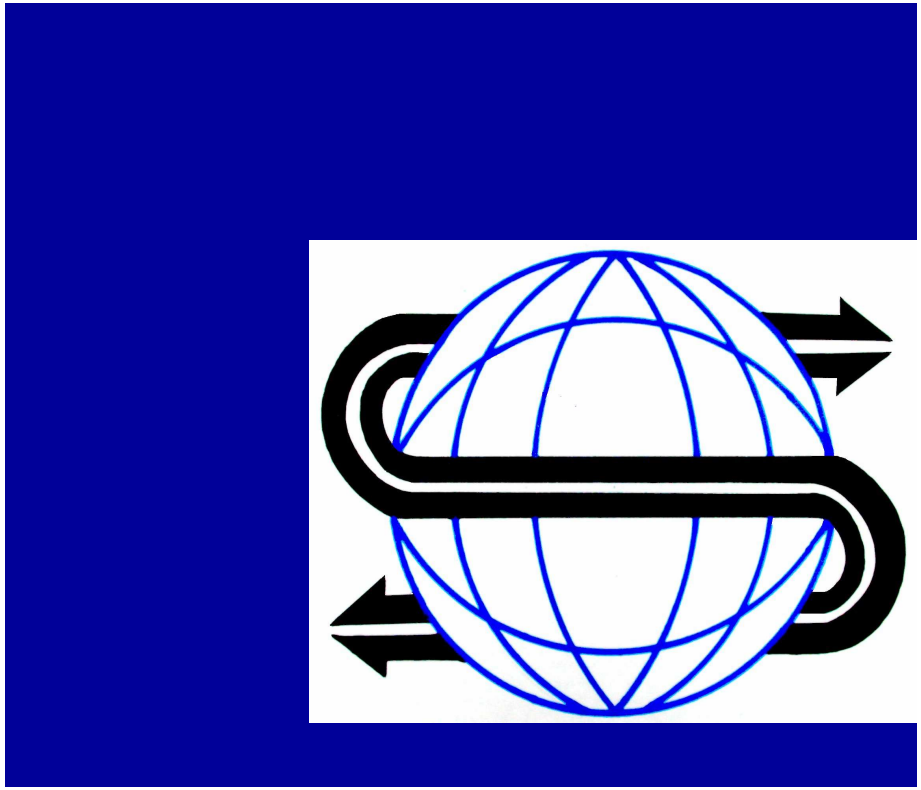


Optimization Services (OS)

-- A Framework for Optimization Software

-- A Computational Infrastructure

-- The Next Generation NEOS

-- The OR Internet

Jun Ma

Industrial Engineering and Management Sciences
Northwestern University

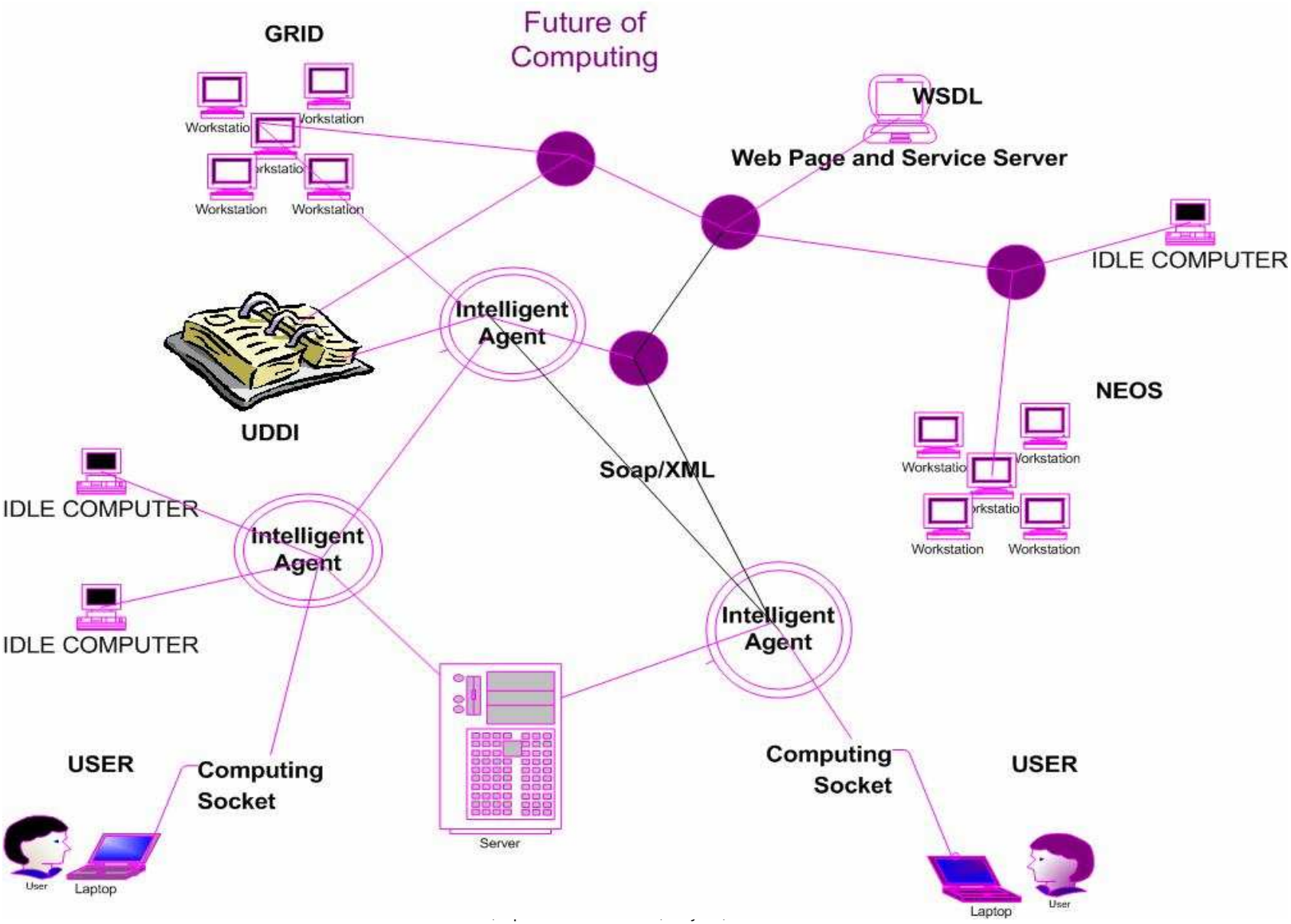
Lindo System Inc. 05/18/2005

OUTLINE

1. **Motivations**
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. Future and Derived Research

Motivation

Future of Computing



Motivation

But how… with so many type of components

1. Modeling Language Environment (MLE)

   (AIMMS, AMPL, GAMS, LINGO, LPL, MOSEL, MPL, OPL, OSmL)

2. Solver

   (Too many, e.g. Lindo)

3. Analyzer/Preprocessor

   (Analyzer, MProbe, Dr. AMPL)

4. Simulation

   (Software that does heavy computation, deterministic or stochastic)

5. Server/Registry

   (NEOS, BARON, HIRON, NIMBUS, LPL, AMPL, etc.)

6. Interface/Communication Agent

   (Lindo Black-Box Interface, COIN-OSI, CPLEX-Concert, AMPL/GAMS-Kestrel, etc.)

7. Low Level Instance Representation

   (Lindo's Instruction List, and others on next page)

Motivation

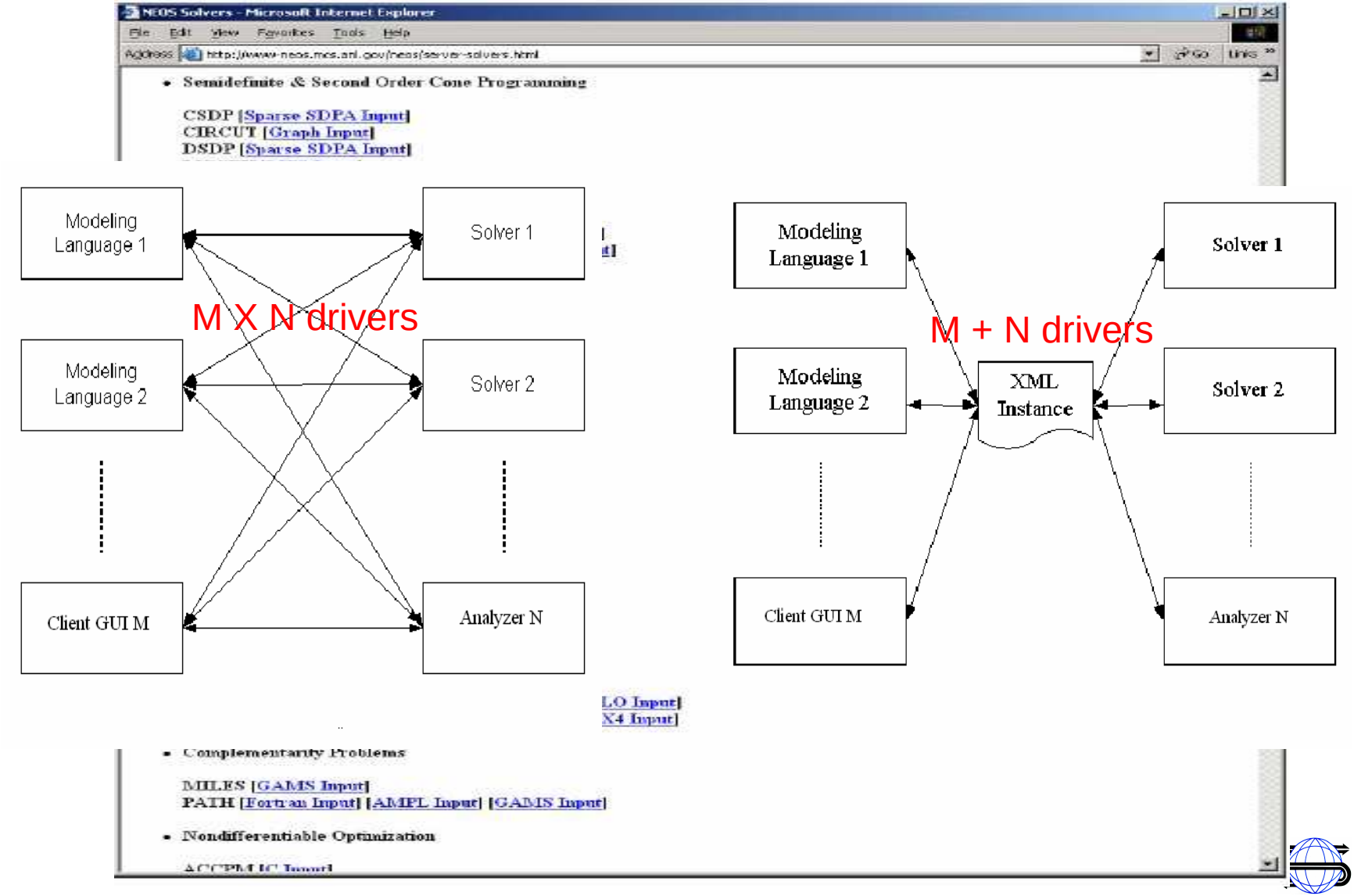
But how… with so many optimization types and representation formats

Linear Programming Quadratic Programming Mixed Integer Linear Programming	MPS, xMPS, LP, CPLEX, GMP, GLP, PuLP, LPFML, MLE instances
Nonlinearly Constrained Optimization Bounded Constrained Optimization Mixed Integer Nonlinearly Constrained Optimization Complementarity Problems Nondifferentiable Optimization Global Optimization	MLE instances SIF (only for Lancelot solver)
Semidefinite & Second Order Cone Programming	Spars[illegible]SDPA, [illegible]PL[illegible]
Linear Network Optimization	NETGEN, NETFLO, DIMACS, RELAX4
Stochastic Linear Programming	sMPS
Stochastic Nonlinear Programming	None
Combinatorial Optimization	None (except for TSP input, only intended for solving Traveling Sales Person problems.
Constraint and Logic Programming	None
Optimization with Distributed Data	None
Optimization via Simulation	None

OSiL

Motivation

Look at the NEOS server Web site



Motivation

As if it's not bad enough …

1. Tightly-coupled implementation (OOP? Why not!)
2. Various operating systems
3. Various communication/interfacing mechanisms
4. Various programming languages
5. Various benchmarking standards

Motivation

Now…

- **The key issue is communication, not solution!**
- **… and Optimization Services is intended to solve all the above issues.**

OUTLINE

1. Motivations
2. **Demonstration**
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. Future and Derived Research

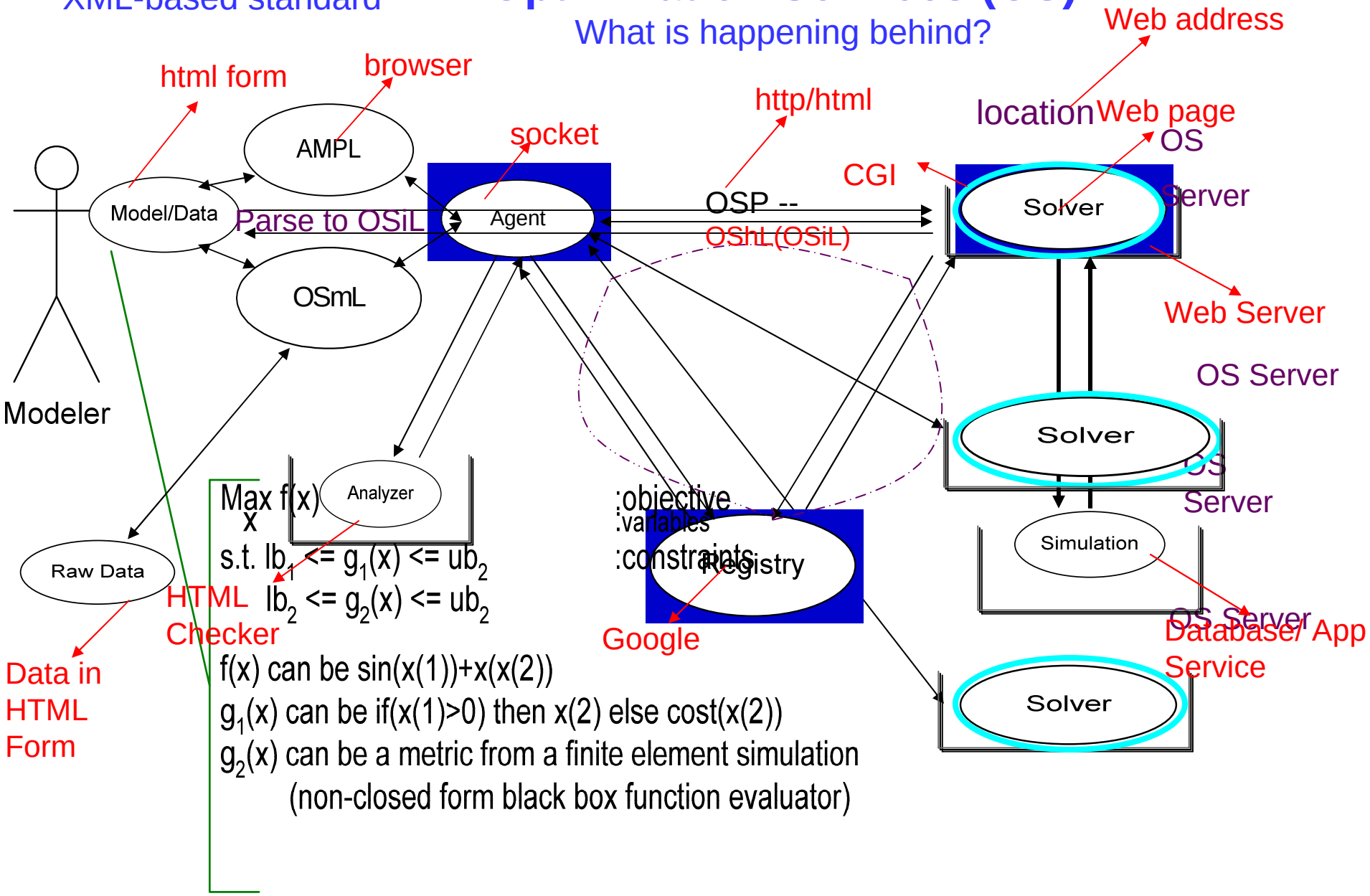
OUTLINE

1. Motivations
2. Demonstration
3. **Optimization Services and Optimization Services Protocol**
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. Future and Derived Research

Optimization Services (OS)

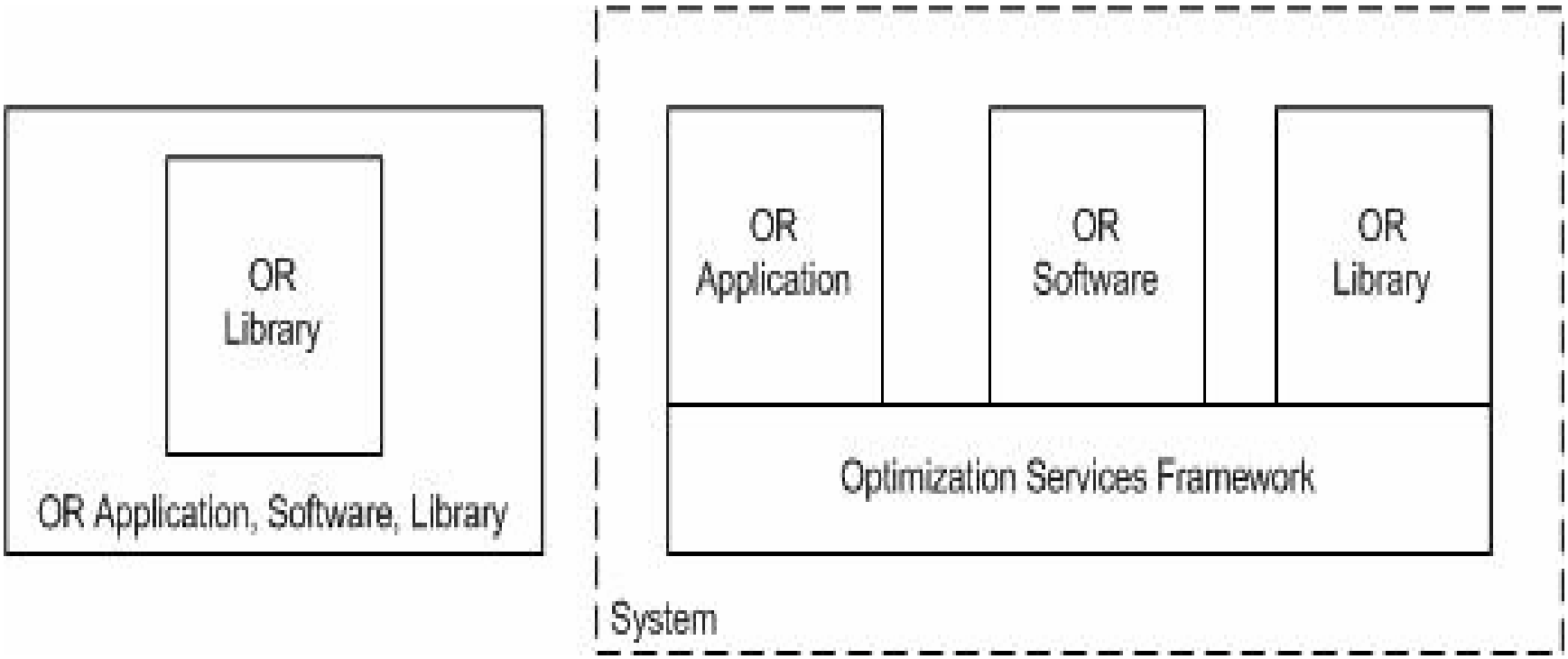
XML-based standard

What is happening behind?



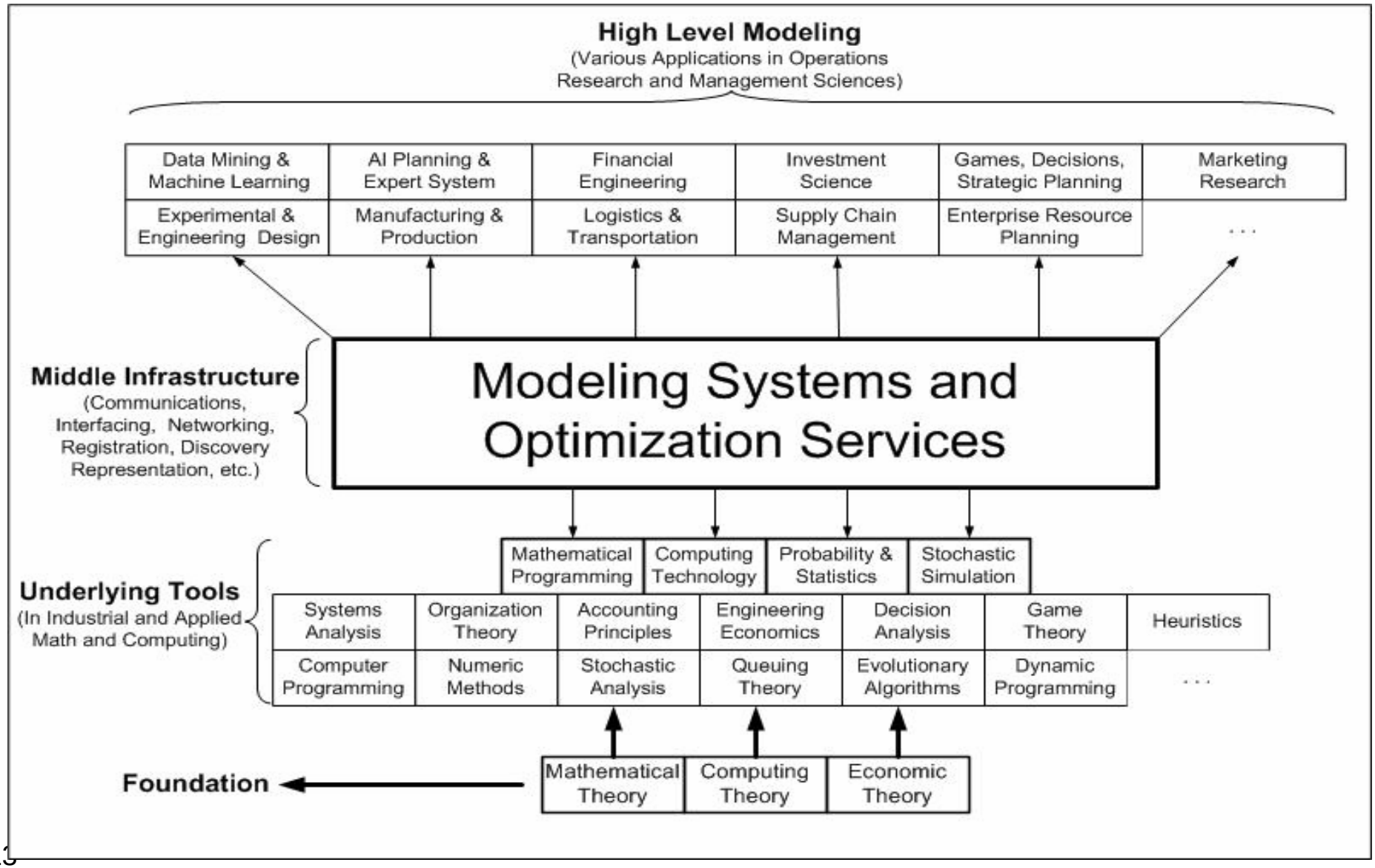
Optimization Services

What is it? – A framework for optimization software



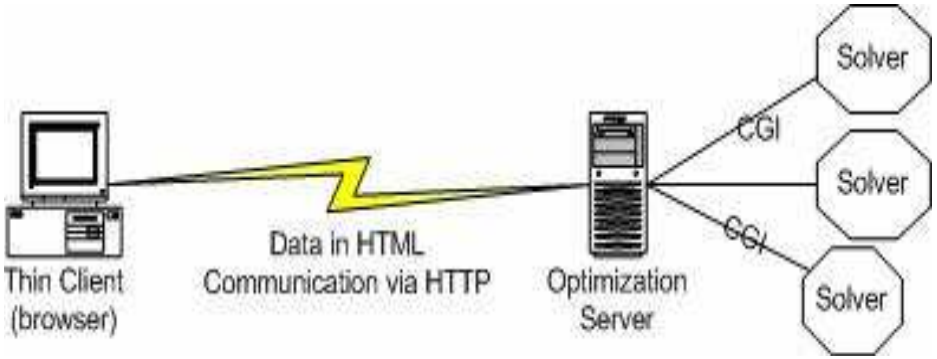
Optimization Services

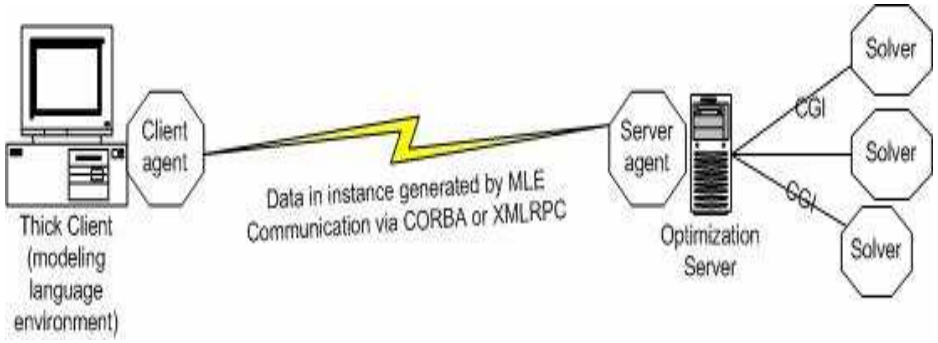
What is it? – A computational infrastructure



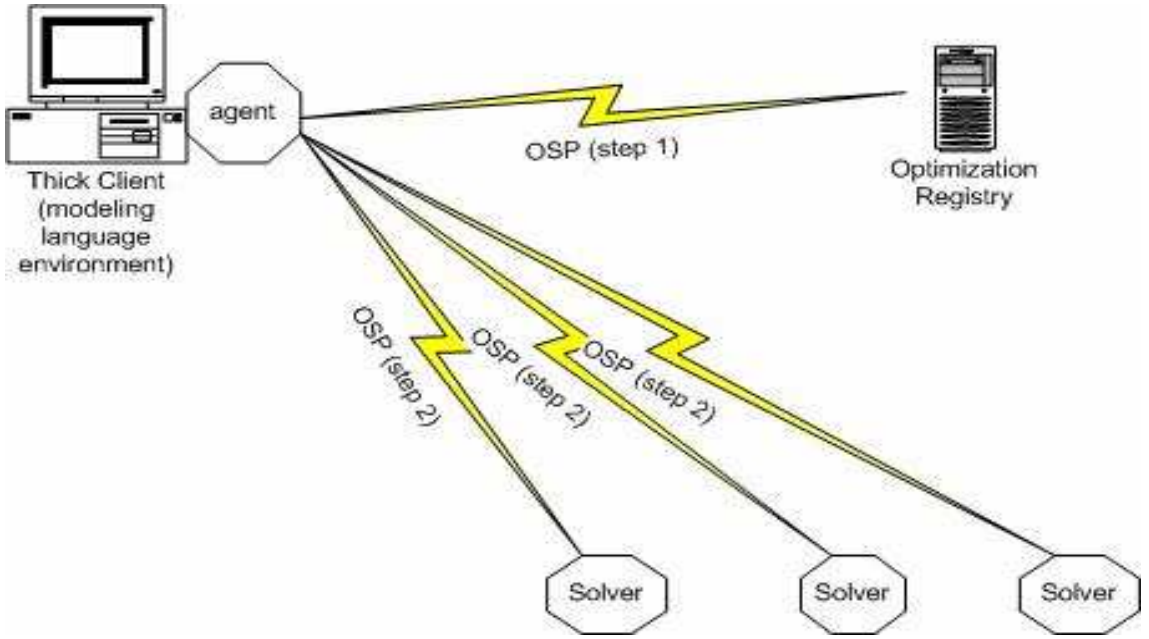
Optimization Services

What is it? – The next generation NEOS



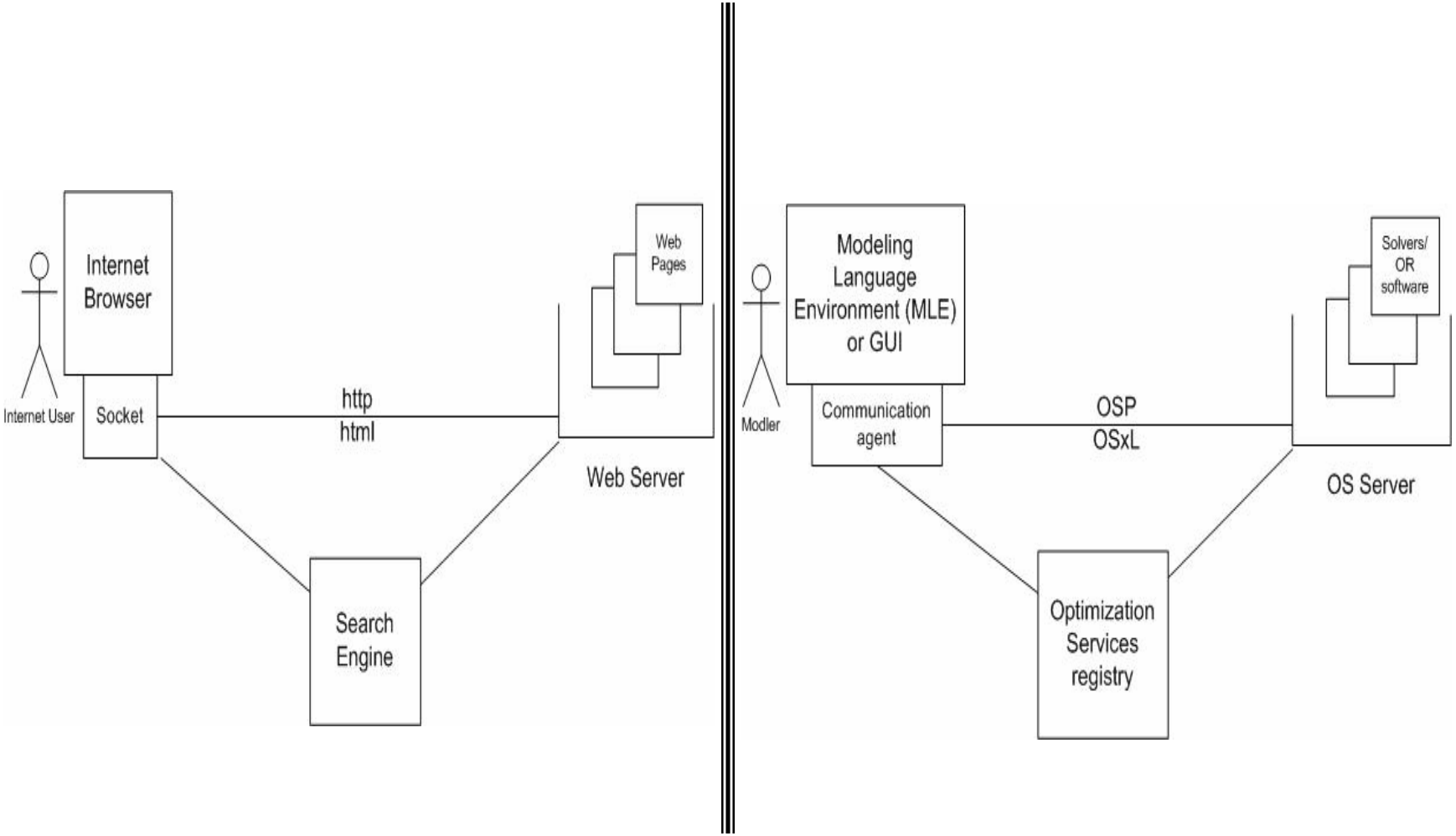


- The NEOS server and its connected solvers uses the OS framework.
- NEOS accepts the OSiL and other related OSP for problem submissions
- NEOS becomes an OS compatible meta-solver on the OS network
- NEOS hosts the OS registry



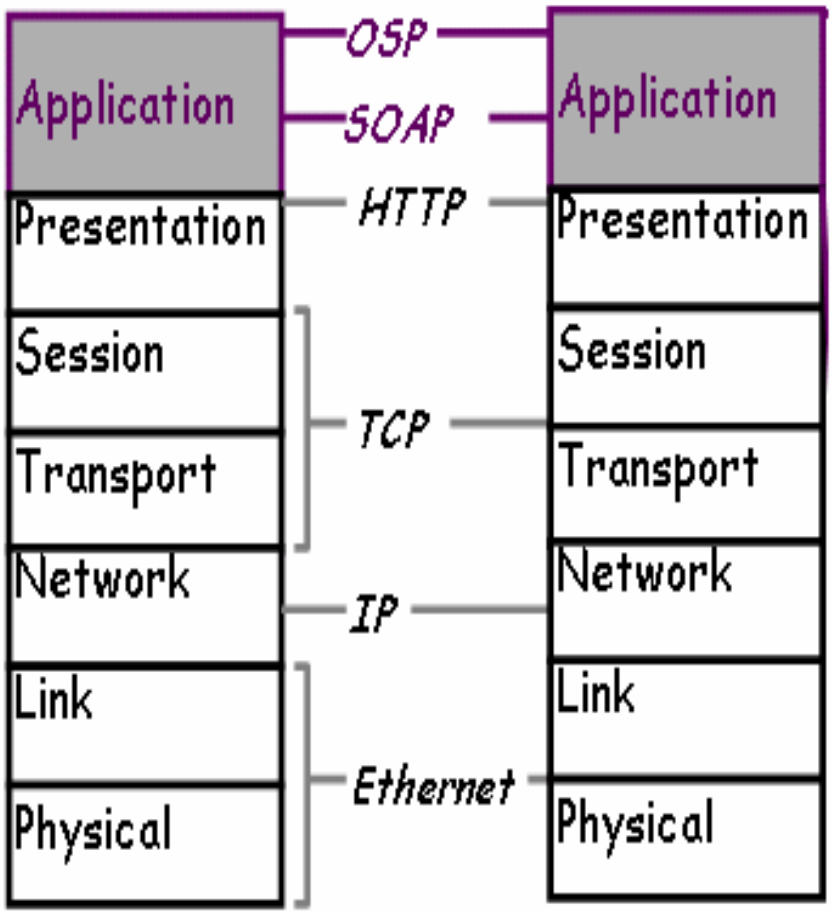
Optimization Services

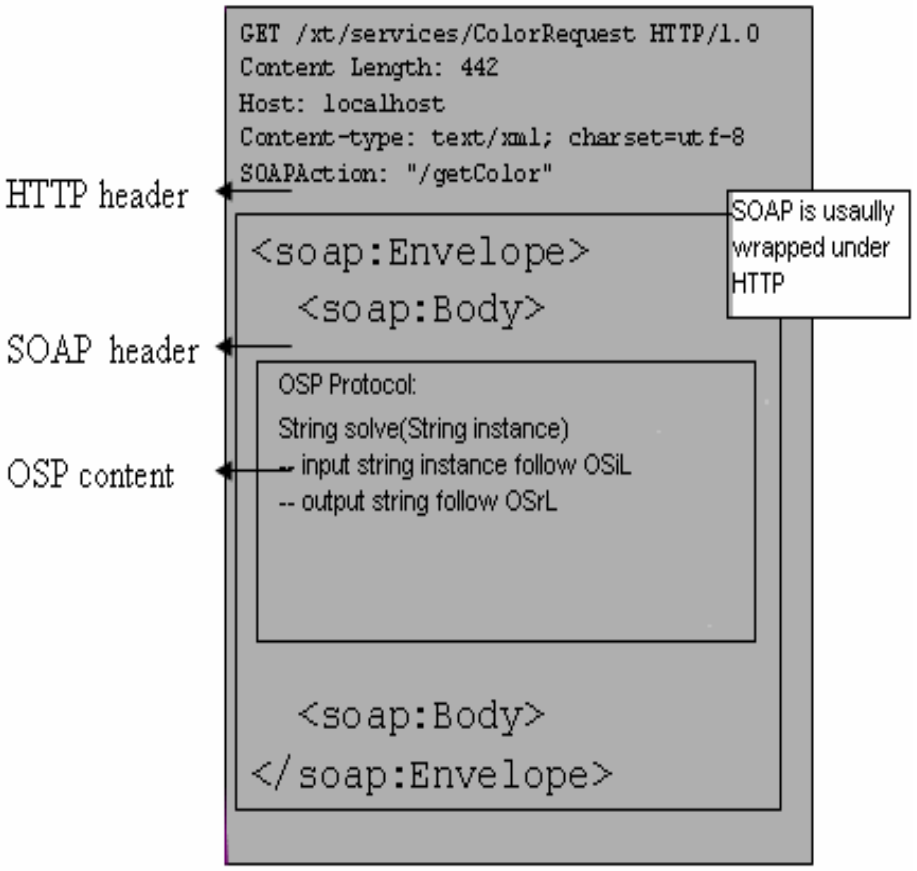
What is it? – The OR Internet



Optimization Services Protocol (OSP)

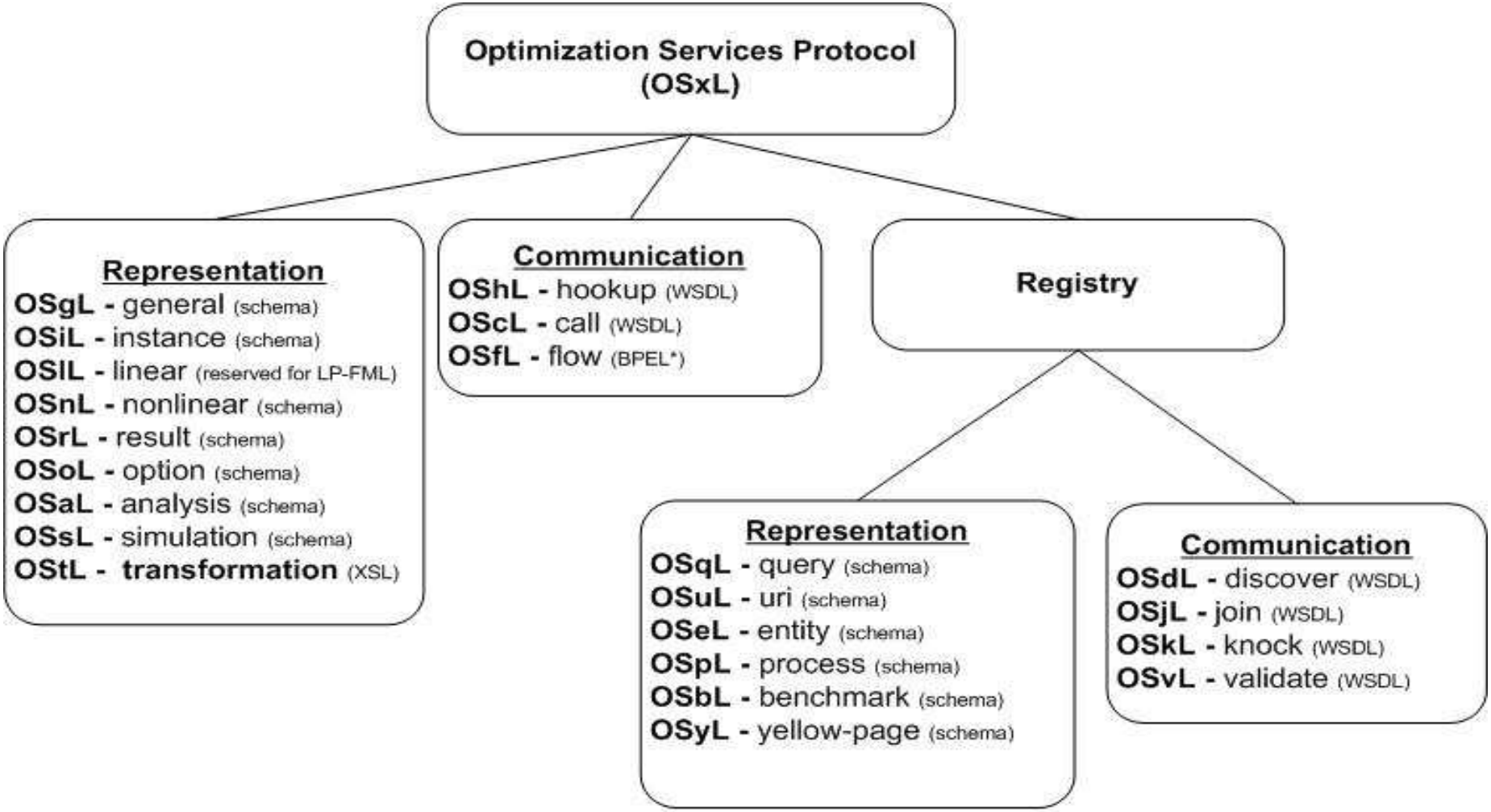
What is it? – Application level networking protocol
– Interdisciplinary protocol between CS and OR





Optimization Services Protocol (OSP)

What does the protocol involve? – 20+ OSxL languages



***OSmL:** a modeling language and NOT an Optimization Services Protocol
***Letters not currently used:** w, z
***BPEL:** Business Process Execution Language for flow orchestration.

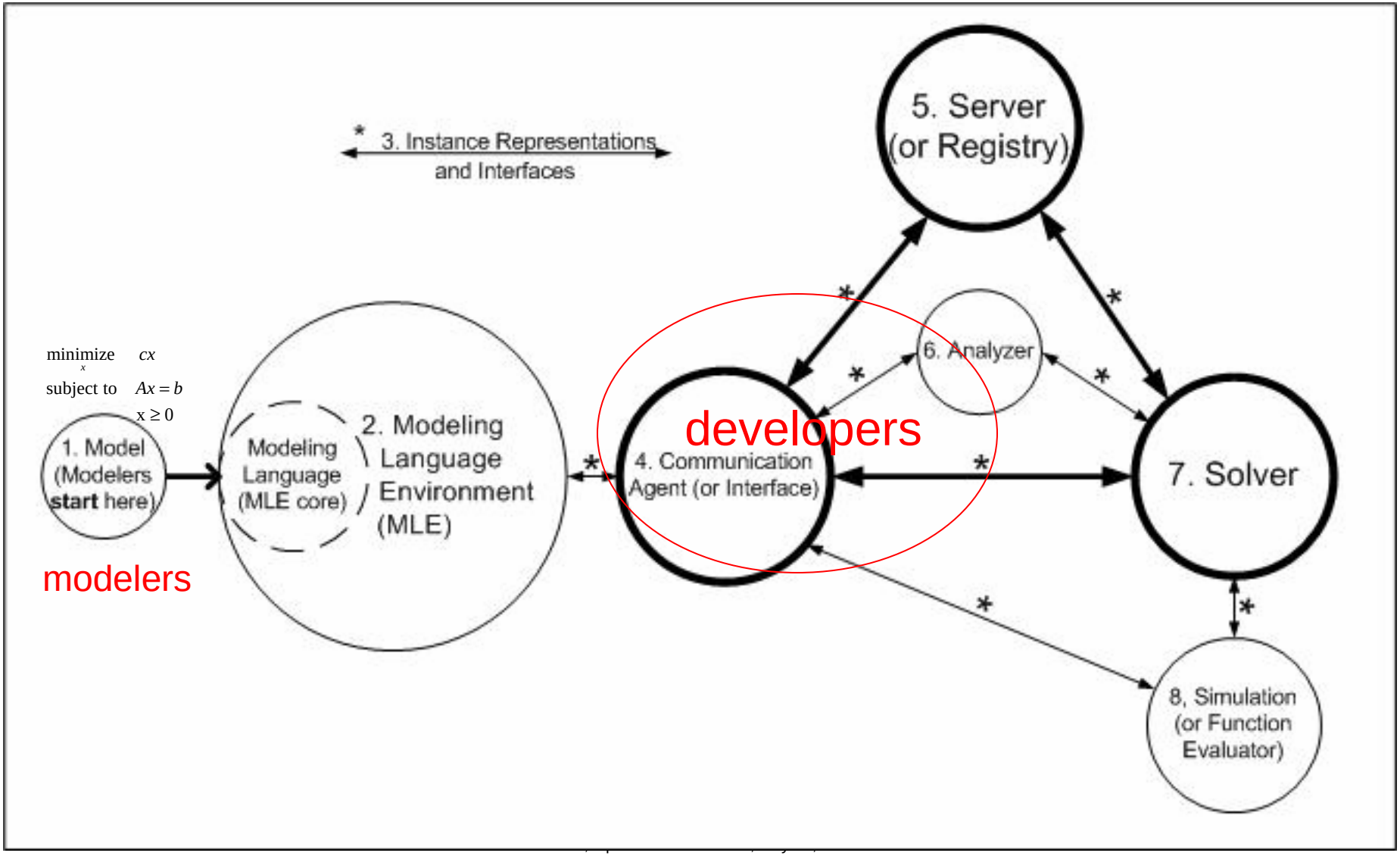
OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. **Optimization System Background**
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. Future and Derived Research

Optimization System Background

What does an optimization system look like?

users



Optimization System Background

What is the difference between a model and an instance?

model — high-level, user-friendly, symbolic, general, concise, understandable

compile →

instance — Low-level, computer-friendly, explicit, specific, redundant, convenient

```
set NUTR ordered;
set FOOD ordered;

param cost {FOOD} >= 0;
param f_min {FOOD} >= 0, default 0;
param f_max {j in FOOD} >= f_min[j], default Infinity;

param n_min {NUTR} >= 0, default 0;
param n_max {i in NUTR} >= n_min[i], default Infinity;

param amt {NUTR,FOOD} >= 0;

# ------------------------------------

var Buy {j in FOOD} integer >= f_min[j], <= f_max[j];

# ------------------------------------------------------

minimize Total_Cost:  sum {j in FOOD} cost[j] * Buy[j];

# ------------------------------------------------------

subject to Diet {i in NUTR}:
n_min[i] <= sum {j in FOOD} amt[i,j] * Buy[j] <= n_max[i];
```

AMPL

```
NAME          qpEx
ROWS
 N  obj
 G  c1
COLUMNS
    x1        c1        6
    x2        obj       -1
    x2        c1        7
    x3        c1        -8
RHS
    rhs       c1        9
QSECTION      obj
    x1        x1        2
    x1        x3        -3
    x2        x2        4
    x3        x3        5
ENDATA
```

MPS

$$\underset{x}{\text{minimize}} \quad cx \qquad \text{subject to} \quad Ax = b, \; x \geq 0$$

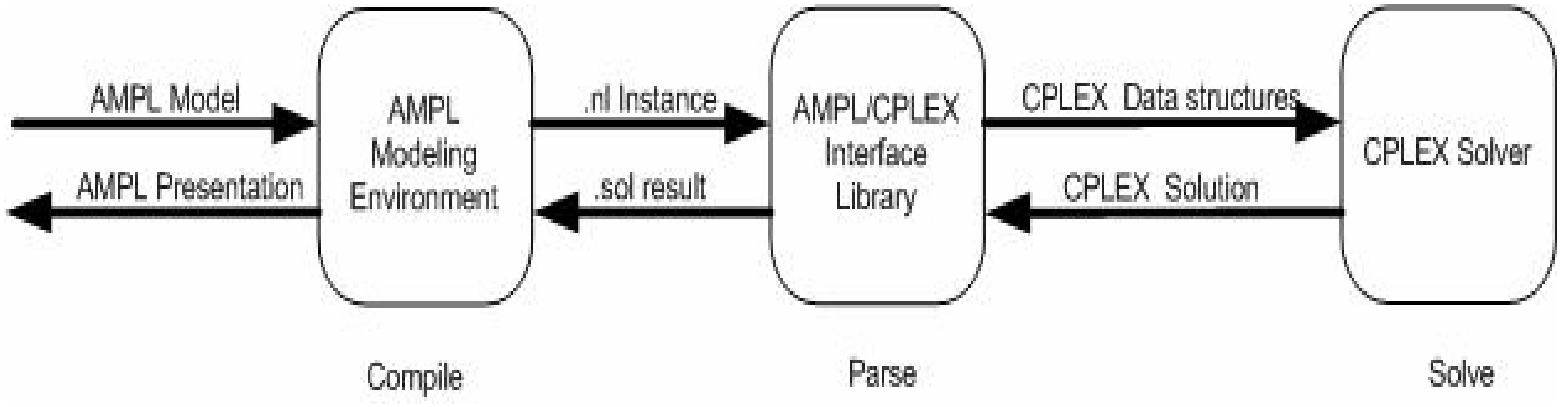
$$\underset{x}{\text{minimize}} \quad -x_2 + 1/2(2x_1^2 - 3x_1x_3 + 4x_2^2 + 5x_3^2)$$
$$\text{subject to} \quad 6x_1 + 7x_2 - 8x_3 \geq 9, \quad x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

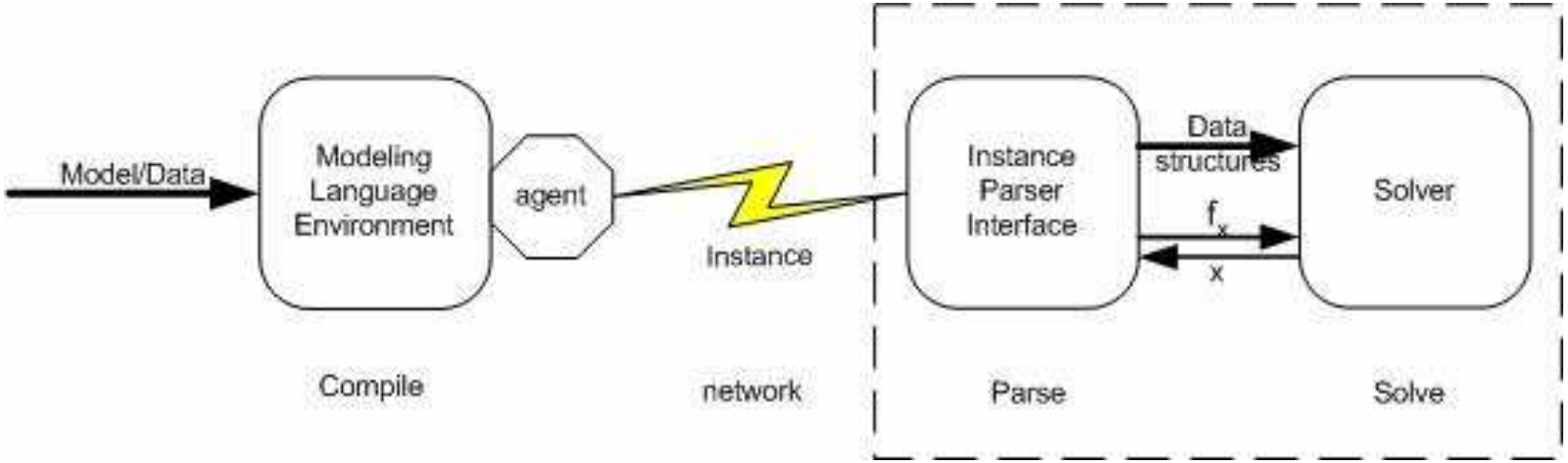
```
<OSiL>
  <programDescription>
    <source>Optimization Services, Jun Ma's Thesis</source>
    <description>Adapted from an example of Rosenbock (1960)</description>
    <objName>adaptedRosenbrock</objName>
    <maxOrMin>min</maxOrMin>
    <objConstant>0.0</objConstant>
    <numberObjectives>1</numberObjectives>
    <numberConstraints>2</numberConstraints>
    <numberVariables>2</numberVariables>
  </programDescription>
  <programData>
  . . .
  </programData>
</OSiL>
```

OSiL

Optimization System Background

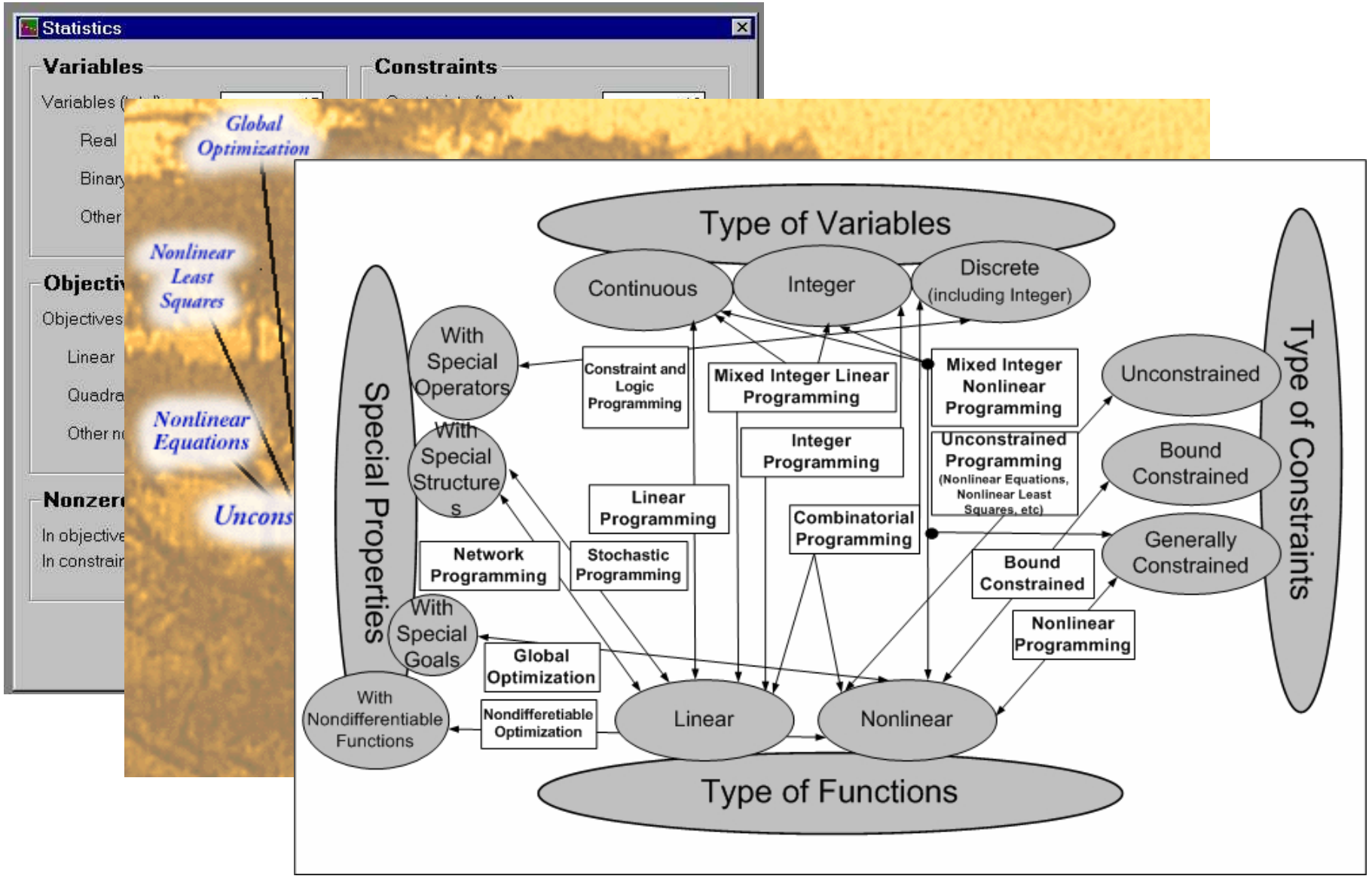
What's the difference between local interfacing and communication agent





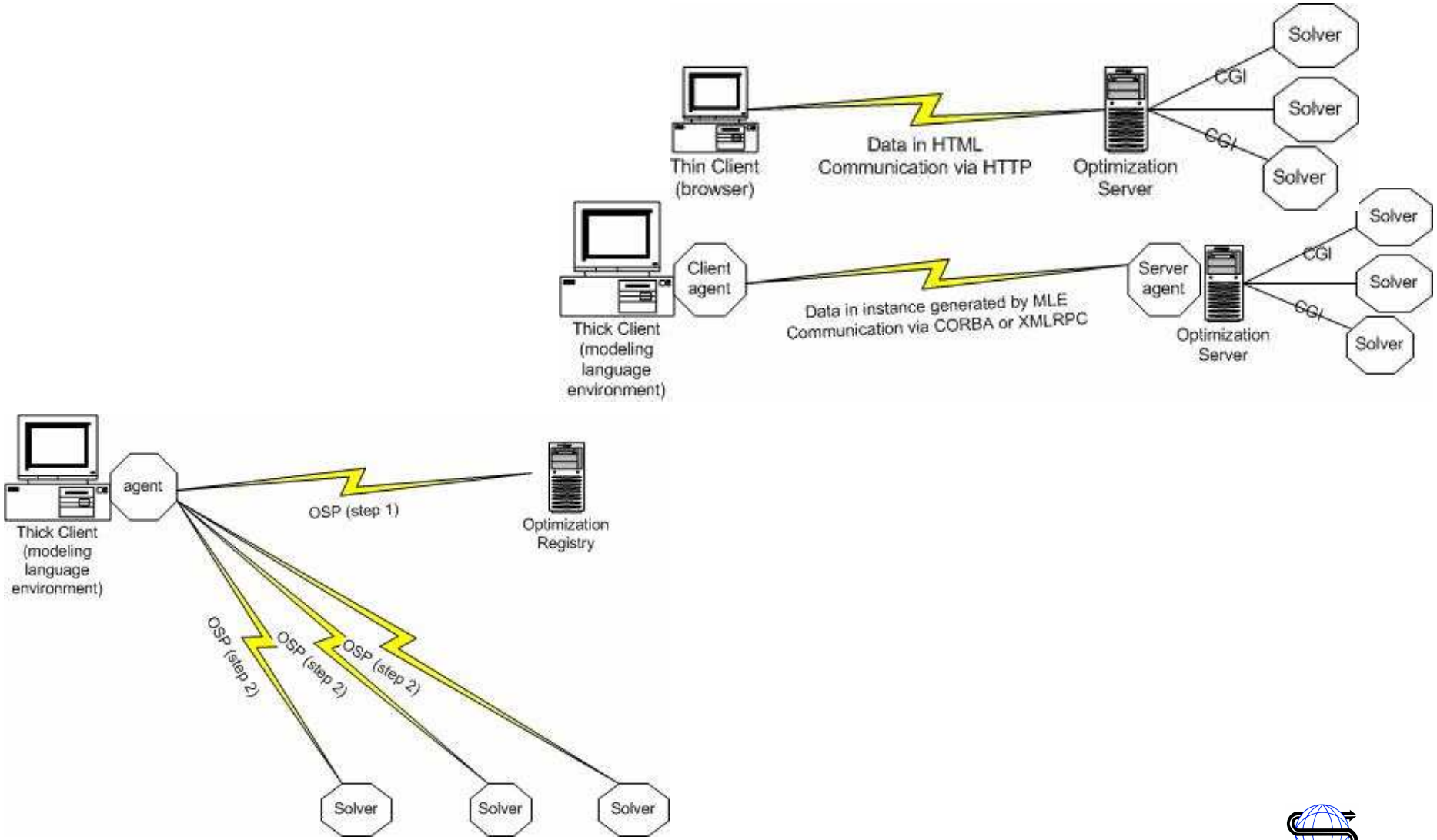
Optimization System Background

Why is analyzer important?



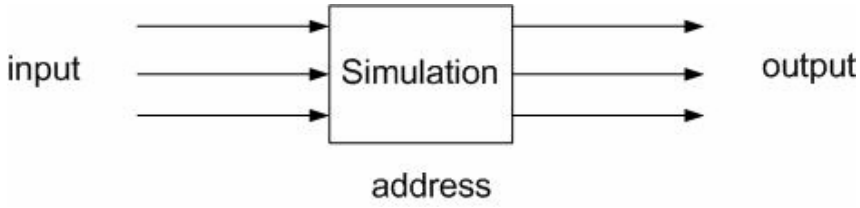
Optimization System Background

What's the difference between a server and a registry



Optimization System Background

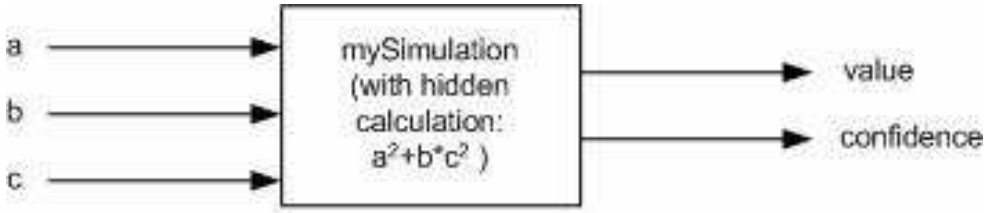
What's a simulation?



$$\underset{x}{\text{minimize}} \quad x_1^2 + 2x_2^2$$

$$\text{subject to} \quad 2x_1 + 3x_2 \geq 9$$

$$x_1 \geq 0,\ x_2 \geq 0$$



http://somesite.com/mySimulation

$$\underset{x}{\text{minimize}} \quad mySimulation(x_1, 2, x_2)$$

$$\text{subject to} \quad 2x_1 + 3x_2 \geq 9$$

$$x_1 \geq 0, x_2 \geq 0$$

mySimulation{

address = http : // somesite .com / mySimulation

input :

a

b

c

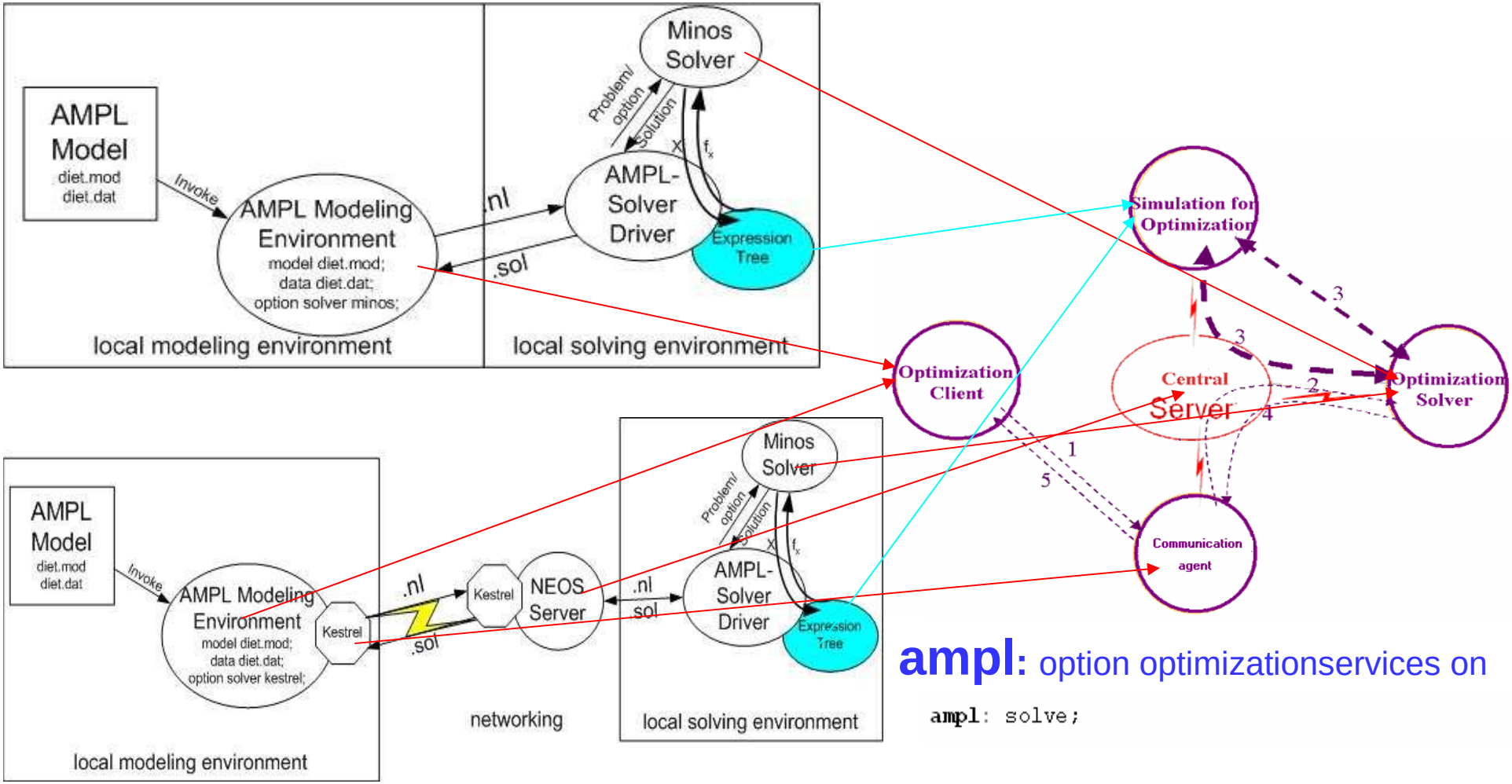
output :

*value + confidence * 0*

}

Optimization System Background

AMPL, NEOS and Kestrel

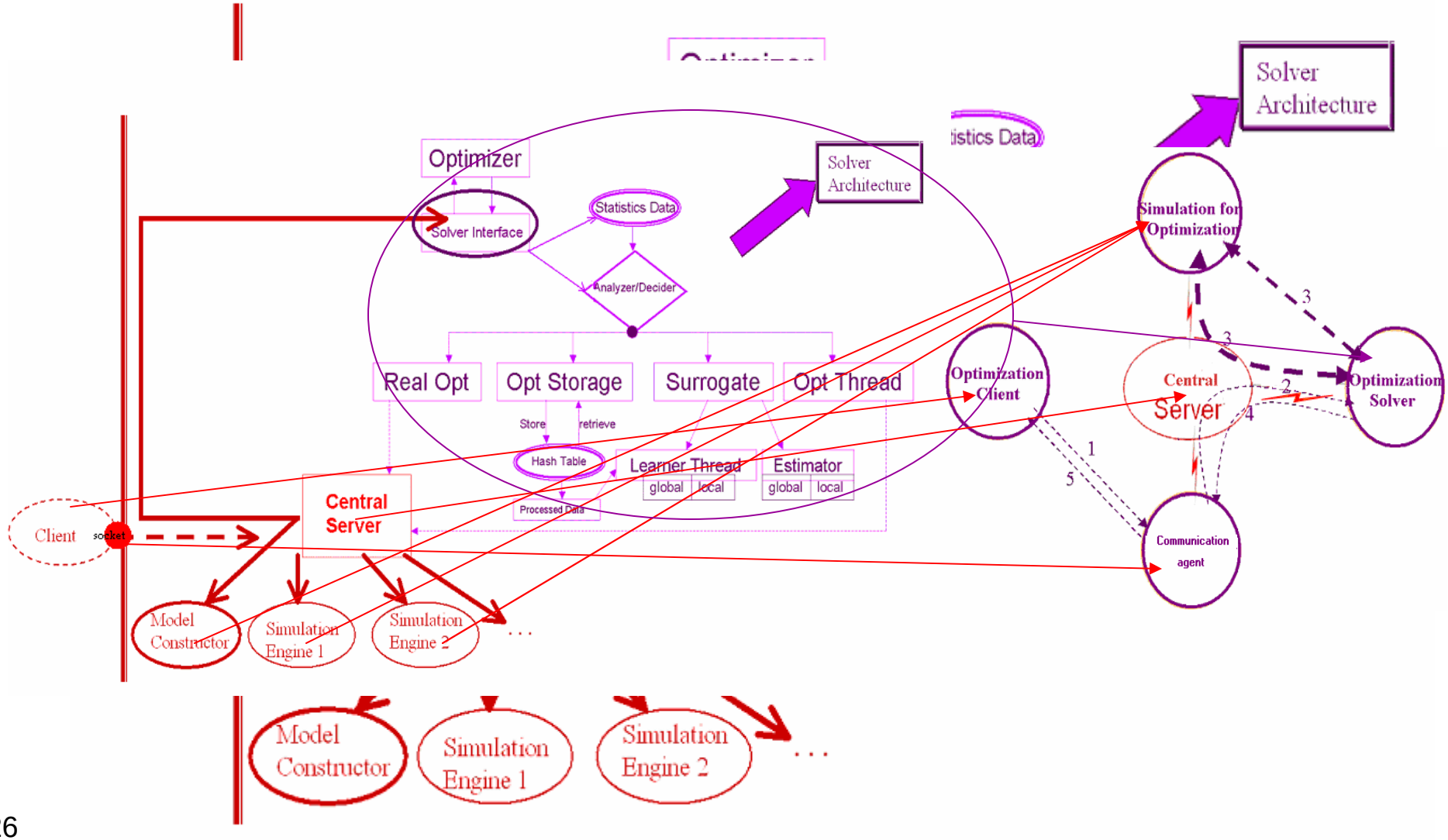


ampl: option optimizationservices on

```
ampl: solve;
```

Optimization System Background

Motorola Optimization System



OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. **Computing and Distributed Background**
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. Future and Derived Research

Computing and Distributed Background

What we used in our implementation

1. Java, Open Source Libraries, Object-oriented Programming (OS library)

2. Networking Protocols: HTTP, SOAP, OSP
   (OS server: Tomcat, Axis, OS library)

3. Eclipse IDE for JAVA development

4. XML Spy for XML Schema design

Computing and Distributed Background

XML and XML Dialect (e.g. MathML, OSiL)

```
<element1 attrName1="value1" attrName2="value2">
    <element1 ...>
        ...
    </element1>
    <element2 ...>
        ...
    </element2>
</element1>

<element .../>
```

OSiL

$(2X_1 + 3X_2)^2$

```
<nl idx="9">
    <power>
        <plus>
            <var idx="1" coef="2"/>
            <var idx="2" coef="3"/>
        </plus>
        <number value="2"/>
    </power>
</nl>
```

MathML

$(2X_1 + 3X_2)^2$

```
<math>
    <apply>
        <power/>
        <apply>
            <plus/>
            <apply>
                <times/>
                <cn>2</cn>
                <ci>X1</ci>
            </apply>
            <apply>
                <times/>
                <cn>3</cn>
                <ci>X2</ci>
            </apply>
        </apply>
        <cn>2</cn>
    </apply>
</math>
```

Computing and Distributed Background

XML Schema

$$\underset{x}{\text{minimize}} \quad 100(x_1 - x_0^2)^2 + (1 - x_0)^2 + 7x_1$$

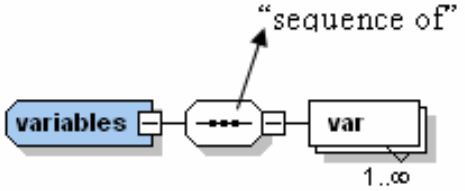
$$\text{subject to} \quad x_0 + 7x_1 \le 10$$

$$\ln(x_0 x_1) + 7x_0 + 5x_1 \le 10$$

$$x_0, x_1 \ge 0$$

```
<variables>
    <var lb="0" name="x0" type="C"/>
    <var lb="0" name="x1" type="C" objCoef="7.0"/>
</columns>
```

```
<xs:element name="variables">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="var" type="var" minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```




```
<xs:complexType name="var">
    <xs:attribute name="name" type="xs:string" use="optional"/>
    <xs:attribute name="init" type="xs:string" use="optional"/>
    <xs:attribute name="type" use="optional" default="C">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="C"/>
                <xs:enumeration value="B"/>
                <xs:enumeration value="I"/>
                <xs:enumeration value="S"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="lb" type="xs:double" use="optional" default="0"/>
    <xs:attribute name="ub" type="xs:double" use="optional" default="INF"/>
    <xs:attribute name="objCoef" type="xs:double" use="optional" default="0.0"/>
    <xs:attribute name="mult" type="xs:positiveInteger" use="optional" default="1"/>
</xs:complexType>
```

Computing and Distributed Background

Other XML Technologies

1. Parsing: SAX and DOM models

2. Transformation: XSL style sheet

3. Lookup: XPath and XQuery

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
    <xsl:template match="/">
        <html>
            <body>
                <hr/>
                <h1>Stocks</h1>
                <p/>
                <xsl:for-each select="stocks/stock">
                    <p/>stock: <xsl:value-of select="@name"/>
                </xsl:for-each>
                <hr/>
            </body>
        </html>
    </xsl:template>
</xsl:stylesheet>
```

Stocks

stock: ge

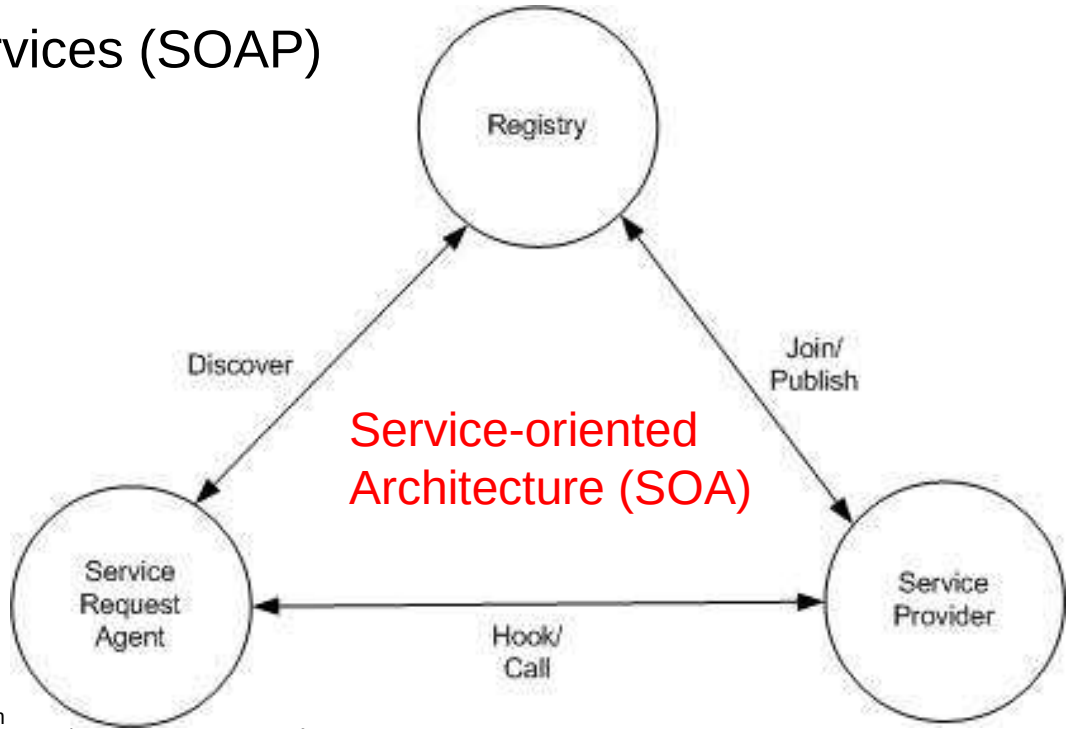
stock: msft

stock: pg

Computing and Distributed Background

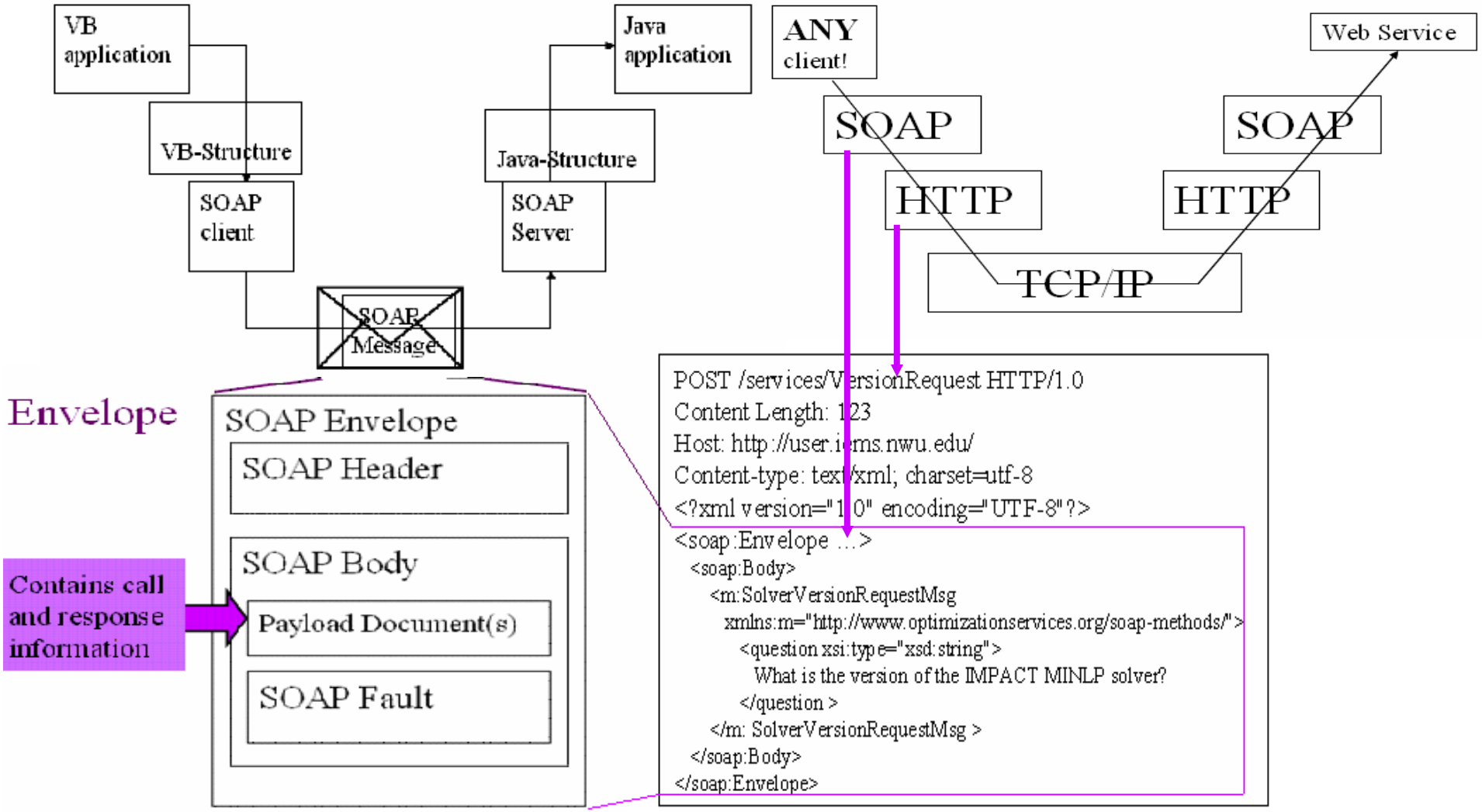
Web services

- **Platform and implementation independent** components
- **Described** using a service description language (WSDL)
- **Published** to a registry of services (UDDI, OS Registry)
- **Discovered** through a standard mechanism (UDDI, OS Registry)
- **Invoked** through a declared API (SOAP)
- **Composed** with other services (SOAP)



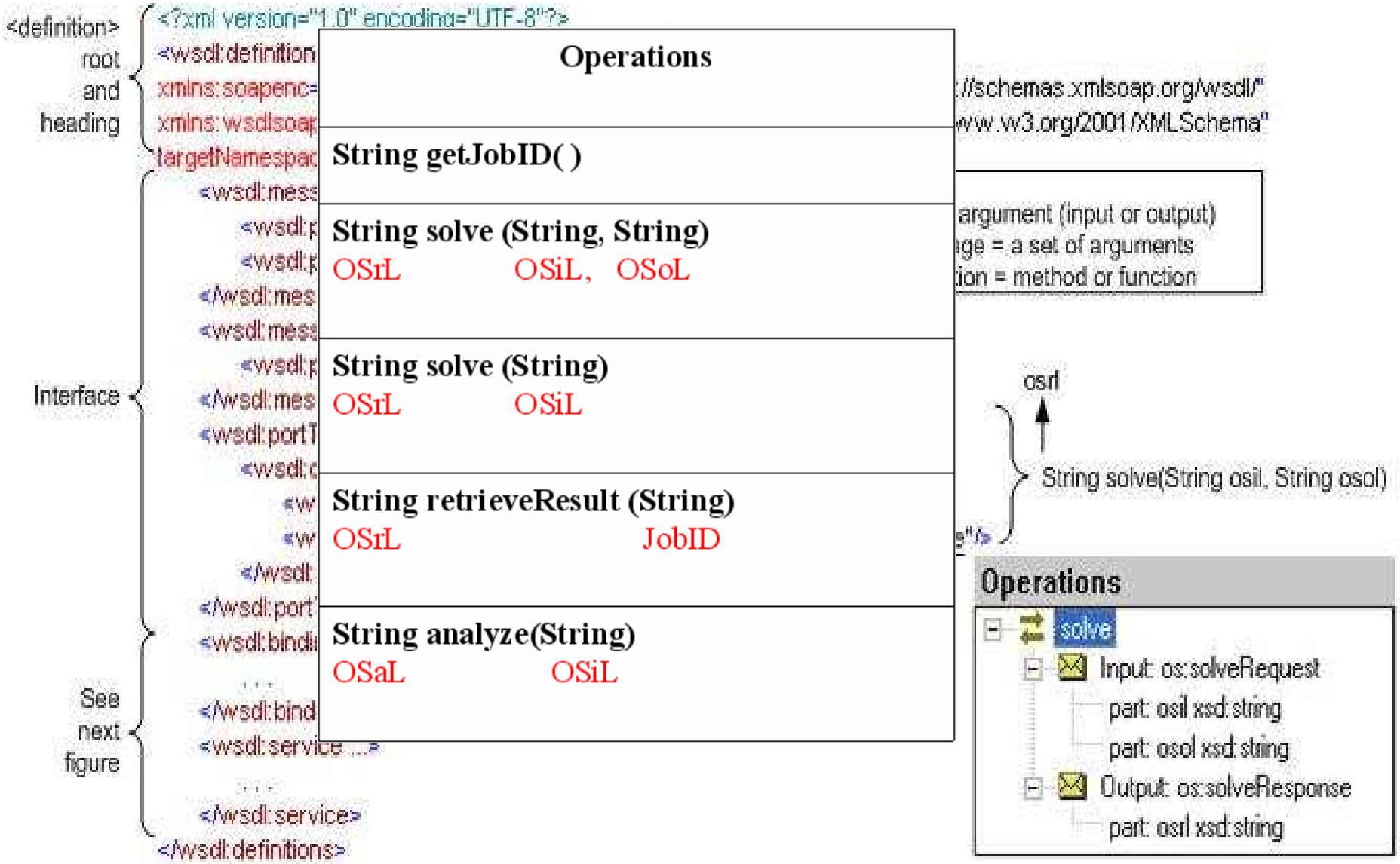
Computing and Distributed Background

Web services and SOAP



Computing and Distributed Background

Web services and WSDL



Computing and Distributed Background

Web services and WSDL

OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. **Optimization Services Protocol - Representation**
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. Future and Derived Research

Optimization Services Representation

Who else did this before?

- Many “standards”
- All limited to problem *input*
- Highly fragmented and *no* general format
- Fourer, Lopes, and Martin’s LPFML **(OSIL)**
- Kristjánsson’s OptML
- Bradley’s NaGML
- We are the first in designing
  - Systematic representation of major optimization types
  - All major instance types (result, analysis, input, query, etc.)
  - Web services (SOAP) based communication standards
  - Optimization registry
  - A universal framework

Optimization Services Representation

Optimization Services general Language (OSgL)

General data structures; Included in other schemas

Optimization Services instance Language (OSiL)

-Linear
-Mixed integer
-Bound constrained optimization
-General quadratic optimization
-Nonlinear unconstrained/constrained
-General mixed integer nonlinear
-General nonlinear with user-defined functions
-Global optimization
-General nonlinear with simulations (black-box functions)
-Optimization over simulation/nondifferentiable optimization
-General nonlinear with xml data (either within OSiL or remotely located)
-General nonlinear with data look up (XPath)
-Network and graph definition
-Network programming
-Constraint programming
-Semidefinite programming
-Semi-infinite programming
-Cone programming
-Complementarity problems
-Stochastic linear/nonlinear (distribution based recourse problem, scenario based recourse problem, chance constrained)
-Combinatorial optimization/Heuristic Optimization (TSP, MST, SP, MF, MCF, VRP, Set Covering, Coloring etc. etc.)

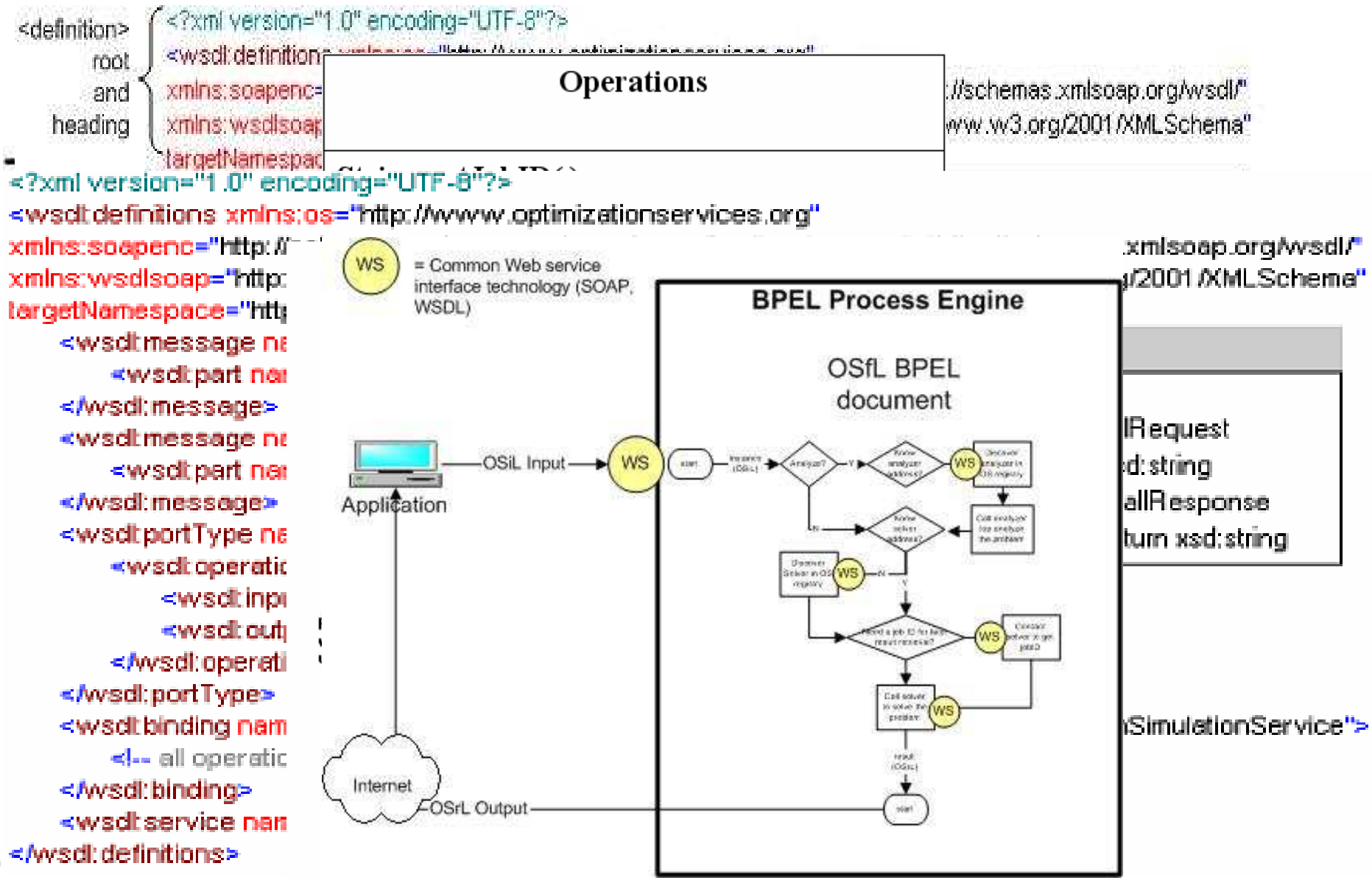
OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. **Optimization Services Protocol - Communication**
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. Future and Derived Research

Optimization Services Communication

Optimization Services hookup Language (OShL)

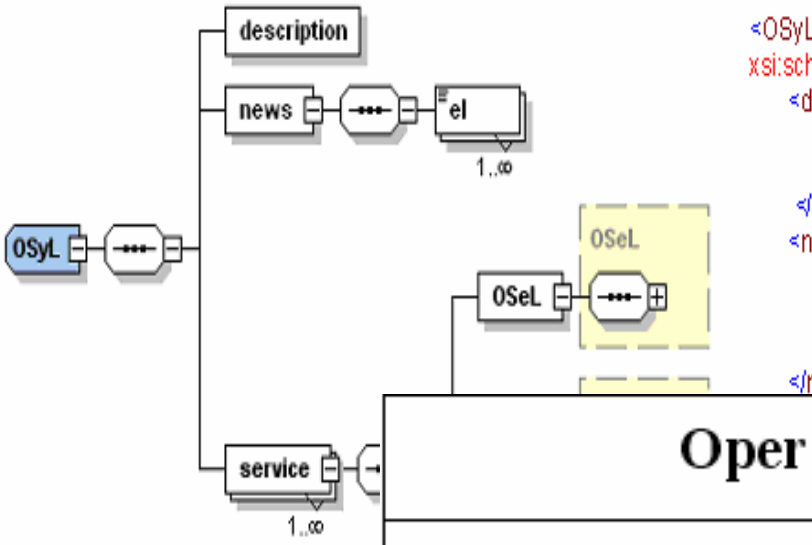
Hookup to solvers, and analyzers



OUTLINE

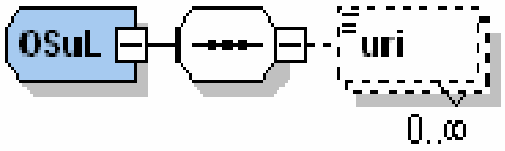
1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. **Optimization Services Protocol - Registry**
9. Optimization Services modeling Language (OSmL)
10. Future and Derived Research

Optimization Services Registry

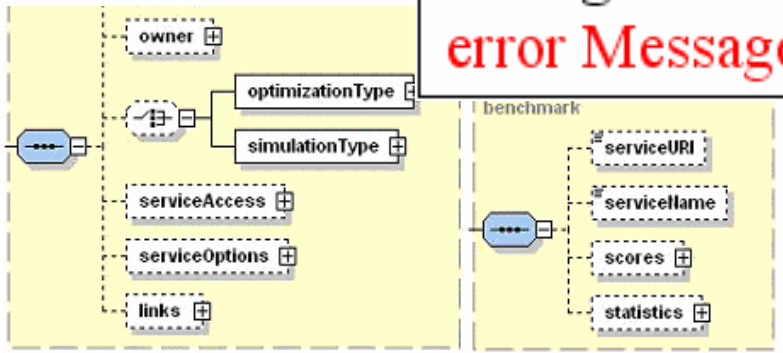



```
<OSyL xmlns="os.optimizationservices.org" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="os.optimizationservices.org http://www.optimizationservices.org/schemas/OSyL.xsd">
    <description>
        OS registry is a native XML data base.
        It contains a sequence of service, each consisting of a triplet (OSeL, OSpL, OSbL).
     </description>
    <news>
        <el date="2005-04-06">Impact Generalized Mixed Integer Solver joins the OS registry</el>
        <el date="2005-03-29">Ziena Knitro Service joins the OS registry</el>
        <el date="2005-02-27">Lindo MINLP Service joins the OS registry</el>
    </news>
                                                                    pactGMIP.jws</uri>
                                                                    teger Solver</name>
        <OSpL> . . . </OSpL>
        <OSbL> . . . </OSbL>
    </service>
    <service>
        <OSeL> . . . </OSeL>
        <OSpL> . . . </OSpL>
        <OSbL> . . . </OSbL>
    </service>
    <service>
        <OSeL> . . . </OSeL>
        <OSpL> . . . </OSpL>
        <OSbL> . . . </OSbL>
    </service>
    . . .
    . . .
</OSyL>
```

Optimization Services Registry




```
<?xml version="1.0" encoding="UTF-8"?>
<OSuL xmlns="os.optimizationservices.org" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="os.optimizationservices.org http://www.optimizationservices.org/schemas/OSuL.xsd">
    <uri>http://www.abc.com/lpsolver.jws</uri>
    <uri match="exact">http://www.pdf.net/lpsolverservice.wb</uri>
    <uri match="moreG
    <uri match="appro
    <uri match="guess
</OSuL>
```




```
            </nonlinear>
        </numberConstraints>
        <numberVariables num="12">
            <continuous num="3"/>
            <integer num="9"/>
            <binary num="0"/>
            <string num="0"/>
        </numberVariables>
    </programDescription>
    <programDataAnalysis> . . . </programDataAnalysis>
</OSaL>
<standard>
    <entity>
        <service>
            <keyWords><key>interior point method</key><key>convex programming</key></keyWords>
        </service>
        <optimizationType>
            <variableType>mixedInteger</variableType>
        </optimizationType>
    </entity>
</standard>
</OSqL>
```

OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. **Optimization Services modeling Language (OSmL)**
10. Future and Derived Research

Optimization Services modeling Language (OSmL)

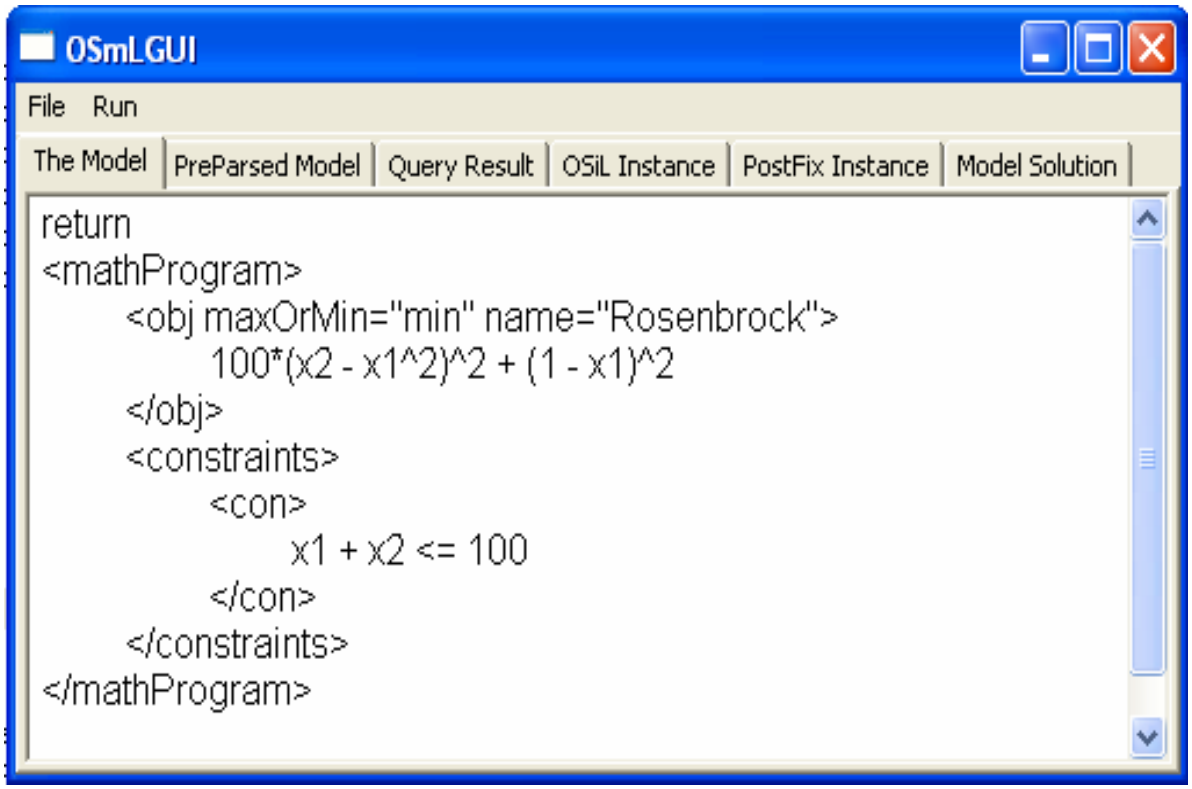
A derived research

Open source and general purpose

Standard based (XQuery input; OSiL output)

Suitable for distributed optimization

XML data is ubiquitous



Optimization Services modeling Language

AMPL

```
#set, parameter, and variable constructior
param T;
set PROD;
set LINKS = {PROD, 1..T};
param HC {PROD} ;
param FXC {PROD} ;
param CAP {1..T} ;
param DEM {LINKS};
param PCOST {PROD, 1..T} ;

#VARIABLE DECLARATION
var x  {PROD, 1..T} >= 0;
var I  {PROD, 0..T} >=0;
var y  {PROD, 1..T}binary;

#OBJECTIVE CONSTRUCTION
minimize Total_Cost:
 sum {i in PROD} I[i, 0]  +  sum {i in PR
(PCOST[i, t]*x[i, t] + HC[i]*I[i, t] + FXC

# INITIAL INVENTORY CONSTRAIN
subject to Init_Inv {i in PROD}:
I[i, 0] = 0.0;

# DEMAND CONSTRAINTS
subject to Balance {i in PROD, t in 1..T}
    x[i, t] + I[i, t - 1] - I[i, t] = DEM[i, t];

# FIXED CHARGE CONSTRAINTS
subject to Fixed_Charge {i in PROD, t in
    x[i, t] <= CAP[ t]*y[i, t];

# CAPACITY CONSTRAINTS
subject to Capacity {t in 1..T}:
   sum {i in PROD} x[i, t] <= CAP[ t];
```

OSmL

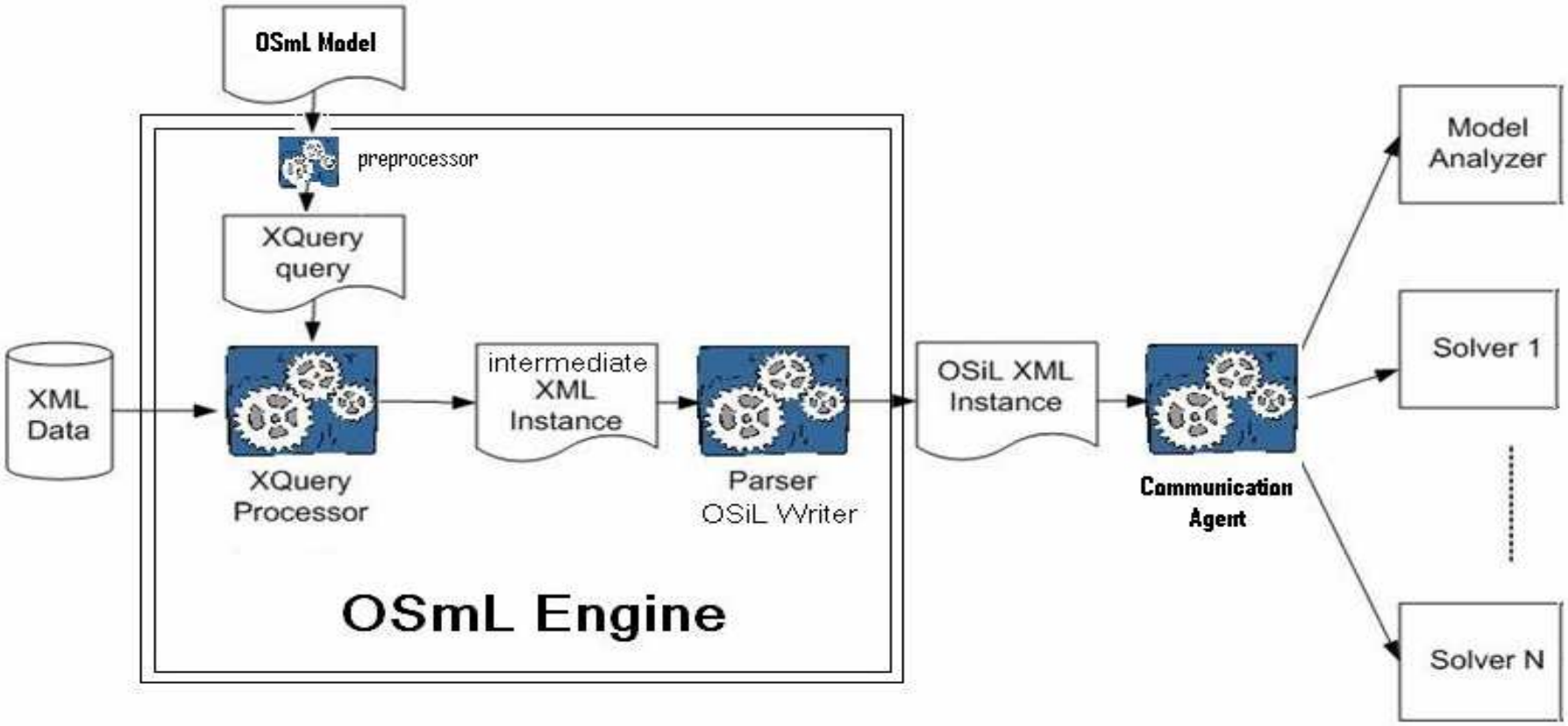
```
<?xml version="1.0" encoding="UTF-8"?>
<lotSizeData>
    <product productID="1" holdCost="1" prodCost="7" fixedCost="150">
        <period periodID="1">
            <demand>60</demand>
        </period>
        <period periodID="2">
            <demand>100</demand>
        </period>
        <period periodID="3">
            <demand>140</demand>
        </period>
        <period periodID="4">
            <demand>200</demand>
        </period>
    </product>
    <product productID="2" holdCost="2" prodCost="4" fixedCost="100">
        <period periodID="1">
            <demand>40</demand>
        </period>
        <period periodID="2">
            <demand>60</demand>
        </period>
        <period periodID="3">
            <demand>100</demand>
        </period>
        <period periodID="4">
            <demand>40</demand>
        </period>
    </product>
    <periodCapacity>
        <capacity periodID="1">200</capacity>
        <capacity periodID="2">200</capacity>
        <capacity periodID="3">200</capacity>
        <capacity periodID="4">200</capacity>
    </periodCapacity>
</lotSizeData>
```

```
ty/capacity

$i]}*I[{$i},{$t}])

))    return
| - I[{$i},{$t}] = {$demand} </con>  }

}*Y[{$i},{$t}] <= 0</con> }

{$i},{$t}])<= {$CAP[$t]} </con>}
```

Optimization Services modeling Language

4 ways of combining XML with optimization

1. Use XML to represent the instance of a mathematical program
2. Develop an XML modeling language dialect
3. Enhance modeling languages with XML features such as XPath
4. Use XML technologies to transform XML data into a problem instance

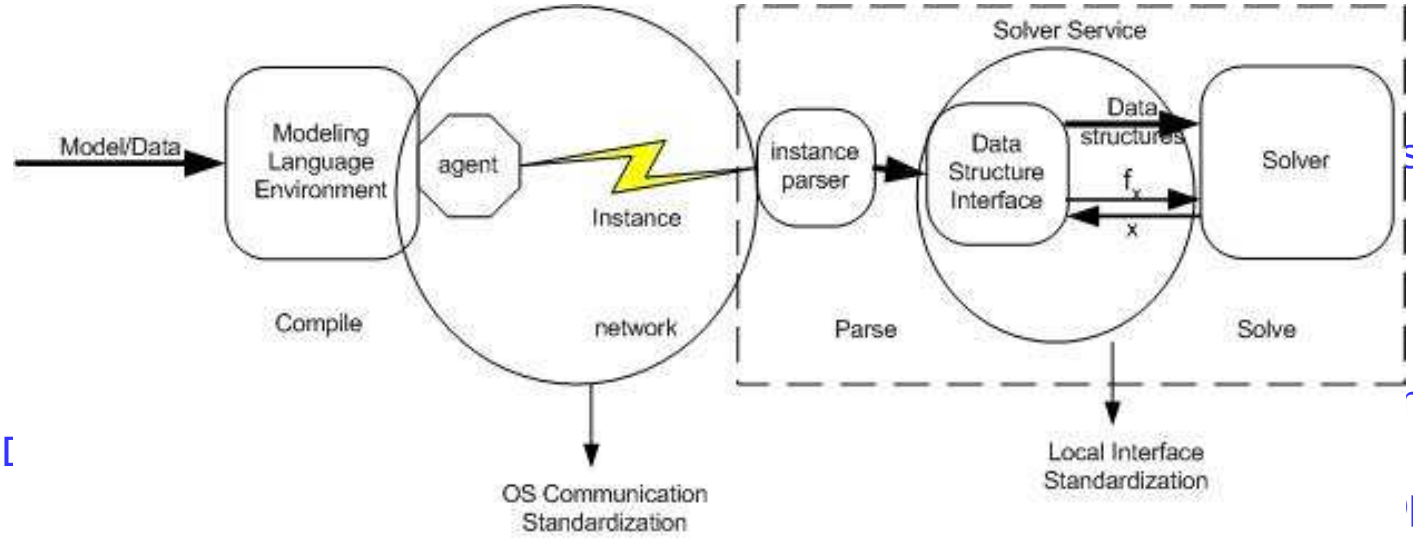


OUTLINE

1. Motivations
2. Demonstration
3. Optimization Services and Optimization Services Protocol
4. Optimization System Background
5. Computing and Distributed Background
6. Optimization Services Protocol - Representation
7. Optimization Services Protocol - Communication
8. Optimization Services Protocol - Registry
9. Optimization Services modeling Language (OSmL)
10. **Future and Derived Research**

Derived Research and Business Models

- The Optimization Services project
- Standardization
- Problem repository building
- OS server software, library enhancement
- Derived research in distributed systems (coordination, scheduling and congestion control)
- Derived research in decentralized systems
  (registration, discovery, analysis, control)
- Derived research in local systems (OSI? OSiI, OSrI, OSoI?)
- Derived research in optimization servers (NEOS)
- Derived research in computational software
  (AMPL, Knitro, Lindo/Lingo, IMPACT, OSmL, MProbe, Dr. AMPL, etc.)
- Derived research in computational algorithm
  Parallel computing



stry.

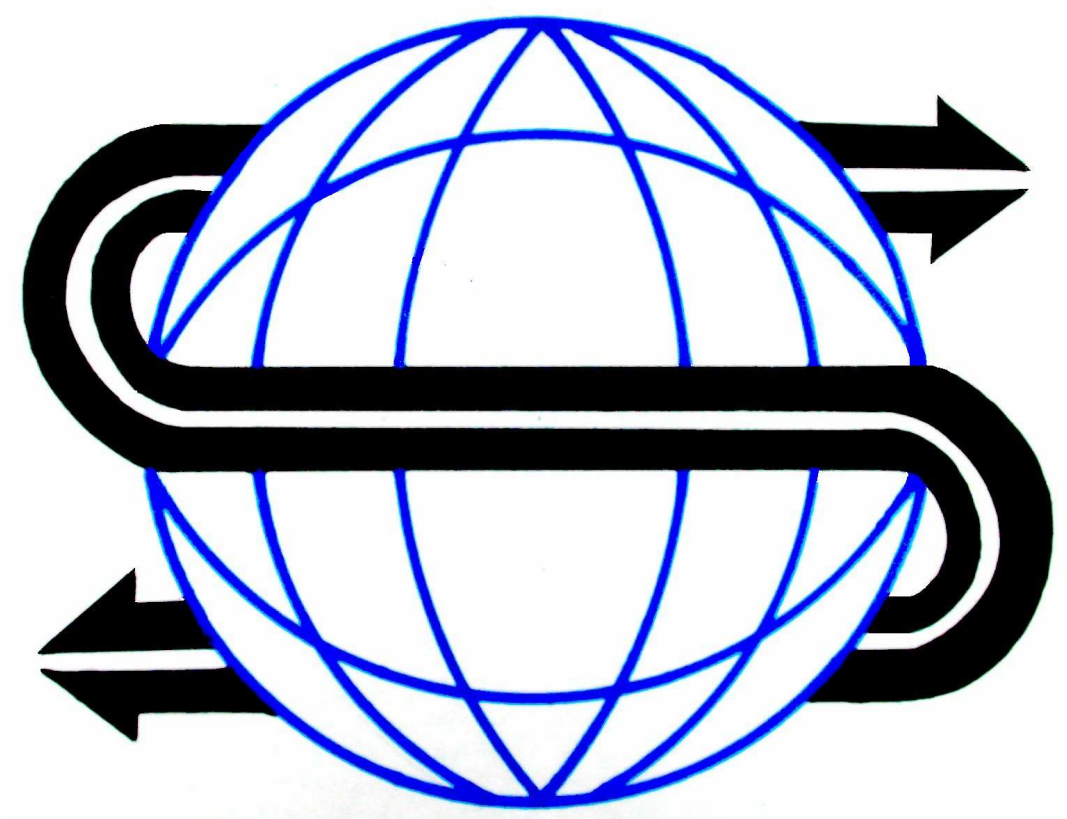
ntation schemes

- [

pers

  Library developers, registry/server developers, and other auxiliary developers
  Computing on demand and "result on demand"

End



http://www.optimizationservices.org (.net)