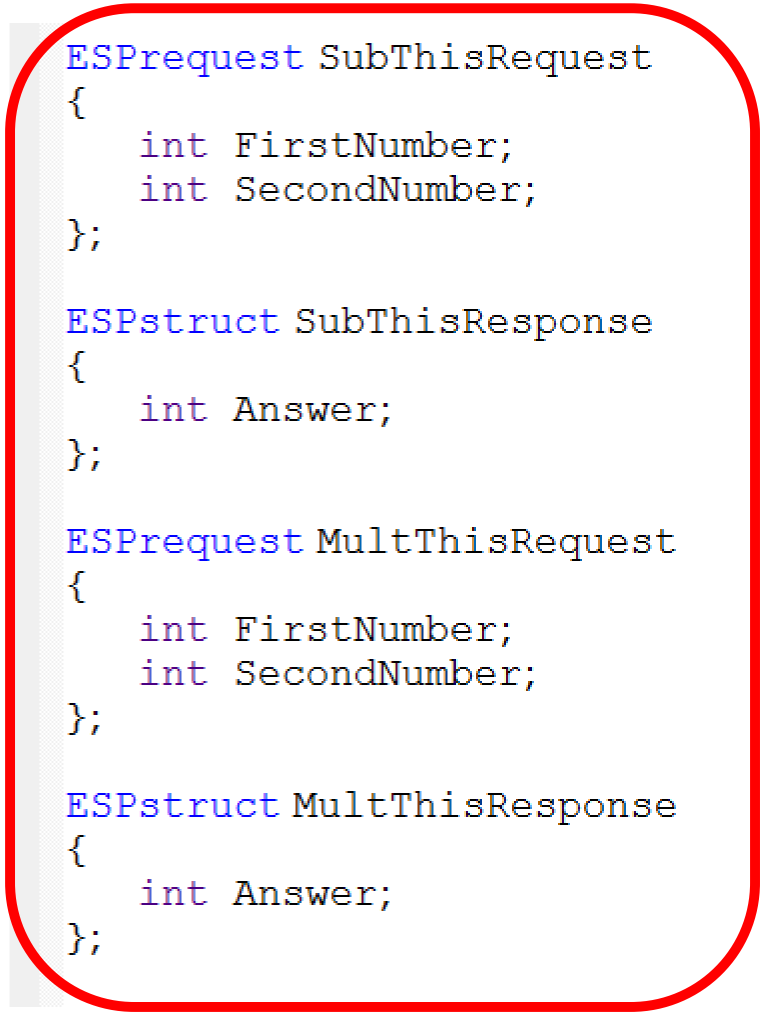
☑ FirstNumber:

☑ SecondNumber:

Demo - Let’s expand the service - Dynamically

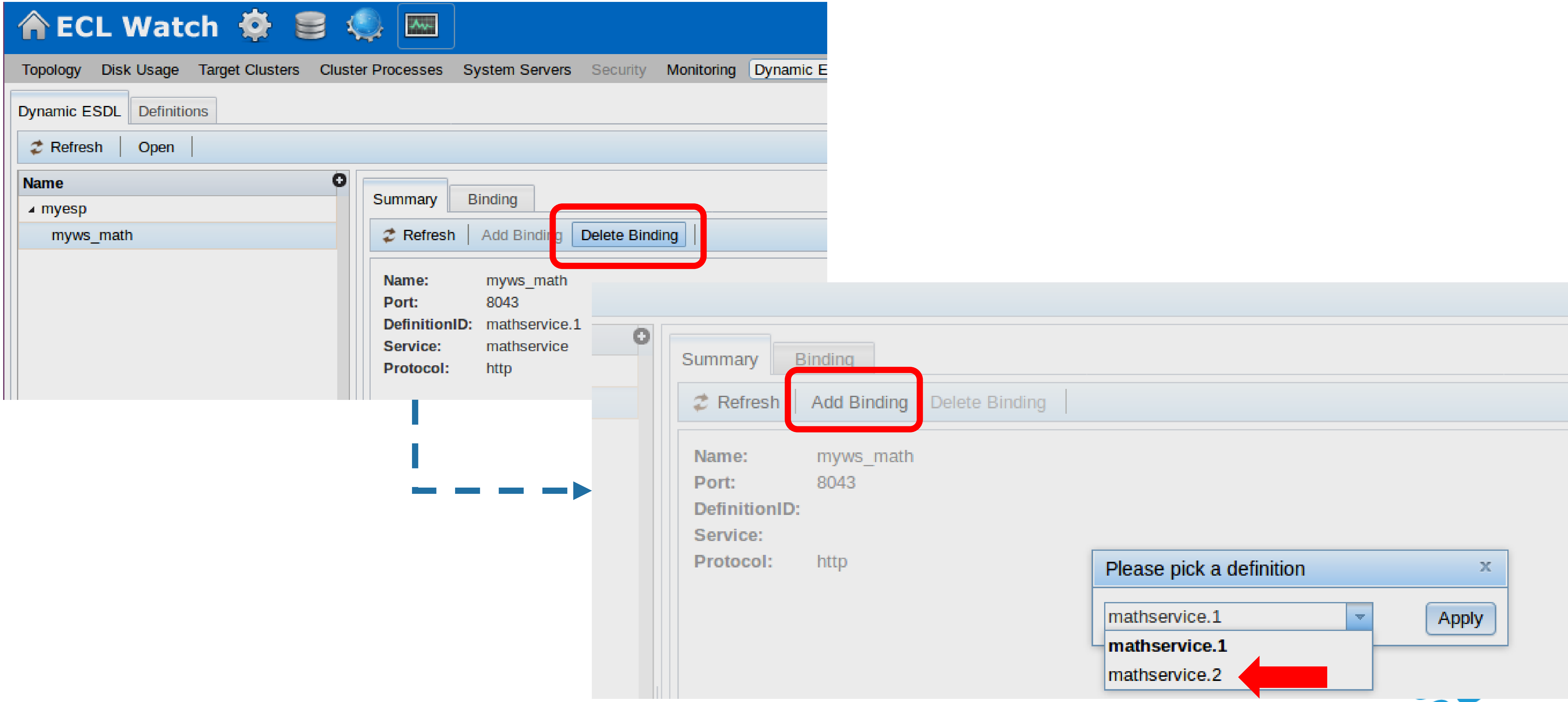


```
ESPservice MathService
{
    ESPmethod AddThis(AddThisRequest, AddThisResponse);

    ESPmethod MultThis(MultThisRequest, MultThisResponse);
    ESPmethod SubThis(SubThisRequest, SubThisResponse);
};
```



```
ESPrequest SubThisRequest
{
    int  FirstNumber;
    int  SecondNumber;
};

ESPstruct SubThisResponse
{
    int  Answer;
};

ESPrequest MultThisRequest
{
    int  FirstNumber;
    int  SecondNumber;
};

ESPstruct MultThisResponse
{
    int  Answer;
};
```

Demo - Let’s expand the service – Publish/Bind

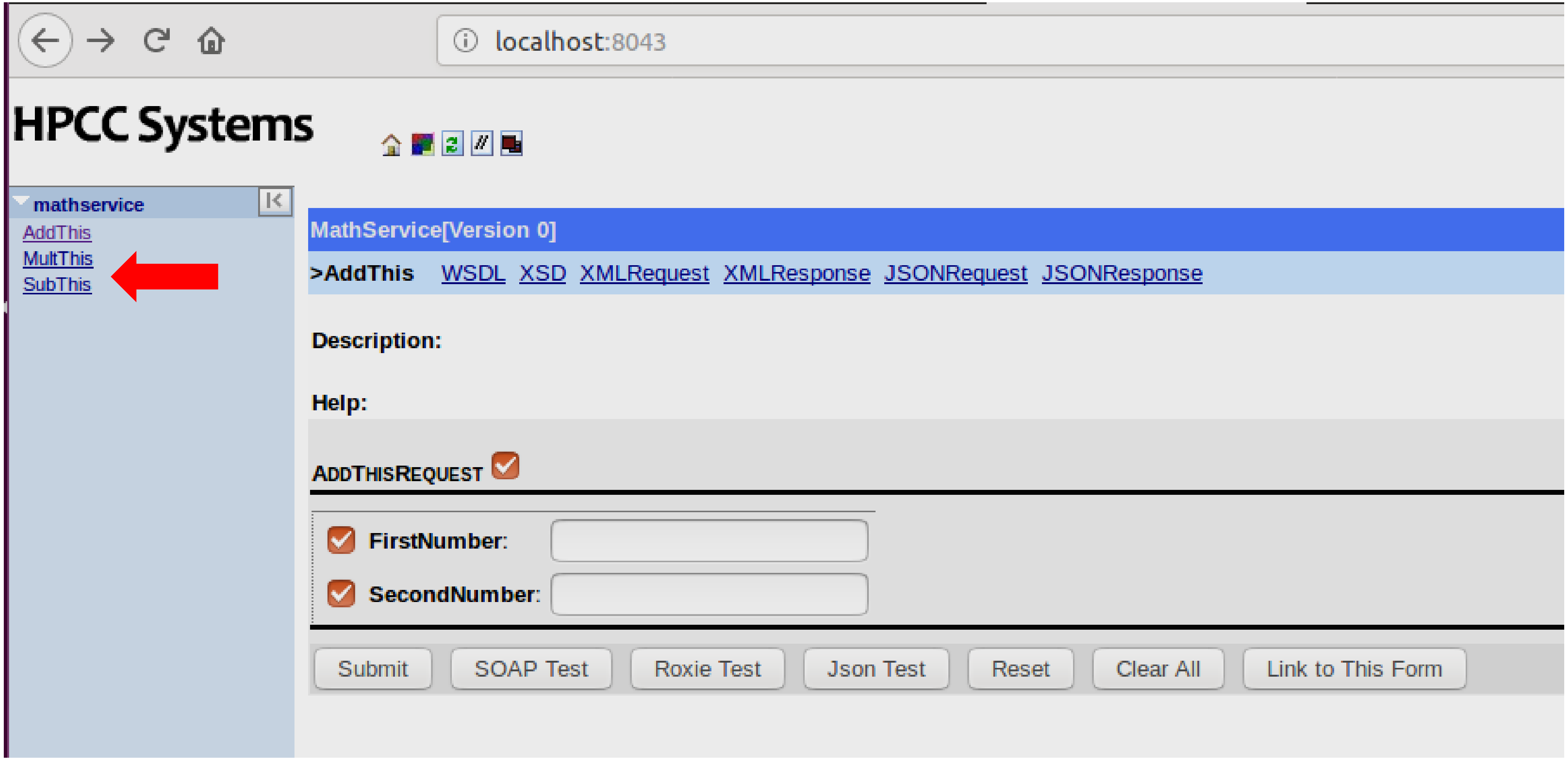


Name: myws_math
Port: 8043
DefinitionID: mathservice.1
Service: mathservice
Protocol: http

Name: myws_math
Port: 8043
DefinitionID:
Service:
Protocol: http

mathservice.1
mathservice.1
mathservice.2

Demo - Let’s expand the service – Publish/Bind



localhost:8043

HPCC Systems

mathservice

AddThis

MultThis

SubThis

MathService[Version 0]

>AddThis WSDL XSD XMLRequest XMLResponse JSONRequest JSONResponse

Description:

Help:

ADDTHISREQUEST ☑

☑ FirstNumber:

☑ SecondNumber:

Further tasks to consider

- Security is configured at the ESP Process level
  - LDAP, HTPassWord out of the box, custom security plugins supported
  - Youtube video -> https://www.youtube.com/watch?v=lNVwEOFkKgY
- Java Implementation
  - Make Java classes available to the ESP “/opt/HPCCSystems/classes/”
  - Create your WS in ESDL and follow process discussed earlier
  - Bind WS method to Java class method:

```
<Methods>
    <Method name="JavaEchoPersonInfo" querytype="java“
                javamethod ="EsdlExample.EsdlExampleService.JavaEchoPersonInfo"/>
</Methods>
```

Further tasks to consider - continued

- Transaction level logging – For billing, accounting, monitoring, etc
  - Fault-tolerant transactional information mapping to alternate backend architectures
  - Support for mysql, Cassandra out of box – Plug-ins supported
- Create Client Application
  - Application that consumes your Web Service output
  - Automatically generate C++ client stub code (from 2 lines of ESDL)!
- Protocols
  - You might be required to support non-HTTP protocol
  - ESP provides plug-in framework for custom protocols
  - Custom protocol and service logic are separate

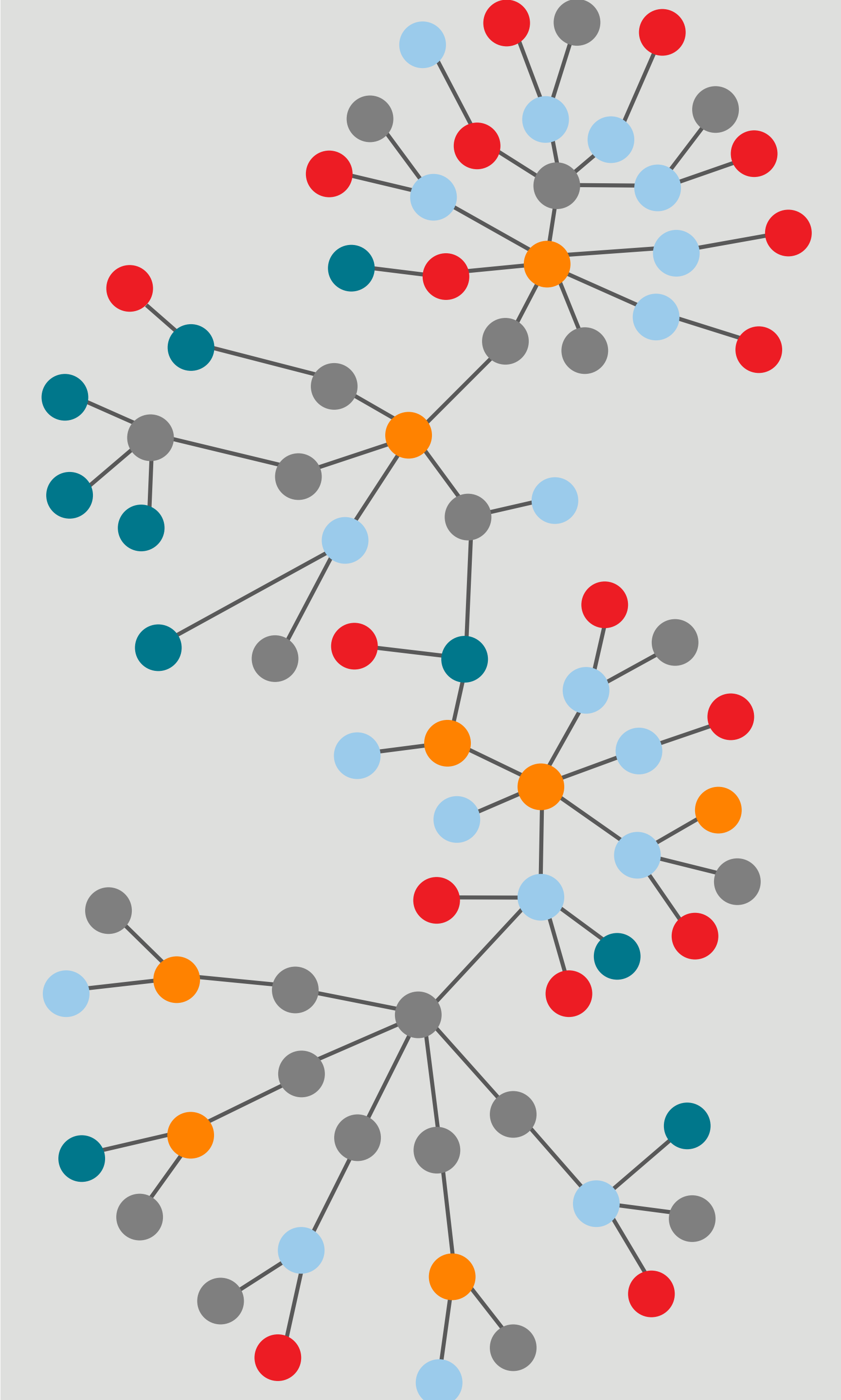
Thank you!

- Lots of documentation on hpccsystems.com portal
  - ESP Overview
    - https://hpccsystems.com/enterprise-services/modules/esp
  - Configuration Manager
    - http://cdn.hpccsystems.com/releases/CE-Candidate-%7Bcurrent_version%7D/docs/UsingConfigManager-%7Bcurrent_version_full%7D.pdf
  - ESDL Language
    - http://cdn.hpccsystems.com/releases/CE-Candidate-%7Bcurrent_version%7D/docs/ESDL_LangRef-%7Bcurrent_version_full%7D.pdf
  - Dynamic ESDL
    - http://cdn.hpccsystems.com/releases/CE-Candidate-%7Bcurrent_version%7D/docs/DynamicESDL-%7Bcurrent_version_full%7D.pdf
  - Security Manager
    - http://cdn.hpccsystems.com/releases/CE-Candidate-%7Bcurrent_version%7D/docs/HPCCSecurityManagerGuide-%7Bcurrent_version_full%7D.pdf
- Code base available on github:
  - https://github.com/hpcc-systems/HPCC-Platform/tree/master/esp

HPCC SYSTEMS®

Quick poll:
What do you consider to be the most important aspect of a Web service?

See poll on bottom of presentation screen



Questions?

Rodrigo Pastrana
Software Architect
LexisNexis Risk Solutions
Rodrigo.Pastrana@lexisnexisrisk.com

THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

ECL Tips and Cool Tricks

HPCC SYSTEMS®

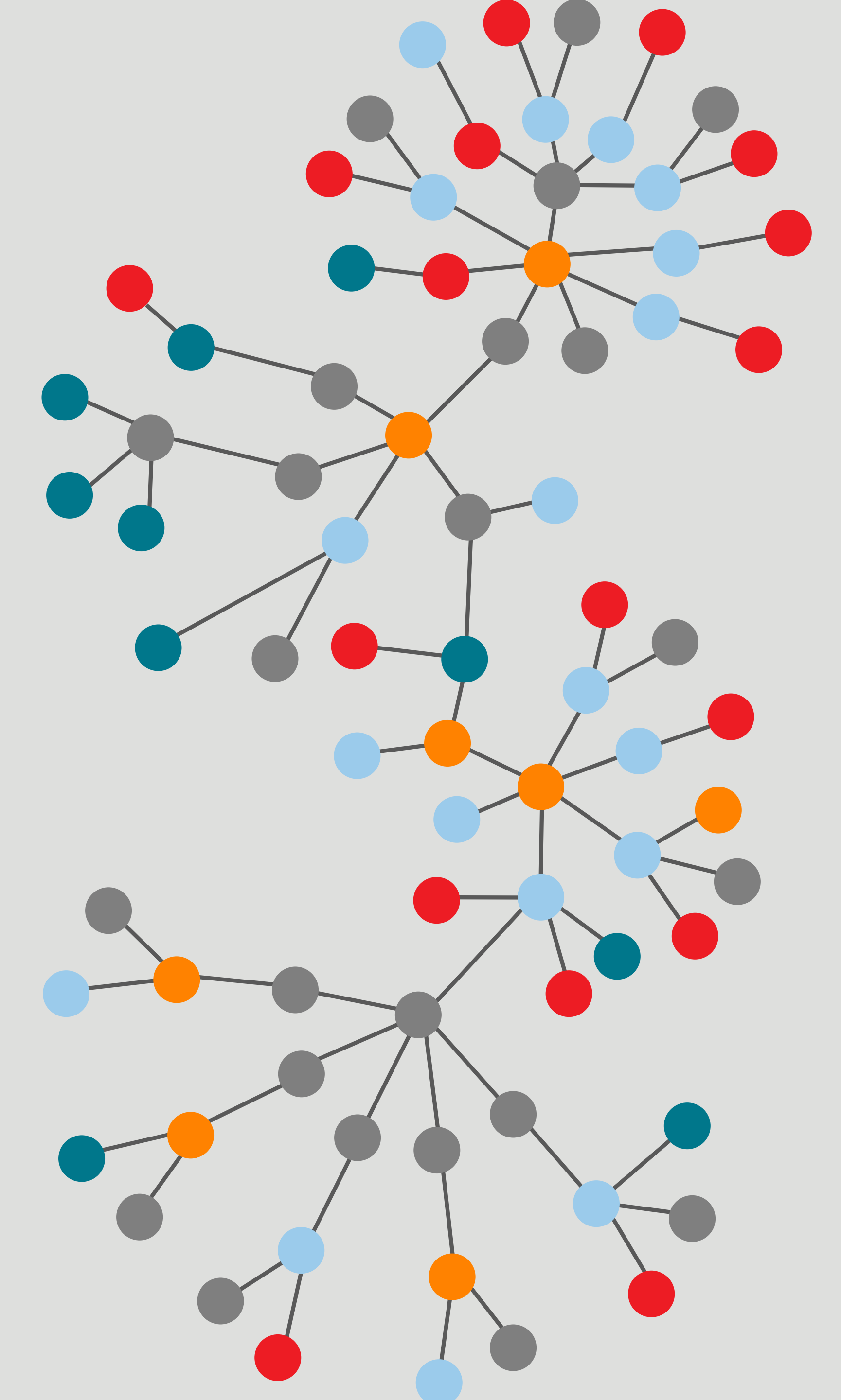
Richard Taylor
Chief Trainer, HPCC Systems

LexisNexis®
RISK SOLUTIONS

Quick poll:

Have you used PARSE already in your ECL code?

See poll on bottom of presentation screen



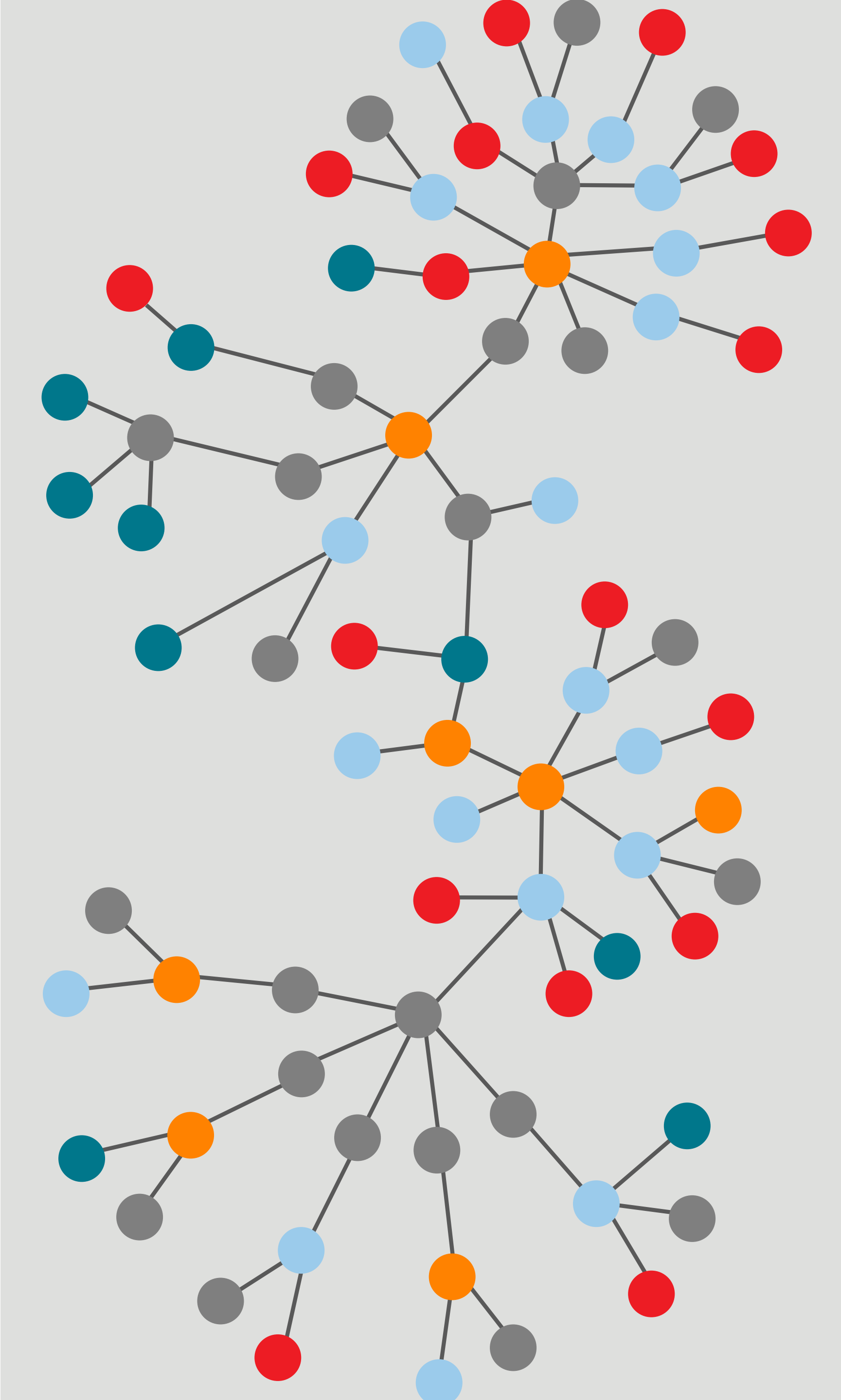
HPCC SYSTEMS®

Demo

Let’s take a look at how the PARSE function works…

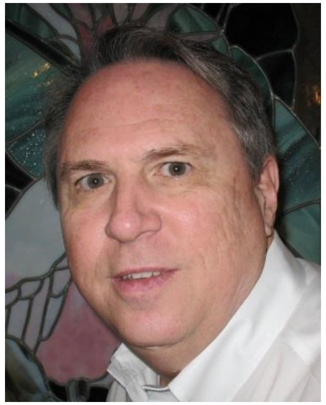
Quick poll:
Will these techniques be useful to you in non-date parsing code?

See poll on bottom of presentation screen



HPCC SYSTEMS®

Questions?



Richard Taylor
Chief Trainer, HPCC Systems
richard.taylor@lexisnexisrisk.com

Submit a talk for an upcoming episode!

- Have a new success story to share?
- Want to pitch a new use case?
- Have a new HPCC Systems application you want to demo?
- Want to share some helpful ECL tips and sample code?
- Have a new suggestion for the roadmap?
- Be a featured speaker for an upcoming episode! Email your idea to Techtalks@hpccsystems.com
- Visit The Download Tech Talks wiki for more information: https://wiki.hpccsystems.com/display/hpcc/HPCC+Systems+Tech+Talks

Mark your calendar for the February 15 Tech Talk -
Topics include the latest development on our Spark connectors!
Watch our Events page for details.

HPCC SYSTEMS®

Thank You!

HPCC SYSTEMS®

RELX Group

A copy of this presentation will be made available soon on our blog:
hpccsystems.com/blog