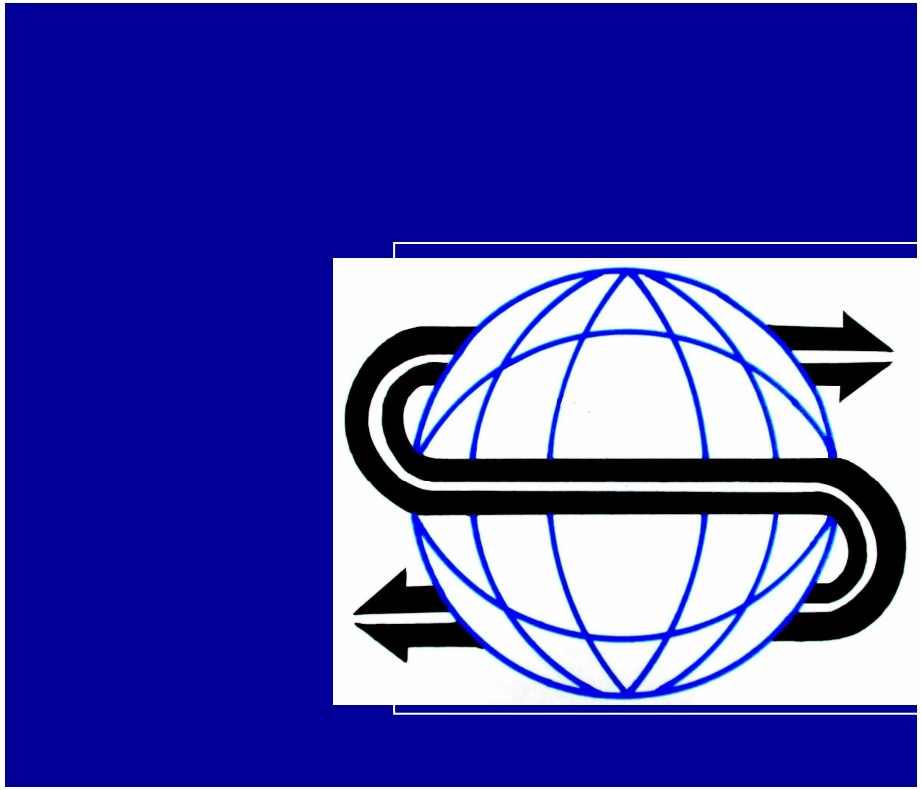


Optimization Services hookup Language (OShL)

-- API for invoking Remote Solver Service

-- Think of HTTP/HTML, JDBC/ODBC

-- Open Source

-- Platform Independent

-- Programming Language Independent

Jun Ma
Robert Fourer
Northwestern University
Kipp Martin
University of Chicago

Jun Ma

INFORMS, Pittsburgh
11/08/2006

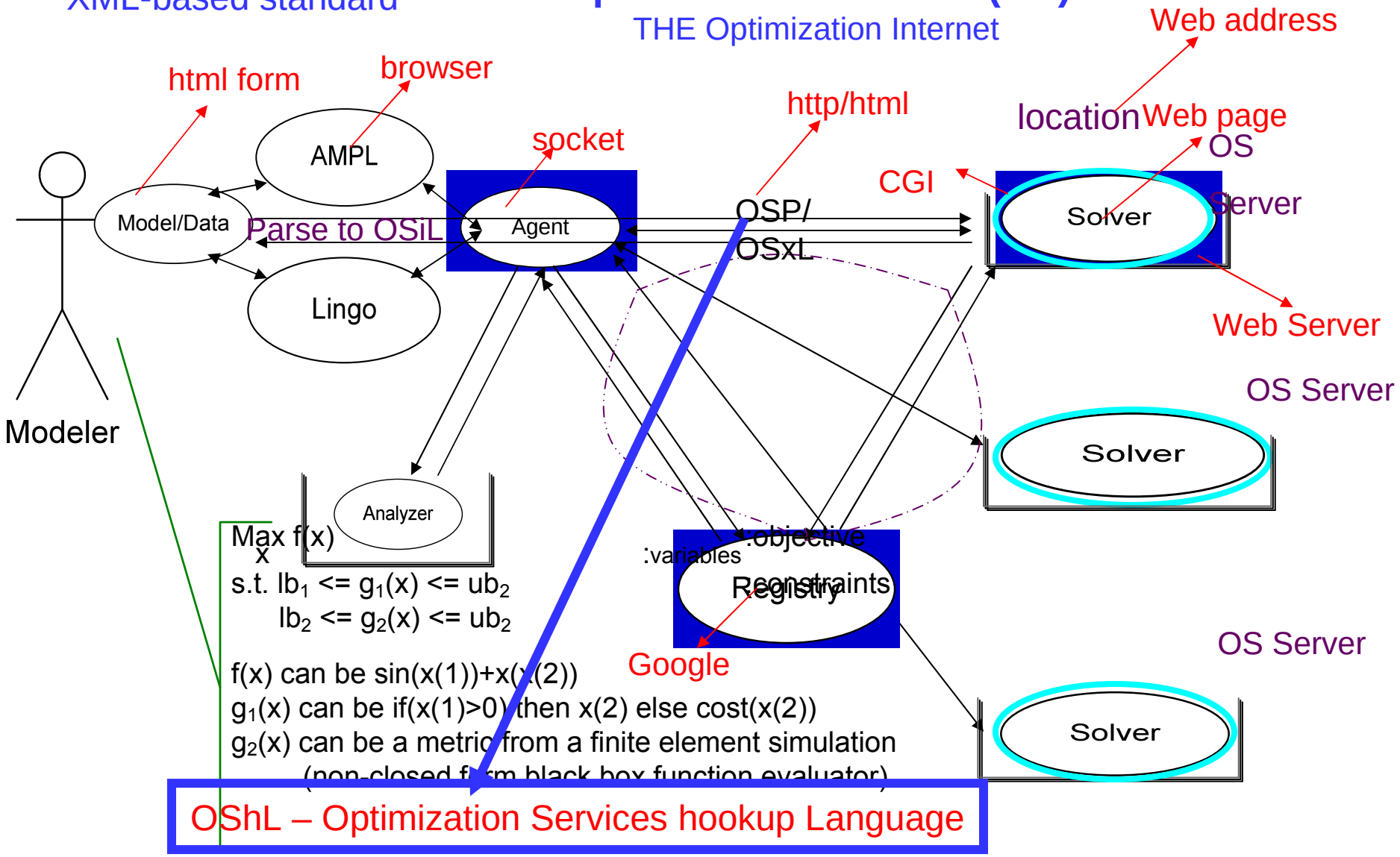
OUTLINE

1. Optimization Services and OSP (OSxL's)
2. Invoking a remote solver service
3. Optimization Services hookup Language (OShL)
4. Conclusion

Optimization Services (OS)

XML-based standard

THE Optimization Internet



Robert Fourer, Jun Ma, Kipp Martin

Invoking remote solver service (1)

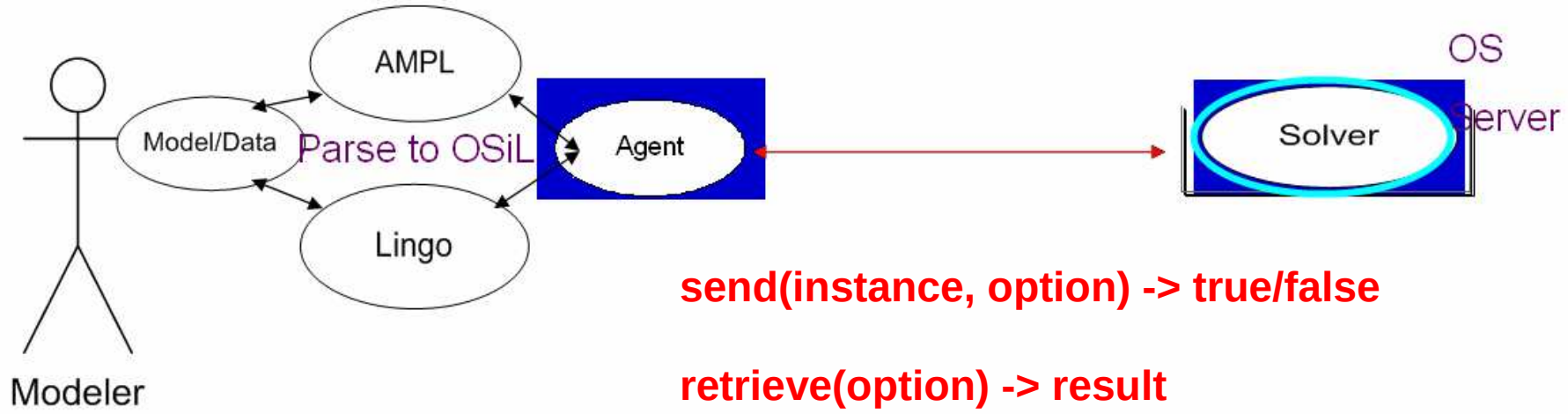
solve(instance, option) -> result

getJobID(option) -> jobID

Requirement	Solution
1. Platform Independent	OS is Web services based
2. Language Independent	OShL is XML Based WSDL
3. Protocol Independent	OShL is in SOAP envelope
4. Type compatibility	Uses all strings as arguments
5. Simple	6 methods, 2 args, only 1 method impl., no network knowledge req.
6. Built-in state/session maintenance	jobID
7. Request and response/Blocking	solve(instance, option) -> result

Robert Fourer, Jun Ma, Kipp Martin

Invoking remote solver service (2)



send(instance, option) -> true/false

retrieve(option) -> result

kill(option) -> killProcessResult

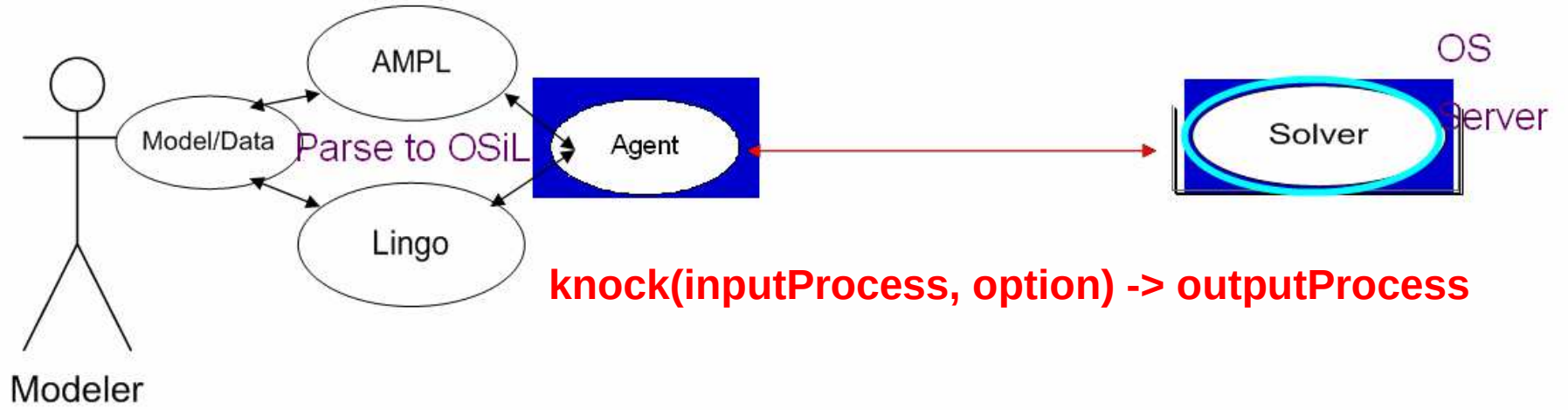
Requirement	**Solution**
8. Communication only	No specification on arguments

Argument specification -> xml schemas -> OSiL, OSoL, OSrL etc.

9. Truly asynchronous (server side)	send(instance, option) -> true/false
10. Retrieve anytime, anywhere	jobID, service-oriented architecture
11. Must stop remote process	Kill (≠ **stop** on the browser)

Robert Fourer, Jun Ma, Kipp Martin

Invoking remote solver service (3)



Requirement	Solution
8. Dynamic process information (heartbeat)	Knock
9. Extendable	2 inputs, 1 output (leverages on OSpL)

ping getServiceStatistics setServiceStatistics

getJobStatistics setJobStatistics

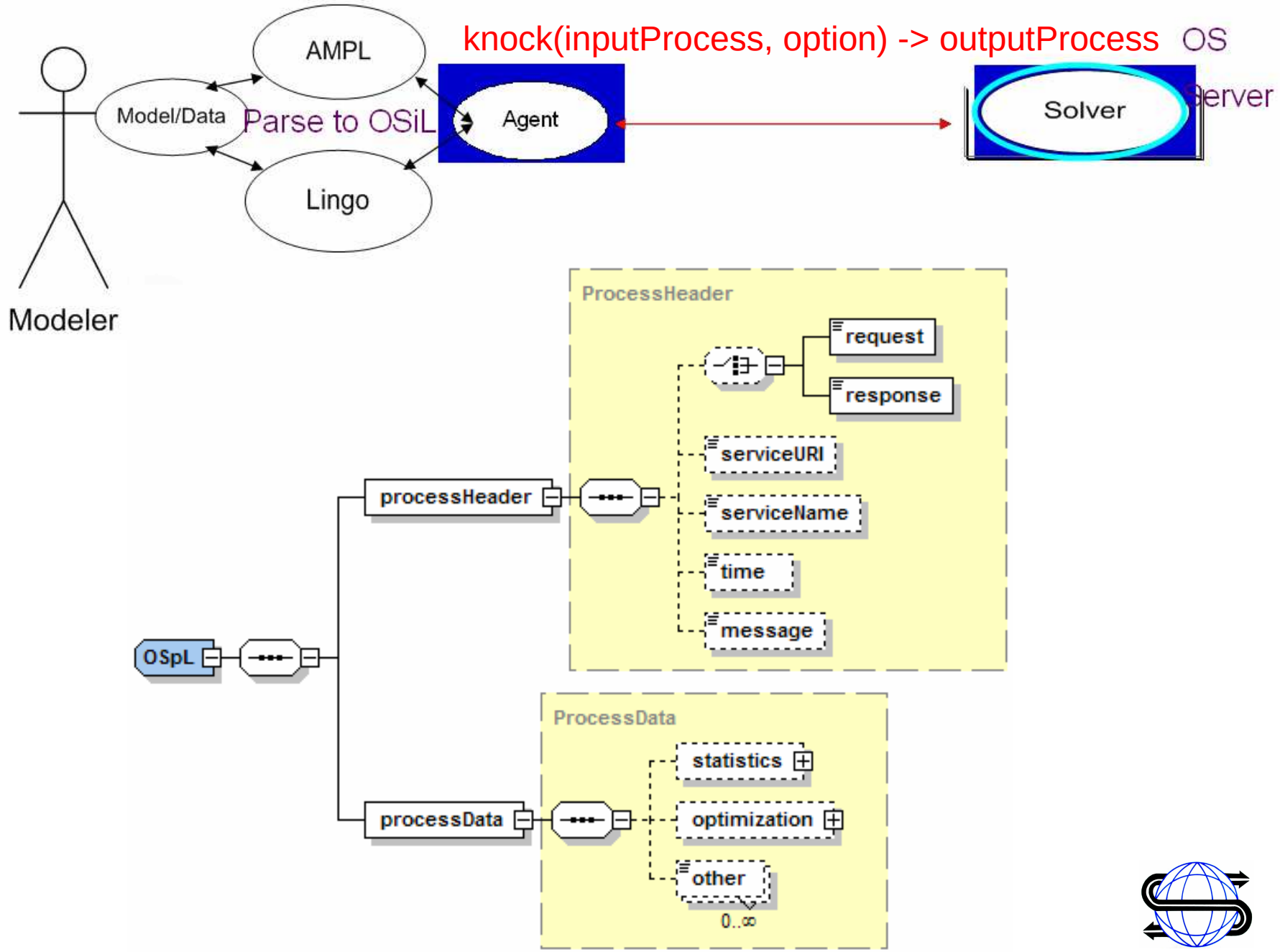
getOptmizationStatistics setOptmizationStatistics

getAll setAll

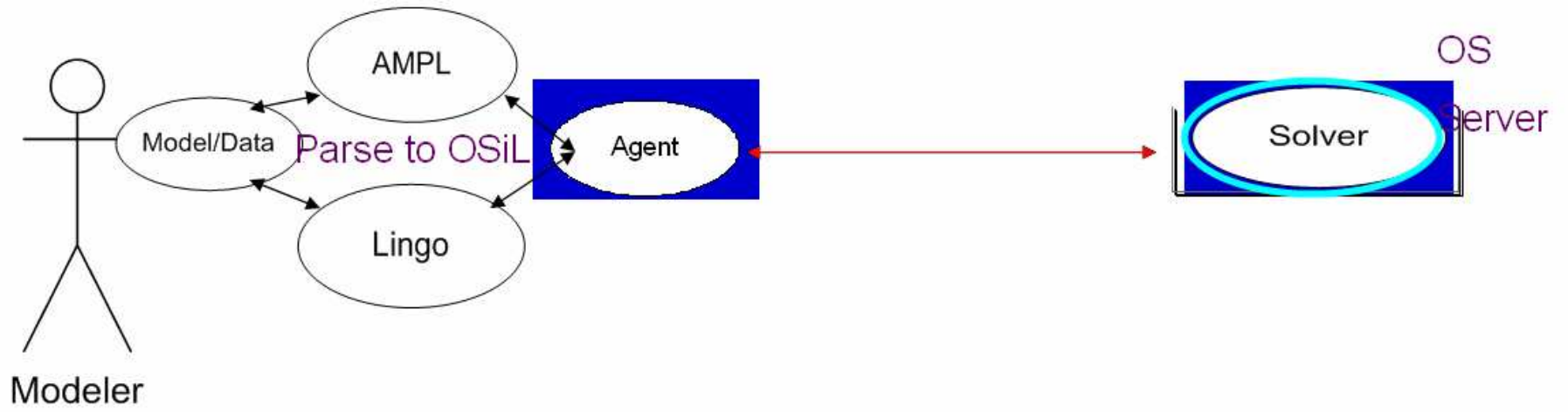
notifyJobCompletion requestJob

Robert Fourer, Jun Ma, Kipp Martin

Invoking remote solver service (4)



Optimization Services hookup Language (OShL)



getJobID (String OSoL) -> jobID

solve (String OSiL, String OSoL) -> OSrL

send (String OSiL, String OSoL) -> true/false

retrieve (String OSoL) -> OSrL

kill (String) -> OSpL

knock(String OSpL, String OSoL) -> OSpL

Robert Fourer, Jun Ma, Kipp Martin

Our OShL-compatible solver hosting SERVER reference impl.

Remote job submission, management and control
Remote retrieval of previously submitted jobs
Session and state maintenance
Synchronous and asynchronous solver invocation
Killing long jobs over the remote server
Checking and managing service status and job statistics
Automatic job completion notification via most common protocols including emails
Persistence between service starts
Service logging
Automatic notification of critical service information to admin
Centralized user configuration
Directory and file cleanup
Disk, memory and process cleanup
Critical data backing up
Waiting job queue management
Long computational job handling
Job dependency handling
Keeping track of service utilization and preparing periodic report
Automatic input and output validation and processing
Support of machines with multiple CPUs
Support of all major operating systems
Authentication and authorization
Security

Conclusion

- Optimization Services, OS Protocols (OSP => OSxL's)
- Design Requirement in Invoking Remote Solver Service
- Optimization Services

- Optimization Services will be released end of the year or beginning of next year.
- Almost all major parties (commercial, open source, research projects) are adopting it (It's a private process now!)
- Next generation NEOS
- Critical role in Cyber-infrastructure
- A Mega COIN project – touches nearly all the major COIN optimization-related projects
- "Run-Time" COIN (COmputational INfrastructure)
- Optimization Internet
- Contact us in private